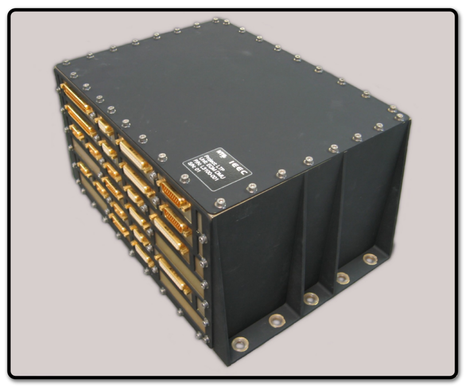
From LPF to eLISA: new approach in payload software

Lluís Gesa, Víctor Martín, Carlos Sopuerta
Institut de Ciències de l'Espai (CSIC-IEEC)

Introduction

The IEEC team has developed the **LISA Pathfinder** **D**ata **M**anagement **U**nit (**DMU**). During several years, our software team has implemented its embedded software following the **ESA** guidelines and standards ensuring and documenting each development stage along its quality.



Picture of the EQM model of DMU, used in hardware and software tests. The DMU is composed by two sets (Nominal and Redundant) of :

- *Two PDUs: Power Distribution Unit*
- *Two DPUs: Data Processing Unit*
- *Two DAUs: Data Acquisition Unit*

Figure: DMU: LPF Data Management Unit

Now it's time to, from lessons learned from more that 5000 hours of operativity, start to define the new software approach for the future space based gravitational wave observatory **eLISA**.

LPF Software conclusions

The **LPF DMU** Software has shown its completely robustness during the **LPF** operations. Few issues appeared, none of them critical in terms of compromising the mission. The proposed workarounds ran fine for some of them, and others "use as it" without any inconvenient.

But from the operative and monitoring we have learned that things could be better:

- Not all hardware is monitored similarly. Hardware status could be better informed to ground and a best internal diagnostic would be useful.
- More flexible upgrade/modification mechanism. Current upgrade system is based in move system to basis application and logically; stopping science. In the observatories the maintenance time is money! The non-productive time should be avoided.
- Historicals. Internal status history could help to analyze problems.

In any future software development for the space observatory, any improvement in those points will be appreciated in order to maximize the support of the payload software for the scientific return.

The key : ARINC-653

Our team is under study of the possible use of the standard **ARINC-563**. This Avionics Application Standard Software Interface is a software specification for **T**ime and **S**pace **P**artitioning (**TSP**) in safety-critical avionics **Re**al-**T**ime **O**perating **S**ystem (**RTOS**). It allows the hosting of multiple applications of different software levels on the the same hardware in the context of an **I**ntegrated **M**odular **A**vionics (**IMA**) architecture but recently is under **ESA** studies to be applied in space.

Several software applications with its own characteristics and security levels can co-exist over same hardware exchanging (or not) information and the failure of one of them doesn't affect the operativity of the others.

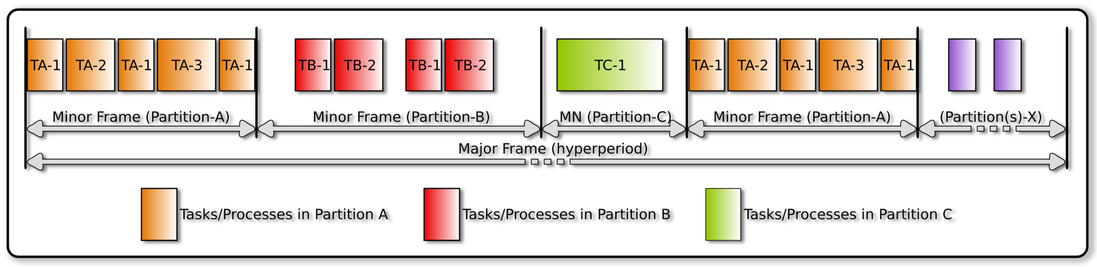


Figure: **TSP** scheduling diagram

In each general process period (hyperperiod) a fixed sequence of partitions executions is performed, and once a partition is active, its internal system is executed isolated from the rest. Communications between partitions and hardware are managed via **TSP** kernel. One partition can be executed several times in the same hyperperiod, so if a task needs to be executed at 100 Hz, its container partition needs to cover at least this 100 Hz in the hyperperiod.

Approach

Based on above idea, next figure deploys our approach:

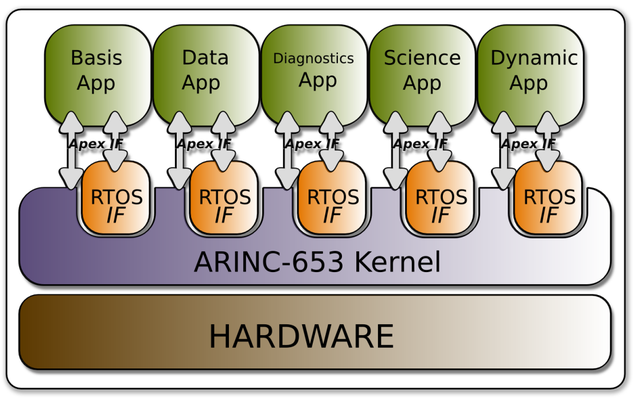


Figure: Layered Architecture under study

Multiple applications running together in parallel sharing hardware resources and data. Each one with its own operating system and isolated security. And each one covering some specific functionality needed to control the payload system.

