

Timing, memory protection and error detection in OSEK systems

by Peter Liebscher, Vector Informatik

A powerful certified OSEK implementation or AUTOSAR-conformant embedded operating system, together with a universal tool-chain, will make it possible to master the complexity of future electronic development.

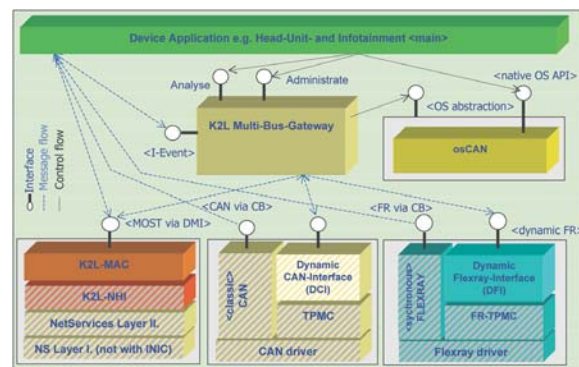


Figure 1. Representation of gateway, system and bus APIs

■ The real-time and multitasking operating system OSEK is the standard in automotive embedded developments today. Its most important properties include low consumption of processor resources and memory, and event-driven task management that effectively handles both cyclic and non-cyclic program blocks. Continuing advances in automotive electronics and the new HIS and AUTOSAR standardization initiatives also place new demands on the operating system. Areas of emphasis in future operating system versions will be timing and memory protection.

In the 1990s the large automotive OEMs introduced the OSEK/VDX operating system specification with the goal of establishing a uniform standard for software in electronic control units. The wide variety of proprietary embedded operating systems at numerous suppliers had proved to be an obstacle to smooth integration in light of the growing significance of automotive electronics. Besides defining the actual operating system core, OSEK also defines communication services and functions for network management.

Since most automotive suppliers had already committed to a preferred operating system beforehand, automotive OEMs had to introduce OSEK persuasively in some cases. Daimler-Chrysler, for example, made OSEK mandatory

as a standard for new developments, both for in-house developments and those at suppliers. The company organized OSEK training courses, created OSEK design guides and supported operating system producers. It also financed one OSEK licence per supplier, whereby only certified OSEK operating systems were permitted. The costs of introduction reached a high point in 2002 and then dropped significantly. In the meantime OSEK has generated its own success, and now most ECUs in the automotive field run on OSEK operating systems.

Efforts have paid off: applications based on OSEK have fulfilled expectations by their improved code quality, structuring and the ability to integrate components from different suppliers. At gateway producer K2L solutions with osCAN, the OSEK/VDX-conformant operating system from Vector Informatik, have demonstrated fast reaction times and precise timing. Last but not least, what has proven to be decisive for OSEK in conjunction with a table-driven interpreter concept are flexibility gains leading to shorter development times, higher quality and functionality and ease in creating variants. Leading the way here were comparisons between OSEK with its preemptive scheduling and a fixed coded static approach with cooperative scheduling. The "osCAN" OSEK implementation has proven itself, not only in ECUs but also in the interface hardware

of the same company. In the MOST Interface VN2600 osCAN's capabilities were put to the test under a 133 MHz Altera Excalibur controller: its processing of up to 35,000 Events/s corresponds to a data throughput of 1.7 MB/s.

Requirements of an operating system grow in parallel with the continuing penetration of electronic technologies. In particular the advance of safety-relevant applications in the vehicle, such as fully electronically controlled steering and braking systems (X-by-wire), make deterministic behaviour essential under peak loads and fault conditions. For example, the specification of OSEKtime will supplement the event-driven OSEK with a time-triggered variant.

Fault tolerance, error detection mechanisms and memory protection also play an important role in achieving system reliability. This has acquired special relevance, because in the future an ECU will handle multiple applications running simultaneously. In "Hersteller Initiative Software" (HIS; Hersteller = producer or OEM) the large German automotive OEMs have come to an agreement on the standards needed to implement the named functions. Working committees have been established in the areas of software, software testing, process assessment, simulation and tools as well as flash programming. The AUTOSAR (automotive open system architecture) consortium is responsible

