NASA/GSFC’s Flight Software Architecture: Core Flight Executive and Core Flight System

Alan Cudmore
Software Engineering Division
NASA/Goddard Space Flight Center
Alan.P.Cudmore@nasa.gov
301-286-5809
• Core Flight System (CFS)
  – A Flight Software Architecture consisting of the cFE Core, CFS Libraries, and CFS Applications

• core Flight Executive (cFE)
  – A set of mission independent, re-usable, core flight software services and operating environment
SAMPEX (launched 8/92)

SWAS (launched 12/98)

TRACE (launched 3/98)

WIRE (launched 2/99)

SMEX-Lite

Triana (waiting for launch)

Swift BAT (12/04)

XTE (launched 12/95)

TRMM (launched 11/97)

IceSat GLAS (01/03)

MAP (launched 06/01)

ST-5 (5/06)

SDO (2007)

GPM, MMS, RBSP, LADEE, Morpheus, Etc..

JWST ISIM (2011)

SDO (2007)

LRO (2009)

Core FSW Executive

Core FSW System
# cFE/CFS Customers

<table>
<thead>
<tr>
<th>Organization</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASA / GSFC</td>
<td>LRO Spacecraft</td>
</tr>
<tr>
<td>NASA / GSFC</td>
<td>DSILCAS Project</td>
</tr>
<tr>
<td>NASA / GSFC</td>
<td>MMS Spacecraft</td>
</tr>
<tr>
<td>NASA / GSFC</td>
<td>GPM Spacecraft</td>
</tr>
<tr>
<td>JHU / APL</td>
<td>RBSP Spacecraft</td>
</tr>
<tr>
<td>JHU / APL</td>
<td>Solar Probe Plus Spacecraft</td>
</tr>
<tr>
<td>NASA / ARC</td>
<td>LADEE Spacecraft</td>
</tr>
<tr>
<td>NASA / JSC</td>
<td>Project Morpheus</td>
</tr>
<tr>
<td>NASA / GSFC</td>
<td>ATLAS Instrument Simulator</td>
</tr>
<tr>
<td>NASA/GSFC and JHU/APL</td>
<td>Memory Protection IRAD</td>
</tr>
<tr>
<td>NASA/GSFC and JHU/APL</td>
<td>Multi-Core IRAD</td>
</tr>
<tr>
<td>NASA/GSFC</td>
<td>Virtualization IRAD</td>
</tr>
<tr>
<td>NASA/GSFC</td>
<td>Various test beds and experiments</td>
</tr>
</tbody>
</table>
CFS Flight Software Layers

- CFS App 1
- CFS App 2
- CFS App N
- Mission App 1
- Mission App 2
- Mission App N

Mission and CFS Application Layer

- CFS Library
- Mission Library

Mission and CFS Library Layer

- cFE Core

CFE Core Layer

- OS Abstraction Layer
- cFE Platform Support Package

Abstraction Library Layer

- Real Time OS
- Board Support Package
- PROM Boot FSW

RTOS / BOOT Layer

Legend:
- Mission Developed
- GSFC Developed
- 3rd Party
• **PROM Boot Software**
  – PROM resident software that does early initialization and bootstraps the Operating System
  – Provides ground based EEPROM/Flash loader
  – Keep it as simple as possible to minimize PROM changes
  – Commonly used Boot Software
    • RAD750 – BAE SUROM
    • Coldfire – Custom GSFC developed
    • LEON3 – uBoot – or Gaisler MKPROM

• **Real Time Operating System**
  • Pre-emptive priority based multi-tasking
  • Message Queues, Semaphores
  • Interrupt handling, Exception Handling
  • File systems, and shell
  • Supported Real Time Operating Systems
    • vxWorks
    • RTEMS
    • Linux ( Not real time, but used for desktop development )
CFS Flight Software Layers

- CFS App 1
- CFS App 2
- CFS App N
- Mission App 1
- Mission App 2
- Mission App N

