Implementing a Software-Defined Radio Scheme for a CubeSat Ground Station

Spectral Ocean Color (SPOC) and Multi-view Onboard Computational Imager (MOCI)

Sydney Whilden*[1], Nicholas Neel**[1], David L. Cotten^[1,2], Deepak Mishra[1,2]
*sydney.whilden25@uga.edu, **13ncn@uga.edu, ^dcotte1@uga.edu,
[1]Small Satellite Research Laboratory, University of Georgia,
[2]Center for Geospatial Research, University of Georgia

Overview

The Center for Orbiting Satellite Mission Operations (COSMO), composed of a combination of open source applications and off-the-shelf components, will be a fully functional command center, sustained entirely by the Small Satellite Research Laboratory. One way to drive down cost is to home-brew solutions; software-defined radio virtually eliminates the need for many costly pieces of hardware, and maximizes user control. For an undergraduate-operated mission, scheduling commands or downlink is difficult due to both orbital and monetary constraints. The approach described here could suffice for either testing or for actual nominal operations. Most hardware components are replaced by computer-based DSP.

Requirements

The chief requirements driving the development of the SDR scheme are as follows:

1) Must be capable of both transmitting and receiving over UHF, and receiving over S-band.
2) The station must be capable of encoding and decoding unnumbered information (UI) frames in AX.25 format. AX.25 is a communication protocol designed to ensure link layer compatibility between stations, and conforms to ISO standards.
3) Telecommand and telemetry packets must be delivered reliably using GMSK modulation at 9600 baud.
4) Science data must be received using QPSK IESS-308 encoding and framed in CCSDS packets.

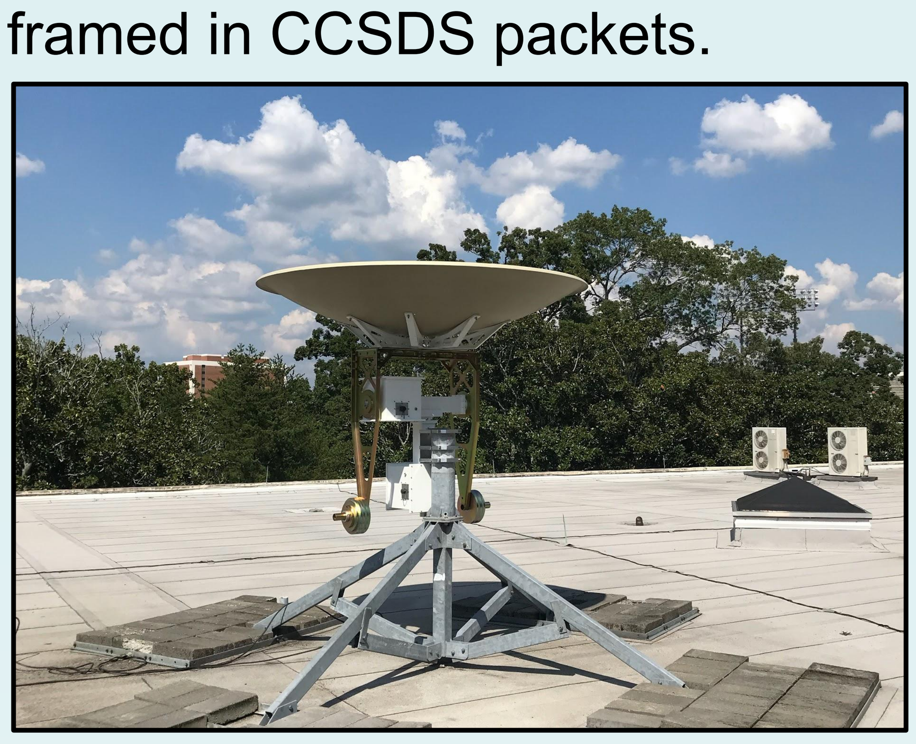

Figure 1: COSMO ground station dish,Yagi antennas no yet installed

Equipment

- GNURadio performs most of the DSP.
- AX.25 encoding is done by a collection of Python scripts (transitioning to non-deprecated GR blocks). This is essentially a software TNC.
- RTL-SDR dongle
- HackRF
- Ettus USRP

