AIR

AIR – ARINC 653 Interface in RTEMS

(ITI - Proof of Concept)

ESA Technical Meeting

May 2007, ESTEC, Noordwijk, The Netherlands

	Name	Title	Signature	Date
Prepared:	José Rufino	Senior Researcher (FCUL)		
	Sérgio Filipe	Technical Manager (Skysoft)		

PRESENTATION SUMMARY

- ✓ **AIR Fundamental Concepts**
  Eng. Sérgio Filipe (Skysoft)
- ✓ **AIR Design and Architecture**
  Prof. José Rufino (FCUL)
- ✓ **Spatial and Temporal Segregation**
  Prof. José Rufino (FCUL)
- ✓ **Proof of Concept Demonstrator: *AIR PMK***
  Prof. José Rufino (FCUL)
  Eng. Manuel Coutinho (FCUL)
- ✓ **APEX Interface: Concept and Design**
  Eng. Sérgio Filipe (Skysoft)
- ✓ **Proof of Concept Demonstrator: *APEX Interface***
  Eng. Sérgio Filipe (Skysoft)
- ✓ **Future Challenges**
  Eng. Sérgio Filipe (Skysoft)
  Prof. José Rufino (FCUL)

AIR Project Summary:

In both avionics and space industries, the safety concept is of paramount importance. The ARINC 653 standard was developed with the purpose that all safety critical software embedded in a system must follow very strict and demanding rules both in terms of operation and certification.

ARINC 653 and Integrated Modular Avionics (IMA) are the answers provided by the civil aviation world to problems that are also identified in the space world. The space world is looking for a standardized interface for the Operating Systems (OS) located on board the spacecrafts. Most of the requirements from the civil aviation world that led to the definition of ARINC 653 are also requirements from the space world and thus the adaptation of the specification to the space world needs can be performed with minor changes, keeping its basic principles.

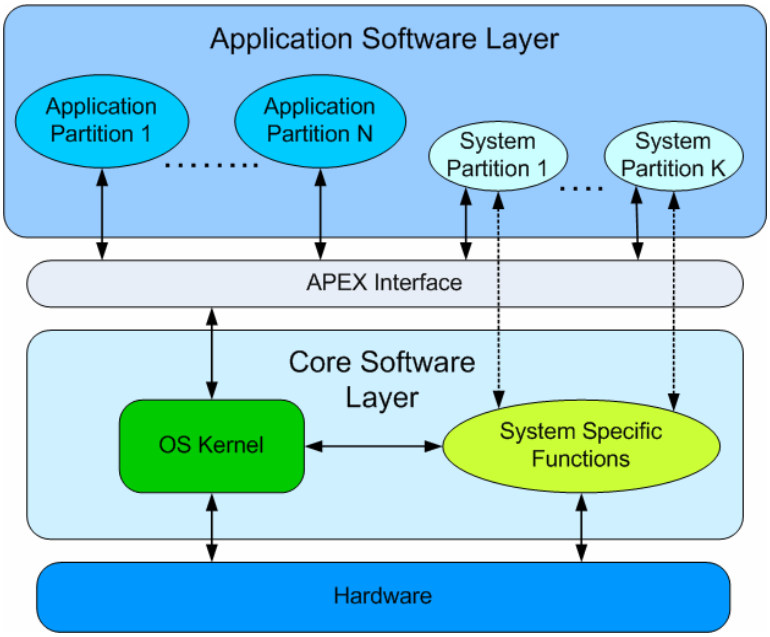
The adoption of the **ARINC 653 concept in space on-board software** will not only provide the space industry the same benefits the aviation industry has already profited with by adopting the standard – software portability and modularity, partitioning and less certification effort, etc. It will also promote the reusability of Research and Development (R&D) efforts already invested in the scope of another industry domain, further increase the synergies in the development of software for the parallel domains of civil aviation and space and potentiate reduction in the development costs of on-board software. Finally, the space world will benefit from ARINC 653's improvement in the development framework available for both application developers and integrators.