Mission and CFS Application Layer

- CFS Library
- Mission Library

Mission and CFS Library Layer

- cFE Core

CFE Core Layer

- OS Abstraction Layer
- cFE Platform Support Package

Abstraction Library Layer

Real Time OS

Board Support Package

RTOS / BOOT Layer

PROM Boot FSW

Legend:
- Mission Developed
- GSFC Developed
- 3rd Party

3rd Party

NASA
Abstraction Library Layer - OSAL

- The Operating System Abstraction layer (OSAL) is a small software library that isolates our Flight Software from the Real Time Operating System.
- With the OS Abstraction Layer, flight software such as the Core Flight Executive can run on several operating systems without modification.
- Current Implementations of the OSAL include:
  - RTEMS - Used on the RHCF 5208 Coldfire CPU
  - vxWorks - Used on RAD750
  - Linux / x86 - Used to run software on Desktop PC with Linux

Open Source release at: http://osal.sf.net
Abstraction Library Layer –
Platform Support Package

• **Platform Support Package (PSP)**
  – A Platform Support Package is all of the software that is needed to adapt the cFE Core to a particular OS and Processor Card.
  – A Platform Support Package also includes all of the toolchain specific make rules and options
  – Each mission is expected to customize a Platform Support Package

• **Functions included**
  – Startup code
  – EEPROM rand Memory read, write, copy, and protection functions
  – Processor card reset functions
  – Exception handler functions
  – Timer functions

• **Common PSPs**
  – Desktop Linux for prototyping
  – Power PC MCP750 / RAD750 – vxWorks 6.x
  – Coldfire - RTEMS
cFE Core - Overview

• A set of mission independent, re-usable, core flight software services and operating environment
  – Provides standardized Application Programmer Interfaces (API)
  – Supports and hosts flight software applications
  – Applications can be added and removed at run-time (eases system integration and FSW maintenance)
  – Supports software development for on-board FSW, desktop FSW development and simulators
  – Supports a variety of hardware platforms
  – Contains platform and mission configuration parameters that are used to tailor the cFE for a specific platform and mission.
cFE Core - Executive Services (ES)

- Manages the cFE Startup
- Provides ability to start, restart and delete cFE Applications
- Manages a Critical Data Store which can be used to preserve data (except in the case of a power-on reset)
- Provides ability to load shared libraries
- Logs information related to resets and exceptions
- Manages a system log for capturing information and errors
- Provides Performance Analysis support
• Provides a portable inter-application message service
• Routes messages to all applications that have subscribed to the message.
  – Subscriptions are done at application startup
  – Message routing can be added/removed at runtime
• Reports errors detected during the transferring of messages
• Outputs Statistics Packet and the Routing Information when commanded
cFE Core - Event Services (EVS)

• Provides an interface for sending asynchronous informational/error messages telemetry to ground
  – Provides a processor unique software bus event message containing the processor ID, Application ID, Event ID, timestamp, and the request-specified event data (text string including parameters)

• Provides an interface for filtering event messages
• Provides an interface for registering an application’s event filter masks, types, and type enable status
• Provides an interface for un-registering an application from using event services
• Provides an interface for enabling/disabling an application’s event filtering
• <optional> Provide an interface for logging event into a local event log
cFE Core - TIME Services

- Provides a user interface for correlation of spacecraft time to the ground reference time (epoch)
- Provides calculation of spacecraft time, derived from mission elapsed time (MET), a spacecraft time correlation factor (STCF), and optionally, leap seconds
- Provides a functional API for cFE applications to query the time
- Distributes of a “time at the tone” command packet, containing the correct time at the moment of the 1Hz tone signal
- Distributes of a “1Hz wakeup” command packet
- Forwards tone and time-at-the-tone packets
cFE Core - Table Services

- Manages all CFS table images
- Provides an API to simplify Table Management
- Table Registry is populated at run-time eliminating cross coupling of Applications with flight executive at compile time
- Performs table updates synchronously with the Application that owns the table to ensure table data integrity
- Shares tables between Applications
- Allows Non-Blocking Table updates in Interrupt Service Routines
- Provides a common ground/user interface to all tables
CFS Flight Software Layers

- CFS App 1
- CFS App 2
- CFS App N

- CFS Library
- Mission Library

- cFE Core

- OS Abstraction Layer
  - cFE Platform Support Package

- Real Time OS
  - Board Support Package

- PROM Boot FSW

Mission and CFS Application Layer

Mission and CFS Library Layer

CFE Core Layer

Abstraction Library Layer

RTOS / BOOT Layer

- Mission Developed
- GSFC Developed
- 3rd Party
**Health and Safety App / Housekeeping App**

