

Trends in Embedded Development

Requirements and future concepts in hardware, software and tools

In the future continual advances in the development of automotive electronics will place significant new demands on underlying base technologies. In spite of growing functionality automotive OEMs must keep costs under control; one way to achieve this is by limiting the number of ECUs in a car. At the same time extended safety and reliability concepts will be key areas of interest. In light of the challenges and multitude of electronic components, powerful tools for development, data management and transfer of software modules into a control module's flash memory will continue to gain in importance.



Without further advances in standardization it no longer be possible to master the growing complexity of automotive electronics. In the 1990s automotive OEMs such as Daimler-Chrysler went to great efforts to establish the OSEK^[1] embedded operating system as a binding standard for in-house developments and supplier components. Today the real-time and multitasking capable operating system is used by automotive OEMs and suppliers as the basis for improved code quality, good structuring and integration of the components of different suppliers. In "Hersteller Initiative Software" (HIS)^[2] too the large automotive OEMs have come to an agreement on uniform standards. Working committees have also 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 for the standards of future vehicle generations.

OSEK: Flexible and calculable

A number of OEMs have certified OSEK implementations. Applications of the "osCAN" OSEK implementation, for example, range from normal ECUs to multi-bus gateways and interface hardware. In the Vector Interface for MOST VN2600 the performance capabilities of osCAN were put to the test under a 133 MHz Altera Excalibur controller processing up to 35,000 Events/s, which corresponds to a data throughput of 1.7 MByte/s. At gateway producer K2L osCAN-based solutions have demonstrated fast reaction times and precise timing.

An ECU for multiple applications

A clear trend in automotive networking is reduction in the number of ECUs in a vehicle. Today up to 40 ECUs operate in a luxury automobile. To permit implementation of even more functions, in the future consistent efforts must be made toward running as many applications as possible within the same control module. The OSEK multitasking 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.

Always in demand: The right timing

One of the focal points in advanced development of embedded operating systems, for future OSEK versions and AUTOSAR, is the ability to monitor software behavior relating to timing and memory accesses. The most advanced methods for monitoring timing are provided by AUTOSAR-conformant implementations. Methods such as "Execution Time Enforcement" and "Arrival Rate Enforcement" provide a mandatory minimum time budget for low-priority tasks too; these methods not only detect errors, they can also clearly identify their sources.

Reliable boundaries for memory protection

In the future memory protection functions will restrict a task's access to the memory space assigned to it. This is especially important to prevent write accesses to other data segments, detect stack overruns, and prohibit execution of

incorrect code. On the other hand, tasks belonging to the same application need to be able to access the same memory areas, and special system functions such as drivers could require 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 stored programs that cannot really cause any damage.

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 reliability. A related proposal for handling this issue in AUTOSAR is currently being discussed in a working subcommittee.

Better hardware support in the future?

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 kByte in size. In the automotive embedded field, however, substantially smaller memory units are needed.

