

Multi-Core Operating System MICROSAR-OS

Features and Configuration

More computing power



- ▶ Similar die size
- ▶ Same or lower CPU clock frequency
- ▶ Parallel processing

Separation of Applications



- ▶ Specialized cores (FPP, DSP)
- ▶ Various operating systems on the same ECU, e.g. OSEK **and** Linux
- ▶ Legal reasons, e.g. software originating from multiple suppliers

Functional Safety



- ▶ avoidance of mutual interference
→ running application software components on separated cores
- ▶ Redundancy → identical application running on 2 cores

Less power dissipation

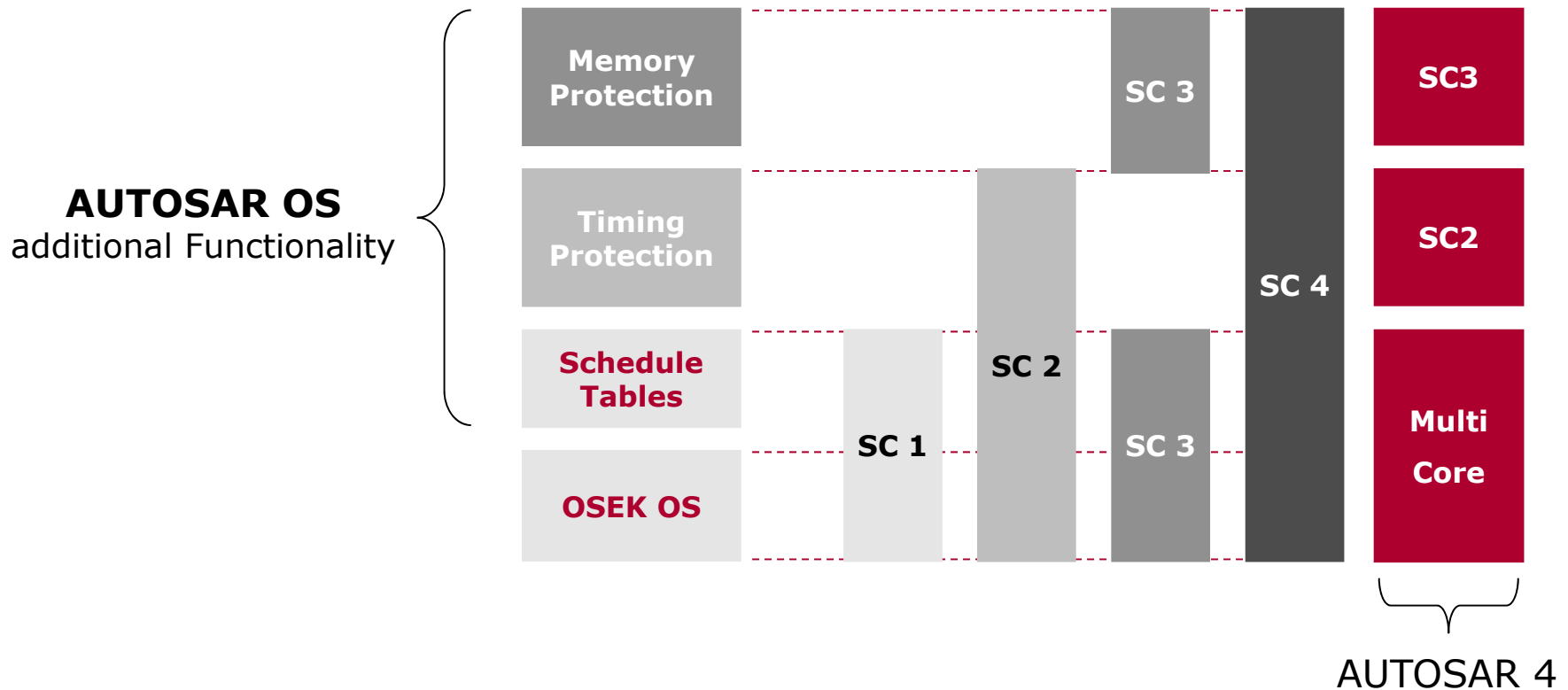


- ▶ Lower CPU clock frequency
- ▶ Computing power on demand, i.e. sleep modes for unused cores

AUTOSAR Multi-Core Operating System

Spezifikationen

- ▶ AUTOSAR **extends** the OSEK/VDX standard of operating systems.
- ▶ AUTOSAR Add-ons are segmented into Scalability Classes (SC).

