Practical Implementation of Mixed-ASIL Systems
A Certified Operating System Simplifies the Development of Safety-related Software

The ISO 26262 standard describes a recognized and standardized process for developing safety-related ECUs in the automotive field. However, only parts of the software in these ECUs are safety-related. The goal is to restrict the additional, intensive development efforts for these safety-related components. It is possible to set up an ISO-conformant mixed-ASIL system, which may contain both ASIL functions and functions without qualification, using an advanced AUTOSAR operating system and two other basic software modules.

The approach of using mixed-ASIL systems provides a solution. In this case, the functional groups are isolated from one another by suitable protection measures, so that they cannot interfere with one another. This reduces the development effort for the individual functional groups to what is required for their specific ASILs.

Assignment of ASILs to all software elements generally results in functional groups with different ASIL classifications. In principle an ECU’s entire software must be developed to the highest ASIL determined for one of these functional groups. This intensifies the development effort to an extreme, because even non-safety related software must be developed to the high requirements of the safety process.

Technical Safety Concept
Developers must ensure that the overall system fulfills their safety requirements. None of the system’s software components may put fulfillment of these safety requirements at risk. Therefore, the only safety requirement for software components without safety-related functionality...
is that they must conform to the principle of freedom from interference ([1] Part 9, section 6.4). Freedom from interference of software components is defined by three properties:
> Safe memory accesses
> Correct time execution
> Safe data exchange

A component’s freedom from interference can be verified by classic verification measures, e.g. by code reviews. There are also approaches in which a specially developed code checker is used to check the freedom from interference of the basic software [2]. Other measures may be taken in the software to ensure protection against hardware-related disturbances as well.

A modern AUTOSAR operating system like MICROSAR OS SafeContext (Figure 1) offers protection against faulty overwriting of memory contents. The protection is achieved by partitioning each functional group into a so-called OS application. Each OS application’s data are allocated in separate memory partitions. Along with the application data, context-related data such as stacks and the contents of important registers are also located in such a memory partition. Access to these memory partitions is prevented by a Memory Protection Unit (MPU), which is part of the microprocessor hardware.

When switching the running task or the Interrupt Service Routine, the operating system executes a context switch. Here, the context data is stored, and the MPU is reconfigured so that it only enables the memory partition for the task or Interrupt Service Routine that is active after the switch (Figure 2). This switch is only executed by the operating system and is safety-related. Therefore, the AUTOSAR operating system MICROSAR OS SafeContext was assigned ASIL D classification and was developed in processes defined for ASIL-D in ISO26262.

**Comprehensive Validation Concept**

It is important to monitor the program flow in safety-related systems. The Watchdog Manager specified in AUTOSAR (Figure 1) is used for this purpose. This module, which supplements MICROSAR OS SafeContext, is available in the form of the SafeWatchdog [3] developed by TTTech Automotive GmbH, which is qualified to ASIL D. As the name suggests, this component controls the hardware watchdog, and it safely ensures a reset of the ECU in case of error. In addition, this component monitors for correct time flow of the application’s tasks. Developers can set a number of parameters for the monitoring such as program flow, cycle times, minimum/maximum execution times, etc.

The third requirement for freedom from interference, which is implementation of reliable communication, is fulfilled by end-to-end protection (Figure 1). With the help of the E2ELib [4] specified in AUTOSAR, the SafeCOM product protects the data to be transmitted using a CRC and sequential message numbers. Strictly speaking, this does not ensure safe communication, rather just “integrity” in communication. Software cannot protect against data failure due to hardware errors, e.g. a bus line break. To ensure safe communication, additional measures must be taken in the hardware, e.g. in the form of redundant buses.
Integration of Application Software and Operating System

In the ISO standard, safety-related components developed by an external supplier and supplied to the ECU manufacturer are referred to as “Safety Elements out of Context” (SEooC). They include the operating system discussed above as well as the watchdog manager and the EZELib. During development, the suppliers of such components must make assumptions about the expected safety goals without familiarity with the ECU project. Therefore, as part of integration work ECU developers must check whether the safety goals assumed for the supplied SEooC are sufficient for achieving the safety goals of their projects. Moreover, the ECU developer is responsible for following special integration instructions for the supplied software module. Therefore, each SEooC is supplied together with a safety manual, which contains the integration instructions and assumptions about safety goals.

At first, this may sound like more work. However, upon closer examination, it becomes clear that a similar level of effort must be planned for integrating components that are developed in-house, but the advantage of supplied components is that the effort required for creating these components is eliminated. Overall, this yields significant work savings.

Outlook

The TÜV Nord organization has certified the MICROSAOS SafeContext operating system developed by Vector for the TMS570 microcontroller, making it the first AUTOSAR operating system to be certified to ASIL D. This implementation is currently being transferred to other platforms. They include multi-core processors which are being used increasingly.

MICROSAR OS SafeContext, used together with the SafeWatchdog and SafeCOM basic software modules, provides an up-to-date and safe development foundation for safety-related ECUs. In particular, it can be used to cost-effectively implement mixed-ASIL systems. Besides protecting the application software, the safety process must also protect all of the basic software. Vector offers a variant developed in conformance to ISO 26262 which is distinguished by its ability to achieve the safety goal “freedom from interference in relation to memory access”.

Translation of a German publication in the special edition “Funktionale Sicherheit” of Elektronik automotive, July/2013

Image rights:
Vector Informatik GmbH

References:

Links:
Website Vector: www.vector.com
Product Information MICROSAOS Safe: www.vector.com/vi_microsar_safe_26262_en.html

Figure 2: The MPU in the hardware protects the memory partitions from unauthorized access. Reconfiguration of the MPU is performed by the operating system.