

SMALL SATELLITE MISSIONS SYMPOSIUM (B4)
Space Systems and Architectures Featuring Cross-Platform Compatibility (7)

Author: Mr. Claas Ziemke
Private, Germany, claas.ziemke@gmx.net

Dr. Toshinori Kuwahara
University of Stuttgart, Germany, kuwahara@irs.uni-stuttgart.de

AN INTEGRATED DEVELOPMENT FRAMEWORK FOR RAPID DEVELOPMENT OF
PLATFORM-INDEPENDENT AND REUSABLE SATELLITE ON-BOARD SOFTWARE

Abstract

Even in the field of small satellites, the on-board data handling subsystem has become complex and powerful. With the introduction of powerful CPUs and the availability of considerable amounts of memory on-board a small satellite it has become possible to utilize the flexibility and power of contemporary platform-independent real-time operating systems.

Especially the non commercial sector such like university institutes and community projects such as AMSAT or SSETI are characterized by the inherent lack of financial as well as manpower resources. The opportunity to utilize such real-time operating systems will contribute significantly to achieve a successful mission. Nevertheless the on-board software (OSW) of a satellite is much more than just an operating system. It has to fulfill a multitude of functional requirements such as: Telecommand interpretation and execution, execution of control loops, generation of telemetry data and frames, failure detection isolation and recovery, the communication with peripherals and so on.

Most of the aforementioned tasks are of generic nature and have to be conducted on any satellite with only minor modifications. A general set of functional requirements as well as a protocol for communication is defined in the ESA ECSS-E-70-41A standard "Telemetry and telecommand packet utilization". This standard not only defines the communication protocol of the satellite-ground link but also defines a set of so called services which have to be available on-board of every compliant satellite and which are of generic nature.

In this paper a platform-independent and reusable framework is described which is implementing not only the ECSS-E-70-41A standard but also functionalities for inter-process communication, scheduling and a multitude of tasks commonly performed on-board of a satellite. By making use of the capabilities of the high-level programming language C/C++, the powerful open source library BOOST, the real-time operating system RTEMS and finally by providing generic functionalities compliant to the ECSS-E-70-41A standard the proposed framework can provide a great boost in productivity.

Together with open source tools such like the GNU toolchain, Eclipse SDK, the simulation framework OpenSimKit, the emulator QEMU, the proposed OSW framework forms an integrated development framework. It is possible to design, code and build the on-board software together with the operating system and then run it on a simulated satellite for performance analysis and debugging purposes. This makes it possible to rapidly develop and deploy a full-fledged satellite OSW with minimal cost and in a limited time frame.