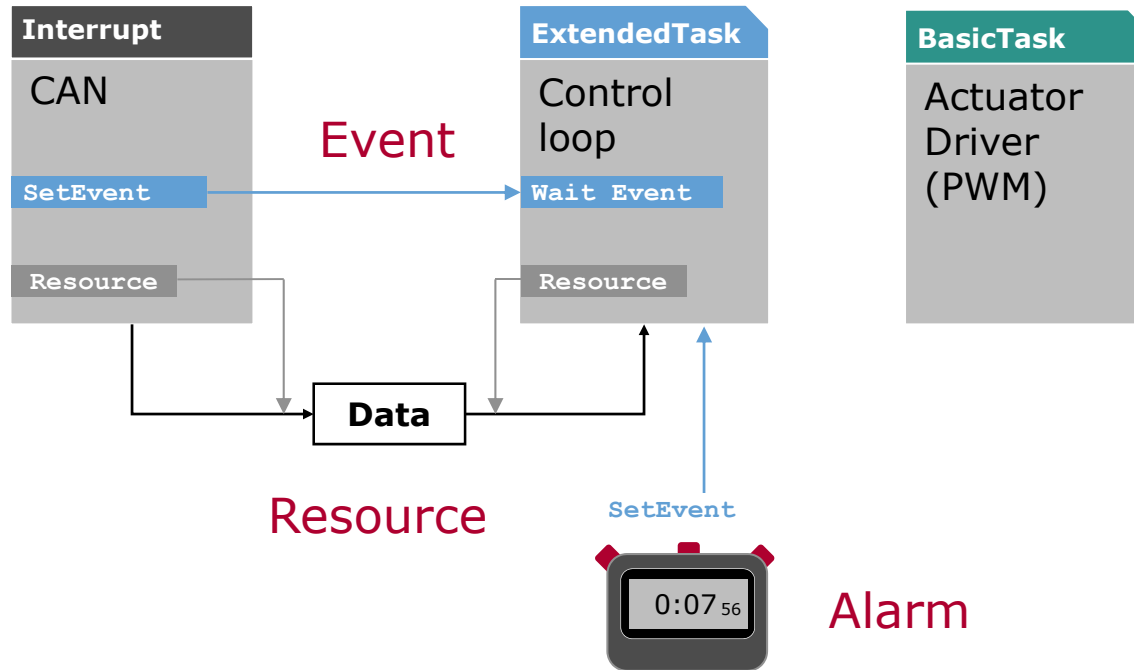


AUTOSAR Multi-Core Operating System

OSEK/VDX System Elements

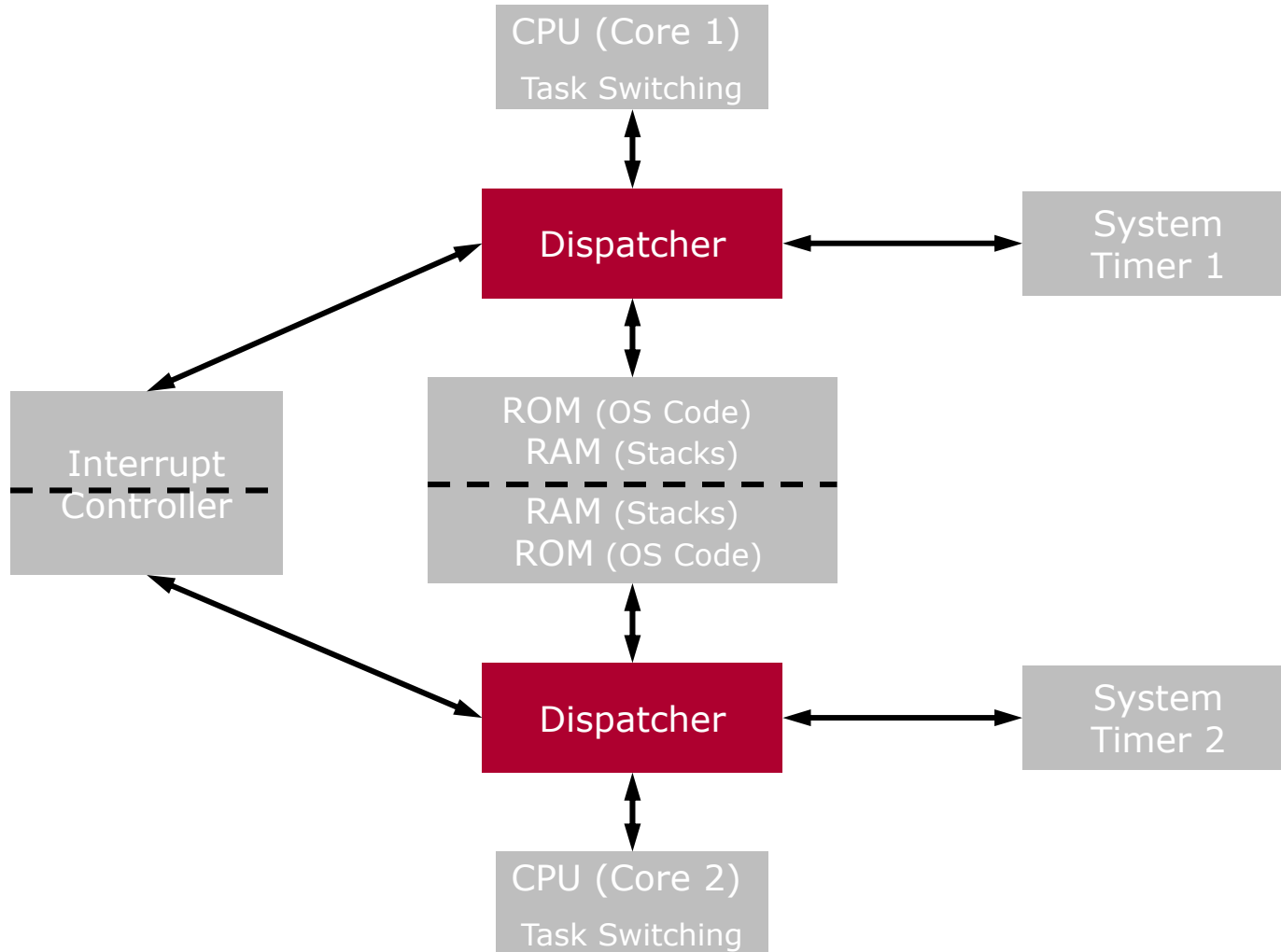
Interrupt

Task



AUTOSAR Multi-Core Operating System

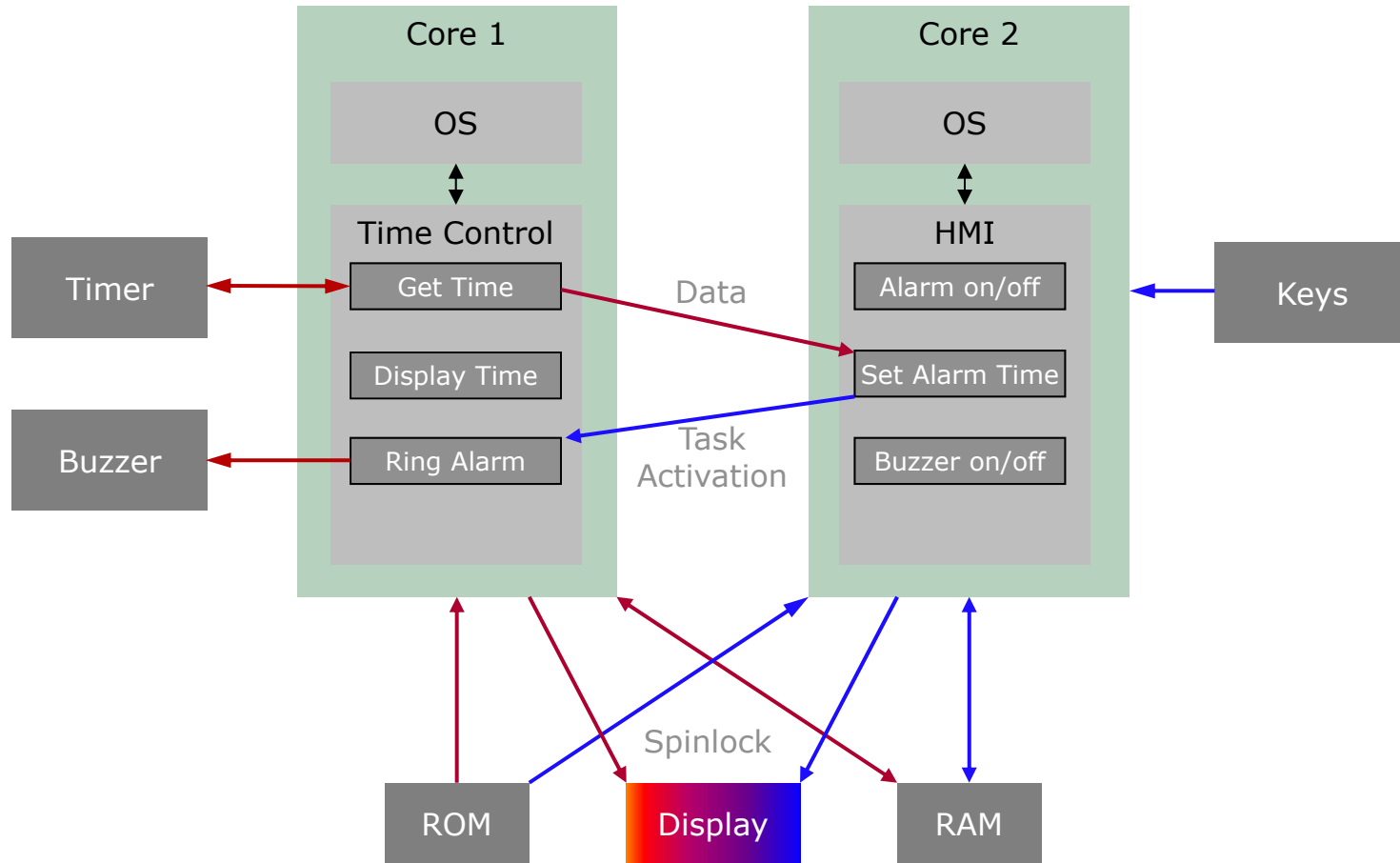