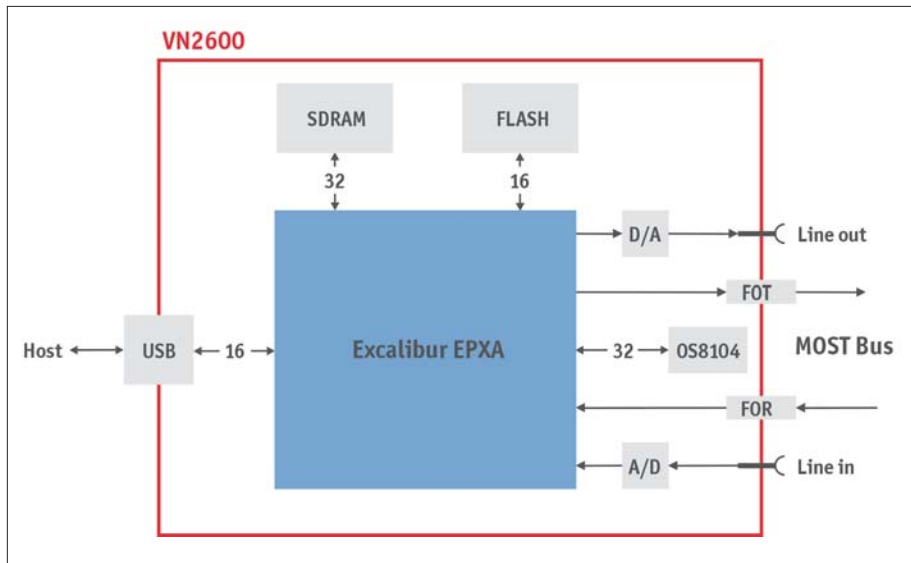


Figure 2. Function block diagram with MOST interface VN2600 and an Altera Excilibur controller



Figure 3. Deadline monitoring to detect deadline violations. The error source of Task 2 is not found in Task 1.

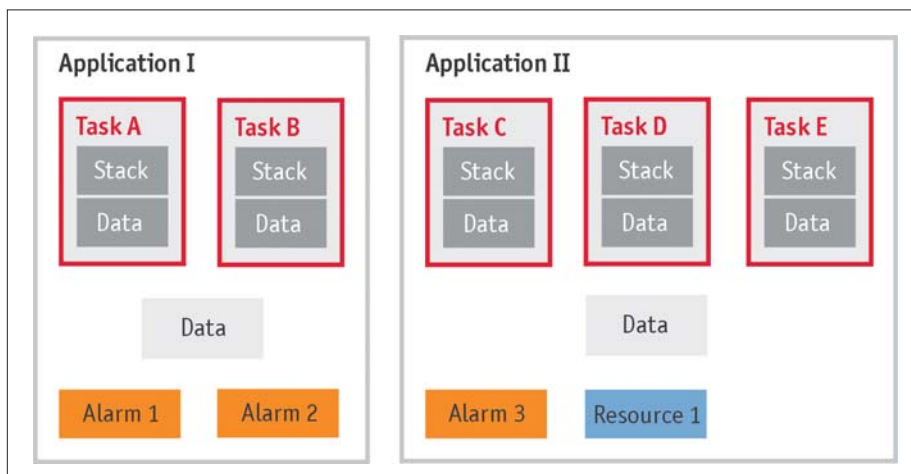


Figure 4. Tasks belonging to the same application must be able to access the same memory areas.

for the latest standards for future vehicle generations, whereby the HIS group is integrating its results into AUTOSAR and is representing uniformity interests there. When one considers that over 50 ECUs operate on a luxury automobile today and that there appears to be no end to potential new automotive electronic applica-

tions in the foreseeable future, it becomes clear why limiting the number of control modules in the vehicle has become a pressing topic. This objective can only be achieved if multiple applications run on the same control module. The multitasking OSEK operating system was specified for this purpose. However, other

properties are required of the operating system for use in safety-critical systems and to integrate the software of different producers. For example, an application must not disturb other applications running in parallel. To prevent this from happening, new operating system features are aimed at optimal monitoring of the time behaviour of individual tasks and universal memory protection.

Progress has varied considerably in the various development levels as a result of these efforts. OSEKtime, specified since 2001 for time-triggered tasks, with its deadline monitoring, only begins to cover functions that will be necessary in the future. Deadline monitoring detects whether a task has ended by a prescribed point in time. Unfortunately, the method cannot discern the causes for deadline violations. For example, if the monitored task was interrupted by a higher-priority task, then the monitored task is not really responsible for being unable to satisfy the prescribed time window. In HIS-conformant OSEK extensions memory protective functions are defined in addition to deadline monitoring. The most advanced is AUTOSAR with its “execution time enforcement” and “arrival rate enforcement” methods, which give low-priority tasks a mandatory minimum time budget, too. These methods are able to clearly identify error sources. Furthermore, in the various operating system versions, Vector Informatik has integrated options for run-time measurements.

Memory protection functions restrict a task’s access to the memory space assigned to it. This applies especially to preventing write accesses to other data segments, detecting stack overruns, and detecting execution of incorrect code. Tasks belonging to the same application, on the other hand, must be able to jointly access the same memory areas. However, special system functions such as drivers could require full unrestricted memory access.

A distinction is made here between so-called “trusted applications” with full access rights and “non-trusted applications” with restricted access rights. These names can lead to some confusion: non-trusted applications are programs that, due to restricted memory access, cannot cause any damage. The trusted applications, on the other hand, are given quasi blind trust. The latter are easy to use but represent risks to system security and cannot provide identification of errors or error sources. It is good design practice to place functionalities in non-trusted applications whenever possible. Vector Informatik therefore offers implementations that also permit calling of non-trusted functions. They are intended for safety-critical applications and offer a maximum level of monitoring. A related proposal for handling this issue in

AUTOSAR is currently being discussed in a working sub-committee. The cited timing and memory monitoring functions can only be implemented efficiently with suitable hardware support. What are needed for memory protection are memory protection units (MPUs) that are tailored to the needs of automotive applications in terms of options offered for number and sizes of memory blocks. In many cases today the smallest units that can be managed are blocks 16 KB in size. In the automotive embedded field, however, substantially smaller memory units are needed. Essentially,

the hardware requirements of current and future OSEK real-time systems can only be fulfilled by a complete redesign of today's processor cores. Desired features are currently being designed together with semiconductor producers. Among the most important requirements, besides the named monitoring functions, are an interrupt controller for different interrupt levels with low latency times, hardware support in task switching, and processor cores with as few registers as possible. For hardware-related and time-critical automotive applications what counts is the ability to react as quickly as possible.