Modules

Our proposal is to group each needed behavior in separate applications to be executed each one in a **TSP** partition. So initially we should need next listed applications:

- **Basis APP** : Application running at highest security level, in charge of overall system and state-of-health monitoring. Managing the State Machine and System Configuration.
- **Data APP** : Data Handling application, acting as a "database" for both configuration parameters and science results.
- **Diagnostics APP** : Application to manage the diagnostics subsystems (Sensors and Actuators) and monitoring connected subsystems, for example the *Phasemeter* Unit.
- **Science APP** : Main application performing the needed science. Up-to-now is not clear the needs in this aspect, but a primary on-board computing of results, in order to automatically configure observations or signal processing seems possible.
- **Dynamic APP** : Our innovative bet, an application partition enable to:
  - Mirror any other partition, adding the capacity to system upgrades while system is in operational mode. In observatories, both space and ground, the science time should be maximized. Long maintenance and idle periods are detrimental for science return and should be reduced. A full-time up-and-running system with hot upgrades could be a great advantage.
  - Fast and reprogramable application, enable to run scripting sequences. Useful to debugging or in-place testing of new functionalities before to be official.

Of course, a very basis application is needed to be executed just at power-on. Our goal is to build a Basis Application enabled to be executed first directly on hardware as booting system, and execute from it the **TSP** system. Once this will be up-and-running much of the boot software code should be reused/executed as Basis App in a partition.

Implementation Technology

- **Hardware** : **C**ommercial **o**ff-**t**he-**s**helf (**COTS**) based on Multicore platforms with Leon 3/4 technologies seems o be the best solution for the hardware data process necessities, alternatively custom **S**ystem **O**n **C**hip (**SoC**) systems are also under study using **FPGA** technology.

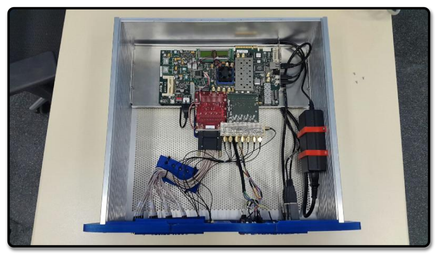


Figure: FPGA Setup at IEEC-CSIC premises

- **Hypervisors**: The **ARINC-563** infrastructure is based in a core layer that acts, in terms of software architecture, as bare metal Hypervisor. Several solutions are already present, none of them in a maturity level to be used in Space Mission, within them, we are under study of: ***AIR***, an **ESA** sponsored project based on ***RTEMS***, and ***Xtratum*** hypervisor system. Other commercial solutions could be ***wxWorks 653***, ***PikeOS***

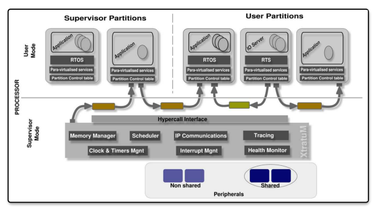


Figure: Overview of the Xtratum Architecture

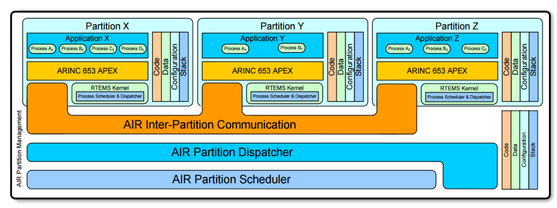


Figure: Overview of the Air Architecture

- **Operating systems**: **LISA Pathfinder** Payload Software has been based on ***RTEMS***, so our first approach is to still using it, but other options like ***Xluna***, or the provided with ***Xtratum***:***LithOS*** are not discarded. Specially the use of standard ***GNU/Linux*** could be useful on the dynamic partition to enable the use of scripting execution for fast development, execution and diagnostic.

Development & Qualification

Development should follow **ESA** quality guidelines, the ECSS standards defines the technological development for space missions. Concretely ECSS-E-40 and Q-80 for software.

In new researches in **TSP** area from a whole system architecture the guidelines and processes proposed by **S**pace **AV**ionics **O**pen **I**nterface a**R**chitecture (**SAVOIR**) should be followed.

Since the proposed Hypervisors are part of several **ESA** studies, projects and researches, full space qualification is not yet reached, our work could be a partner in this process.

Our approach of Dynamic APP partition is a new concept not yet studied, therefore a further work in terms of development and qualification will be required.

Conclusion

While our team is currently immersed in the **LPF** mission, the work for **eLISA** just starts. Many open points and much work ahead, but the challenges of all these new software technologies and technical proposals give us the energy to kick off as soon as possible to start to see results in short.

Study phase is not yet finished, and this is important to enter in the developing phase ensuring the final goal.

Validation and acceptance phase is the expected more complex part. All these technologies are new in space and quality process will take its time.

Acknowledgments

Support from MICINN via contract code AYA2010-15709 is thankfully acknowledged.

WWW: http://gwart.ice.cat

Mail: gesa@ieec.cat / vmartin@ieec.cat

Institut de Ciències de l'Espai (CSIC-IEEC) - Barcelona, Spain