A Flight RTEMS OSAL With Runtime Loader Support and Other Enhancements

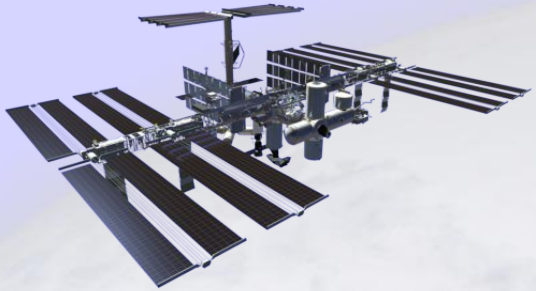
Kevin Gordon, Allen Brown
Odyssey Space Research, LLC

2017 Workshop on Spacecraft Flight Software
Johns Hopkins University Applied Physics Laboratory
Laurel, MD

This presentation does not contain US
export controlled material.

Company Background

- Odyssey Space Research LLC
  - Established 2003
  - Houston TX (Denver CO, Austin TX)
- Core Areas
  - GN&C algorithms, design, analysis, integration, evaluation, test
  - Flight software development, integration, test
  - Simulation development, integration
  - Trajectory / Mission design, analysis
- Current Principal Projects
  - Orion Multi-Purpose Crew Vehicle
  - Commercial Crew
  - Commercial Resupply Services 1 & 2
  - ISS Visiting Vehicle Integration
  - Exploration Mission analysis and design
  - Flight dynamics for mission operations
  - Commercial launch vehicle development
  - Satellites: LEO and beyond



Company cFS Areas

- NASA Support
  - Integrated FSW Simulations
    - Project Gateway
    - Moon Mission
    - ...and others
  - Class A Certification (Human Rating)
  - New cFS platforms (Xenomai, ARINC 653)
  - cFS Voting Architecture
  - Distributed cFS Integration
  - New cFS Apps/Libs for Commands/Telemetry

- Commercial Applications
  - Integrated FSW Simulations
  - DoD test satellite
  - Commercial Science Satellite
  - Orion BFS

- Internal Research & Development
  - Integrated FSW Simulations
  - Human-in-the-Loop Flight Mockup (displays, vehicle & environment sim, cFS FSW)
  - Integrated Development Environments
  - Command and telemetry UI architecture & development
  - Data management

Company cFS Areas (cont'd) and Current FSW Direction

- Company cFS Areas (cont'd)
  - Full-stack development
    - BSP, custom drivers up through custom cFS applications
    - C&DH, GNC, etc.
  - Ground dev/test and operations support
  - cFS Training & Consulting
    - Internal and for commercial customers upon request
    - Training classes and materials
    - Templates, guidelines, HOWTO's

- Right now:
  - FSW development opportunities growing
  - Both Government and Commercial applications
  - Both cFS and custom solutions

Commercial Project and an RTEMS OSAL

- Supporting LASP
  - Laboratory for Atmospheric and Space Physics, University of Colorado Boulder
  - Development of flight software
  - Spacecraft slated to perform a multi-year deep space mission
- Technical Background:
  - Cobham UT700 LEON3FT 32-bit SPARC v8 processor
  - cFS: current release (6.5.0a, released applications, etc.)
    - Strong objective: Minimal changes
  - Custom: BSP, PSP, new cFS mission apps

Commercial Project and an RTEMS OSAL

Technical Background (cont'd)

- RTOS: RTEMS
  - RTEMS 4.10 and prior: Statically built (cFS limitations)
  - RTEMS 4.11+: Contains Run-Time Linker/Loader (RTL)
- OSAL: Currently released version (4.2.0a): Supports RTEMS 4.10 (no RTL)

Objectives

- Space Mission
  - Runtime loading of cFS applications and libraries
  - Project needs a flight-quality OSAL that uses RTEMS RTL
  - Enhance RTEMS OSAL functionality and robustness
- Community
  - Release updated RTEMS OSAL back to the cFS community
  - Engage RTEMS developers with RTL testing and feedback

Flight RTEMS OSAL
Summary

- Functional Improvements
  - Runtime loading fully implemented using RTL
  - Compatibility with RTEMS versions
  - Task names
- Quality Improvements
  - Error detection, mutex timeouts, string operations, debugging info
- Testing Improvements
  - Test BSPs
  - Functional Tests, Unit Tests
- Lessons learned:
  - Module dependencies on static libs vs. exposed symbols
  - *libm* in RTEMS kernel with loadable modules
  - Options: select symbols in kernel or put lib in cFS library

