OreSat UniClOGS Field Report

Miles Simpson, KI7VEI
Glenn LeBrasseur, KJ7SU

Portland State University

June 2019

Table of Contents

Introduction: OreSat, SatNOGS, and UniClOGS

The Portland State Aerospace Society at Portland State University is the lead organization building Oregon's first satellite, a 2U CubeSat named "OreSat". OreSat was selected as part of the 2017 NASA CubeSat Launch Initiative (CSLI). OreSat is a completely open source CubeSat that has two primary missions: a "video from space" amateur radio-based STEM outreach to Oregon high schools, and a multi-camera system for investigating the global distribution of high altitude Cirrus clouds. More information can be found at https://oresat.org/.

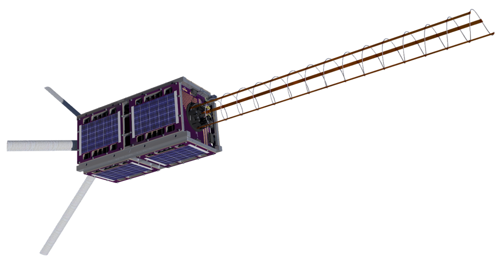

Figure 1: CAD model of OreSat, Oregon's First Satellite

OreSat uses a UHF (436 MHz) transmitter (downlink) and an L band (1.2 GHz) receiver (uplink) radio system. The UHF downlink is in a well used amateur radio band, and like most amateur radio satellites, will beacon every few minutes using morse code (CW) and digital transmissions (G3RUH, which is AX.25 over 9600 bps GFSK).

We expect amateur radio operators around the world to receive and decode the CW transmissions. We also expect that dozens of amateur radio operators all around the world will receive our digital transmissions using the Satellite Open Network Ground Station project, or SatNOGS. SatNOGS is a do-it-yourself, open source, extremely inexpensive and autonomous ground station for tracking and receiving amateur radio satellite transmissions. Literally hundreds of SatNOGS ground stations have been built, and along with a cloud-based database to coordinate the receives, has become the defacto ground station network for amateur radio-based satellites.

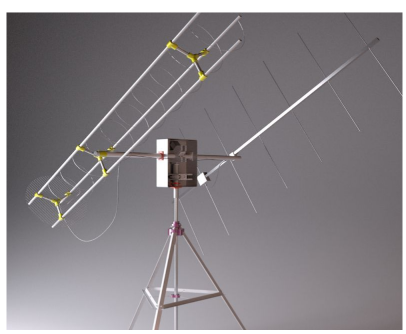

Figure 2: CAD model of SatNOGS Ground Station

The SatNOGS network has one major drawback: because of cost and global regulatory concerns, SatNOGS is receive only. With OreSat on the horizon, the OreSat team decided to create a larger, more rugged, and more capable version of the SatNOGS ground station meant to be operated by a team of licensed amateur radio operators, like at a university. Thus the University Class Open Ground Station, or UniClOGS, project was created.

Figure 3: UniClOGS prototype on the roof of the Portland State University engineering building

UniClOGS is meant to extend the SatNOGS project by adding transmission capabilities. Because each CubeSat and CubeSat program is different, UniClOGS is also built to be extremely flexible by being designed around the high end Myriad RF LimeSDR software defined radio (SDR). The current version of UniClOGS supports:

- **Downlink (Satellite to Ground)**
  - VHF (2 m / 145 MHz)
  - UHF (70 cm / 436 MHz)
- **Uplink (Ground to Satellite**
  - VHF (2 m / 145 MHz)
  - UHF (70 cm / 436 MHz)
  - L band (23 cm / 1265 MHz)
- **Amateur radio modes currently supported**
  - L/u, L/v, U/v, V/u, U/u, and V/v

Like SatNOGS, UniClOGS is fully open source, both in hardware and software. Git repository with full hardware and software information can be found at http://github.com/oresat.

UniClOGS Top Level Design

UniClogs is divided into four logical blocks:

- RF front end: Including LNAs, PAs, filters, and coax switch matrix
- SDR: Myriad RF LimeSDR providing three Tx and two Rx paths
- Server: Linux server that runs the mod/demod flow graphs, controls tracking, hosts power control services, hosts dashboard server, and stores IQ data samples
- User Interface: Any PC anywhere (with proper authentication) that runs a COSMOS control panel and client GNURadio flow graphs.