Furthermore, there is a general demand for the use and re-utilization of commercial off-the-shelf (COTS) components in the design of complex embedded systems, such as those found in aerospace applications. The AIR – ARINC 653 Interface in RTEMS – innovation initiative has emerged complying to this requirement, exploiting the utilization of a COTS licence-free open-source real-time operating system, the Real-Time Executive for Multiprocessor Systems (RTEMS). The use of RTEMS is particularly interesting given its qualification for critical on-board software of unmanned space programs.

However the AIR Project went one step further, defining a design approach that allows the fundamental AIR concept to be applied to other COTS real-time operating system (RTOS) kernels. Different RTOS kernels may even operate in the same system. This definition of a very flexible and versatile architecture, making use of a comprehensive set of systems and tools, opens room for the application of the AIR concept to the different RTOS technologies and to several application sets, thus standing for the general designation: ARINC 653 Interface in RTOS kernels.

AIR Fundamental Concepts:

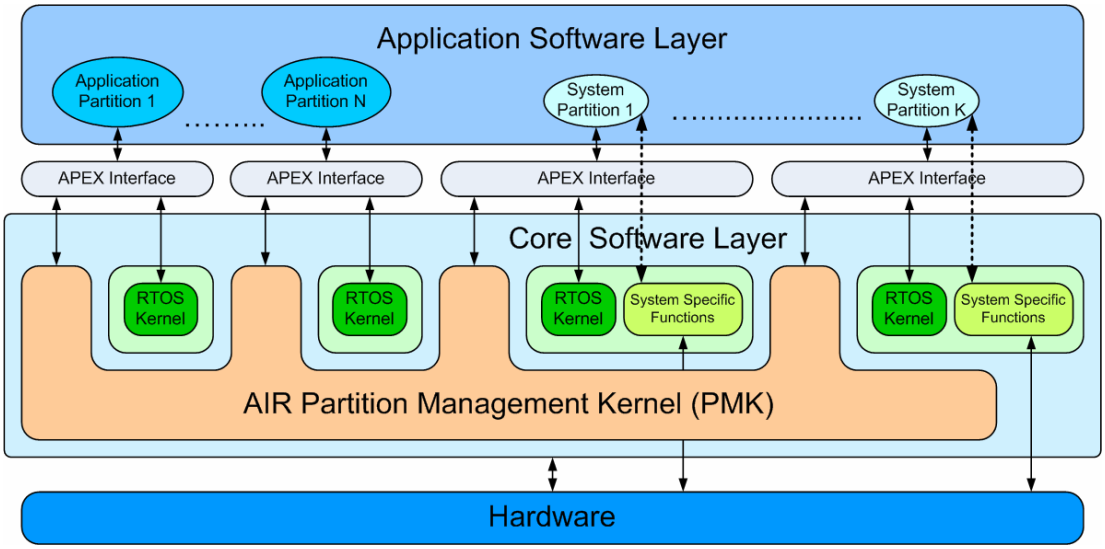
- The IMA (Integrated Modular Avionics) concept aims to develop a new generation of aeronautic systems based on a group of standards and using COTS products that shall be available for multiple aircraft platforms;
- One of the most accepted and stable IMA standard is ARINC 653
- ARINC 653 defines an APEX (APplication EXecutive) that provides a common OS interface and set of services to the avionics application developers
- ARINC 653 implements the Partitioning concept as defined by IMA



AIR Design and Architecture:

The definition of the AIR system overall architecture makes use of:

- a multi-executive core software layer;
- a two-level hierarchical scheduler.
- a different instance of a RTOS kernel is used per partition
- homogenous RTOS integration also implies the use of one RTOS per partition.