Flight RTEMS OSAL
Functional Improvements

- Dynamic loading implemented using RTL for:
  - Loading and unloading/replacing cFS applications and libraries from ELF .o files at runtime
  - Full Symbol table dump including RTEMS symbols (not just module entry points); and unresolved symbols (bad loads)
  - Module information (code, data, and BSS addresses/sizes)
- OS API updated to support RTEMS 4.11.1 & 4.12 (5.0)
- RTEMS task name length support
  - RTEMS: Has a 4 char task name limit and allows duplicates
  - OSAL/cFS: OS_MAX_API_NAME (~20 chars) supports task-by-name access
  - cFE system would map to: “CFE_”, “CFE_”, “TIME”, “TIME”, …
  - Added mapping of cFE names into unique 4-char RTEMS task names

Flight RTEMS OSAL
Extra Runtime Protection

"But what about…?"

- If unresolved symbols exist after module load?
  - (RTEMS RTL allows this for possible future load resolution)
  - This OSAL: automatically unloads module and optionally shows missing symbols
  - cFE controls module load order
  - Conservative stance: *"no undefined symbols"*
- Changing a loaded module's file?
  - Confirmed with RTEMS RTL
    - No effect on in-memory image...
    - ...When a loaded module's file in filesystem is altered.

Other OSs have ***different behavior!***

Flight RTEMS OSAL
Quality Improvements

- Improved: error detection - all RTEMS return values checked
- Identified and added: missing logic (off-nominal cases)
- Added: missing arg checks
- Changed OSAL internal resource mutex use:
  - Modified take/give locations for better protection and avoiding race conditions
  - Added optional and configurable timeouts when appropriate
  - Return values in OSAL API will indicate “timeout” if it occurs (Checked all cFE and cFS apps return value logic)
  - Concern: deadlock when no timeout can be used
- Replaced: unsafe string ops
- Added: significant debug output with OS_DEBUG_PRINTF

Flight RTEMS OSAL
Test BSP Improvements

- Completed pc-rtems (i386) BSP Implementation
  - Added Makefiles and 2-step link for linking RTEMS symbol table
  - Updated BSP start file and volume table
    - tasking-example & OSAL functional tests
    - qemu-system-i386
- Added sparc-rtems-leon3 BSP
  - Added Makefiles and 2-step link for linking RTEMS symbol table
  - Created all other files needed
    - tasking-example & OSAL functional tests
    - qemu-system-sparc (and hardware)

Flight RTEMS OSAL
Test Improvements

- Enabled FP tasks in functional tests
  - To match flight & cFE
- Functional Test Upgrades
  - Added: Loader API Tests
    - Load, unload, function calls, shared resource access, etc.
    - Corner case tests for memory, resources, repeated load/unload
  - Added: File API Tests
    - OS_ShellOutputToFile() - Supported cFE shell command bugfix testing
    - Missing File API function tests for OS_cp(), OS_mv(), OS_rmfs(), etc.
- Unit tests: Added RTEMS support!
  - Only VxWorks, Linux, and ARINC 653 supported previously

Flight RTEMS OSAL
Tested Platforms

- Current: RTEMS 4.11.1 open-source toolchain
  - RTEMS' sparc-leon3 and i386 OSAL Test BSPs
    - sparc-leon3 on QEMU and hardware and i386 on QEMU
  - UT700 LEON3FT Flight BSP
    - Both on hardware and QEMU
- Functional tests confirmed on:
  - Cobham rcc1.3-rc1 (RTEMS Cross Compiler, RCC)
  - RTEMS 4.11.1 release and 4.11.99 HEAD (Oct 2017)
- In work:
  - RTEMS 4.11.99/4.12/5.x Cobham rcc1.3-rc2
- Final destination:
  - RTEMS release on Cobham release toolchain

Feedback to RTEMS Team

- We have tested the RTL through OSAL functional tests
- Reported three RTL-related issues to RTEMS
  - Memory leak after module unload with dlclose()
  - Infinite loop condition in dlopen() resulting from leftover external reference resolution attempt after dlcose()
  - Module can be unloaded with current dependencies
- We want to help improve the RTL and form a relationship with the RTEMS developer community

Flight RTEMS OSAL Current Status

- Functionally complete
  - No API changes
- Testing on various platforms/toolchains continues
- Public release to the cFS & RTEMS communities
  - Planned, cleared internal review
  - In-work: Release date, site, and logistics

Thank You:

LASP

The cFS Community

Special thanks: Alan Cudmore/NASA, for RTEMS 4.10 OSAL

The RTEMS Community

...Q & A?