Software Architecture of the UGA COSMO Ground Station

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

UNIVERSITY OF GEORGIA

Richard Red[*1], Mateen Saki[*1], David L. Cotten[^1,2], Deepak Mishra[1,2]
[*]richardred.q@gmail.com, mateensaki1@gmail.com[^]dcotte1@uga.edu,[1]
Small Satellite Research Laboratory, University of Georgia,[2]
Center for Geospatial Research, University of Georgia

Overview

In order to communicate with satellites, COSMO (Center for Orbital Satellite Mission Operations) is necessary to establish uplink and downlink connections whenever a satellite is passing over and within range of the satellite dish. However, the dish itself, while crucial for receiving downlinked data from the satellite, is merely one variable in the greater equation that includes the other components of the ground station, which are summarized in the below abstract.

Abstract

The software components of the UGA ground station provides the underlying architecture for interacting with the hardware components. The core private mission controller layer consists of a physical computer running Ubuntu 16.04 that will control the hardware components with COSMOS Mission Control Software and GPredict, a real-time satellite tracking application. This computer is connected to another main computer via OpenVPN. The software defined radio portion of this is handled with GNUradio and uses an Ettus USRP X310 with two daughterboards, enabling transceiving operations from 1-6000 MHz; it will be handled primarily with GNUradio. This, along with the hardware defined portion of the ground station, will be able to communicate using Gaussian Minimum Shift Key modulation at 9600 baud. A Green Heron RT-21 AZEL will control the physical rotation of the ground station and will be able to remotely change the azimuth and elevation of the satellite dish, and the cabling will be exclusively LMR-400. The hardware/software defined portions will be toggled with a Raspberry Pi switch.

Software Architecture

Figure 1 shows a very abstracted representation of the above software components. There are two primary software elements: COSMOS, Ball Aerospace's mission control software, and GPredict, a real-time satellite-tracking and satellite orbit prediction application.

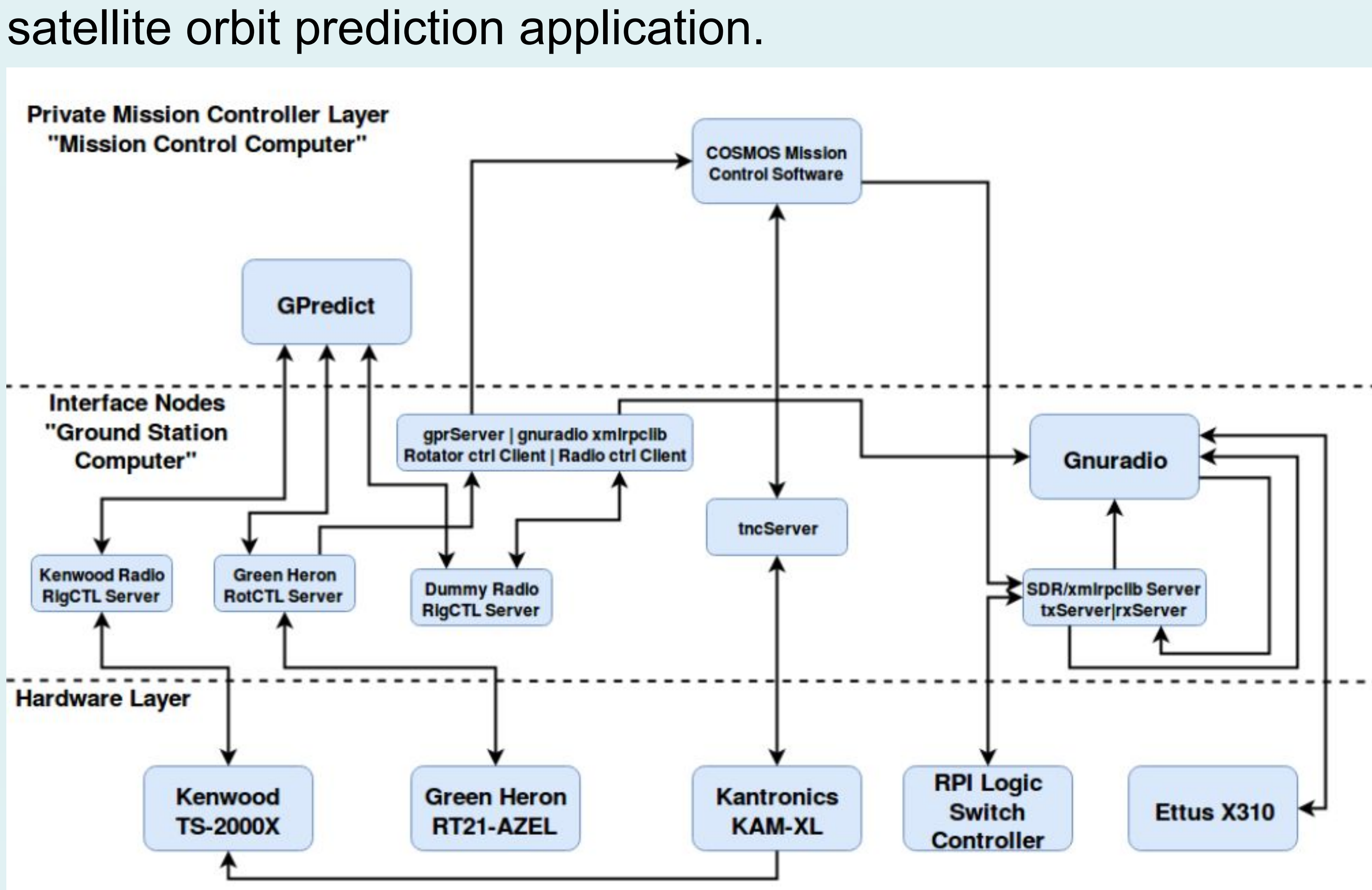

Figure 1: Software Components. Courtesy Nicholas Neel.

