North Dakota State University--Fargo

From the SelectedWorks of Jeremy Straub

April, 2014

Development of a Ground Station for the OpenOrbiter Spacecraft

Jacob Huhn
Alexander Lewis
Christoffer Korvald
Jeremy Straub
Scott Kerlin

Available at: https://works.bepress.com/jeremy_straub/189/
Development of a Ground Station for the OpenOrbiter Spacecraft

Jacob Huhn, Alexander Lewis, Christoffer Korvald, Jeremy Straub, Scott Kerlin

Department of Computer Science, University of North Dakota

About
The OpenOrbiter Small Spacecraft Development Initiative[1] at the University of North Dakota is working to design and build a low cost[2] and open-hardware / opensource software CubeSat[3]. The Ground Station is the user interface for operators of the satellite. The ground station software must manage spacecraft communications, track its orbital location, manage task assignment, provide security and retrieve the data from the spacecraft. This will be presented via a graphical user interface that allows a user to easily perform these tasks.

The goals of this project are to:

i) create an intuitive, powerful graphical user interface for ground-station personnel to send commands to an unmanned small satellite

ii) develop a centralized task / spacecraft data database that allows for multiple concurrent ground-stations to send and receive commands or data to the CubeSat, as it becomes accessible to each ground-station

iii) design a ground-to-space data link that automatically sends commands to and receives data from the unmanned small satellite as it passes over the ground-station

Scientific Merit
A modular, unmanned small satellite is being designed as a fully open-source software / open-hardware university-class satellite that aims to reduce the costs of placing a satellite in space significantly[4]. The design allows academic institutions to develop their own modules for additional functionality at a small cost. To increase the usability of OpenOrbiter and decrease the technical requirements of end-users, an intuitive graphical user interface that converts basic commands into powerful commands is strongly desired. Since communications windows are limited, the prioritization of command and data transmissions and the ability to connect multiple ground stations over a wide geographic area are of paramount importance.

In order to satisfy these requirements, a centralized database will be created, allowing multiple ground stations to:

i) see the current prioritization queue

ii) add to the prioritization queue

iii) see commands that have been sent to the OpenOrbiter and are expecting results

iv) see all results returned by the OpenOrbiter.

The initial graphical user interface will be designed to support the OpenOrbiter’s image-processing mission, the interface will be built to facilitate later extension and enhancement.

Current Progress
i) User Interface — The user interface is a website that was developed with the Django Web Framework. The website has a connection to a MySql database that gives us functionality such as adding and deleting tasks from a given database. A graphical interface on the task creation page makes it easier for the user to get the correct latitude and longitude coordinates.

ii) Web Server — We are utilizing Microsoft’s Azure cloud service to create a Ubuntu virtual machine to house our web server, database, and backend programming. The client program will be able to connect to the VM via a TCP/IP connection. This setup allows us to have multiple different ground stations able to connect and share information with our main server.

iii) Client programming — This programming helps facilitate the connection between all of the information on the webserver and the spacecraft. It connects to the webserver and receives any information that the client needs to send. After receiving, it sends the information through a serial port to the radio transmitter. It also receives information from the satellite, which it turns into usable information and sends to the webserver for input into the database

iv) Hardware—in the process of obtaining the hardware necessary to communicate efficiently.

Future Challenges

i) User Interface — Implement a visual representation of where the spacecraft will be.

ii) Web Server — Refine backend programming to allow seamless connections.

iii) Client programming — Implement hardware control features to tell the radio transmitter where and when it needs to point to allow a connection to the OpenOrbiter satellite.

iv) Hardware — Integrate hardware and software components to allow communication.

Note
This poster is updated and extended from [4, 5].

References