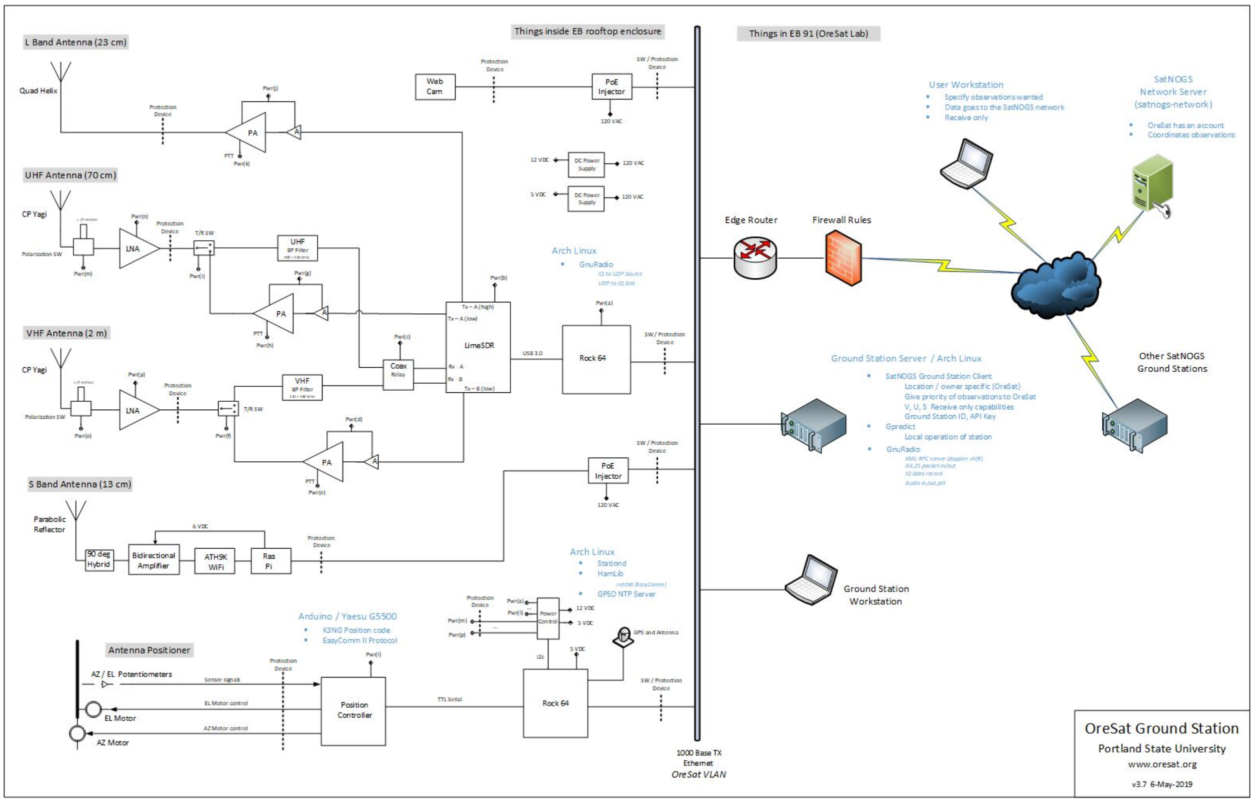

Figure 4: Block Diagram of UniClOGS

Physically, the system is divided into three discrete pieces:

- Station Enclosure
- Server
- User workstation

UniClOGS Detailed Design

The UHF component is centered in the 435 to 438 MHz amateur spacecraft service band and utilizes a circular polarized directional Yagi antenna with a RHCP/LHCP polarization sense switch. The VHF component is centered in the 144.30-144.50 and 145.80-146.00 MHz amateur spacecraft service band and utilizes a circular polarized directional Yagi antenna with a RHCP/LHCP polarization sense switch. The L-Band component is centered in the 1260 to 1270 MHz amateur spacecraft service band designated for uplink only, and utilizes a RHCP circular polarized 2x2 helix array. All of this is mounted on an Yaesu G5500 azimuth/elevation antenna positioner. An RF coax switch connects the UHF antenna to either a Low Noise Amplifier (LNA) for receiving, or a Power Amplifier (PA) for transmitting. The LNA and PA is then connected to a LimeSDR that provides both UHF RF input and output signals. The L-Band antenna is directly connected to a PA for transmitting only, and then to the same LimeSDR to provide the L Band RF signal. Figure 4 illustrates this system.

