Growing Flight Software Hands-On Experience: CFS and Educational Outreach

Allen Brown, Mathew Benson
Odyssey Space Research, LLC

2015 Workshop on Spacecraft Flight Software
JHU APL - Laurel MD

This presentation does not contain US export controlled material.
Company Background

- Odyssey Space Research LLC
  - Established 2003

- Core Areas
  - GN&C design, analysis, integration, evaluation, test
  - Flight software development, test
  - Simulation development, integration

- Current Principal Projects
  - SpaceX Dragon ISS Resupply/Commercial Crew
  - Orbital ATK Cygnus ISS Resupply
  - Orion Multi-Purpose Crew Vehicle
  - Visiting Vehicle Integration with ISS
  - Exploration Mission analysis and design
  - Flight Dynamics for Mission Operations
Company Background

- LM Orion (MPCV) Flight Software
  - Development, Integration, Verification, Test
  - GNC, ECLSS, EPS, BFS...
- FSW Safety IV&V
  - Dragon, Cygnus
- Core Flight System (CFS)
  - Internal Research & Development
    - Unmanned Aircraft Systems (UAS) Platforms
    - Integrated FSW Simulations
    - Integrated Development Environment
  - NASA
    - Class A Certification (Human Rating)
    - Commanding/Telemetry Upgrades
    - New CFS platforms (Xenomai, ARINC 653)
    - CFS Voting Architecture
- Other S/W Projects with Outreach
  - SpaceLab for iOS

2015 Workshop on Spacecraft Flight Software (FSW-15)
Flight Software Education

• Flight Software vs. “software that flies”
  - A mindset
  - Touches all aspects: flight vehicle, ground systems, communications, operations, development, ...

• Why?
  - $$ Cost? Yes and no... $$
  - Remote vehicle
  - Communication latency, outages
  - Environment stresses → failures
  - Loss of mission, loss of vehicle (and/or loss of crew!)
Flight Software Education

• Usually on the job...
  – ...Education before a career starts?
  – Benefits

• A great way to learn...
  … learn by doing.
  Safely? No/little cost for failure?

• Now, a new opportunity...
Purpose

• Supporting NASA JSC Engineering Directorate
• Create a set of course material
  – University-level
  – Assume software development exposure
  – “Learn by doing”
• Illustrate use of (CFS)
  – Now open source (NOSA)
  – Focus on CFS application development, mods
  – Include and provide references on CFS
    • Including layers: cFE, OSAL, PSP
Purpose

• Illustrate a FSW development process
  – “Test like you fly” & “Fly like you test”
  – CFS with integrated Trick Simulation Environment
    • Now open source (NOSA)
  – Use the same command/telemetry tools between test and flight

• Package it all together:
  – Interesting, usable, representative, available.
Approach

- Create a single Virtual Machine (VM) Linux image
- Use existing open source products
- Craft “hands-on” tutorials
  - Include new outreach CFS applications
  - Include new outreach sim software models
- User:
  - Laptop with WiFi
  - (Parrot SA) AR.Drone 2.0 quadcopter
Components

- **Linux VM Image**
  - VirtualBox VM software (open source)
  - RHEL → CentOS (open source) for outreach
    - x86_64 guest image
  - CFS (framework and apps) for Linux/POSIX
    - open source: SourceForge
  - Generic Drone Sim (Trick)
    - open source: GitHub
  - x86_64, x86_32 Linux toolchain (open source)
  - ARM cross-compile toolchain (open source)
Components

- Eclipse development environment (in VM) (open source)
  - Eclipse CDT (C/C++ development)
    - editing, debugging, building, profiling, etc.
    - CFS and Sim development
  - Displays (via Eclipse plugins) (are/will be open source)
    - Commanding and telemetry
    - Drone flight control
- Documentation: Trac (in VM) (open source)
  - Wiki, issue tracking – default Firefox browser
  - All course materials as wiki pages
  - User-modifiable
Components

- AR.Drone 2.0 (COTS)
  - 1 GHz ARM Cortex A8, Linux
  - WiFi a/b/n
  - 2 cameras, gyro, accelerometer
  - magnetometer
  - barometric pressure sensor
  - ultrasound altitude sensor

- NOT manipulating GNC algorithms
  - Respecting Parrot, SA IP
  - Utilizing existing (unmodified) UDP sockets (described in SDK)
    - CFS interaction, display interaction

- NOT endorsed by manufacturer
  - Running CFS (off and) ON the drone
    ...Voiding your warranty there...
Current Status

- VM: created, using for dev work
  - VM config with bridged host WiFi adaptor to drone
- CFS drone apps running in VM Guest
  - Running in prototype Trick drone sim
  - Running with “drone in-the-loop” (HWIL)
- CFS drone apps running ON drone
- Basic flight via CFS interfaces
- Prototyping Displays
  - Includes Odyssey IRAD efforts
Prototype Display

This presentation does not contain US export controlled material.

2015 Workshop on Spacecraft Flight Software (FSW-15)
Current Status

- Building tutorial outline
- Capturing necessary support docs
  - Host, drone installation & configuration
  - VM setup
  - Connecting to the AR.Drone
  - CFS build and deploy instructions
  - Debugging CFS on the AR.Drone with Eclipse
  - etc.
Current Status

- Prototyping: VM development process
  - Repeatable packaging process (CI, CD)
    - Creation, provisioning OS
    - Provisioning/updating outreach content
    - Packaging and testing release image
  - Currently a Git, Vagrant, VirtualBox, Jenkins recipe
Future

- Complete “dev/test” environment
  - Sim, displays, CFS apps
- Complete course materials
- Refine packaging process, user testing

- Release planned for 2016 (?)
Special Thanks

- Dr. Lorraine Prokop, NASA JSC
- David McComas, NASA GSFC
- CFS and Trick software supporters
Thank you.

Q & A