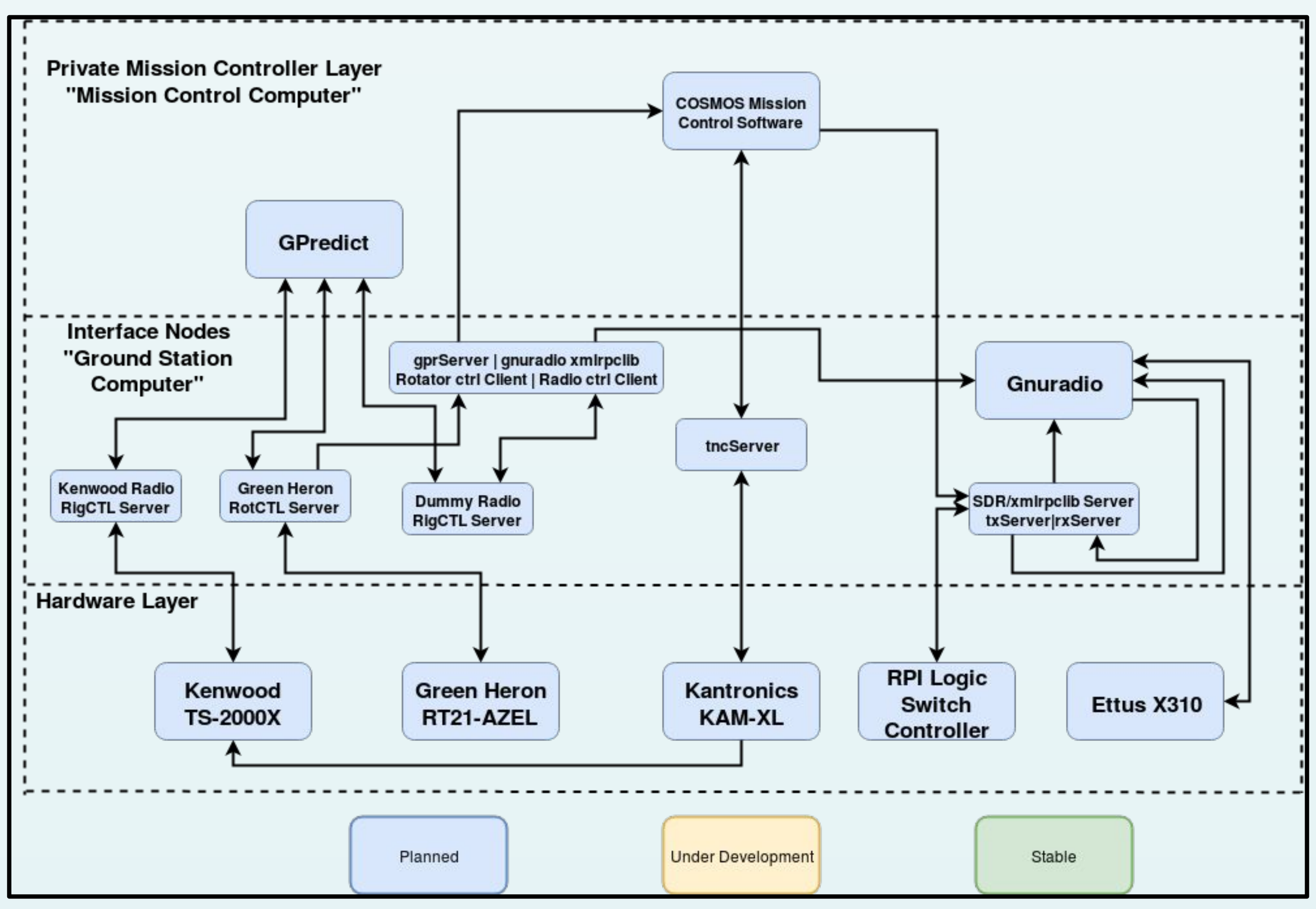

Figure 2: Ground station software plan

Methodology

The transmission process consists of analog-to-digital (ADC) conversion, packetization in AX.25 format, GMSK modulation, and output from the computer terminal through the transmitting antenna (usually the HackRF or the dongle for testing purposes). Conversely, the receiving setup comprises digital-to-analog (DAC) conversion, demodulation, de-packetization, and output to the computer terminal.

GNURadio's basic AX.25 packetization functions are deprecated. This was solved using a simple set of Python-based packet framing and de-framing tools, coded in-house. Each packet must be converted to binary format, padded with ones and zeroes, fitted with flags at intervals, and given the appropriate control field. A checksum is added to the end of each frame. Another significant challenge was achieving reliable packet transmission at the required data rate of 9600 baud. Rather than GR's basic Rational Resampler block, the Polyphase Arbitrary Resampler block proved more effective because of its more configurable filter setup, as well as its more transparent methodology. Resampling is easily among the trickiest steps of signal processing; even when conceptually clear to the operator, it remains finicky, and requires close attention.