The LimeSDR used for VHF, UHF, and L-Band is from LimeMicro, and utilizes GNURadio for modulation and demodulation. The emission type used by OreSat is GFSK at 50 kb/s, and was chosen for the capability of the embedded transceiver on the OreSat spacecraft, an acceptable performance indicated by the link budget, and is somewhat common in the amateur satellite community. Other emission types are easily possible by use of GNURadio.

RF Path Design

The components in the RF path after the LimeSDR include the following:

Tx Amplifiers

In the transmit path is a MiniCircuits GALI49 (TB-409-49+ dev board) linear amplifier on both the VHF and UHF paths, and a MiniCircuits ZRL-1150LN in the L-Band path. These are needed to provide the signal levels required by the Power amplifiers that follow. The Power amplifiers are Monolithic modules from Mitsubishi or Hitachi, each providing about 30 watts of linear output power before P1dB compression. The PA's used are as follows; For VHF is the Hitachi SAV33 MOSFET; For UHF is the Hitachi SAU83L MOSFET; For L-Band is the Mitsubishi RA18H1213G. These are packaged into functioning PA components using products from Down East Microwave in the 2m30pack, 7025pack, and 2330pack kits respectively.

Rx LNAs

In the receive path is an Advanced Receiver Research GaAs LNA, MSP144VDG-160 in the VHF path, and MSP432VDG-160 in the UHF path. When power is not applied to these LNAs, they are bypassed by an internal coax relay allowing up to 160 W to be transmitted through them. These LNAs provide 24 and 18 dB of gain respectively at a noise figure of 0.55 dB, and are actually mounted out on the antenna boom to be as close to the antenna as possible.

Rx Filters

Following the LNAs are bandpass filters intended to suppress out of band signals that may be picked up by the antennas. For VHF there is a MiniCircuits ZABP-141-S+ and for UHF there is a ZABP-450-S+, each providing 30 dB or greater suppression outside of the VHF and UHF bands respectively.

Coax switches

The RF paths between transmit and receive are switched using an RF coax switch from DowKey Microwave. The DowKey 403-2208 is used for both VHF and UHF providing the 2 pole coax switch function. To switch the receive path between the two LimeSDR Rx(Low) inputs, a DowKey 411-2208 transfer switch is used. This allows the combination of full duplex satellite operation of L/u, L/v, U/v, V/u, U/u, or V/v to be realized with the LimeSDR.

Antennas

The circular polarized Yagi antennas used are from M2 Antenna Systems, where the model 2MCP14 is used for VHF, and the 436CP30 for UHF. The L-Band antenna is a quad 12 turn Helix providing RHCP only.

Polarization sense switch

The default polarization sense for the CP Yagi antennas is RHCP, and thereby a polarization switch from M2 Antenna Systems, PS2M for VHF and PS70CM for UHF is used to switch to LHCP should propagation conditions require it.

Power board

A power control board developed by the UniClOGS design team is used to switch all of the amplifiers, power amps, LNAs, coax relays, and polarization sense switches. Here all functionality of station operation can be controlled. Details of this board is on the UniClOGS Hardware GitHub repo. The board provides both high-side and low-side switching of the DC voltages controlled over an I2C bus.

