AIR Hypervisor using RTEMS SMP
- 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**
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.
What is AIR

- 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
### AIR overview

#### Structure

<table>
<thead>
<tr>
<th>AIR APP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition 1</td>
<td>Task1</td>
</tr>
<tr>
<td>Partition 2</td>
<td>Task1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POS</th>
<th>RTEMS 5</th>
<th>RTEMS 4.8i</th>
<th>Bare</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LIBS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBIOP</td>
<td>CAN</td>
</tr>
<tr>
<td>IMASPEX</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PMK</th>
<th>Core</th>
<th>Arch</th>
<th>BSP</th>
</tr>
</thead>
</table>

#### TOOLS

- Partition Assembler
- Configurator

---

- **Supported Architectures:**
  - SPARC V8
  - ARMv7

- **Supported RTOS:**
  - RTEMS 4.8i
  - RTEMS 5
- 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

**AIR overview**

### AIR APP

<table>
<thead>
<tr>
<th>Partition 1</th>
<th>Partition 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task1</td>
<td>Task1</td>
</tr>
</tbody>
</table>

### POS

- RTEMS 5
- RTEMS 4.8i
- Bare

### LIBS

<table>
<thead>
<tr>
<th>LIBIOP</th>
<th>LIBAIR</th>
<th>IMASPEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAN</td>
<td></td>
<td>Trap Handlers</td>
</tr>
<tr>
<td>GRSpw</td>
<td></td>
<td>LIBPRINTF</td>
</tr>
<tr>
<td>1553</td>
<td></td>
<td>LIBTEST</td>
</tr>
</tbody>
</table>

- **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**

**AIR overview**

- 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

**AIR APP**

<table>
<thead>
<tr>
<th>Partition 1</th>
<th>Partition 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task1</td>
<td>Task1</td>
</tr>
</tbody>
</table>

**POS**

| RTEMS 5 | RTEMS 4.8i | Bare |

**LIBS**

<table>
<thead>
<tr>
<th>LIBIOP</th>
<th>LIBAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAN</td>
<td>Trap Handlers</td>
</tr>
<tr>
<td>1553</td>
<td></td>
</tr>
<tr>
<td>IMASPEX</td>
<td>LIBPRINTF</td>
</tr>
<tr>
<td></td>
<td>LIBTEST</td>
</tr>
</tbody>
</table>

**PMK**

<table>
<thead>
<tr>
<th>Core</th>
<th>Arch</th>
<th>BSP</th>
</tr>
</thead>
</table>

Hardware
Supported RTOS
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

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

- 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.

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 and 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 and Positioning

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

THANK YOU

daniel.silveira@gmv.com
bmgomes@gmv.com