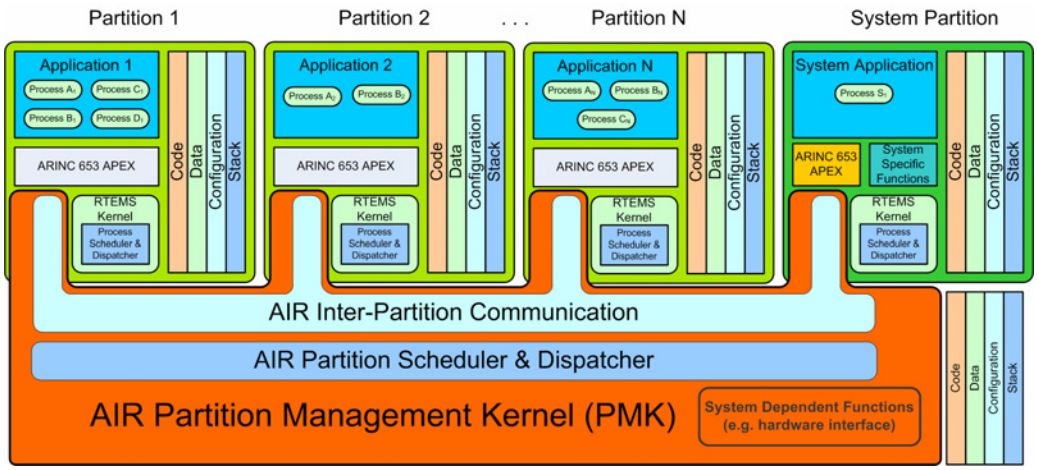


The fundamental AIR design components are the following:

- **application executive (APEX) interface**
- **native RTOS kernel**
- system partitions do an additional use of **system specific OS functions**
- **AIR Partition Management Kernel (PMK)**

The main components of the AIR Partition Management Kernel (PMK) include the following functional modules:

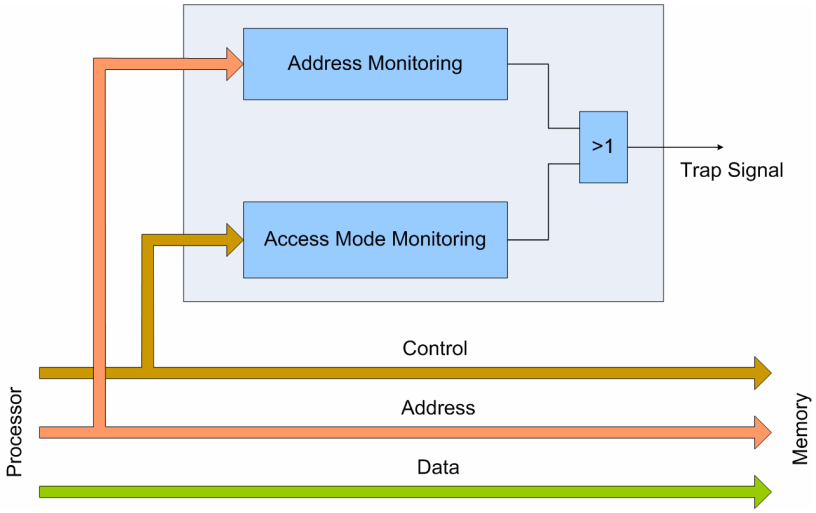
- **AIR inter-partition communication**
- **AIR partition dispatcher**
- **AIR partition scheduler**
- **AIR time manager**
- **AIR Hardware Abstraction Layer (HAL)**



AIR Spatial and Temporal Segregation:

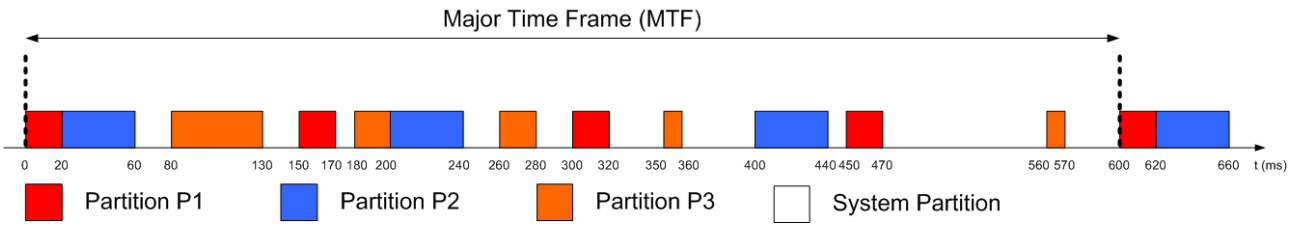
Securing Spatial Segregation

- Effective support to spatial partitioning requires the use of specific memory protection mechanisms;
- Supported by memory management unit (MMU) hardware
- Protection of memory addressing spaces but also a
- Functional protection concerning the management of privilege levels and restrictions to the execution of privileged instructions.



Securing Temporal Segregation

- Restricted processing time assigned to each partition;
- A single partition cannot monopolize the usage of the processor infrastructure;
- Scheduling of partitions is strictly deterministic over time;
- Fixed temporal window in which the partition has control over the computational platform;
- Partition is scheduled on a fixed, cyclic basis.



AIR Proof of Concept Demonstrator - AIR PMK:

- Proof of concept prototypes built using the RTEMS 4.6.6 version enhanced with a graphical window manager, dubbed VITRAL (VITRAL is the Portuguese word for stained glass window).

- AIR Multi-Executive Core (MCE) proof of concept illustrates:
  - Partition and Task Scheduling using multiple RTEMS Kernels (one per partition)
  - Partition scheduling uses a fixed cyclic algorithm;
  - Process (task) scheduling uses the native RTEMS scheduler.

Partition	Function
Partition X	Attitude Control
Partition Y	Telemetry Tracking and Command
Partition Z	On-Board Data Handling
System Partition	Communications

AIR APEX Interface – Concept and Design:

In this phase, the APEX Interface design does not aim to demonstrate the total compliance of AIR implementation with the ARINC 653 standard. The purpose is to show evidence that basic APEX features can be implemented over RTEMS and that **correct** behaviour is achieved.

The APEX Interface Demonstrator implements and uses the following APEX resources:

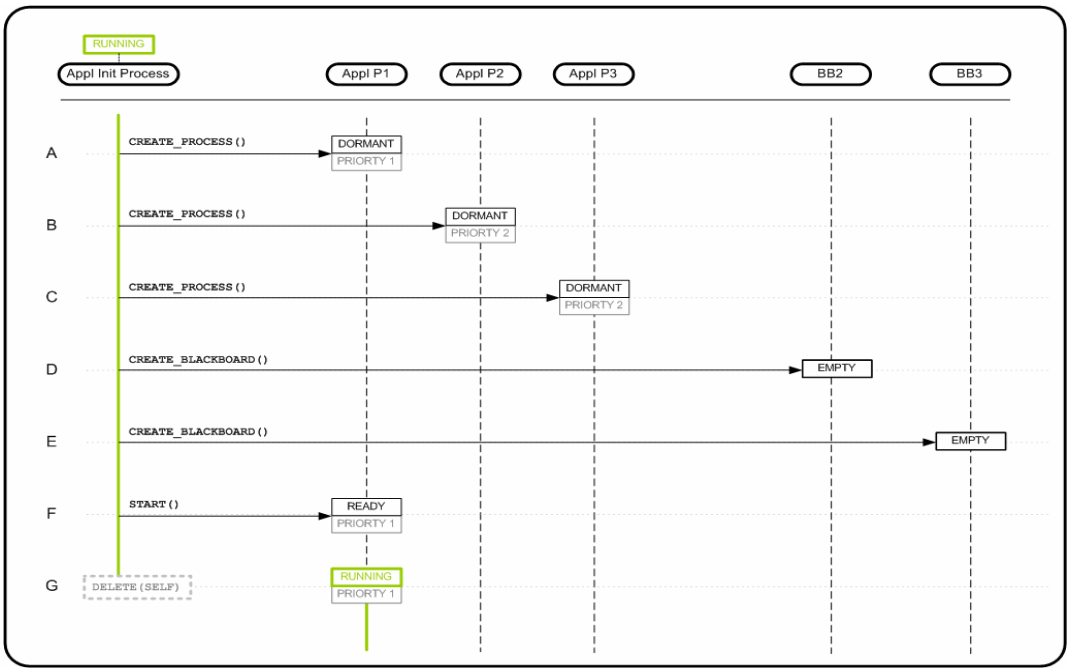
- **Process P1** – this is the main process that sends messages to the blackboards. It has the lowest priority of the three processes used;
- **Process P2** – this process waits for a message to be available on blackboard BB2. When this condition is met, it reads the message and clears the blackboard. It then reacts according with the received message. Its priority his above P1.
- **Process P3** – this process is similar to P2 but uses blackboard BB3 instead of blackboard BB2. Also, it has the same priority as P2.
- **Blackboard BB2** – this blackboard shall be used for P1 to send messages to P2;
- **Blackboard BB3** – this blackboard shall be used for P1 to send messages to P3.