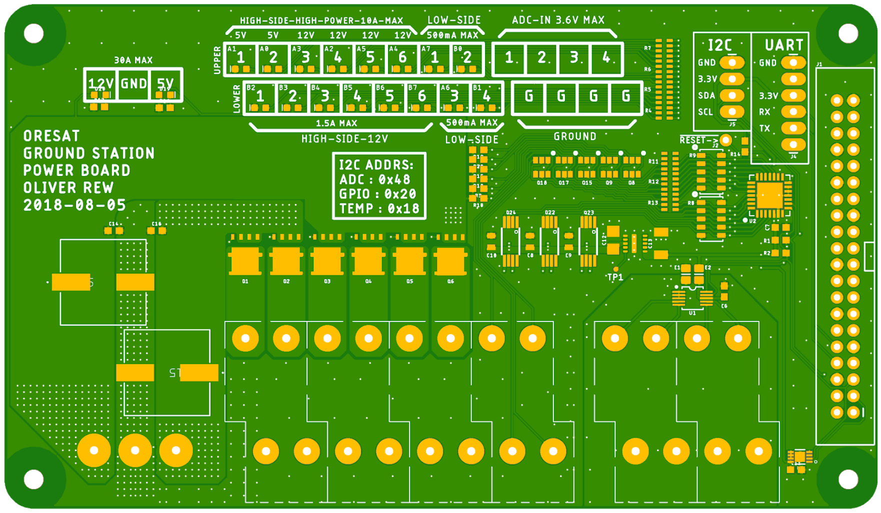

Figure 5: UniClOGS Power Control Board

Station Enclosure

All of the components of the UniClOGS station are contained inside of a NEMA-4 electrical enclosure. This enclosure is mounted right below the antenna mast to provide the shortest path to the antenna boom system. The antennas are connected using 1/2" 50 Ohm Heliax Coax Cable (Andrew FSJ4-50B), and pass into the enclosure through protection devices from PolyPhaser. All components are listed in the bill of materials on the GitHub repo.

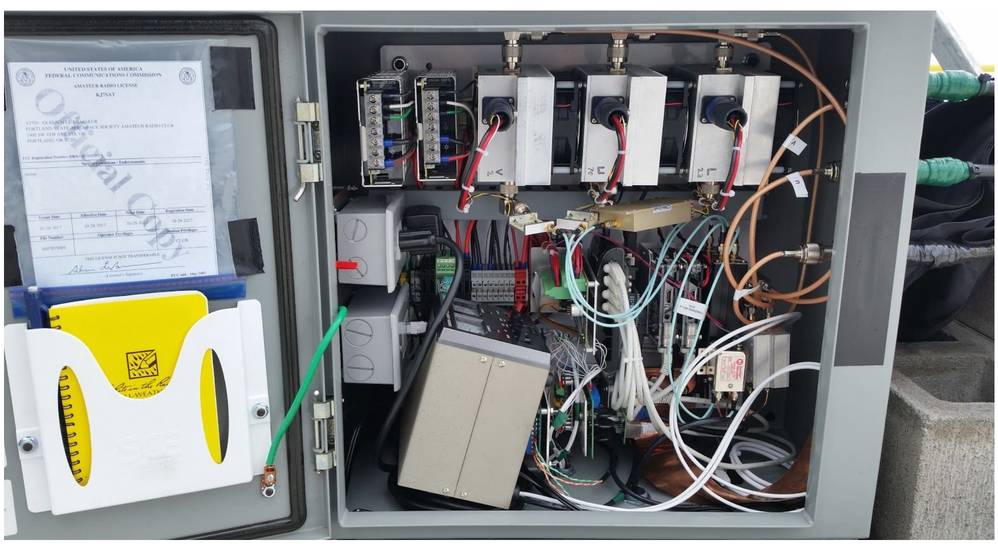

Figure 6: Station Enclosure

UniClOGS Ground Station Software

The ground station uses several different software packages in order to support operation. For power and state management, we developed some inhouse software to interface with our custom power board. We use GNURadio to develop our SDR flowgraphs for the remainder of the RF path. In addition, we use GPredict and HamLib, along with K3NG, to control antenna orientation and for calculating doppler shift. Finally, we use Ball Aerospace's COSMOS to integrate it all together, and monitor system health with Zabbix.

Stationd

Our power board is controlled via I2C with a daemon we call "stationd" that implements state control for our ground station hardware through a network interface. Through this interface we can command systems on and off, and get status information, temperature, and RF power measurements.