Hardware View



AUTOSAR Multi-Core Operating System

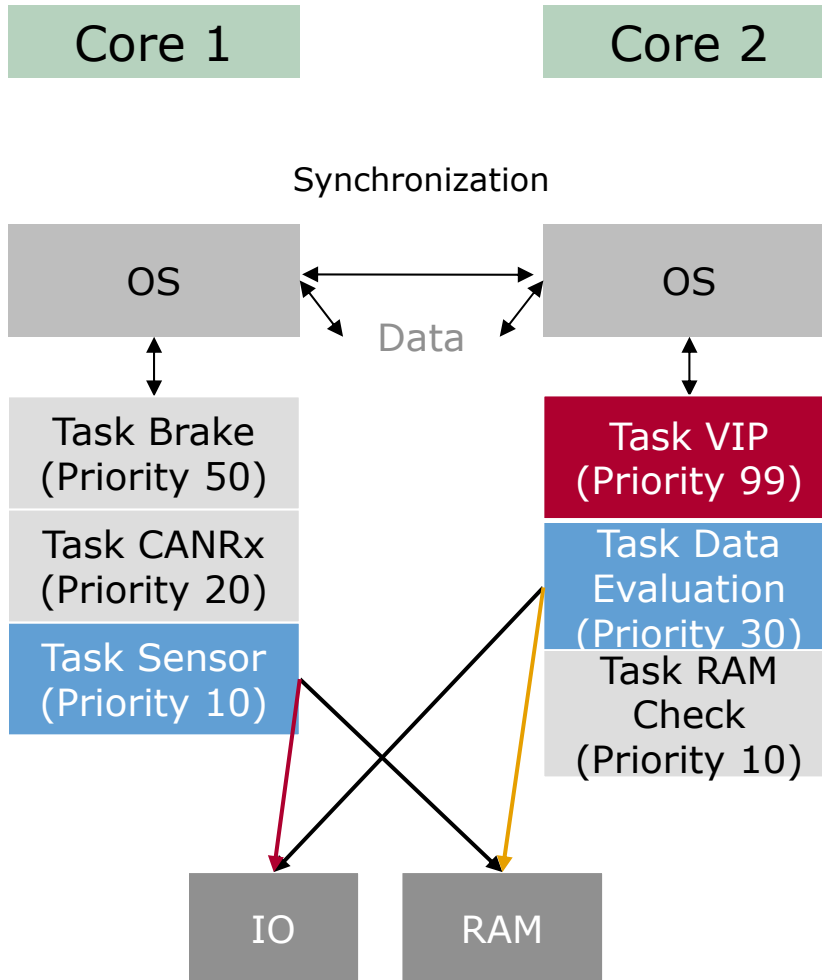
Sample: Alarm Clock

- > Data transfer between cores
- > OS services across core boundaries
- > Shared peripherals (RAM, ROM, Display)

