Tools for Software Based Validation and Verification of Small Satellites

Matt Grubb
Matthew.d.grubb@nasa.gov

Small Sat Workshop 2016
Overview

What is NOS$^3$?

- A software test bed for small satellites – Currently a Functional Beta
- Based upon STF-1 hardware, but sufficiently generic
- Easily-interfaces to CFS, but CFS not required
- Openly distributed solution Ready-to-Run (RTR)
- A collection of Linux executable and libraries
- Test as you fly

What is it used for?

- FSW early-development – NOS$^3$ provides real-world inputs to FSW
- FSW V&V – Testing FSW, invalid inputs, behavior, stress conditions
- FSW Integration – Used for early-app development and payload team integration
- Mission Planning – Example: power analysis
NOS$^3$ Components

- Virtual Machine – for running NOS$^3$
- NOS Engine Middleware
- Hardware Simulators
- FSW Hardware Abstraction Layer
- Orbit Inview & Power Prediction (OIPP) Tool
- CFS – Flight Software
- 42 – Dynamics Simulation and Visualization
- COSMOS – Commanding & Telemetry
NOS³ Architecture

Ground System Software (COSMOS)

STF-1

Commanding & Telemetry

OS Abstraction Layer (OSAL)

NOS Engine

NOS³ UI & Control

Hardware Models

42

Linux (x86, ARM/Pi)

FreeRTOS (AVR32)

Flight Hardware

Hardware Adapter i2c / SPI

COSMOS

NASA Operational Simulator for Small Satellites
NOS$^3$ Components

NOS Engine Interface -> time, input bytes -> Hardware Model

NOS Engine Interface -> time, input bytes -> Hardware Model

NOS Engine Interface -> time, input bytes -> Hardware Model

Environment Data Provider

FWS
Virtual Machine Auto Generation

• Install *Vagrant* and *VirtualBox*
• Run `vagrant up`
• Developer build tools installed
• Convenience scripts for building/running
• Ready-to-run after unpacking a .tar
NOS Engine Middleware

- ITC developed middleware
- Common server to communicate to all data nodes (CFS, Hardware simulators, Timeticker, Command terminals)
- C API
- I2C, UART and SPI protocols
- Asynchronous and Synchronous
Hardware Simulators

• Modeled based on characteristic data, or manufacturers data specifications
• Currently have modeled
  – Novatel GPS
  – Clyde EPS
  – Honeywell Magnetometer
  – ISISpace Antenna System
  – A3200 support chips (FRAM, Gyro
Flight Software (CFS)

• Open source flight software developed by GSFC

• Includes an OS Abstraction Layer
  – Allows building for flight and NOS³ targets on same machine without source code changes

• Additional Platform-Support-Package (PSP) added to sync CFS time with NOS³
GSFC Open Source Dynamics Simulator

- NOS$^3$ TCP/IP Socket Integration
- Simulation time synchronized with NOS$^3$
- Moving toward closed loop
COSMOS

• Open Source for embedded system commanding and telemetry
• Currently connects to CFS TO_lab
  – Future plan is to have radio simulator to replace TO_lab
• Can be used for operator training, testing table loads to SC, verifying command and telem databases, etc.
COSMOS

NASA Operational Simulator for Small Satellites
Orbit, Inview, and Power Prediction

• Web page: Generated daily by cron job
• TLE Data pulled from http://celestrak.com as obtained from NORAD
• Time Periods (configurable)
  • Yesterday, Today, Future
• Displays
  • Ground station in-views
  • Sunlight and Eclipse times
Orbit, Inview, and Power Prediction (OIPP)
Backup Slides
NOS$^3$ Utilization Example for STF1