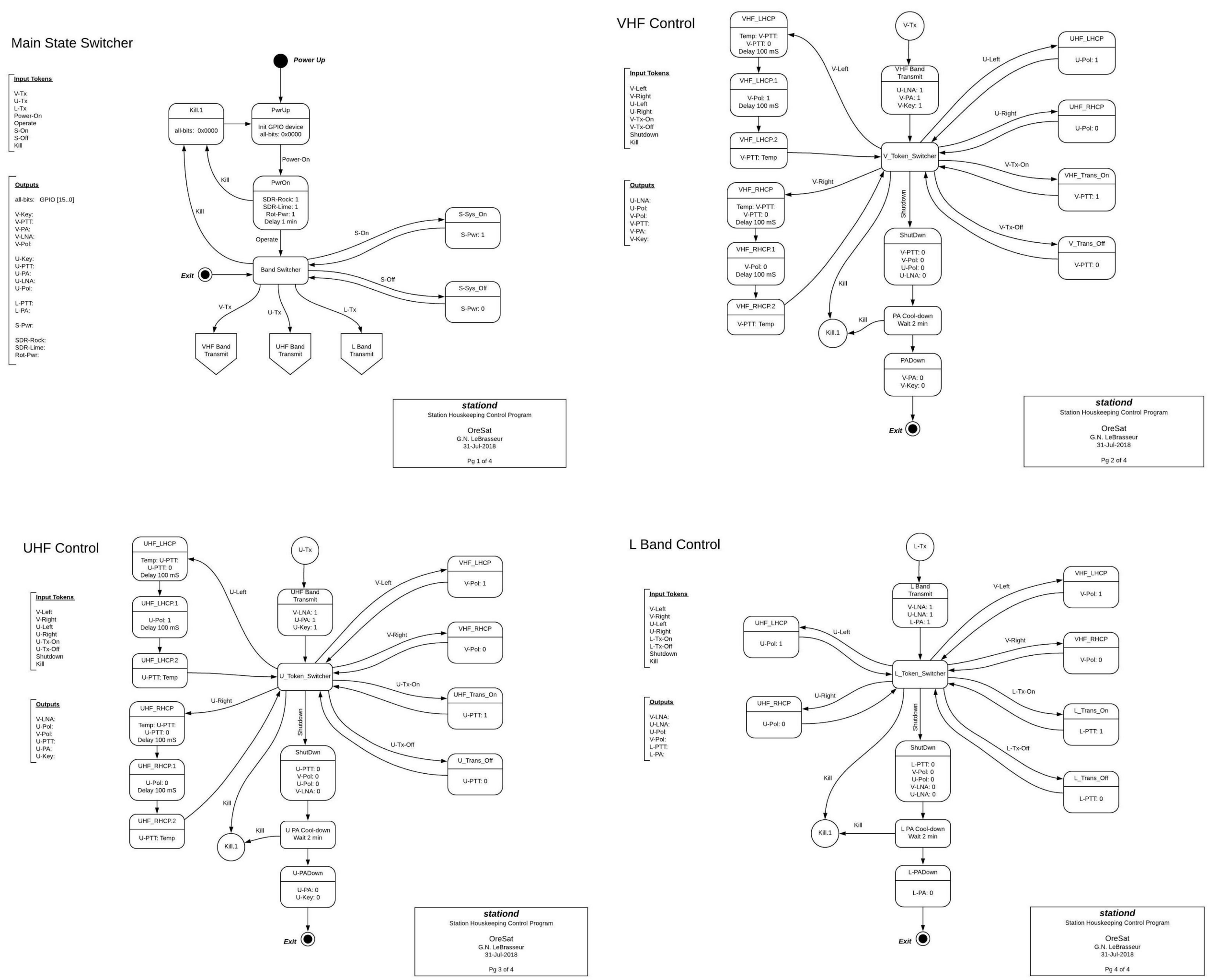

Figure 7: Stationd State Diagram

GNURadio Flowgraphs

Testing Flowgraphs

In order to test the software side of our radio subsystem, we used a couple of simple GNURadio Companion flowgraphs, utilizing MyriadRF's gr-limesdr LimeSuite Source and Sink blocks and ZeroMQ PUB and SUB blocks. The LimeSDR side of the flowgraphs are headless, and simply publish the raw IQ samples to the network. On our workstation we did a simple test by subscribing directly to the radio IQ samples and attempting an FM demodulation on a specified frequency.