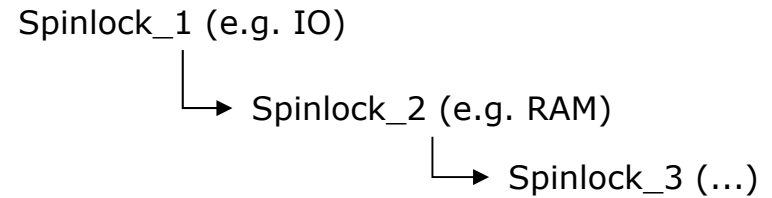


AUTOSAR Multi-Core Operating System

Avoidance of Deadlocks



Configuration of Spinlock Hierarchy



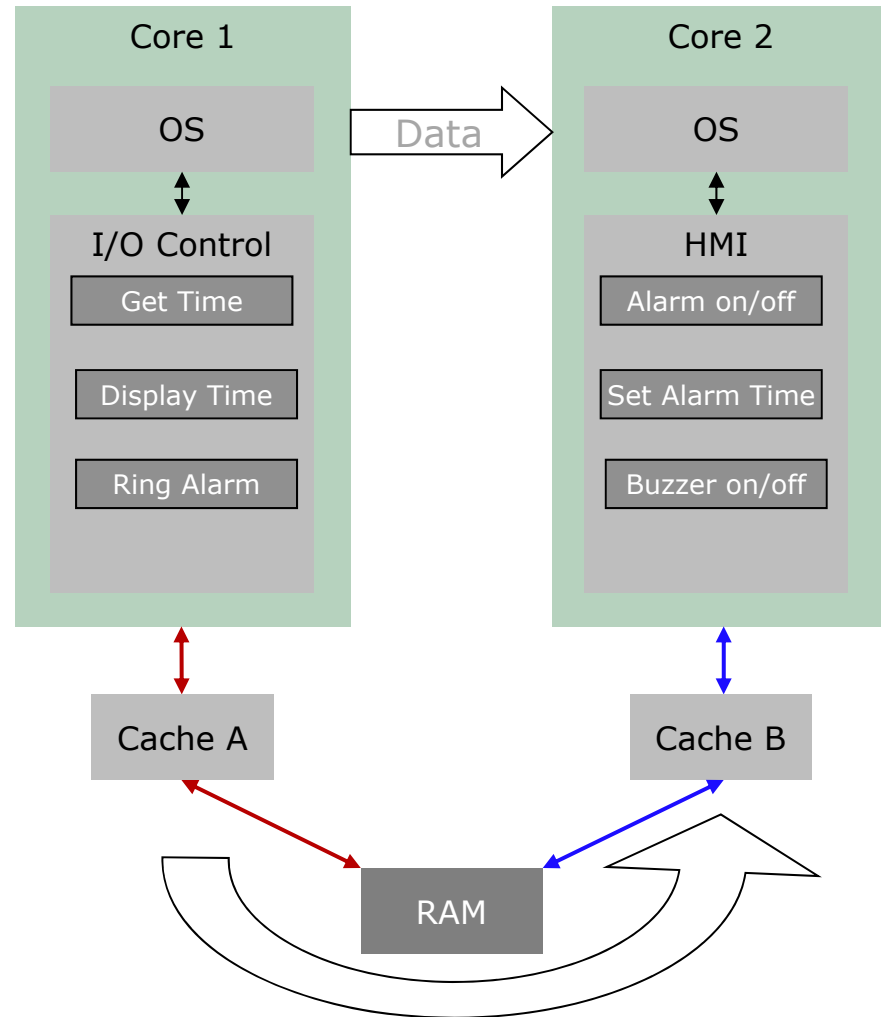
Task	Status	Spinlock
TaskData	active	Get Spinlock IO
TaskSensor	active	Get Spinlock RAM
TaskData	waiting (*1)	Get Spinlock RAM
TaskSensor	Not permitted!	Get Spinlock IO
TaskSensor	active	Release Spinlock RAM
TaskData	active	Get Spinlock RAM