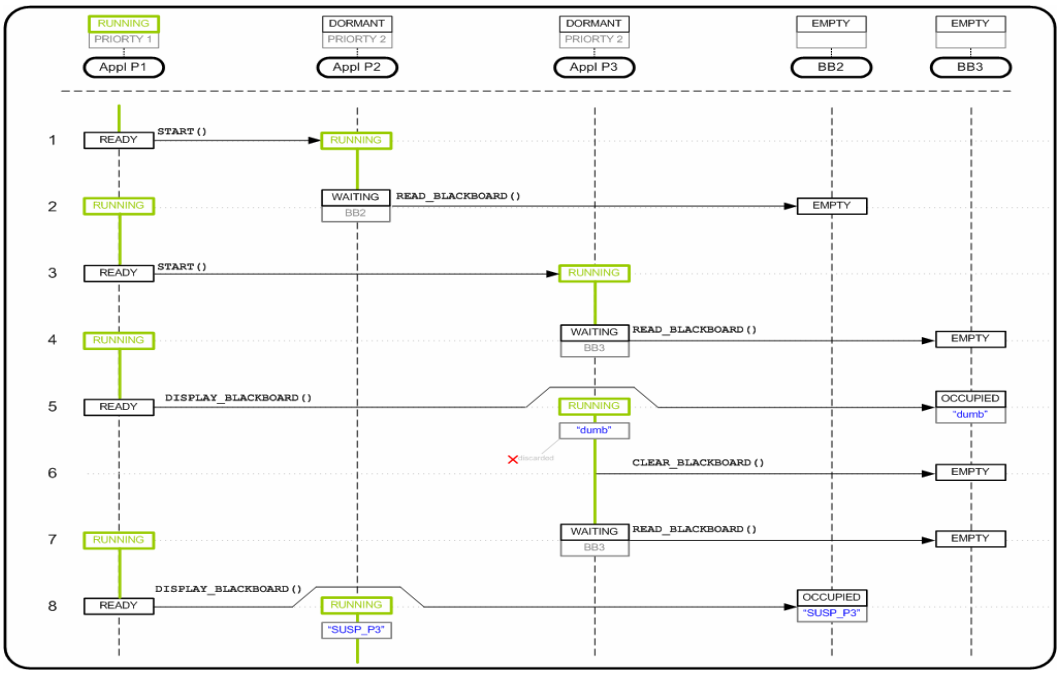
The demonstrator main focus relies on the following issues using the listed above resources:

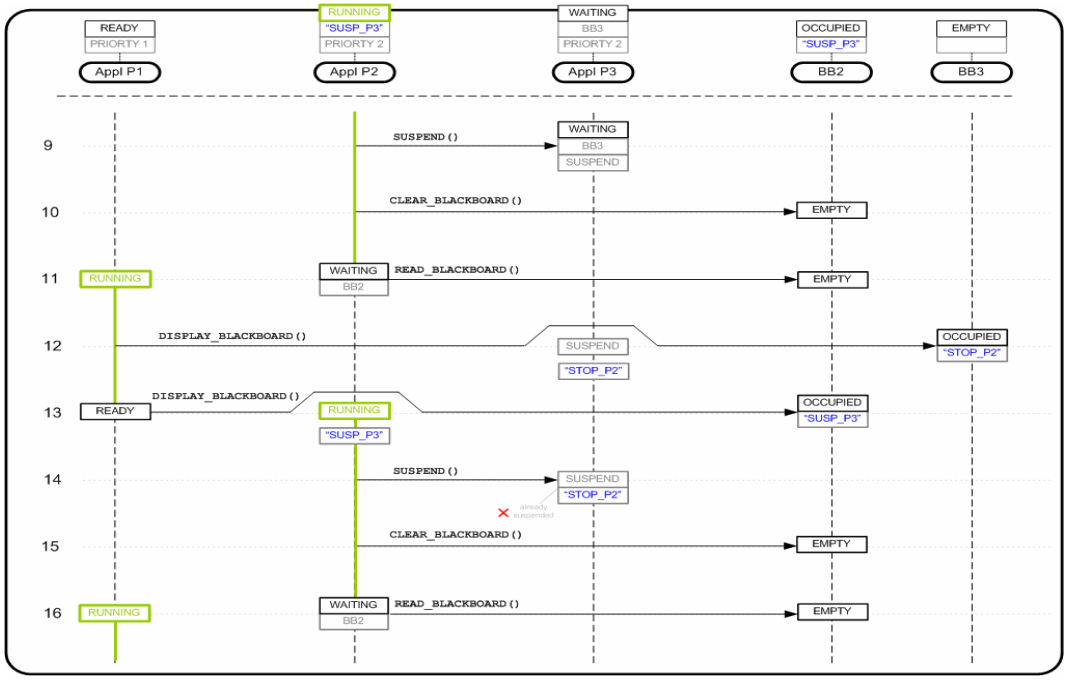
- Basic operations over APEX processes like **`CREATE`**, **`START`**, **`STOP`**, **`SUSPEND`**, **`RESUME`** using RTEMS tasks to implement APEX processes;
- Intra partition communication between AIR processes by using the APEX Blackboard service;
- The implementation of the APEX priority driven process scheduler by using the RTEMS native task scheduler

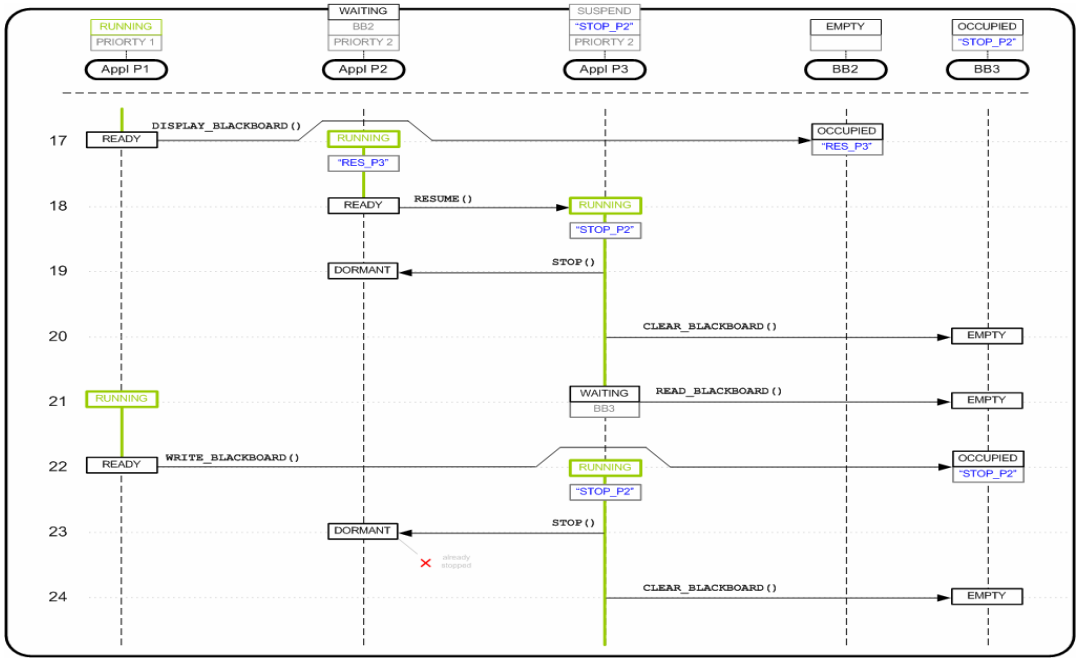
AIR Proof of Concept Demonstrator – APEX Interface:

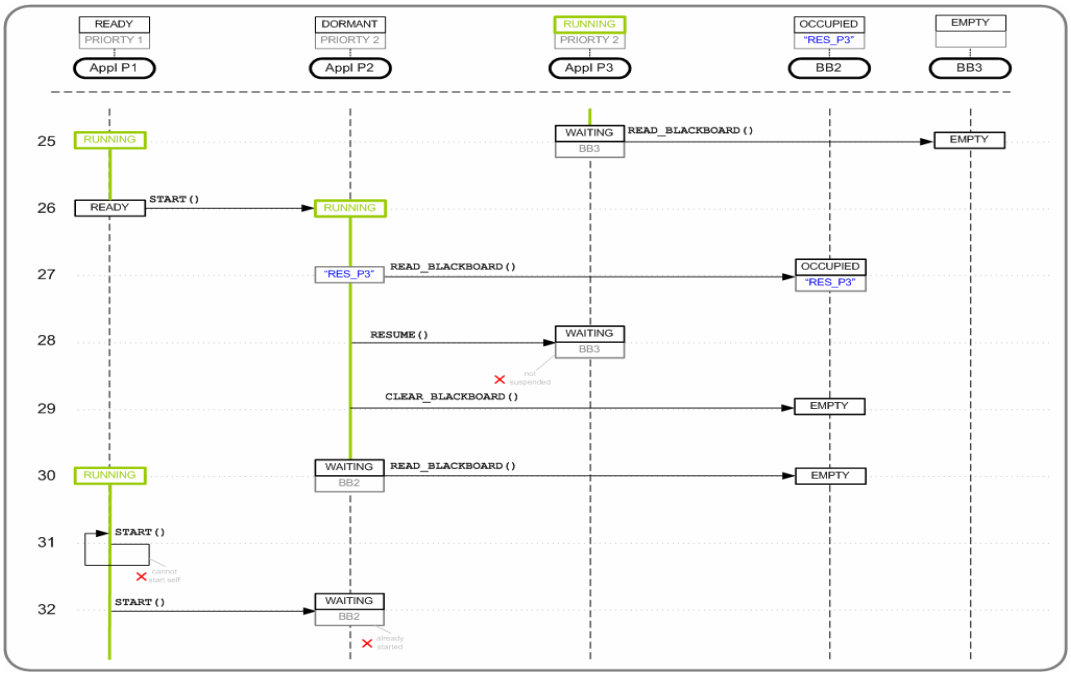
- Activity diagrams for the AIR APEX Interface Demonstrator:

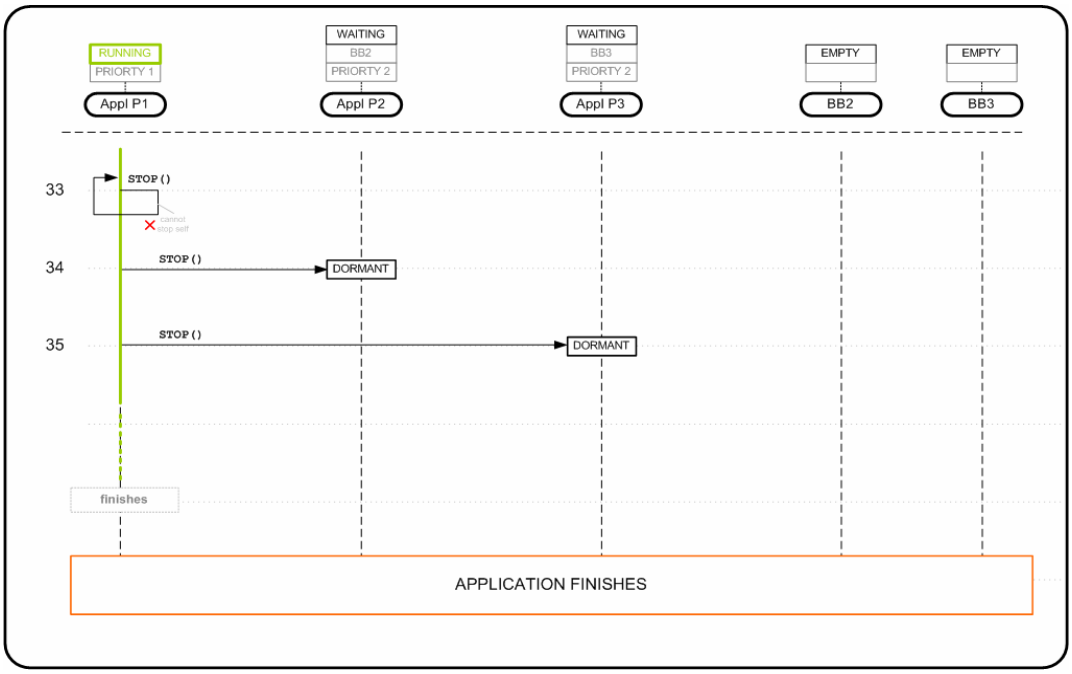












AIR Future Challenges:

A set of issues must be carefully assessed before a future full implementation of AIR Technology can be developed and used as a product:

- **Inter partition communication** – besides completely defining the interface between the APEX implementation and the PMK, the low level memory devices needed to achieve spatial segregation are hardware dependent and as such, the design of the inter partition communication must be carefully evaluated for each kind of hardware configuration;

- **Definition of the APEX interface with the PMK** – this is an AIR system issue that was not 100% solved because it was not on the scope of the project. Although an open issue it should be not a big challenge to solve. This issue was identified for Partition Management but in the future other needs may arise;

- **Implementing periodic processes over RTEMS** – the implementation of periodic processes over the RTEMS native API was informally analyzed among the AIR team and it was evident that some issues might need an extra effort to solve as to make it 100% complaint with the APEX specification. This issue must be formally analyzed as to find the correct solution;

- **Health monitoring** – the health monitor was out of the AIR scope but it is no question that it must be present on any RTOS suitable for space systems. As such, sooner or later it must be implemented over AIR. This should be done always having on mind the issue we address next;

- **ARINC 653 in space specificities** – the adoption of the ARINC 653 standard on space, and as such its transition from the aviation to the space market, need a detailed analysis as to understand the inherent differences and specificities. This could imply some changes over the APEX interface that must be reflected over on the AIR system. A very meaningful example of this is, as referred before, the health monitoring;

- **Optimization** – even after having a 100% compliant implementation of the APEX over RTEMS, any implementation should be carefully tested as to avoid bottlenecks on the system and make it as efficient as possible;

- **Certification** – before proceeding to a fully available RTOS suitable for space usage, the AIR system must be certified as to guarantee the demanded safety level.