Is This What the Future Will Look Like?
Implementing fault tolerant system architectures with AUTOSAR basic software

Highly automated driving adds new requirements to existing safety concepts. It is no longer sufficient to simply deactivate a function to reach a safe state. In the future, a safe state will require energy and active functionality. This article shows available mechanisms and explains how they can be modularly combined to attain an effective safety concept. It also aims to create an awareness of the challenges of future fault tolerant systems and shows that they can be overcome effectively with AUTOSAR.

In safety-relevant systems of today’s vehicles, the most frequent reaction to a fault is to deactivate or reset the faulty function. This is referred to as fail-silent. It is easy to implement this type of solution, and it is effective for achieving a safe state and maintaining it. However, E/E systems in the vehicle are increasingly assuming other functions that must remain available in case of a fault, e.g. when a microcontroller fails. This behavior is referred to as fail-operational and in the following as fault tolerant. In the future, the demand for fault tolerant systems will increase substantially in manufactured automobiles. One example: in some of today’s heavy SUVs, it is necessary to keep steering assist systems active to assure that drivers can handle steering safely. While development for fail-silent systems is now mastered quite well with ISO 26262, issues in the design of fault tolerant systems are still difficult to resolve with ISO 26262. In particular, attempts to come up with a precise definition of a safe state are still causing headaches in this context. The second edition of ISO 26262, whose publication is planned for 2017 or 2018, will not achieve final clarity either. Aside from the requirements set in standards, the following chapters show how existing safety concepts can be extended to fault tolerant systems using AUTOSAR technology.

Modular Safety Concept for Fail-Silent Systems

Safety engineers use a modular concept to efficiently tailor the various safety mechanisms to a specific project (Figure 1). Here they make a rough distinction between measures for microcontroller integrity, measures for functional monitoring and comprehensive measures.

Measures for establishing integrity of microcontrollers are selected according to the highest Automotive Safety Integrity Level (ASIL) of the software that is used. They are independent of the function to be performed, and they are determined by the required diagnostic coverage (DC) for a specific ASIL.
Microcontroller manufacturers often set specific requirements based on their safety analyses. For example, a DC for ASIL D requires built-in self-tests (BITs) that are executed periodically by the software. Generally, starting with ASIL B the probability of occurrence of so-called Single Event Upsets (SEUs) must be considered. Microcontrollers in lock-step mode and memory with error detection and correction codes (ECC RAM, ECC ROM) offer effective protection against SEUs. Both safety mechanisms are realized in the hardware, are nearly transparent to software development, and are therefore very efficient solutions.

The developer normally implements additional mechanisms in the application to perform functional monitoring. They include monitoring tasks for sensors and actuators, as well as limiters and program flow monitoring (logic monitoring). Program flow monitoring can be achieved with an AUTOSAR watchdog, for example.

Functional monitoring and microcontroller integrity are defined and implemented according to the specific project. However, there are also mechanisms that are used in nearly every safety-related ECU and are independent of functionality and ASIL. Almost every ECU with ASIL software also executes QM software. To ensure

Figure 1:
A modular concept enables efficient tailoring of safety mechanisms to a specific project.

Figure 2:
AUTOSAR timing protection offers early detection of a violation of the allowed time budget for tasks and interrupts.
coexistence according to ISO 26262, memory separation and monitoring of time constraints are needed [1]. Memory partitioning is realized by an AUTOSAR operating system with Scalability Class 3 (SC3) that controls a memory protection unit (MPU) with the required ASIL. The watchdog usually handles monitoring of time requirements by deadline monitoring. As soon as safety-related data is exchanged between more than one ECU, communication protection comes into play. AUTOSAR offers an effective safety mechanism for this purpose in the form of end-to-end protection (E2E). Products from Vector that are certified up to ASIL D are available for implementing these comprehensive measures.

**Transition to Fault Tolerant Systems**

For cost reasons, today’s hardware is designed to be nearly redundancy-free. Therefore, a hardware fault generally leads directly to a serious functional degradation up to complete failure. On the other hand, mature methods exist for quantifying hardware failures, such as those defined in IEC 62380 and SN29500, which permit predictions of the target failure rate [2].

It is often difficult to quantify software faults, since they are exclusively systematic [3]. Timing protection is a suitable safety mechanism for boosting fault tolerance with respect to software faults. Timing protection guards against such faults as infinite loops in software components that prevent execution of the actual functionality. In timing protection, the developer assigns time budgets for the execution times of tasks and interrupt routines and for the blocking times of interrupts and resources. The time intervals between tasks and interrupt routines are also monitored (Figure 2). In case of a fault, the AUTOSAR operating system can terminate the task or interrupt routine that is causing the fault and exclude it from further execution. However, timing protection is only a first step toward the fault tolerant systems that will be needed in the future.