*1 = Task RAM is blocked, Task VIP can become active

Multi-Core Performance

OS service

IOC
Inter OS application Communication



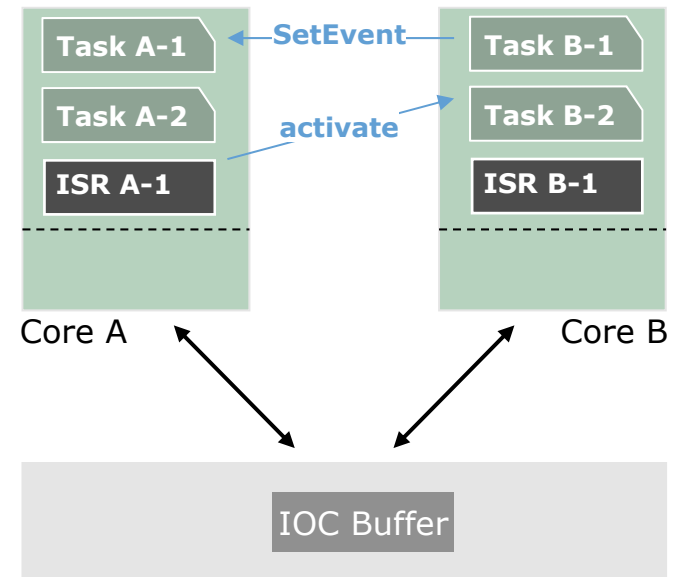
Global variable

- access coordination (Spinlocks)
- content synchronization

AUTOSAR Multi-Core Operating System

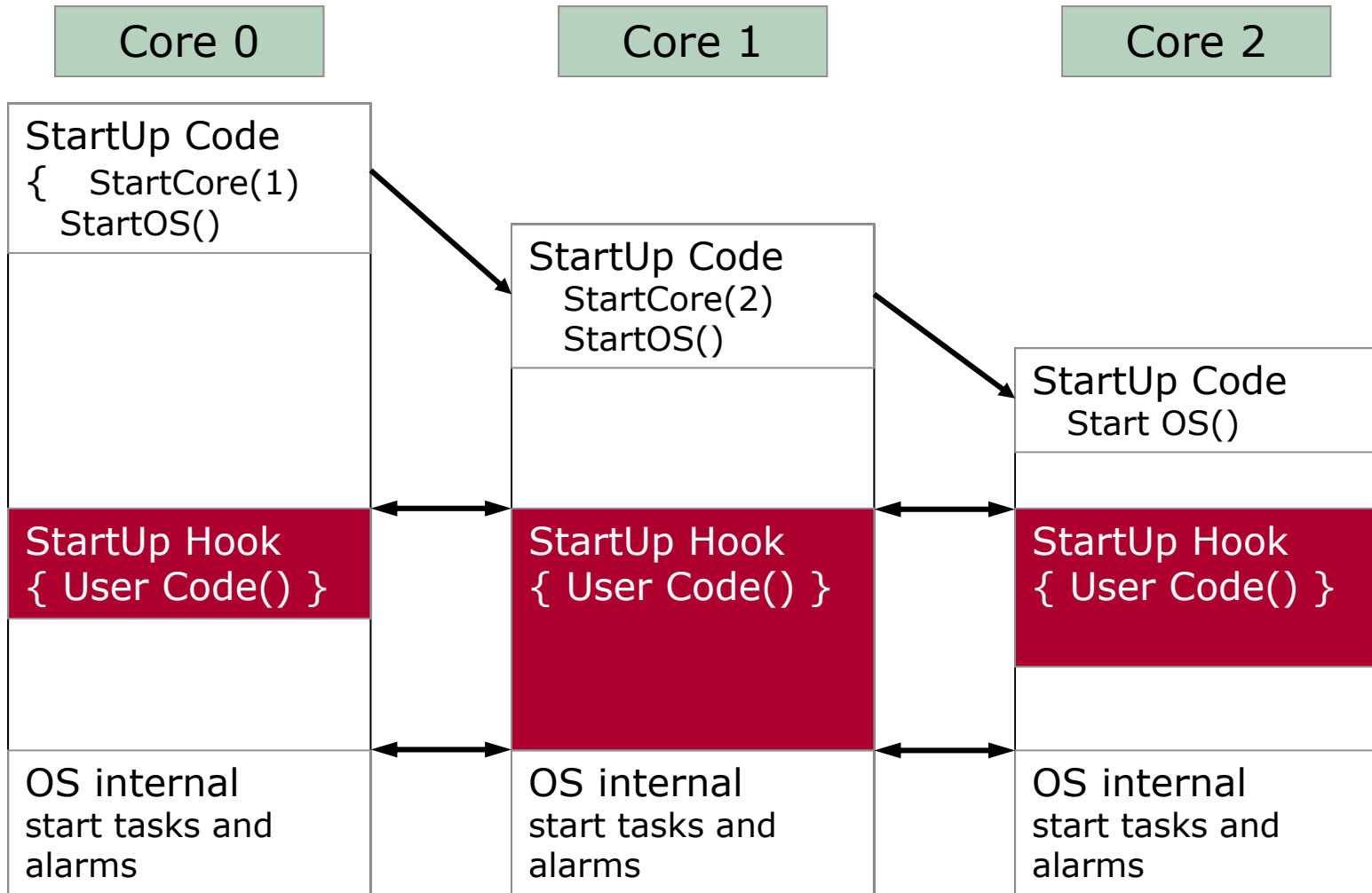
Operating System View

- ▶ Static assignment to a core
 - > Tasks
 - > Interrupts
 - > Alarms & Counters
 - > Resource
- ▶ Inter-Core OS Services
 - > Task Activation
 - > Set / Get Event
 - > Start / Cancel Alarms
 - > Start / Stop ScheduleTables
- ▶ Coordination of access to shared resources by
 - > Resource (Intra-core)
 - > Spinlocks (Inter-core)
- ▶ Consistent Data exchange with OS control
 - > **I**nter **O**S-**A**pplication **C**ommunication
- ▶ Start-up synchronization of all cores



