Software Development for MUSE - CubeSat

Michael Baker
Mentors: Dr. George Corliss and Dr. Dennis Brylow

Background

The CubeSat standard was developed in 1999 by California Polytechnic State University. One such satellite is pictured.

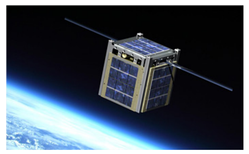


Photo by NASA

NASA provides opportunities for these small satellites to be an auxiliary payload on designated missions. This allows high schools and colleges around the country to be able to design, build, test, and deploy real satellites in space.

Our satellite is part of Nasa's ELaNa program (**E**ducation **La**unch of **Nan**osatellites). It is scheduled for launch on January 17, 2017 aboard a Delta II rocket. Pictured below is the actual rocket on which our satellite will fly.

Photos by NASA

Along with NASA, a company called Tyvak is assisting in the preparation of our satellite for launch. They help coordinate with NASA and ensure that we are prepared to actually put a satellite into orbit.

Purpose

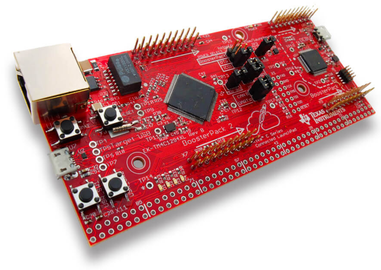
One of the main purposes of the MUSE program from a software perspective is to foster interest in embedded systems using a concrete, real-world experience like building a satellite. Having your code in a satellite that is literally orbiting the earth is a very cool experience and thus the MUSE program lends itself well to an exciting atmosphere for learning about embedded systems.

Methods and Materials

Our CubeSat uses a real-time operating System provided by Texas Instruments known as TI-RTOS. TI-RTOS provides things like an advanced scheduler. We can define what we want our different interrupts to be and what priority they should have and TI-RTOS handles scheduling them.

The use of TI-RTOS eliminates the need to create a lot of basic system functions from scratch.

We also use TI-RTOS Launchpad devices for developing and testing code. These devices have similar processors to the one we will use on our final custom CDH board and make for a good development tool.



ARM Cortex-M4F based MCU TM4C1294 Connected LaunchPad
Picture by Texas Instruments

Progress

Coming into this project a lot of code had been prototyped, but none of it was in the context of TI-RTOS or any sort of timing/cyclical process. A lot of what we want the satellite to be able to do will be happening periodically. This is one of the uses of TI-RTOS as it provides easy-to-use clock functions for scheduling periodic events. One of the things that we have made progress on is translating these drivers to work within the TI-RTOS framework.

We have also prototyped the software for the burn wire that will deploy the radio antennas.

Future Work

Code for most of the major systems in the satellite has been written. The various separate drivers for things like the gyroscope and radio must still be implemented into one cohesive operating system.

Another thing that must happen in order for the final flight software to be ready is for the code to be transferred from its development board state to its final flight state. This involves changing pin mappings and overall making sure that the software is correctly configured for the hardware on which it is running.

Acknowledgements

The work on this project was done under the mentorship of Dr. George Corliss and Dr. Dennis Brylow at Marquette University. My time here was made possible in part by the funding of the National Science Foundation.
NSF AWARD ACI - #1461264