The Core Flight System (cFS) Community: Providing Low Cost Solutions for Small Spacecraft

David McComas, Jonathan Wilmot, Alan Cudmore

NASA Goddard Space Flight Center

August 9, 2016
Morpheus – An Early cFS Adopter

14 Months between receipt of cFS and successful tethered flight test

“The cFS… It just worked.”
– Morpheus Software Lead
A NASA Controlled Product Line

DOD and US industry
• Potential for standardization though the CCSDS and the Space Universal MOdular Architecture (SUMO) team sponsored by Office of the Director of National Intelligence

Kirtland AFB – Onboard Autonomous Planning System

JPL – Evaluating architecture for robotic missions and ESTO missions

Commercial - Moon Express (Lunar X-Prize)

JSC - Used Successfully on Morpheus. Using on AES projects, Habitats, Waypoint, Certified for Class A (human rated).

GRC – CPST and Advanced suit

APL - RBSP. Proposing use on Solar Probe, DoD programs.

GSFC: LRO, MMS, GPM, NICER, OPIS and many others.

ARC - LADEE

KSC - Evaluating for AES, sounding rockets and UAV’s

European Space Research and Technology Centre

JAXA’s Engineering Digital Innovation Center
Next generation software architecture research

Korea Aerospace Research Institute
Lunar program

Monthly

3
cFS Heritage

SAMPEX (8/92)
SWAS (12/98)
TRACE (3/98)
WIRE (2/99)
SMEX-Lite
Triana/DSCOVR
Swift BAT (12/04)

XTE (12/95)
TRMM (11/97)

IceSat GLAS (01/03)

MAP (06/01)

ST-5 (4/06)

SDO (2/10)

LRO (6/09)

GPM (2/14)

JWST ISIM (2018)
Lifecycle Artifact Reuse

Project Requirements

FSW Subsystem Requirements

Detailed Requirements

Build Test (Tester)

System Test (Tester)

Integration Test (Developer)

Unit (Developer)

Code

CFS Repository

CFS Repository
cFS Layer Architecture

- Application Layer
  - Application Library
  - Mission Library

- Executive Services Layer
  - cFE API
  - cFE Core

- Platform Abstraction Layer
  - OS Abstraction API
  - Platform Support Package API
  - OS Abstractions (Linux, RTEMS, VxWorks)
  - cFE Platform Support Packages

- Open Source
- Mission Specific
## OS Abstraction Layer Platforms

<table>
<thead>
<tr>
<th>Operating System</th>
<th>OSAL Version</th>
<th>Status</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSIX/Linux</td>
<td>4.1.1</td>
<td>Production</td>
<td>Desktop Dev. use CentOS 6.x/Ubuntu 14.04 32 bit</td>
</tr>
<tr>
<td>RTEMS</td>
<td>4.1.1</td>
<td>Production</td>
<td>Flying on MMS Mission RTEMS 4.10/Coldfire</td>
</tr>
<tr>
<td>VxWorks</td>
<td>4.1.1</td>
<td>Production</td>
<td>Flying on GPM Mission vxWorks 6.4/PowerPC</td>
</tr>
<tr>
<td>FreeRTOS</td>
<td>4.2.x</td>
<td>In Dev.</td>
<td>GSFC Dellingr CubeSat Mission FreeRTOS/Arm</td>
</tr>
<tr>
<td>VxWorks 6.x SMP</td>
<td>4.3.x</td>
<td>In Dev.</td>
<td>vxWorks 6.7 LEON3 Dual Core</td>
</tr>
<tr>
<td>ARINC653</td>
<td>4.3.x</td>
<td>In Dev.</td>
<td>Green Hills Integrity OS</td>
</tr>
<tr>
<td>RTEMS 4.12+SMP</td>
<td>Future</td>
<td>Future</td>
<td>Future Release</td>
</tr>
<tr>
<td>Xenomai Linux</td>
<td>Future</td>
<td>Future</td>
<td>Future Release</td>
</tr>
</tbody>
</table>

