# Table of Contents

1. Operating Systems for ECUs ................................................................. 3
2. MICROSR.OS – The Real-Time Operating System for the AUTOSAR Standard ................................................................. 3
   2.1. Overview of Advantages .............................................................. 3
   2.2. Properties ................................................................................. 4
   2.3. Optional Extensions ................................................................. 4
   2.3.1. Synchronization with the global system time ......................... 4
   2.3.2. Memory protection (SC3, SC4) .............................................. 4
   2.3.3. Timing Protection (SC2, SC4) .................................................. 4
   2.3.4. Execution time measurements (SC2, SC4) .............................. 4
   2.3.5. High Resolution Timer ........................................................... 4
   2.4. Operating system for applications according to ISO 26262 ........... 4
   2.5. Multi-core Operating System ..................................................... 5
   2.6. Graphical configuration and generation tool ............................... 5
   2.7. Scope of Delivery ..................................................................... 5
3. osCAN - The real-time operating system based on the OSEK/VDX standard ................................................................. 6
   3.1. Overview of advantages ............................................................ 6
   3.2. Application Areas ................................................................... 6
   3.3. Functions ................................................................................. 6
   3.4. Option: "High Resolution Timer" - Improve performance with high-precision timers for operating systems .................. 6
   3.5. Configuration .......................................................................... 7
   3.6. Scope of Delivery ................................................................... 8
   3.7. Availability .............................................................................. 8
4. Additional Information ...................................................................... 8

Please consider the environment before printing this document.

V1.4 June 2015
1 Operating Systems for ECUs

The usage of an operating system supports you in the realization of complex tasks regarding processing, steering and control. Further than this, it helps you to achieve a much better expandability, possibility to port and re-use of your application software.

Vector offers operating systems for all standards:

> Operating system according to the AUTOSAR standard

MICROSAR.OS is the AUTOSAR-conformant operating system from Vector. It is backward compatible to the OSEK-/VDX-standard and supports all scalability classes, thus also time synchronization and memory protection. Numerous extensions are available as an option, e.g. variants certified according to ISO26262 / ASIL D.

> Operating system according to the OSEK/VDX standard

As supplier of the first certified OSEK/VDX conformant operating system worldwide, Vector bears longtime experience in realization and implementation of this standard.

2 MICRO SAR.OS – The Real-Time Operating System for the AUTOSAR Standard

MICROSAR.OS is a pre-emptive real-time multitasking operating system with optimized properties for use on microcontrollers. Vector’s many years of experience in developing operating systems and drivers for microcontrollers are bundled into this small, robust operating system core.

![Figure 1: MICRO SAR.OS module according to AUTOSAR 4.2](image)

2.1 Overview of Advantages

> Small, quick, resource-economizing operating system and with short boot times

> MICRO SAR.OS is available for AUTOSAR 4.x and 3.x
> Optional: Available as multi core operating system
> Optional: Safe context switching per ISO26262 / ASIL-D
> Quality process per SPICE Level 3
> Graphic configuration tool for easy configuration
> Available for many 8, 16, 32 and 64 bit microcontrollers

### 2.2 Properties

MICROSAR-OS is based on the AUTOSAR OS specification, an extension of the practice-proven OSEK/VDX-OS operating system standard. This standard was extended to include functions for time monitoring and memory protection.

MICROSAR. OS is fully conformant to the AUTOSAR OS specification and supports all scalability classes:

> SC1: Real-time operating system implemented per the OSEK/VDX-OS standard and extended to include schedule tables
> SC2: Real-time operating system with time synchronization and monitoring of the time behavior of individual tasks and interrupt service routines
> SC3: Real-time operating system with memory protection mechanisms on microcontrollers with suitable hardware support
> SC4: Combines scalability classes SC2 and SC3

### 2.3 Optional Extensions

#### 2.3.1 Synchronization with the global system time

Schedule tables can be synchronized with the global system time, e.g. the time transmitted over the FlexRay bus. This makes synchronized and simultaneous execution of tasks possible in a distributed system.

#### 2.3.2 Memory protection (SC3, SC4)

Memory protection assures that application components do not mutually destroy data. This makes the integration of applications easier and more reliable.

#### 2.3.3 Timing Protection (SC2, SC4)

Timing protection ensures that assumptions made in the early design phase related to execution times are preserved during run-time as well. As a result, a defective application section cannot impair the execution times of other running processes.