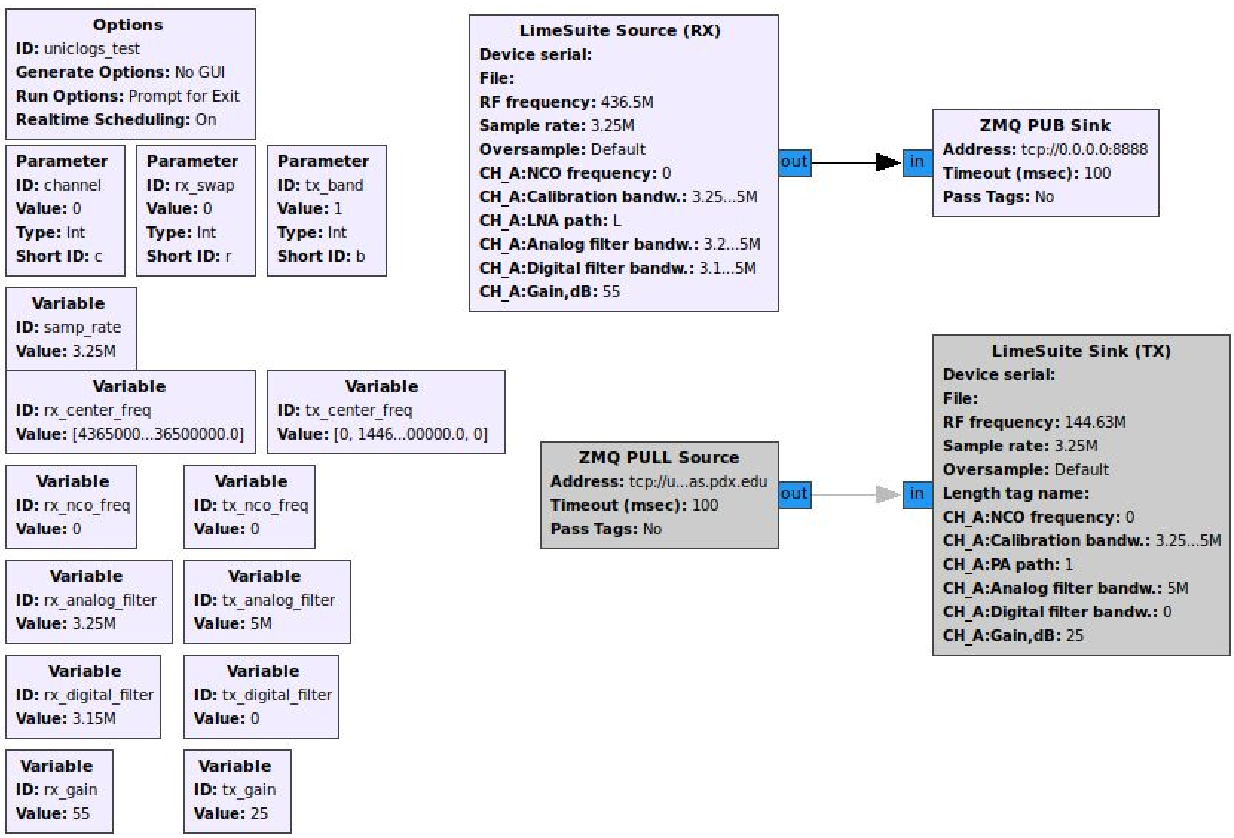

Figure 8: Headless LimeSuite Source Test

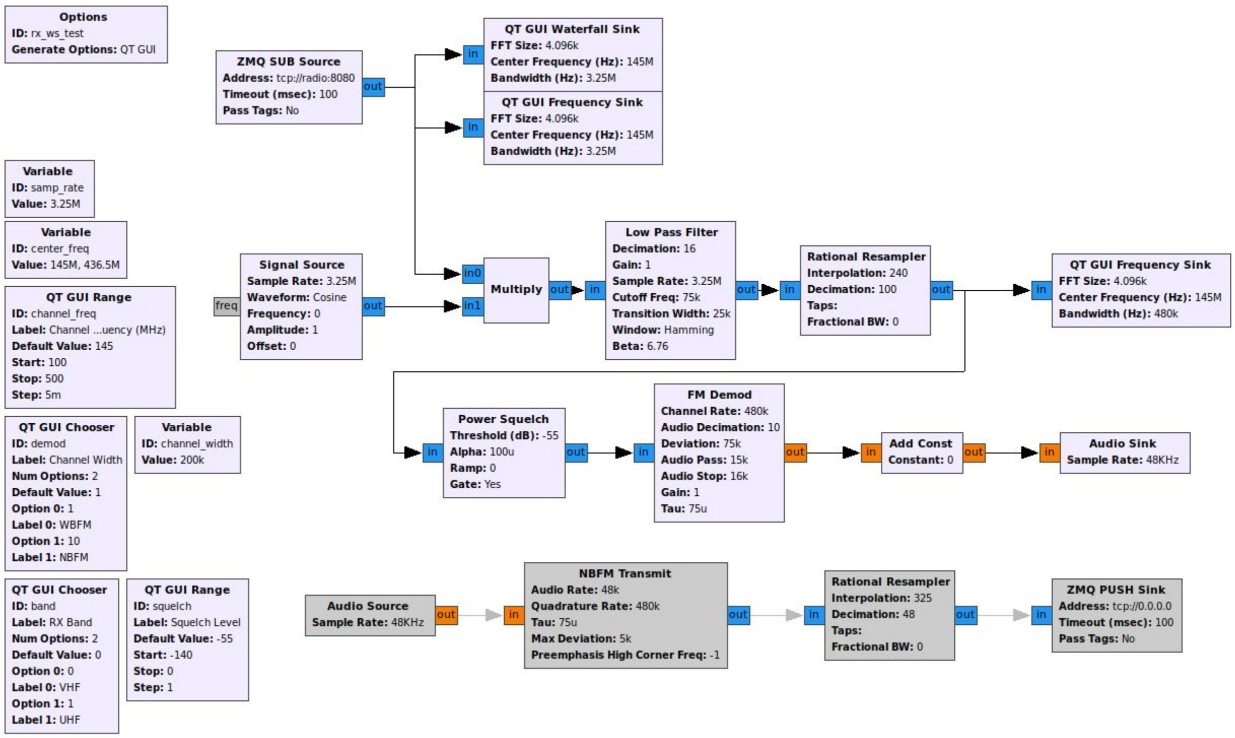

Figure 9: FM Demodulation Test

Future Flowgraphs

Our next step is to develop more sophisticated C++ implementations of GNURadio flowgraphs and move Modulation and Coding (ModCod) to the server, with the server acting as a headless control point and intermediary for any client workstations that wish to operate the ground station. This will centralize and secure control of the ground station radio components, and permit autonomous operations and observations of satellite passes. We will be implementing ModCod blocks for CCSDS specifications, as well as frequently used Amateur protocols.

Antenna Control

An antenna position controller is used to aim the antenna array and is interfaced serially using the EasyCommII control protocol accessible through the HamLib control library. The position controller is based on the Open Source hardware/software project by K3NG, and provides the ability to control satellite tracking through HamLib's *rotctld*.

GPredict

On a client workstation, a user can track satellites via *GPredict*, which is a program that uses TLE data to predict the position of a satellite in the sky given the position of the ground station. This software also provides an interface for commanding rotator control and doppler shift adjustments to hardware. In the case of UniClOGS, these commands will be arbitrated through the server.

HamLib (*rotctld*)

The actual antenna orientation is controlled via HamLib's *rotctld* software, which provides a network interface that accepts commands on how to move the antenna. This is running as a simple daemon on our station controlling Rock64.

Tracker controller