The table above lists the operating systems and their versions, along with their status and target platforms. Each entry indicates whether the system is in production, development, or future planning.
<table>
<thead>
<tr>
<th>Board/Platform</th>
<th>OSAL Operating System</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CentOS/Ubuntu Linux Desktop</td>
<td>POSIX/Linux</td>
<td>Used on a balloon mission</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Common initial development/test environment</td>
</tr>
<tr>
<td>MMS Custom C&amp;DH Coldfire</td>
<td>RTEMS</td>
<td>1 year in flight on MMS Mission</td>
</tr>
<tr>
<td>GPM RAD750</td>
<td>VxWorks</td>
<td>2 years in flight on GPM Mission</td>
</tr>
<tr>
<td>Gomspace Nanomind ARM CubeSat</td>
<td>FreeRTOS</td>
<td>Under development for GSFC Dellingr CubeSat Mission</td>
</tr>
<tr>
<td>GSFC MUSTANG Dual Core LEON3</td>
<td>VxWorks SMP</td>
<td>Under development for GSFC MUSTANG Dual Core LEON3 architecture</td>
</tr>
</tbody>
</table>
The Power of Community

1993 - Microsoft releases digital encyclopedia called Encarta

2001 - Wikipedia launched

2009 - Microsoft terminates Encarta
core Flight System (cFS)
cfs.gsfc.nasa.gov
Coming soon…
coreflightsystem.org
• News
• Email sign up
• Documentation
• Discussion forums
• Links to software
• Submit trouble tickets
• Software Exchange
• Collaborative projects
cFS Starter Kits

- NASA Operational Simulator for Small Satellites (NASA’s Independent V&V Facility)
  - Ground System: Ball Aerospace’s COSMOS
  - Dynamic Simulator: NASA Goddard’s 42

- The Hammers Company
  - Ground System: Hammer’s Integrated Test and Operations System (ITOS)
  - Dynamic Simulator: Hammer’s VIRTUALSAT®

- NASA Johnson Space Center
  - Ground System: Hammer’s ITOS
  - Dynamic Simulator: NASA JSC Trick
Future Directions

• Model-Based Application Development
  - Simulink Interface Layer (SIL) allows cFS applications to be generated from Matlab Simulink models

• End-user certification

• Hardware vendor supplied device drivers & verification test scripts

• Embedded software school curriculums using cFS kits
Summary

- The cFS is an open source embedded software solution
- Opening our community: coreflightsystem.org
- Starter kits simplify adoption
- Enhancing component architecture for device plug-ins
- FSW Workshop, December 13-15, 2016 in Pasadena, CA
  - Hosted by NASA JPL, Aerospace Corporation, and Johns Hopkins University Applied Physics Laboratory
  - flightsoftware.jhuapl.edu
  - cFS Workshop on December 12th, 2016

Democratizing Space
Backup Slides
State of the Community Communication

• Mailing Lists
  - cfs-community@lists.nasa.gov
    • Contains all members
  - cfs-community-ccb@lists.nasa.gov
    • CCB members

• Public Websites
  - https://cfs.gsfc.nasa.gov/
    • General information and links to all open source code and documents on Sourceforge
  - https://sourceforge.net/projects/xxx
    • Multiple projects for different cFS components

• Restricted access (requires NDC account)
  - https://nsckn.nasa.gov/Community
    • NESC hosted server containing discussion forums, documents, meeting notes…
    • Approved for ITAR and Sensitive But Unclassified (SBU) material
  - https://babelfish.arc.nasa.gov/
    • ARC hosted server used for inter-center collaboration
    • Git andTrac used for source code configuration management and change requests
    • Not approved for ITAR material
Questions? Contact:

- CFS 582 Branch Management
  - David McComas - NASA GSFC/Code 582 Flight Software Branch
    - david.c.mccomas@nasa.gov
    - 301.286.9038

- CFS Lead Architect
  - Jonathan Wilmot - NASA GSFC/Code 582 Flight Software Branch
    - jonathan.j.wilmot@nasa.gov
    - 301.286.2623

- CFS Product Development Lead
  - Susie Strege - NASA GSFC/Code 582 Flight Software Branch
    - susie.strege@nasa.gov
    - 301.286.3829

- OSAL Product Development Lead
  - Alan Cudmore - NASA GSFC/Code 582 Flight Software Branch
    - alan.p.cudmore@nasa.gov
    - 301.286.5809
Example Mission Code Metrics
Global Precipitation Measurement (GPM)

- Noteworthy items
  + cFE was very reliable and stable
  + Easy rapid prototyping with heritage code that was cFE compliant
  + Layered architecture has allowed COTS lab to be maintained through all builds
    - Addition of PSP changed build infrastructure midstream

- Lines of Code Percentages:

<table>
<thead>
<tr>
<th>Source</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAE</td>
<td>0.3</td>
</tr>
<tr>
<td>EEFS</td>
<td>1.7</td>
</tr>
<tr>
<td>OSAL</td>
<td>2.1</td>
</tr>
<tr>
<td>PSP</td>
<td>1.0</td>
</tr>
<tr>
<td>cFE</td>
<td>12.4</td>
</tr>
<tr>
<td>GNC Library</td>
<td>1.6</td>
</tr>
<tr>
<td>CFS Applications</td>
<td>23.5</td>
</tr>
<tr>
<td>Heritage Clone &amp; Own</td>
<td>38.9</td>
</tr>
<tr>
<td>New Source</td>
<td>18.5</td>
</tr>
</tbody>
</table>
## cFS Metrics

<table>
<thead>
<tr>
<th>cFE/ App</th>
<th>Logical Lines of Code (non-table)</th>
<th>Config. Parameters</th>
<th>EEPROM (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cFE</td>
<td>12,930</td>
<td>General: 17</td>
<td>341,561</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Executive Service: 46</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event Service: 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Bus: 29</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Table Service: 10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time Service: 32</td>
<td></td>
</tr>
<tr>
<td>CFDP</td>
<td>8,559</td>
<td>33</td>
<td>85,812</td>
</tr>
<tr>
<td>Checksum</td>
<td>2,873</td>
<td>15</td>
<td>35,242</td>
</tr>
<tr>
<td>Data Storage</td>
<td>2,429</td>
<td>27</td>
<td>40,523</td>
</tr>
<tr>
<td>File Manager</td>
<td>1,853</td>
<td>22</td>
<td>16,272</td>
</tr>
<tr>
<td>Health &amp; Safety</td>
<td>1,531</td>
<td>45</td>
<td>15071</td>
</tr>
<tr>
<td>House-Keeping</td>
<td>575</td>
<td>8</td>
<td>8,059</td>
</tr>
<tr>
<td>Limit Checker</td>
<td>2,074</td>
<td>13</td>
<td>31,026</td>
</tr>
<tr>
<td>Memory Dwell</td>
<td>1,035</td>
<td>8</td>
<td>8,617</td>
</tr>
<tr>
<td>Memory Manager</td>
<td>1,958</td>
<td>25</td>
<td>15,840</td>
</tr>
<tr>
<td>Scheduler</td>
<td>1,164</td>
<td>19</td>
<td>35,809</td>
</tr>
<tr>
<td>Stored Command</td>
<td>2,314</td>
<td>26</td>
<td>104,960</td>
</tr>
<tr>
<td></td>
<td>(124 command sequences)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# cFS Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFDP</td>
<td>Transfers/receives file data to/from the ground</td>
</tr>
<tr>
<td>Checksum</td>
<td>Performs data integrity checking of memory, tables and files</td>
</tr>
<tr>
<td>Command Ingest Lab</td>
<td>Accepts CCSDS telecommand packets over a UDP/IP port</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Records housekeeping, engineering and science data onboard for downlink</td>
</tr>
<tr>
<td>File Manager</td>
<td>Interfaces to the ground for managing files</td>
</tr>
<tr>
<td>Housekeeping</td>
<td>Collects and re-packages telemetry from other applications.</td>
</tr>
<tr>
<td>Health and Safety</td>
<td>Ensures that critical tasks check-in, services watchdog, detects CPU hogging, and calculates CPU utilization</td>
</tr>
<tr>
<td>Limit Checker</td>
<td>Provides the capability to monitor values and take action when exceed threshold</td>
</tr>
<tr>
<td>Memory Dwell</td>
<td>Allows ground to telemeter the contents of memory locations. Useful for debugging</td>
</tr>
<tr>
<td>Memory Manager</td>
<td>Provides the ability to load and dump memory.</td>
</tr>
<tr>
<td>Software Bus Network</td>
<td>Passes Software Bus messages over Ethernet</td>
</tr>
<tr>
<td>Scheduler</td>
<td>Schedules onboard activities via (e.g. HK requests)</td>
</tr>
<tr>
<td>Scheduler Lab</td>
<td>Simple activity scheduler with a one second resolution</td>
</tr>
<tr>
<td>Stored Command</td>
<td>Onboard Commands Sequencer (absolute and relative).</td>
</tr>
<tr>
<td>Telemetry Output Lab</td>
<td>Sends CCSDS telemetry packets over a UDP/IP port</td>
</tr>
</tbody>
</table>
cFS Community Purpose

