OBDP2019

AIR Hypervisor using RTEMS SMP

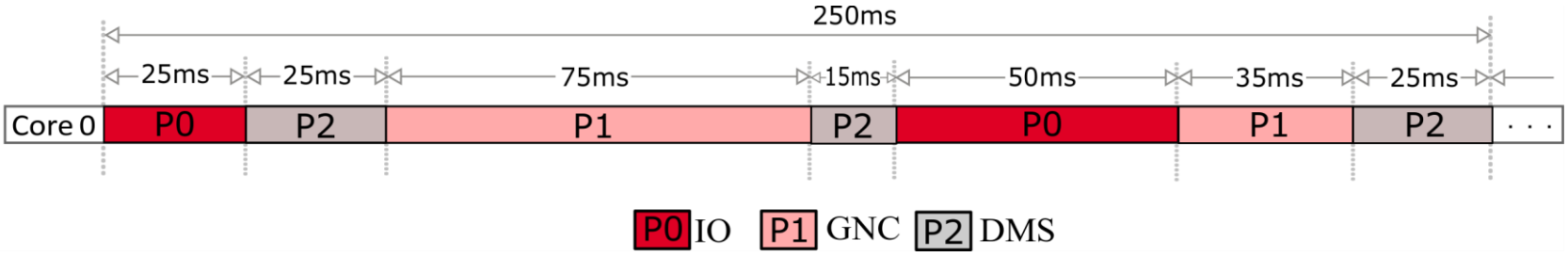
gmv
INNOVATING SOLUTIONS

CONTENTS

Activity

Context

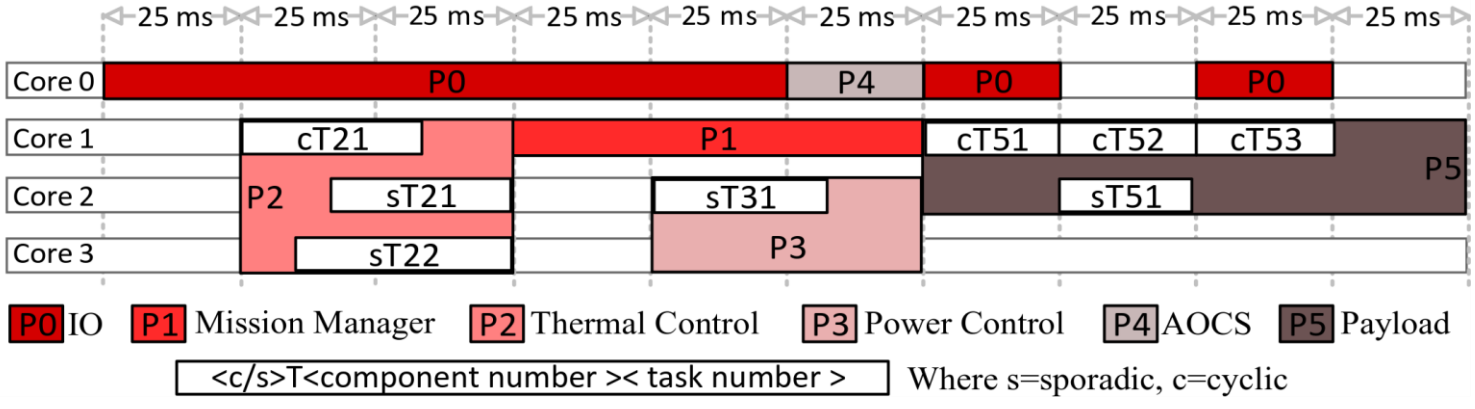
TSP



- Avionics functions using the same resource, **safely isolated** in time and space
- Applications with **different criticality levels** can co-exist in the same processing hardware
- Use of standards promotes **software reuse**, portability and modularity
- Less hardware, thus less weight and power consumption
- Overall: **more capabilities, less costs**

SMP

- Symmetric Multiprocessing (SMP), a tightly coupled multiprocessor system with identical processors running independently from each other where each processor shares the same memory and I/O devices
- The SMP operating system is able to manage tasks according with predefined schedule algorithms



