ESTCube-2 On-Board Computer: File Systems with Low Overhead

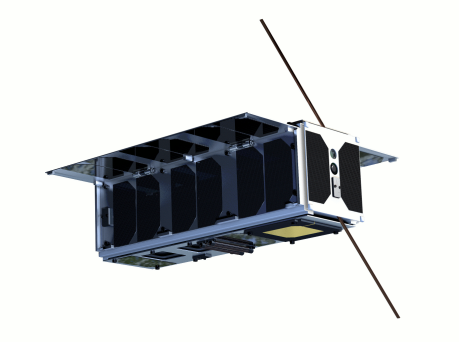

Figure 1: Artist's impression of ESTCube-2 by Taavi Torim.

ESTCube-2 is a student satellite project to test the electric solar wind sail technology in the plasma brake configuration. Among other subsystems such as the electrical power system, communication, attitude determination and control system, the satellite has an On-Board Computer (OBC). The job of the OBC is to coordinate the mission by scheduling operations, distributing commands, gathering measurements and housekeeping data from other systems, and storing them in on-board non-volatile memory. Two types of non-volatile memory are used: non-volatile RAM and Flash. Non-volatile RAM is used for mission-critical data such as configuration files, error logs, executable code, etc. Flash memory is used for temperature, voltage, current measurements, camera images, etc.

There are several well-known RAM file systems out there (ramfs, pramfs), which could also be used for non-volatile RAM. However, they are typically designed for a specific operating system, have a relatively large overhead or pack too many features.

There is an even wider range of Flash file systems (JFFS2, UBIFS, YAFFS2, etc.). However, they often have either too great an overhead or require too much RAM.

While there are also dedicated file systems available for embedded systems with low code and data memory, the situation has changed since the end of the world (2012). In 2012, there was not much of a choice and custom file systems "ECRFS" and "ECFFS" were implemented. By now, good alternatives are probably available. A new survey of embedded file systems for non-volatile RAM and Flash should be conducted.

Goal

The goal of the thesis is to perform the following:

1. Survey of file systems fulfilling specific criteria:
   (a) Maturity (years in development, active support, comprehensive documentation).
   (b) Portability to the arm-none-eabi target (specifically ARM Cortex-M3, Cortex-M4, Cortex-M7).
   (c) Portability to FreeRTOS.
   (d) Availability (cost, license).
   (e) Code footprint, when built for ARM Cortex.
   (f) Data footprint, when built for ARM Cortex.
   (g) Metadata overhead and how it scales.
2. Writing a test case for the file systems.
3. Selection of a few candidates for testing on hardware.
4. Comparison of code and data footprints, access time.

Contact

Supervisor / contact	Indrek Sünter
E-mail	indrek.sunter@estcube.eu
Fleep	indrek.sunter@fleep.io
Phone	+372 56161367