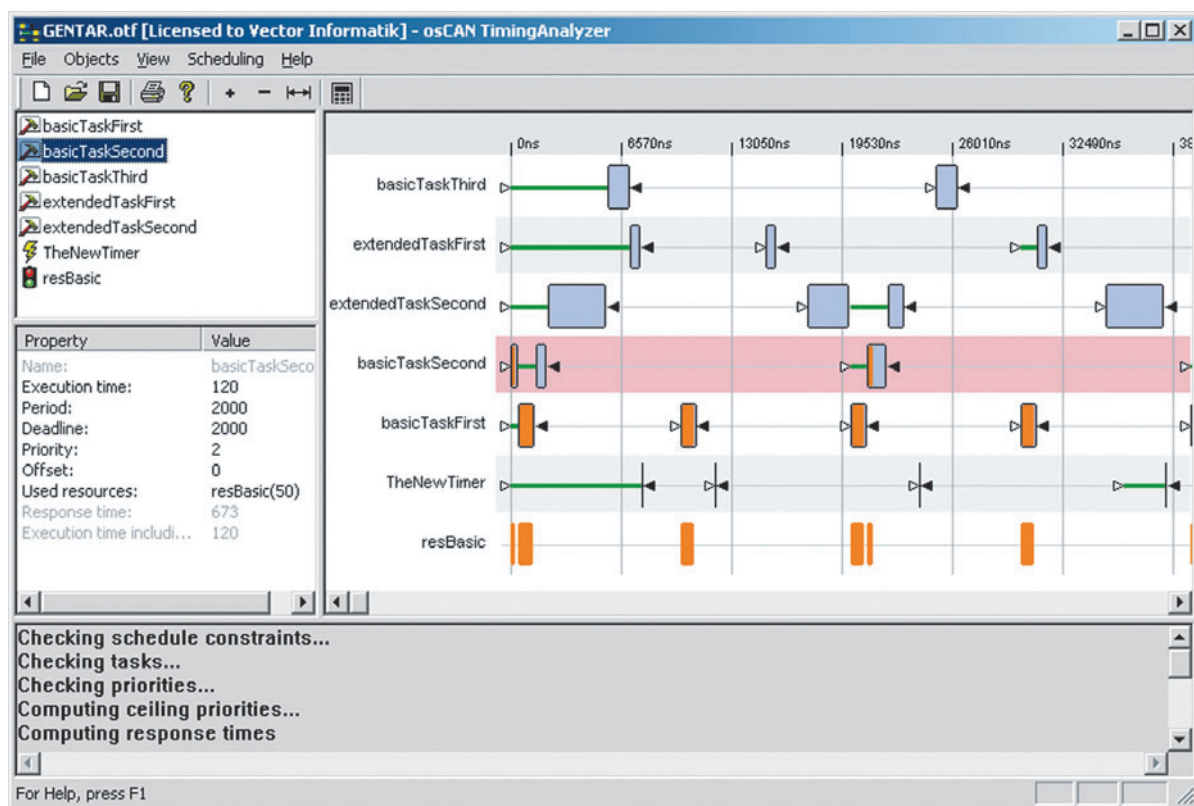


Figure 2: osCAN TimingAnalyzer enables simulation of scheduling tables (schedules) and computation of schedulability

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 negotiated 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.

Can error-free embedded systems be developed?

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 enable detection of errors in real-time systems that would otherwise not even be revealed in extensive testing. Scientists on the Verisoft project have taken this one step further in concluding that it is possible to develop error-free embedded systems and electronic components. Entire systems consisting of hardware, software, operating system communication, applications, etc., might be universally verified by methods of formal verification.

Reprogramming ECUs

Reprogramming of ECUs by flexible flash programming is continuing to draw greater interest. The issues here do not relate to the actual flash programming technology as much as they do to the organization and handling of the overall process. Constraints vary from project to project, whereby in addition to OEM and model specific requirements, other requirements must be considered such as hardware properties (bootloader, flash initiation), flash formats, transport and diagnostic protocols. If flash data pass through various gateways, for example, it must be assured that no data can be lost there. These and similar questions must be answered individually for each particular situation. In practice it is not really possible to implement a simple automatic approach here. Given these constraints, the rational handling of flash processes continues to gain in importance. Therefore, one trend leads in the direction of uniform management of flash processes in standardized formats. For this purpose, tools

from Vector Informatik save the flash data together with references to the flash jobs in a ODX flash data container.

In some cases it may be necessary to enable modification of flash data in the field by means of a post-build process. In this context it is important to recalculate checksums and signatures, in addition to other parameters, and to send them to the flash bootloader during a flash update. The CANape Graph and CANdito tools, both online and offline post-build processes can be handled in an elegant way, whereby a script language optimized for flash and diagnostic tasks is very helpful.

Managing different memories in the ECU

An important topic in reprogramming is the management of different memory types in an ECU. Rising complexity, e.g. in multiprocessor or distributed systems, goes hand in hand with greater memory requirements and the use of different types of memory. Some conventional nonvolatile memory devices have very different physical characteristics. Among the important differentiating characteristics of nonvolatile



Figure 4: Flash programming

memory types are: Size of the write segment, size of the erase segment, maximum number of programming cycles, and times required to program and erase.

The latest flash memory is based on NOR Stacked Gate and MONOS (Metal Oxide Nitride Oxide Silicon) technologies. Beginning about 2008 new memory products are expected that are based on FeRAM (ferro-electric), MRAM (magneto-resistive) and other technologies, which could permit unlimited numbers of write/erase cycles.

HIS-standardized Memory Driver Interface

With the goal of achieving uniform memory management, the HIS Automotive Group defined a standard for the Memory Driver Interface that is experiencing growing support from semiconductor producers. The interface provides functions for initializing, de-initializing, erasing, programming and reading data. In an implementation based on the HIS interface a Multiple Memory Type I/O Manager used to access various types of memory. The memory configuration can be defined conveniently with the Geny tool from Vector. The user benefits from maximum flexibility in accessing different memory types, including access by SPI (Serial Peripheral Interface).

Time is money: Accelerating flash processes

Depending on the number of ECUs, it may take a full hour or more to transfer data in the production environment. Therefore automotive OEMs and tool suppliers are considering adding greater bandwidth to the transport media as well as data compression. Scientific studies indicate that for compression purposes a combination of the LZ77 method and an arithmetic coding method would be ideal and might reduce data volume by up to one half.

[1] OSEK stands for "Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug" (Open Systems and their Interfaces for Electronics in Motor Vehicles).

[2] "Hersteller" = Producer or OEM



Author:

Peter Liebscher (Graduate Engineer) studied telecommunications engineering at the Technical College in Esslingen, Germany. Since 2002 he has been employed as Business Development Manager at Vector Informatik GmbH where he is responsible for the Embedded Software Components product line.
Tel. +49-711/80670-413,
Fax +49-711/80670-111,

E-mail: peter.liebscher@vector-informatik.de