Next are all of the pieces that collectively form the physical "ground station computer", which are the parts that provide interfaces between the components in the mission control layer and the hardware components. More specifically, it consists of GNU Radio, which handles the software defined radio portion of the entire ground station system, and a bunch of servers that provide interfaces for the radio hardware and the rotator controller which will eventually move the satellite dish.

Other Components

While the focus of this presentation is on the software of the ground station, it is the hardware that actually has the physical capabilities to move and track satellites currently in orbit. The Green Heron RT-21 AZEL rotator controller will be located in a control room very close to the satellite dish along with the ground station computer, which will be remotely controlled by a main mission control computer located in a slightly more accessible location.

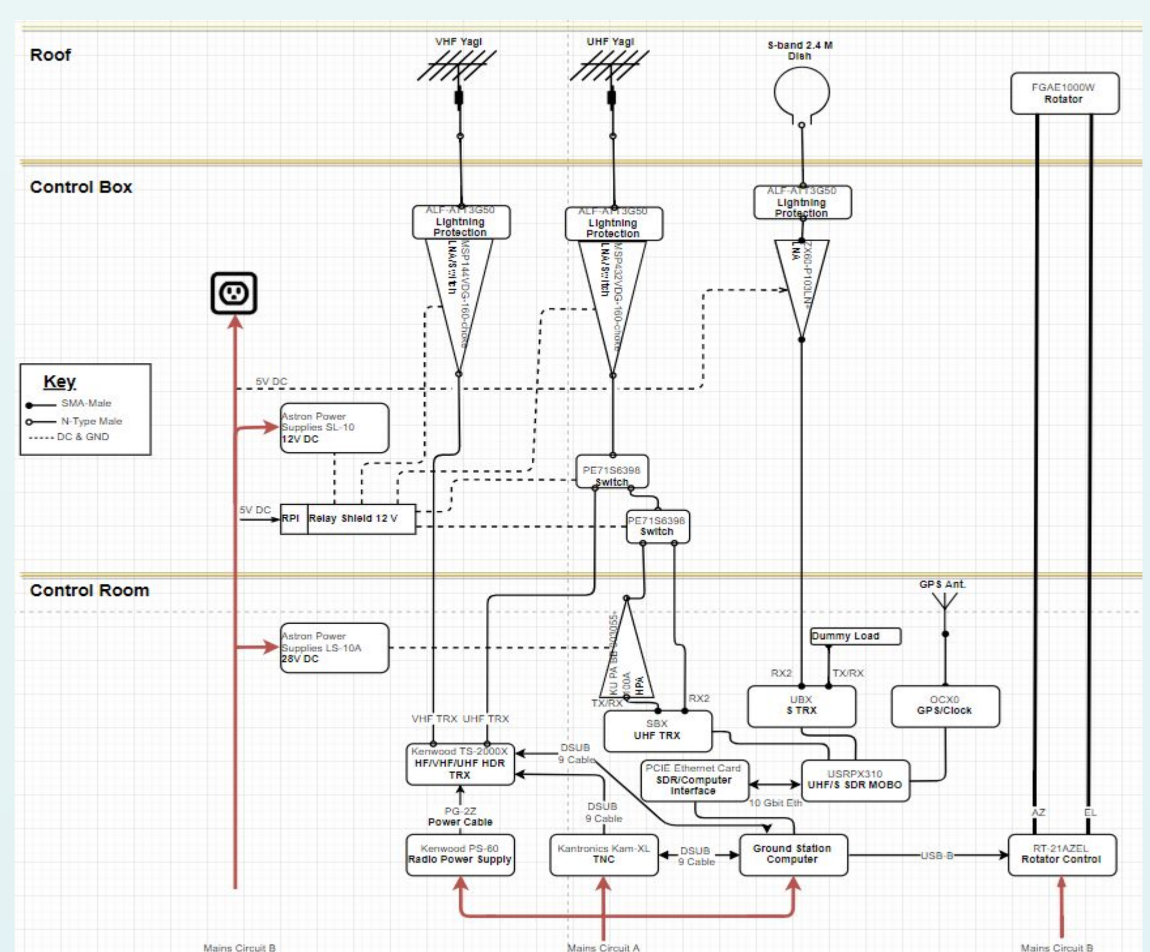

Figure 2: An outline of the hardware components, with the aforementioned software components embedded in the bottom layer. Courtesy Nicholas Neel.

The connection between the two computers requires an extremely high level of security. In order to achieve this, an OpenVPN network was configured with an Apache web server in order to completely hide the ground station computer's presence, even within UGA's internal network; this way, it can only be accessed via direct connections (SSH, VNC) from the primary mission control computer.

Future Considerations

Because of the flexible nature of software, certain additions that don't utilize any of the hardware components can be made to the system. For example, there is a weather sensor within close proximity to the satellite dish that posts data online, so it was possible to simply pull data from that API directly into our mission control software to provide additional insights and information about the status the dish.

References

Feilner, Markus. Openvpn: Building and Integrating Virtual Private Networks. Packt Publishing Limited, 2016.

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

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