Many of these applications consist of drivers and interrupt service routines (ISRs) which in contrast to the workstation field belong to the application here. It is problematic that on today's controllers the ISRs often can only be disabled completely. In general, disabling mechanisms must be made more efficient in implementations, since this basic

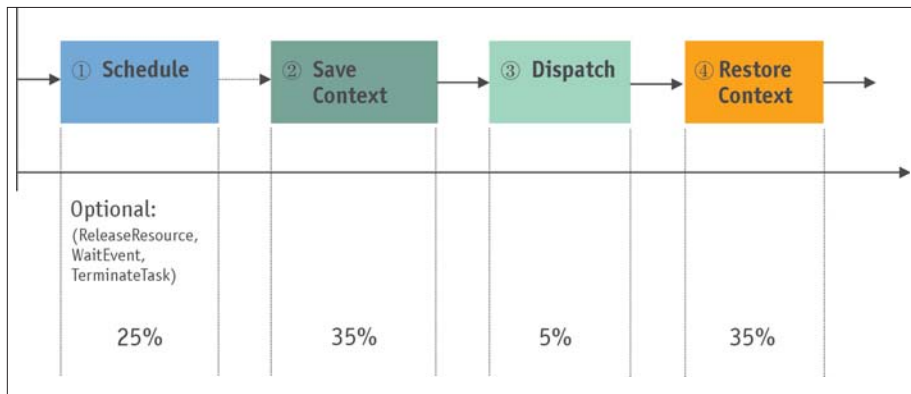


Figure 5. Phases of a task switch

Click-for-More

Interested in more information?
 Visit our specific website with links to:

- ▶ Real-time operating system for automotive control units
- ▶ Protocols and drivers for CAN communication
- ▶ Flash programming via CAN and LIN
- ▶ Solutions for AUTOSAR by Vector

Simply type-in Reader Service #: **783** at Embedded-Control-Europe.com/know-how

Embedded Control europe

the online information for design engineers

[product news](#) [companies](#) [events](#) [know-how](#) [newsletter](#) [ICOMedia](#)
[e-ce](#) [time](#) [b&S magazine](#) [contact](#) [search](#) [home](#)

Read service

Type in know-how ID from ECE magazine

Analysis tool and bus interface for FlexRay networks
 Vector Infomatik supplies tools for an optimized analysis of FlexRay networks. These are the FlexRay Option of the worldwide-used CANoe development tool and FlexCard, a compact bus interface.

[read more](#)

function is executed very frequently. The quick task switches by which real-time embedded systems “thrive” are currently given just rudimentary support in hardware. The majority of resources is consumed in saving and restoring the context. The context is made up of core registers, register banks, memory access registers, floating point and arithmetic registers of the stack pointers, special peripheral units and a number of operating system variables. A fully hardware-supported context switch would be ideal here.

Furthermore, it has been shown that processors with a low number of registers offer better performance. Many registers can only be used meaningfully in typical workstation environments, because that is where individual program sequences run for a relatively long time without interruption. A potential trend here could lead in the direction of so-called softcore processors and to compilers that permit configuration of the registers used. Interesting possibilities for mastering industrial complexity are being researched in ongoing projects involving the intensive application of mathematics to the engineering sciences. Systematic analytical methods can be used to reveal critical situations in the time behaviour of an OSEK system that would otherwise not be found even by extensive testing. In this context, tools from the SympTAVision company enable targeted searches for “bottlenecks” and “hot spots”, informing users of worst-case situations. The advantages of systematic analysis lie in

reduced testing effort, productivity gains, quality improvement and comprehensive system optimization. Scientists on the Verisoft project have taken this one step further in concluding that it is possible to develop absolutely error-free embedded systems and electronic components. They turn to the methods of formal verification to verify entire systems consisting of hardware, software, operating system communication, applications, etc. In cooperation with project partner Infineon, the TriCore 2 processor, the future flagship of the company’s 32-bit microcontroller device line, offered the first evidence that this innovative technology could be applied to highly complex designs. The long-term Verisoft project set up under the project leadership of Prof. Dr. Wolfgang J. Paul, of the University of Saarbrücken, is being supported by the German Federal Ministry for Education and Research.

Foundations and base technologies have been created for achieving reliable electronic systems in the automobile. Specific challenges must still be overcome by controller producers. Apart from that it is the responsibility of automotive OEMs and suppliers to utilize the available resources optimally. A powerful certified OSEK implementation or AUTOSAR-conformant embedded operating system, together with a universal tool-chain, will make it possible to rationally master the complexity of future electronic development. ■