Nuttx in space
a pocketQube satellite application

PRESENTED BY:
Fabio Balzano
Lead Engineer at Stara corp.
Today's Topics

· Overview of our platform
· Why Nuttx?
· Nuttx duties
· Challenges
· Our building system
Our platform
Overview

- PocketQube standard form factor
- “Heterogeneous” system
- Automotive standards for electronic circuits
- Innovative
- Cheapest as possible
- Easy to assembly and efficient to test
Overview

- Solar panels
- Energy harvesting
- Power Regulators
  - SBC NXP MC35FS6500 Fail-safe state machine
- LiOn batteries
- Super capacitors

- ADCS
  - EMS
    - NXP IMXRT1062 Nuttx OS
  - “FOTA” MCU

- RF - services
  - OBC Arch Linux
  - RF - Telemetry

- Deploy
Why Nuttx?
Why Nuttx

We evaluated:

- FreeRTOS
- TizenRT
- ChebiOS
- briefly others (Zephyr)
Why Nuttx

Nuttx:

- Support many platforms (more developers)
- Protected build
- POSIX
- Proven reliability
- One of the most active opensource RTOS
- Community support is good
Nuttx duties
2 phases:

1. After deploy
2. In mission duties
1. After satellite deploy:

- Timing the panels deploy
- Enable the power supply buses
- Enable OBC with Linux
- Enable RF modules
- Send telemetry with alive signal
2. During the mission:

- Monitor current consumptions
- ADCS: sensors reading
- ADCS: drive the actuators
- Temperature monitoring
- Watchdog for the Linux OBC
- Send telemetry
Challenges
Challenges

- Correct prioritization of tasks
- Safe userland in protected environment
- Lowest power consumption
- Develop new drivers
Challenges

Correct prioritization of tasks:

- The case of the MARS pathfinder spacecraft
  https://www.rapitasystems.com/blog/what-really-happened-to-the-software-on-the-mars-pathfinder-spacecraft
Challenges

Protected build:

- Separate the userland from the kernel space
- → “FOTA”
  - Increase recovery capabilities
  - Continuous optimization
    → more robust runtime
Lowest possible power consumption:

- Critical feature in space missions
  - Less weight $\rightarrow$ cheaper to launch
  - Bigger power budget $\rightarrow$ Increase capabilities of the spacecraft
  - Safer
Challenges

Development of new drivers:

- submit patches for the approval
  - to Nuttx mainstream
  - Expose new code to the Nuttx community
  - → A lot more of testing
Our testing system
Testing system
Testing system

Buildbot features:
https://buildbot.net

- Python based
- GIT friendly
- Adding custom script is easy
- De-localization of builds
- Logic organized in recipes
How we use Buildbot:

- Not CI!
- Great for automation
- Building test for the kernel space + userland
- Distribution of releases
Testing system

Buildbot:

→ automate the building
→ automate the hardware testing

- Track differences with hardware measurements
- It alerts if the procedure goes wrong
Testing system

Hardware tests:
- Power consumption
- Functions safety
- Actuators drivers
- Timing

Internet → Buildbot daemon → Postgresql DB → Flash via USB → Testing board → Analog + Digital
Nuttx is at the heart of our platform.

100% trustful implementation is required.

We follow the Nuttx mailing list.