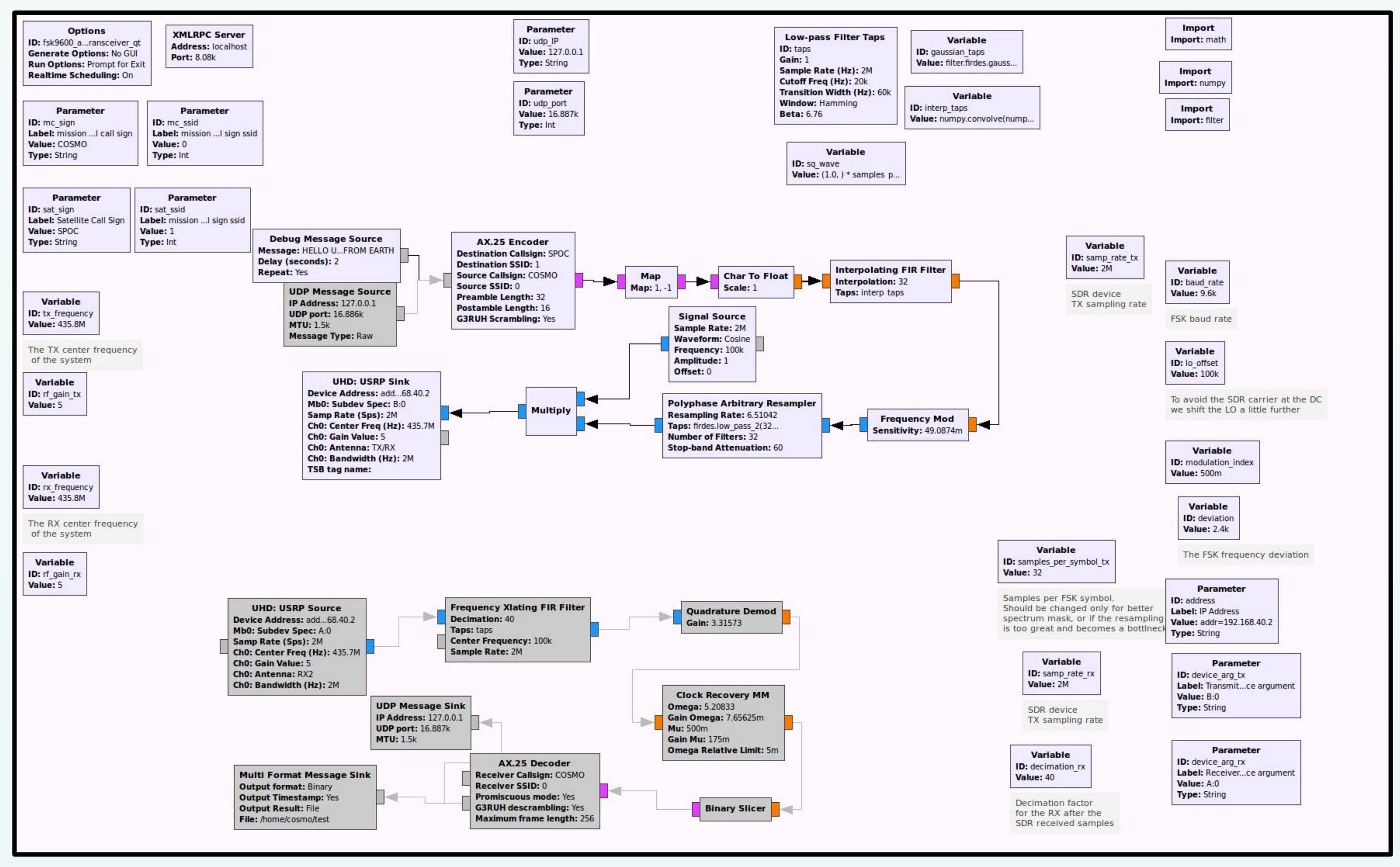

Figure 3: GNURadio test loop flow diagram

Results & Future Considerations

SDR was found to be flexible enough to support most ground station needs. Software-defined radio is a viable option for cost-limited missions like those run by undergraduates. It does, however, require the operators to have a firm grasp of basic signal processing steps and techniques, especially resampling. As the number of university CubeSat missions increase, demand for ground station use increases, and so does the need for accessible and inexpensive radio equipment. As methods involving resources like GNURadio are test-driven, so to speak, tried-and-true approaches emerge, and become usable without sacrificing quality or reliability.

References

Bloessl, B., Segata, M., Sommer, C., & Dressler, F. (2013, August). An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio. In *Proceedings of the second workshop on Software radio implementation forum* (pp. 9-16). ACM.

Ge, F., Chiang, C. J., Gottlieb, Y. M., & Chadha, R. (2011, December). GNU Radio-based digital communications: Computational analysis of a GMSK transceiver. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE* (pp. 1-6). IEEE.

Schmid, T. (2006). Gnu radio 802.15. 4 en-and decoding. *UCLA NESL TR-UCLA-NESL-200609-06, Tech. Rep.*

For more information about our research/missions, visit www.smallsat.uga.edu

Small Satellite Research Laboratory
Franklin College of Arts and Sciences
UNIVERSITY OF GEORGIA