AIR

Overview

What is AIR

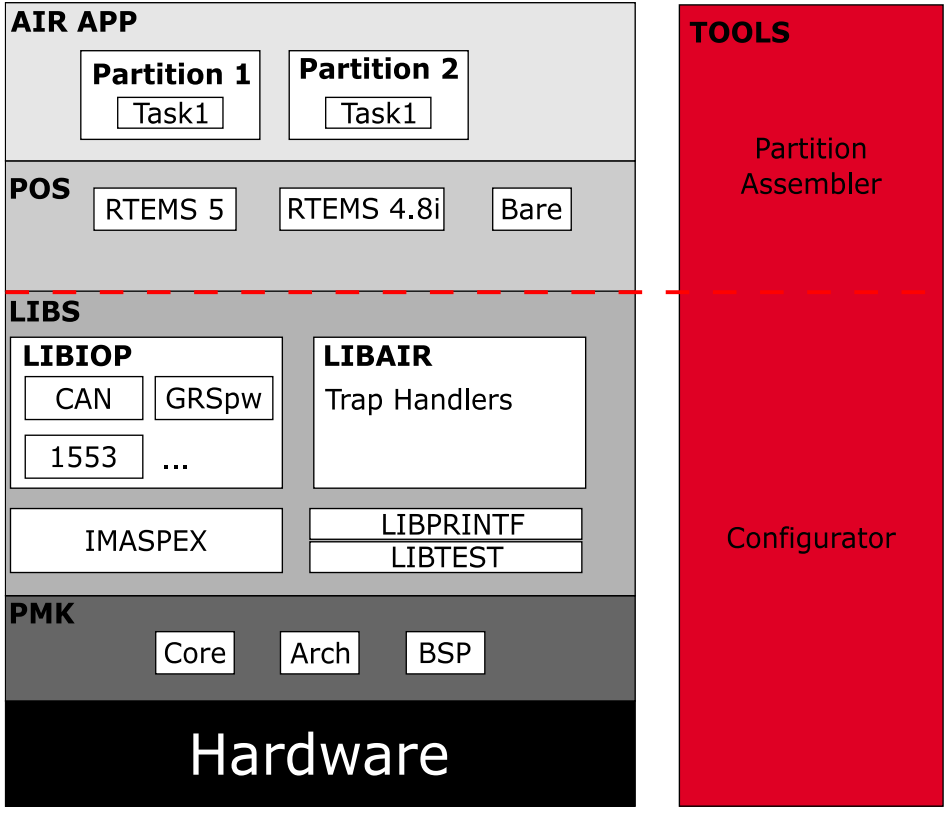
air

REAL-TIME HYPERVISOR
COMPLIANT WITH ARINC 653 and IMA-SP

- AIR is a type-1 hypervisor that guarantees **TSP** in the target hardware
- Allows multiple avionics applications with **different criticalities** to run independently in the same hardware resource