**Fault Tolerant System Architectures**

Fault tolerant system architectures have been used for many years now in the aerospace industry. For the safety-critical flight control systems, three or four ECUs are redundantly combined to form a complex system. This redundancy in the hardware is of course extremely cost-intensive. Therefore, new approaches must be sought for using fault tolerant systems in the automotive industry. This industry also benefits from lower severity of the consequences of failure in risk assessments.

A feasible system architecture (Figure 3) always consists of at least two channels. In this example, one channel comprises a sensor, a logic unit and an actuator [4]. It is obvious that when the microcontroller of one channel fails, the associated software and its functionality fail as well. Due to their complexity, microcontrollers frequently have the highest failure rates in an ECU. Therefore, proper execution of a function cannot be assured for even a very short time period.

Figure 3: Example of a fault tolerant system architecture achieved by redundant design.
To make this two-channel system fault tolerant, each channel must detect all individual errors for itself and switch itself to passive [5]. Without this requirement, both channels are needed for safe operation of the function. However, in this case the failure rate would double and not be halved as desired. Of course, this system architecture requires a redundant energy supply for the two channels, just as it requires a redundant communications path to relevant partners. The IEC 61508 standard identifies such a system as a 1-out-of-2 with diagnostics (1oo2D).

Software Architecture with AUTOSAR for Fault Tolerant Systems

In principle, introducing redundancies into the hardware also increases complexity in the application. This creates new challenges in the area of control engineering, such as how to achieve controller stability and control actuators when redundant controllers are active at the same time. It is also necessary to reassess data consistency in networks (“Byzantine Generals problem” [6]). From a system architecture perspective, this complexity could be limited by a hot-standby mode, for instance. In this case, only one of the two channels controls the actuators at any given time. If an error occurs on this channel, the other channel immediately assumes control. The AUTOSAR basic software (Figure 4) is useful for simplifying the application development process for the following reasons.

> Re-use: The AUTOSAR components presented above that are used to achieve a modular safety concept can also be re-used by the developer for error detection.

> Use of existing mechanisms: There are two philosophies when it comes to implementing the software of redundant channels: diversity or homogeneity. The diversity philosophy uses different software on the two channels. With the same types of channels and microcontrollers, it is possible to use the same software that is simply parameterized differently for each channel. This is done with the Post-Build Selectable Mechanism of AUTOSAR, which is normally used to develop ECU variants. The use of the same types of channels requires examining errors with the same cause [7].

To enable a quick switchover of control to the other channel in case of a fault, sensor and actuator values as well as status information on the channel are exchanged between the channels (Figure 3). The mechanisms of AUTOSAR make it possible to implement just one configuration of the basic software. The developer can either implement channel switching application-specifically as a software component (SWC) in hot-standby mode or exploit the flexible configuration options of the AUTOSAR manager components Basic Software Mode Manager (BSWM) and ECU State Manager (ECUM). Today, application-specific software is implemented to exchange the error status between the channels. In the future, however, it is conceivable to specify standardized basic software components for exactly this purpose.

---

**Figure 4:** MICROSA R software architecture for fault tolerant systems based on AUTOSAR
Tool Support

To overcome the added degree of complexity in the future, effective and comprehensive tool support needs to begin in an early development phase. This frees up resources for focusing on application development and relieves engineers of tedious and error-prone work such as manual consistency checking of redundant signals in the system model.

Outlook

Today, on-board AUTOSAR capabilities already permit efficient implementation of safety-related projects. If increased fault tolerance of E/E systems is required, new system architectures will be needed that can handle the failure of the microcontroller as well. This presents new challenges for the application and the basic software. Nonetheless, the complexity of these types of systems can be mastered using AUTOSAR methodology. Here, AUTOSAR basic software offers an excellent starting point due to its configurability. The associated tool chain simplifies management of the necessary redundancy.

In the future, assistance will be further improved by the basic software and its tools. A first step, however, is to realize that fault tolerant systems also require new approaches in the system architecture.

Translation of a German publication in Elektronik automotive, 11/2015

Figure Credits

Lead figure: Fotolia #7650668, Originator: Jakub Krechowicz, with modifications by Vector
Figures 1 - 4: Vector

References:
[1] Definition of the fault tolerant time interval (FTTI) in ISO 26262-1, 1.45
[2] ISO 26262-5:9 Evaluation of safety goal violations due to random hardware failures
[3] ISO 26262-10:4.3 Relationship between faults, errors and failures
[4] Definition of a system in ISO 26262-1, 1.129
[5] ISO 26262 Single-point fault metric (SPFM) for ASIL D

Links:
Website Vector: www.vector.com
Product Information MICROsAR: www.vector.com/microsar

Contact information of the Vector Group:
www.vector.com/contact

Jonas Wolf (Dip. Ing.), Vector
has a degree in Aerospace Engineering from the University of Stuttgart. He has been working at Vector since 2012; currently he works in the role of Senior Product Management Engineer for Functional Safety and Cyber Security.