#### 2.3.4 Execution time measurements (SC2, SC4)

You can use functions of scalability classes 2 and 4 to measure the execution times and interrupt disable times of applications. These measured data can later be used as practice-based values in designing and integrating future applications.

#### 2.3.5 High Resolution Timer

The high resolution timer mechanism offers time resolutions of less than 1ms without increasing the interrupt load. Depending on the controller, this may enable resolutions into the microseconds range.

### 2.4 Operating system for applications according to ISO 26262

For safety-relevant applications per ISO 26262, you can obtain Vector’s operating system variant MICROSAR.OS SafeContext that was developed according to ASIL D. It is based on the AUTOSAR scalability classes SC3 and SC4 and is responsible for memory protection and safe context switching. To preserve “freedom from interference” with regard to memory protection, you need a suitable processor, e.g. with a Memory Protection Unit (MPU).
You can use MICROSAR.OS SafeContext to use safety-relevant application components together with standard components on the same CPU.

Scope of delivery for MICROSAR.OS SafeContext:

- Operating system core as source code
- Graphic configuration tool for Windows 7 / Windows XP
- Command line based generator
- BSW Module Description
- Description files for DaVinci Configurator Pro
- Documentation
- Read-back tool
- Safety Manual

For more details on the Vector solution for safety, please refer to the separate chapter about MICROSAR.Safe.

2.5 Multi-core Operating System

MICROSAR.OS Multi-core is an advanced development of the proven real-time operating system MICROSAR.OS from Vector. You can use it wherever a multi-core system is to be developed according to the AUTOSAR specification. MICROSAR.OS Multi-core is based on AUTOSAR specification 4.x, but it can also be used in AUTOSAR 3.x projects.

The multi-core operating system enables parallel and independent operation of multiple processor cores, each with a separate instance of the AUTOSAR operating system. Thereby, its configuration and system services match those of the single-core operating system. The extensions SC2 to SC4 and High Resolution Timing are available. In addition, MICROSAR.OS Multi-core offers mechanisms for coordinating and synchronizing tasks that run on different cores:

- Synchronized Start-Up: The operating system assures that all cores are started up and initialized before the associated application is started.
- Inter-Core Coordination: Processes on different cores are synchronized by task activations, setting of events, starting and stopping of alarms or ScheduleTables.
- Shared Resource Access: If different cores should access commonly used resources, the operating system provides a coordination mechanism known as Spinlocks.
- Inter-Core Communication: For consistent data exchange between two cores, the operating system provides an efficient interface in the form of the Inter-OS Application Communicator (IOC).

2.6 Graphical configuration and generation tool

For a comfortable configuration we recommend the DaVinci Configurator Pro. It contains consistency checks and the call of the generator. The generator is implemented as a command-line tool to enable its integration in an automated development environment.

2.7 Scope of Delivery

The MICROSAR.OS product is supplied with the following components:

- Operating system core as source code
- Graphic configuration and generation tool for Windows XP/ Windows 7
- DaVinci Configurator Base as basic editor
- Command line based generator
- BSW modules description
- Documentation
3 osCAN - The real-time operating system based on the OSEK/VDX standard

osCAN is a preemptive real-time multitasking operating system with properties optimized for use on microcontrollers. Vector’s many years of experience in developing operating systems and drivers has made osCAN a small and robust operating system core. It is certified to OSEK/VDX specification 2.2. This open standard of the automotive industry provides a foundation for long-term stability and availability.

3.1 Overview of advantages

- Small, fast and resource-saving operating system core
- Short boot times
- Stack monitoring at runtime (deactivatable)
- Internal fault trace (deactivatable)
- Component management for integration of sub-projects
- Supplied as source code
- Quality process fulfills SPICE Level 3
- Graphic configuration tool for easy configuration of the operating system
- Available for many 8, 16, 32 and 64 bit microcontrollers

3.2 Application Areas

Due to its static operating system architecture, osCAN is small and fast, and its use is beneficial wherever resources such as memory and processing time are in short supply. osCAN is especially ideal when extremely short boot times are necessary.

