Software Development for MOSES Flight Operations

Roy Smart, Jackson Remington, David Keltgen, Justin Hogan, Charles C. Kankelborg
Physics Department, Montana State University, Bozeman, MT 59717
roytsmart@gmail.com

Abstract

The Multi Order Solar EUV Spectrograph (MOSES) sounding rocket payload is a slitless spectrograph that allows snapshot imaging spectroscopy of the Sun in extreme ultraviolet (EUV) wavelengths (Fox, Kankelborg and Thomas, 2010 Astrophys.J., 719:1132-1143). The MOSES payload relies on an embedded flight computer and FPGA to control the instrument, command exposures from the cameras, save the experimental data, and send the data with additional telemetry back to Earth. Replacing the previous command and data handling system with a low-power solution necessitated the development of new software that could satisfy the same requirements as last mission while improving on the latencies present in the old flight software. The new configuration is still in development, and is undergoing debugging and optimization in prepara tion for launch. The updated software accomplishes our goal of improvement over the flight software implemented in the previous flight, lower latency between subsequent images, and being able to successfully transmit images back to Earth.

First Launch

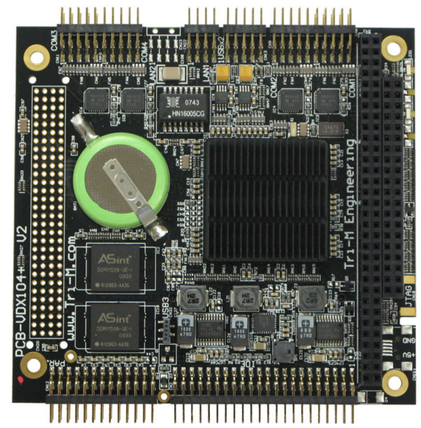
MOSES was first launched on February 8, 2006 on a from White Sands Missile Range on a Black Brant IX sounding rocket. The next launch will include updated optics and electronics and is scheduled for summer 2015.

Data Interface

The new flight computer is designed as a drop-in replacement for the old flight computer, so it must interface with the instrument in the same way. The primary means of controlling the instrument are through **timers**, which are single-ended inputs configured to trigger events (e.g. data start/stop) at predefined times during flight. For redundancy, two other interfaces can also be used to control the instrument, **uplinks** and **HLP packets**.

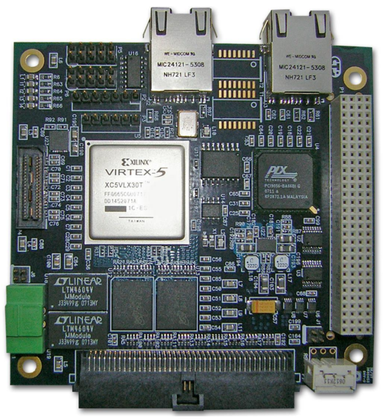
Hardware Solutions

The VDX104+ is the flight computer for the instrument. It provides a GNU/Linux operating system for development, ethernet and RS-232 serial ports for communication, and PCI-104 bus for interfacing with the FPGA. The flight software designed by our team runs on this device.



VDX104 flight computer

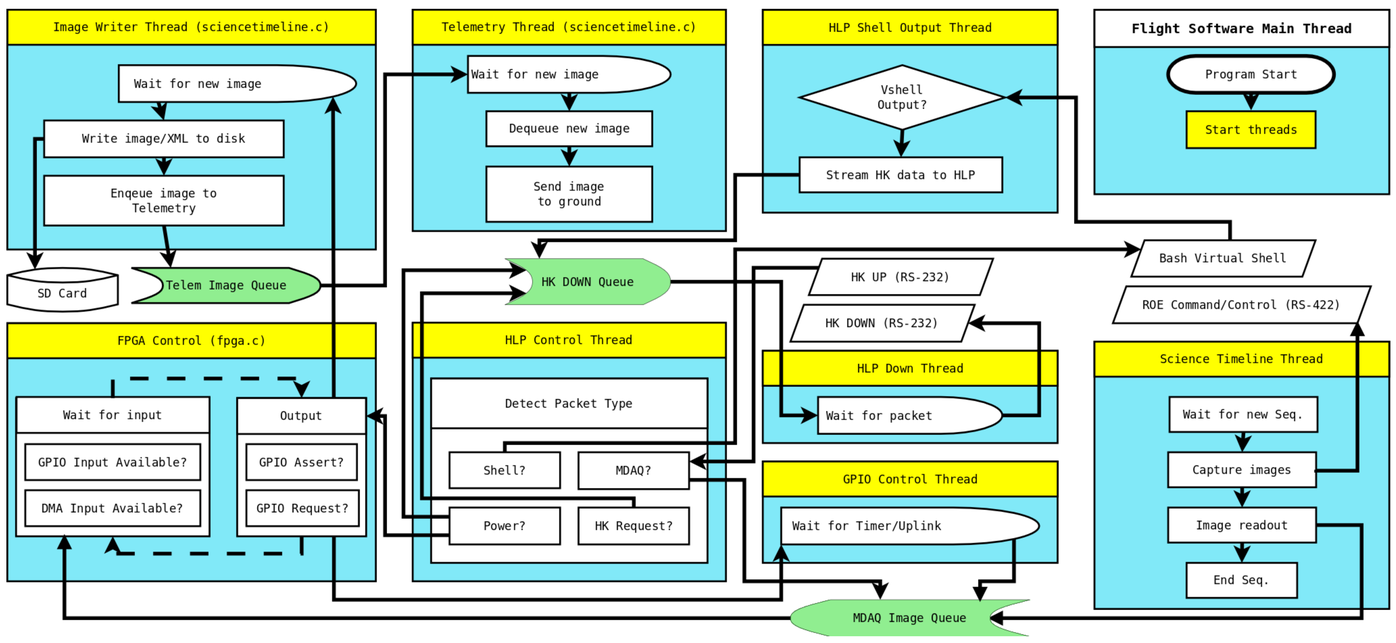
The Virtex5 FPGA captures the 32 Mbit/s parallel data produced by the experiment and various GPIO. It then transfers the data through DMA to the flight computer for transmitting.



Connect Tech FreeForm PCI-104 w/ Virtex5 FPGA

Software Architecture

The flight software relies on a threaded architecture to reliably respond to a variety of inputs, all while the processor is actively capturing data. Several thread-synchronization techniques have been utilized, including mutex-protected queues, Linux Signals, and semaphores.



HK packet on Oscilloscope