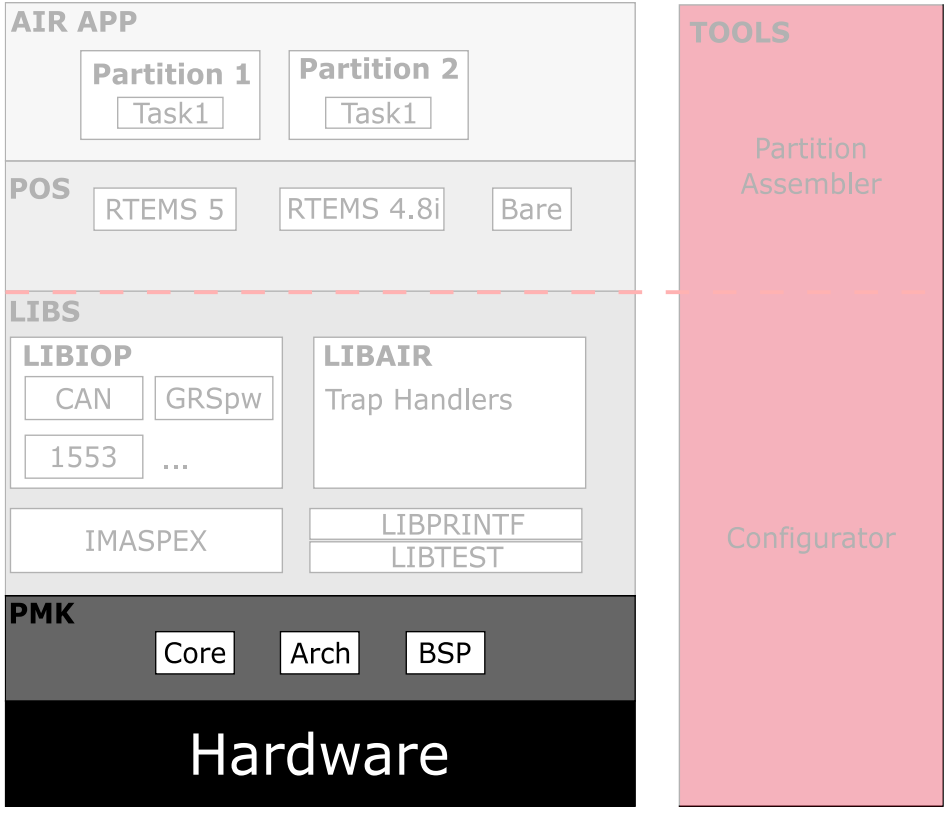
Prototyping | Development | Integration

Structure



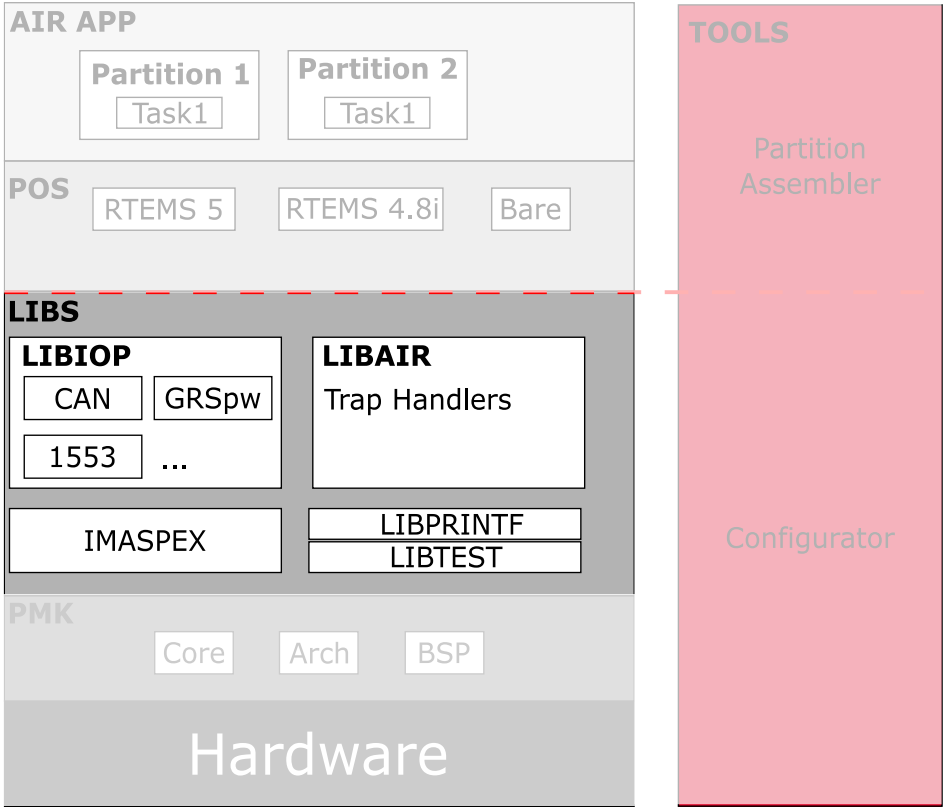
- Supported Architectures:
  - SPARC V8
  - ARMv7
- Supported RTOS:
  - RTEMS 4.8i
  - RTEMS 5

PMK



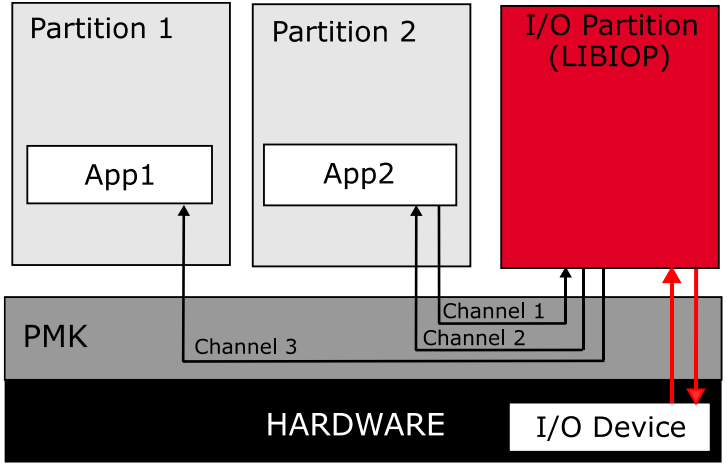
- Partition Management Kernel (PMK) designed with a microkernel approach
  - Core: AIR core functionalities (Partition Management, Scheduling, Health Monitor)
  - Arch: Generic functionality for SPARC v8 and ARMv7
  - BSP: Specific functionalities for LEON2, LEON3, LEON4 and Cortex-A9

Supporting Libs



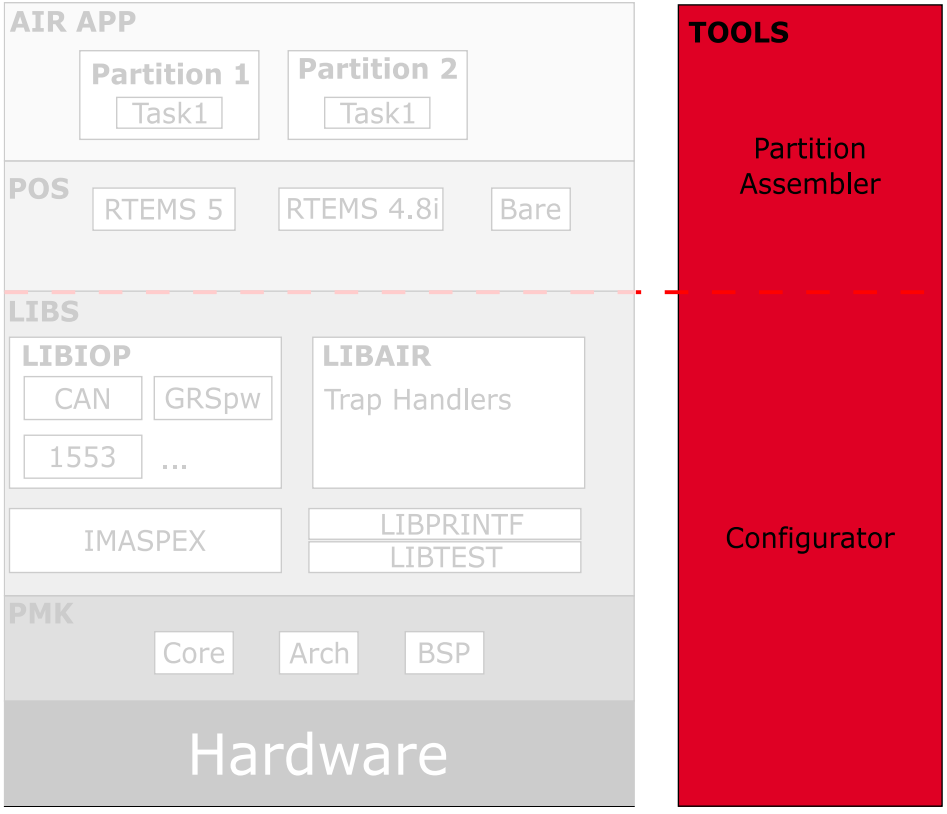
- IMASPEX: Implementation of standard ARINC 653 API enriched by ESA specifications taken in IMA-SP
- LIBAIR: Support to system calls used in RTOS paravirtualization. Provides Health Monitor support
- LIBPRINTF: Implementation of the printf functionality (useful in partitions without RTOS)
- LIBTEST: Auxiliary functions used to execute, integrate and a validation test used to be applied in the test and validation campaign of AIR and device drivers

Supporting Libs - LIBIOP



- Implementation of device drivers based on the same device drivers source code present in RTEMS
- Is generic, decoupled from the application
- Is robust, can be used by more than one partition without interference
- Routes data to its rightful partition
- Is quantifiable, its execution time must be bound and measurable
- Supported I/O interfaces:
  - SpaceWire
  - Ethernet
  - MIL-STD-1553
  - CAN bus

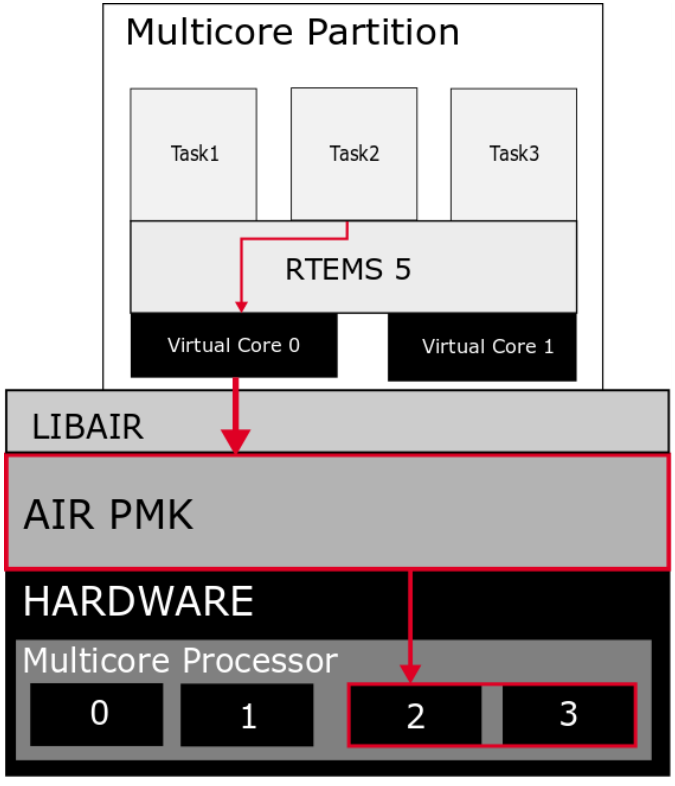
Tools



- Bind the layers to generate the executable
  - Configurator:
    - Auto-generates all the required development environment
    - Validate and auto-generate all applications dependencies given the high level XML system description
  - Partition Assembler:
    - Aggregates all the built partitions into a single executable file. PMK at initialization can effectively manage and deploy the partitions in memory

Supported RTOS

RTEMS



AIR supports paravirtualized RTOS versions for SPARC:

- RTEMS version 4.8i (a space tailored version of RTEMS 4.8 by Edisoft[1])
- RTEMS version 5
  - SMP
  - Flexibility to move tasks across different schedulers and reassign cores to schedulers

[1] - "RTEMS by Edisoft" Edisoft - Defense & Aerospace, www.edisoft.pt/product-list/rtems

Performance

Evaluation

AIR RTEMS 4.8i

Window Slot (s)	Execution Time (s)	Iterations per second	CoreMark[2]	Performance Loss
NO AIR	12.070	166	1.105	-
1.000	12.240	163	1.089	1.38%
0.500	12.240	163	1.088	1.39%
0.010	12.450	161	1.074	2.77%
0.001	12.669	157	1.050	4.88%

- AIR presence overhead grows inversely with window size
- AIR's schedulers major frame should be >1ms

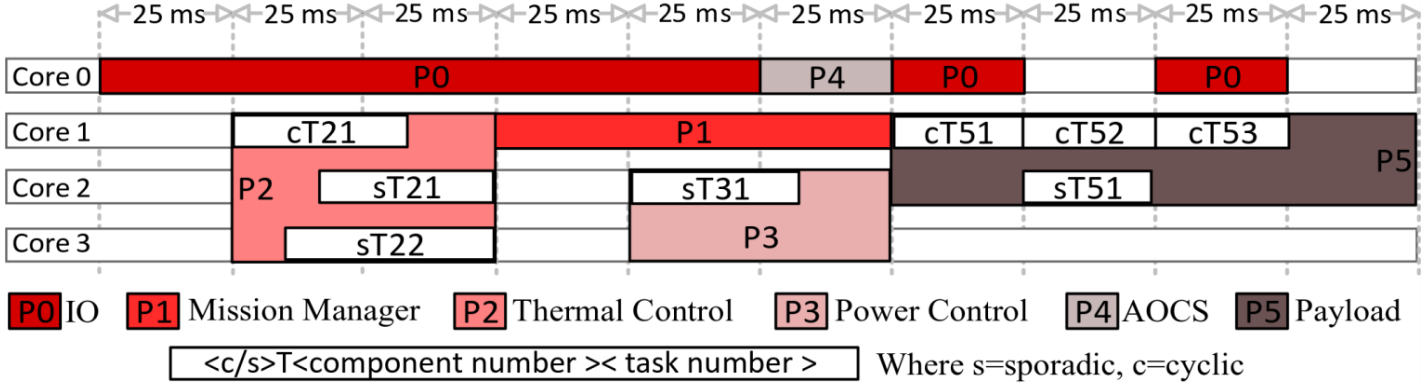
- A performance test will be conducted in order to compare a RTEMS 5 SMP multi-core application versus the same application in an AIR multi-core partition.

[2] - "CoreMark® An EEMBC Benchmark", EEMBC, https://www.eembc.org/coremark/

Use Cases

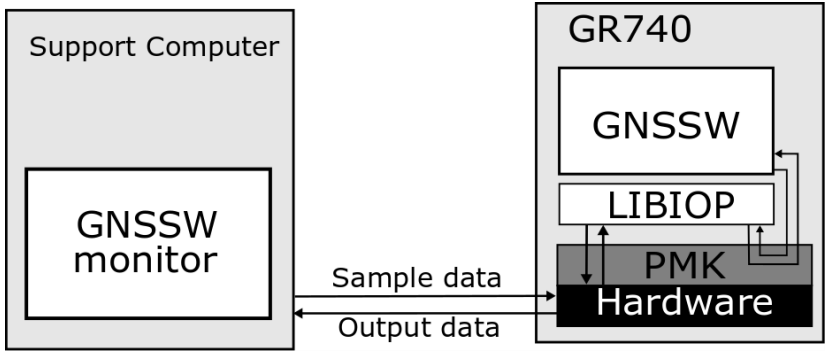
MORA-TSP Project

- Demonstrate the feasibility and performance evaluation of an end-to-end process, tools and building blocks from application level specification using the component based approach of the On-Board Software Reference Architecture (OSRA)
- The scenarios were designed to be generic but the partition names and tasks are derived from EagleEye TSP project
- The generated scenarios exemplified the multitude of scheduling and partitioning options system designers have while using a single multi-core on-board computer



GNSSW-LEON4 Project

- Implement an on-board Software Defined Radio GNSS receiver on a GR-740 on-board computer using RTEMS 5 with SMP
- Challenge of porting an already optimized SMP application into the TSP paradigm. Ensure the execution of all tracking software within a temporal deadline
- The test and simulation setups are still being used as AIR's most demanding testing scenario



Conclusion

Positioning

Conclusion

- AIR hypervisor focus is to deliver an **effective** and simple tool to build systems based on the IMA-SP directives
- Developed by strictly following **ECSS standard 40C** and **ARINC 653 specification**
- AIR is a solid solution to assist system designers and developers in quickly **porting** or **creating** a new TSP system for space
- Allows the developers to keep their **focus on the application** while enabling another layer of safety and possible optimization

Conclusion

- AIR support is in hand with industry increasing multi-core on-board computers offer by supporting new features offered by the RTOS
- AIR IO solution is **flexible** for applications able to execute on-board data handling
- MORA-TSP project is an example of how AIR is able to offer and manage **increasingly complex scenarios** given SMP scheduling flexibility
- Project GNSSW-LEON4 proves AIR is prepared for the increase in computational power of on-board computers and multi-core use in the near future

Positioning

- All AIR versions are provided open source[3] under GPL v2 license
- Currently working to support ARM Cortex-R family: R5 and R52
- Being currently applied in further new use cases (ESROCOS, INFANTE, …)
- RTOS paravirtualization activities

- GMV focus on **providing support service** to the on-board software development requiring expertise on:
  - TSP
  - Multi-core applications
  - RTEMS
  - SPARC LEON and ARM targets

[3] - "AIR git repository" GMV, https://spass-git-ext.gmv.com/AIR/AIR

www.gmv.com

THANK YOU

daniel.silveira@gmv.com

bmgomes@gmv.com

gmv BLOG