AUTOSAR Multi-Core Operating System

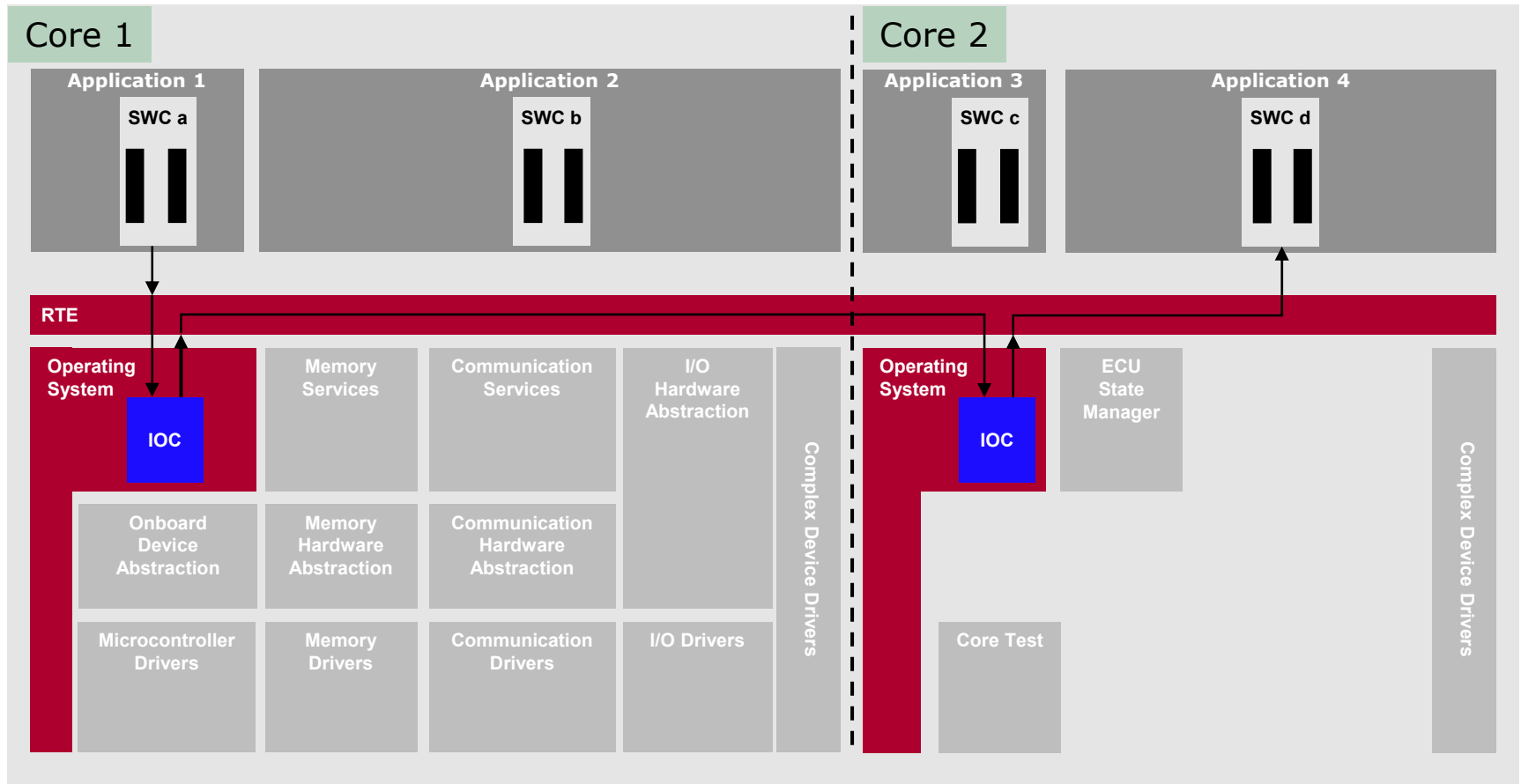
Start-up Behavior



Design Hints

AUTOSAR Multi-Core Operating System

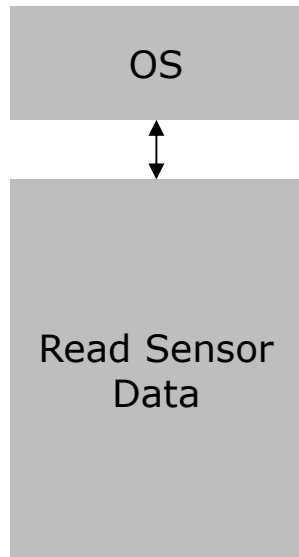
AUTOSAR Architecture



AUTOSAR Multi-Core Operating System

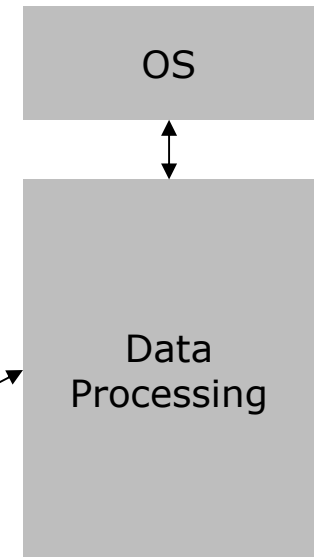
Design

Core 1



- ▶ Execution Time
- ▶ Synchronization points
- ▶ Data transfer between cores
- ▶ Shared resources

Core 2



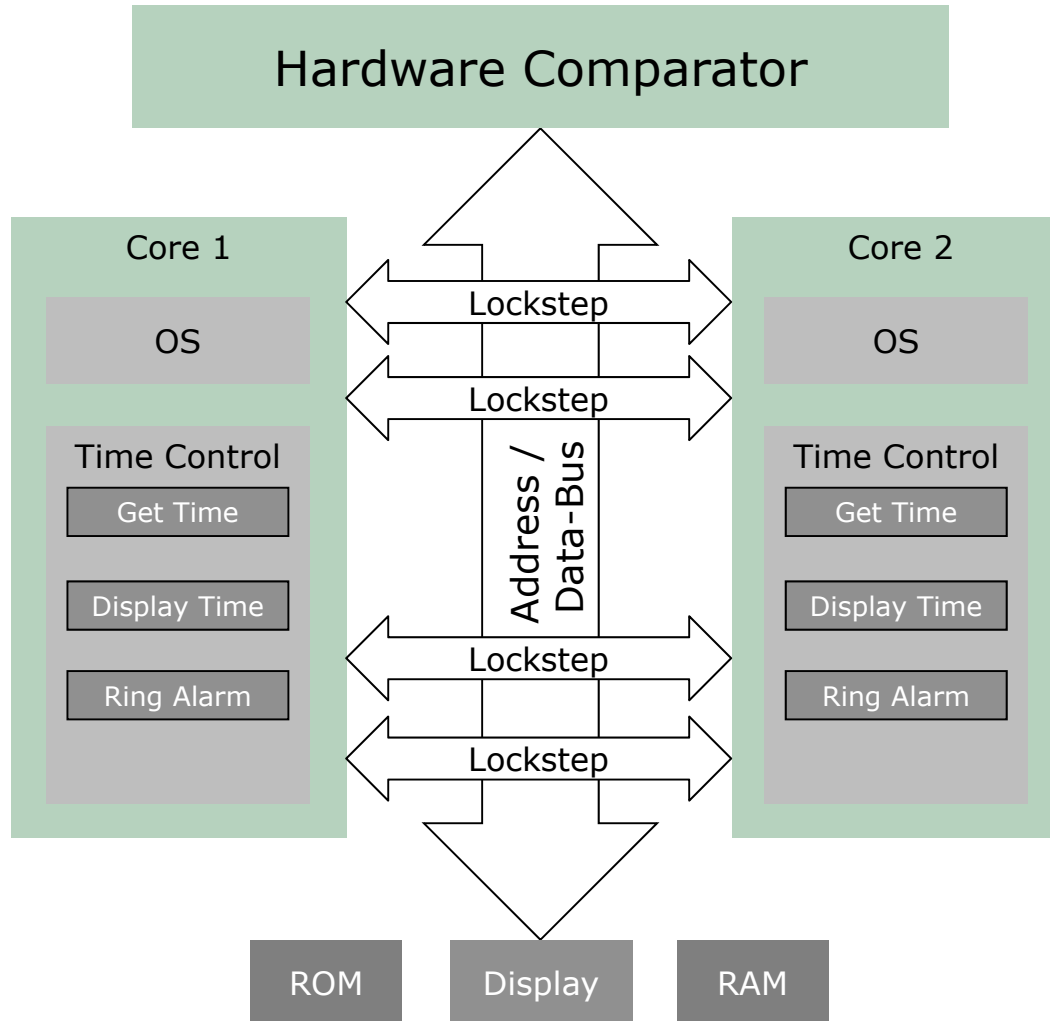
Name	Execution Time	Exchange Data	Sync Requirements
Func A	5%	8 Byte	none
Func B	20%	256 Byte	none
Func C	13%	64 Byte	Func_B