- **Health and Safety App**
  - Monitor Applications
    - Detect when defined applications are not running and take a defined action
  - Monitor Events
    - Detect table defined events and take a table defined action
  - Manage Watchdog
    - Initialize and periodically service the watchdog
    - Withhold periodic servicing of the watchdog if certain conditions are not met
  - Manage App Execution Counters
    - Report execution counters for a table defined list of Application Tasks

- **Housekeeping App**
  - Build combined telemetry messages containing data from applications
  - Notify the ground when expected data is not received
Data Storage App / File Manager App

• Data Storage App
  • Stores Software Bus messages (packets) to data storage files.
  • Filters packets according to packet filter table definition
  • Stores packets in files according to destination table definition

• File Manager App
  • Manages onboard files
    • Copy, Move, Rename, Delete, Close, Decompress, and Concatenate files providing file information and open file listings
  • Manages onboard directories
    • Create, delete, and providing directory listings
  • Device free space reporting
Limit Checker App / Memory Dwell App

• Limit Checker App
  – Monitors Table Driven Telemetry Watchpoints
    • Each watchpoint compares a telemetry data value with a constant threshold value
  – Evaluates Table Driven Actionpoints
    • Each action point analyzes the results of one (or more) watchpoints

• Memory Dwell App
  – Samples data at any processor address
  - Augments telemetry stream provided during development and debugging
  – Dwell Packet Streams are Specified by Dwell Tables
  – Up to 16 active Dwell Tables
  – Dwell Tables can be populated either by Table Loads or via Jam Commands
Scheduler App / Stored Command App

• **Scheduler App**
  – Operates a Time Division Multiplexed (TDM) schedule of Applications via Software Bus Messages
    • Synchronized to external Major Frame (typically 1 Hz) signal
    • Each Major Frame split into a platform configuration number of smaller slots (typically 100 slots of 10 milliseconds each)
    • Each slot can contain a platform defined number of software bus messages (typically 5 messages) that can be issued within that slot

• **Stored Command App**
  – Executes preloaded command sequences at predetermined absolute or relative time intervals.
  – Supports Absolute Time Tagged Sequences
  – Supports Relative Time Tagged Sequences
Checksum App / Memory Manager App

• **Checksum App**
  – Monitors the static code/data specified by the users and reports all checksum miscompares as errors.
  – CS will be scheduled to wakeup on a 1Hz schedule
  – CS will be byte-limited per cycle to prevent CPU hogging

• **Memory Manager App**
  – Performs Memory Read and Write (Peek and Poke) Operations
  – Performs Memory Load and Dump Operations
  – Performs Diagnostic Operations
  – Provides Optional Support for Symbolic Addressing
Other CFS Apps

- **CFDP App**
  - Implements flight portion of CCSDS CFDP Protocol

- **Command Uplink App**
  - Implements flight portion of CCSDS Command uplink
  - Usually mission specific

- **Telemetry Output App**
  - CCSDS Telemetry downlink
  - Usually mission specific

- **Memory Scrub App**
  - Memory Scrub – Scrubs SDRAM check bits
  - Usually mission specific

- **CI Lab & TO Lab**
  - UDP sockets based uplink and downlink apps for lab testing
The CFS has a complete development environment that is designed to manage:

- Builds of images for multiple processors
- Multiple processor architectures
- Multiple operating systems
- Different application loads on each processor
- As little duplication of code as possible
What’s next for the cFE/CFS?

- cFE Core 6.1 Open Source Release
- cFE 6.2 Release
- Ongoing CFS Application Update Releases
- Updated Platform Support Packages
- Research and Development
  - cFE and Memory Protection
    - FY 2010, 2011 IRAD
    - Can run multiple cFE/CFS systems on a single vxWorks OS using RTPs
  - cFE on Multi-Core systems
    - Would like to research running cFE on Multi-Core flight processors
  - Virtualization platform
    - Would like to research running cFE on a hypervisor / VM system