The tracker used with UniClOGS is a Yaesu G5500 commercial product and is interfaced to *rotctld* via the K3NG Rotator Controller. This is essentially an Arduino and simple interface shield to convert the serial "EasyComII" protocol that *rotctld* speaks, into the relay output and potentiometer input expected by the Yaesu G5500 rotator control box. K3NG has provided an open source solution implementing this Arduino code.

The OreSat GitHub at 'https://github.com/oresat/uniclogs-hardware' repo contains some simple instructions on how to get the code, a description of the shield, and the settings needed by the UniClOGS hardware.

COSMOS/Zabbix

The final components of UniClOGS software are Ball Aerospace's COSMOS and Zabbix.

COSMOS is a simple to use Ruby application that is designed to provide Telecommand and Telemetry interfaces for embedded systems. We will be using this to control the ground station itself, as well as for controlling our OreSat satellite project. COSMOS is what will be wrapping everything together and providing a nice interface for people to use to control the ground station. It is also what will allow us to construct different payloads to send via our GNURadio ModCod interfaces and implements the various higher level protocols.

Zabbix is a simple network and system health monitoring tool. We use it to monitor the status of the various linux systems used in our ground station, and to report health and errors to system administrators.

Further Information

For those wishing to take a look at UniClOGS, or who wish to implement a similar system themselves, our hardware design and documentation is available at https://github.com/oresat/uniclogs-hardware and our software and documentation available at https://github.com/oresat/uniclogs-software.