Functional Safety

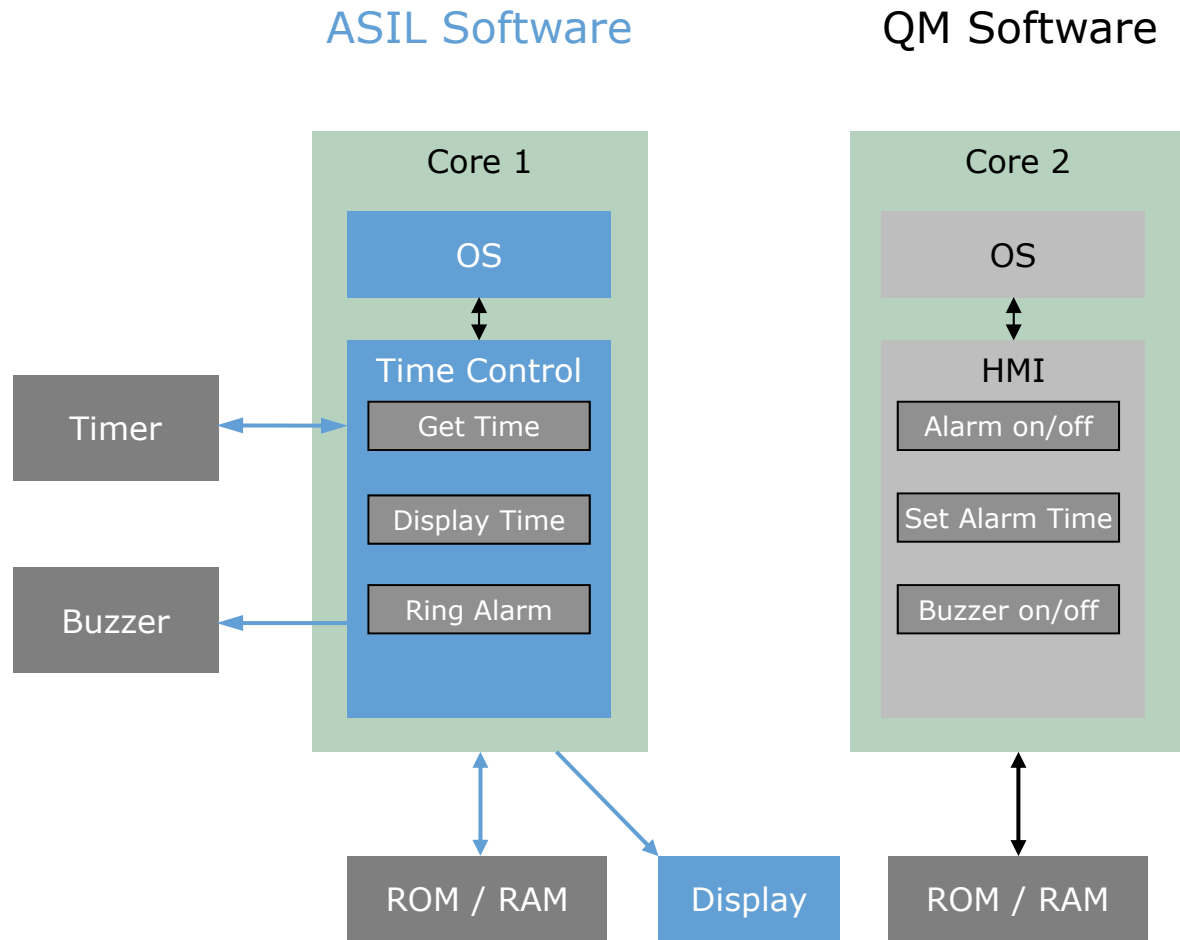
AUTOSAR Multi-Core Operating System

Safety - Redundancy



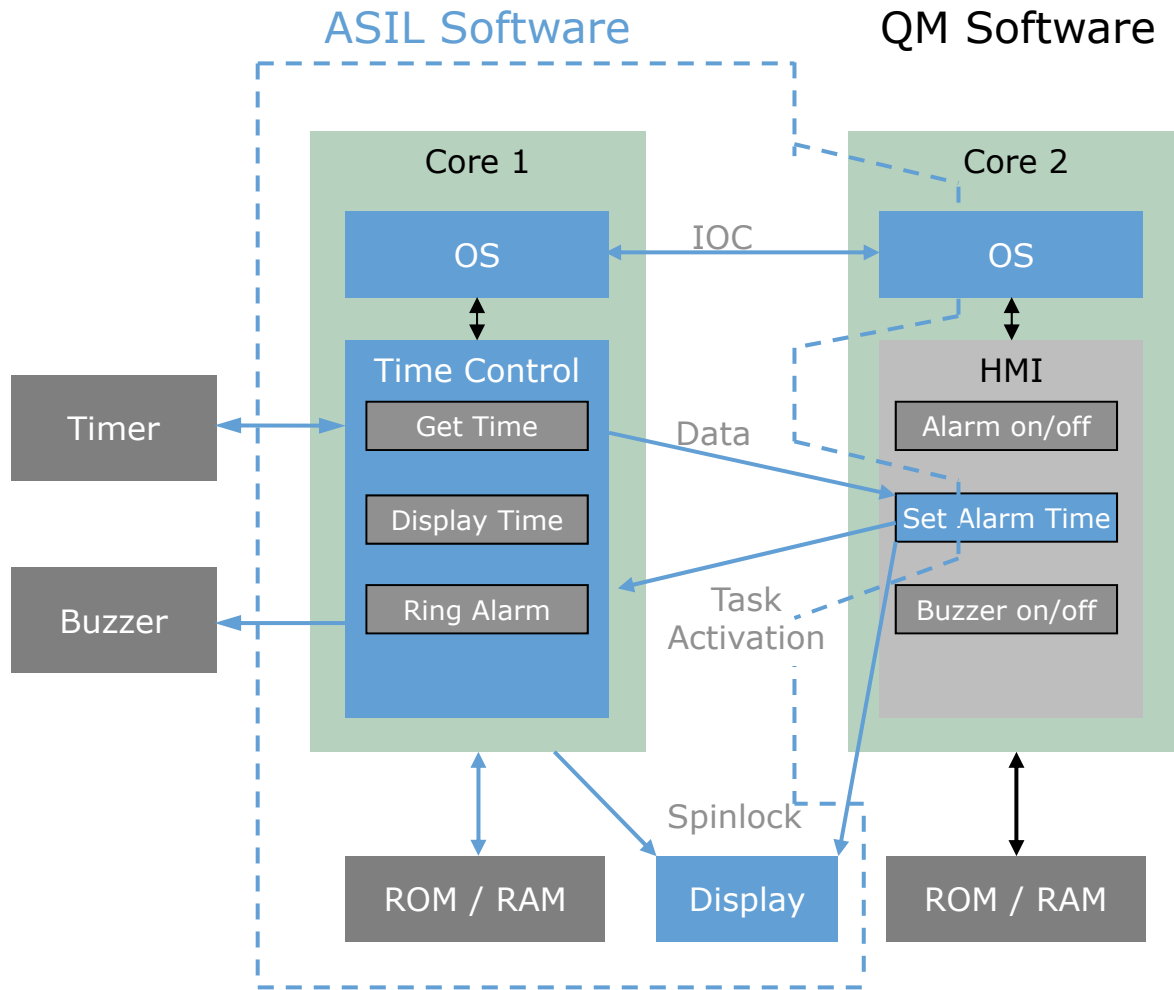
AUTOSAR Multi-Core Operating System

Safety – Freedom from Interference



AUTOSAR Multi-Core Operating System

Safety – Freedom from Interference



AUTOSAR Multi-Core Operating System

Presentation of Configuration

The screenshot displays the Vector DaVinci Configurator Pro interface for the project 'DLCOMP_ecuc'. The main window is titled 'Vector DaVinci Configurator Pro - DLCOMP_ecuc' and features a menu bar (File, Edit, View, Project, Build, Tools) and a toolbar. The 'Phase' is set to 'PreCompile'. The 'Project Explorer' on the left shows the project structure: 'DLCOMP_ecuc' containing 'Architecture', 'Str9_Str912faw44', and 'System', with 'Os' selected under 'System'. The main area shows the 'Architecture' view for 'Os', with an 'Alarm' container selected. Below this, there are 'Insert Alarm' and 'Delete Alarm' buttons, and a table listing existing alarms:

Name	Counter
TestAlarm	/Os/SystemTimer
DataLoggerAL...	/Os/SystemTimer

Below the table, the 'Alarm Properties' section is visible, with tabs for 'Autostart' and 'StaticAlarm'. The 'StaticAlarm' tab is active, showing the following configuration:

AlarmName	TestAlarm
Counter	/Os/SystemTimer
AccessingApplication	<Empty>
Action	OsAlarmSetEvent
Event	/Os/EvDelay
Task	/Os/Test

At the bottom, the 'Messages' pane shows a log of system events:

Time	Source	Code	Message
2010-10-11 14:02:16	System	0001	Welcome visbk!
2010-10-11 14:02:16	System	0225	Package license folder set to 'C:\OSSstick\STR9 v5.01\STR9\SIP\Components\'
2010-10-11 14:02:19	System	1000	New project.
2010-10-11 14:02:43	System	1008	Project closed.
2010-10-11 14:02:43	System	1002	Open project.