• Advance the creation, evolution, promotion, and support of a NASA Class B flight software system
  - Important we stay focused on our domain

• Cultivate both an open source community and an ecosystem of complementary products, capabilities, and services.
  - All inclusive in terms of organizations
  - No constraints on complementary products
cFS Community

NASA Product Management

Ground Systems
- ITOS
- ASIST
- COSMOS
- Unit Test
- Performance
- EDS
- Table

Development Tools & APIs
- cFE Apps
- Core Lab Apps
- App Library
- Apps
- App Libraries

Applications

Executive
- cFE API

Platform Abstraction
- OS Abstraction API
- OS Abstract
- OS Abstract
- Platform Support Package API
- PSPs
- PSPs

Members
- Government
- International
- Academia
- Industry

Contributors: Bug Fixes, Verification

Users: Feedback, Feature Requests, Bug Reports

User Support

Contributors: Bug Fixes, Verification

NASA Controlled Assets

Member Sponsored Assets

External to cFS
Electronic Data Sheets

- Designer and/or tools
  - Flight SW components
  - Models
  - Simulators

- Component (software) EDS

- Device EDS

- Test/Simulation tool/consoles
  - Ground System Database(s)
  - Test procedures
  - Operations procedures

- Ground Systems

- Page Displays
Model-based Application Development

Iterate Early and Often

Verification

Requirements
- Design/Algorithm Development
- Flight Software Modeling
- Vehicle & Environment Modeling
- Workstation Simulations (e.g., Simulink)
- Hand Developed Apps
- Code Generation
- Integrated Tests
  - Processor-in-the-Loop
  - Hardware-in-the-Loop
- Unit Tests

Analysis

Heritage Models
- Develop Models of FSW, Vehicle, and Environment
- Automatically generate High-Level Control Software
- Integrate with hand-written and heritage software.
- Iterate while increasing fidelity of tests – Workstation Sim (WSIM), Processor-In-The-Loop (PIL), Hardware-in-the-Loop (HIL)
- Automated self-documenting tests providing traceability to requirements
Automated Testing

1. **Generic Setup Function (Initialization)**
   - Set Up Test Case
   - Execute Function Under Test
   - Verify Results
   - Generic Teardown Function