3.3 Functions

As a fully preemptive multitasking operating system, osCAN offers priority-driven task management and a wide variety of synchronization mechanisms. It can generate cyclic and singular time intervals as well as behave in an event-driven way. Its flexible interrupt strategy makes it possible to use system services even within interrupt service routines. In addition to functions defined in the standard, osCAN offers:

- Monitoring of stack utilization at runtime
- Internal tracing of system calls
- Extended error checking at runtime
- Clear and concise documentation of the configuration in HTML

With the optional High Resolution Timer, you can implement resolutions higher than 1ms for the timers, without simultaneously increasing interrupt load. Depending on the controller, resolutions in the microseconds range are possible.

3.4 Option: "High Resolution Timer" - Improve performance with high-precision timers for operating systems

An important property of operating systems is the precision of alarms; this in turn affects the resulting interrupt load. The alarm mechanism of an OSEK/VDX or AUTOSAR operating system is based on counters which are incremented by either software or hardware timers. When a counter reaches a preset alarm value, a pre-defined action is executed. The typical time resolution of such timers is 1 ms.

If timer resolutions with a higher precision than 1 ms are required, the High Resolution Timer mechanism can achieve these enhanced precisions without simultaneously increasing interrupt load. Resolutions in the microsecond range are possible depending on the controller used.
Features and advantages

- The High Resolution Timer for the osCAN and MICROSAR OS operating systems permits high-precision alarms with low system interrupt load (depending on hardware). An interrupt is only triggered when an alarm actually becomes due.

- The precision of the alarm time depends only on the resolution and clock frequency of the hardware timer. It may range from 100 ns to 100 μs depending on the microcontroller.

Application areas

- The High Resolution Timer is designed for applications requiring high-precision timers but in which additional interrupt loads are unacceptable.

- The High Resolution Timer can be used on all microcontrollers which have a free-running timer.

- The timer must also be able to trigger an interrupt when its value matches a user-definable compare value (compare-match). The clock speed at which the hardware timer is operated should be adjustable so that the desired level of accuracy is achieved.

Functions

- “Collective” processing of several consecutive alarms can be controlled by defining a minimum interval between timer interrupts. This reduces interrupt load, especially when several alarms with similar intervals expire almost simultaneously.

Configuration

- The High Resolution Timer is configured using the configuration tool for the Vector operating systems. This tool is used to adjust the prescalers according to the clock frequency of the microcontroller.

3.5 Configuration

An easy-to-use graphic configuration tool lets you define all operating system resources before compiling. To assure portability, the configuration is saved in the standardized OIL format (OSEK Implementation Language).

The configuration tool offers component management, which lets you implement sub-applications simultaneously in parallel processes. Next, the sub-applications are integrate into a total application. This enables short development times and spatially separate developments. It offers the following properties:
> Easy to operate graphic user interface
> Automatic checking for completeness and consistency
> Convenient system scaling
> Component management
> ORTI support: Many emulators already offer the option of observing application behavior on the operating system services level (OSEK Awareness). osCAN supports the ORTI interface required for this. Some UML and SDL modeling tools give you the option of configuring models for use with an OSEK/VDX operating system. This is possible, for example, with:
  > MATLAB/Simulink from Mathworks
  > TargetLink from dSPACE

3.6 Scope of Delivery
> Operating system core as source code
> Graphic configuration tool (OIL) for Windows 7
> Instruction line based generator
> Documentation
> Sample programs

3.7 Availability
The real-time operating system osCAN from Vector is available for many commonly used 8, 16 and 32 bit microcontrollers. For more information on the Internet please go to: www.vector.com/oscanscan. osCAN conforms to OSEK/VDX specifications
> OSEK/VDX-OS 2.2
> OSEK/VDX-OIL 2.3
> OSEK/VDX-COM 2.2.2
> All conformity classes (BCC1, BCC2, ECC1, ECC2, CCCA, CCCB)
> Size: 1–10 kB ROM, depending on platform and configuration

4 Additional Information
Vector provides you a completely tuned solution for developing ECUs. Along with the operating system, you get protocols and drivers for CAN, LIN or FlexRay. For more detailed information, please refer to the relevant Product Information.

For further information on our products and our configuration tool DaVinci Configurator Pro, please see our internet site http://vector.com/vi_embedded_software_en.html.
Get more Information!

Visit our Website for:
> News
> Products
> Demo Software
> Support
> Training Classes
> Addresses

www.vector.com