2010-10-11 14:02:44	System	1013	Project 'DLCOMP_ecuc' opened.

The status bar at the bottom indicates 'Container: Test [0:x]', 'CfgClass: Unknown', and 'Additional Checks: Unique Within Container, Is Valid Short'.

- ▶ OS
 - > How many cores shall be used?
- ▶ Application
 - > Which core shall execute the OS-Application?
 - > Which OS elements (Task, ISR, Alarm, etc.) shall be assigned to the respective OS-Application?
- ▶ Spinlocks
 - > Name of the subsequent spinlock
- ▶ IOC
 - > Configuration of the communication parameters
 - > List of senders

- ▶ The webinar series about operating systems
 - ▶ 2011-10-25 MICROSAR OS - a pre-emptive realtime multitasking operating system
 - ▶ 2011-11-09 TimingAnalyzer – schedulability analysis of task runtime
 - ▶ 2011-11-22 Memory and runtime protection of the MICROSAR OS operating system
 - ▶ 2011-11-29 Introduction into the multi-core operating system from Vector
 - ▶ 2011-12-06 MICROSAR Safe: AUTOSAR basic software for safety-relevant ECUs according to ISO 26262

- ▶ Registration to the upcoming Webinars and the list of recorded Webinars:
http://www.vector.com/vi_webinars_en.html

- ▶ The overview of Vector's training services:
http://www.vector.com/vi_training_en.html

- ▶ We stay online for some more minutes to answer your questions. Please write your questions in the **Q&A window** and submit them to **all participants**.

- ▶ Contact data for additional questions, product information or presentation :
 - ▶ helmut.brock@vector.com
 - ▶ +49 (0) 711 80670 385
 - ▶ embedded@de.vector.com

Thank you for your attention.

For detailed information about Vector
and our products please have a look at:

www.vector.com

Author:

Dr. Helmut Brock

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart