

## Why does the automotive industry need AUTOSAR?

The AUTOSAR partnership was set up with the objective of defining an open standard for an automotive E/E (electrical/electronic) architecture. There are several major issues that need to be addressed:

- ▶ Managing the growing complexity of automotive E/E systems
- ▶ Flexibility for product modification, upgrade and update
- ▶ Scalability of solutions within and across product lines
- ▶ Quality and reliability of E/E systems

The standard for automotive software now under development will be a first step towards tackling these problems. The concept for the standard is a layered software architecture with standard APIs. Establishing a software standard will be a big step forward, but on its own it is not enough. Therefore standardization will apply not only to the software, but also to the whole development process from functional description to software testing. In future, design engineers developing a new E/E architecture will adhere to the AUTOSAR design process. The process uses a top down approach:

1. Complete system definition (definition of software components, available hardware and system constraints)
2. Allocation of software components to each ECU
3. Configuration of the software on each ECU
4. Conformance testing

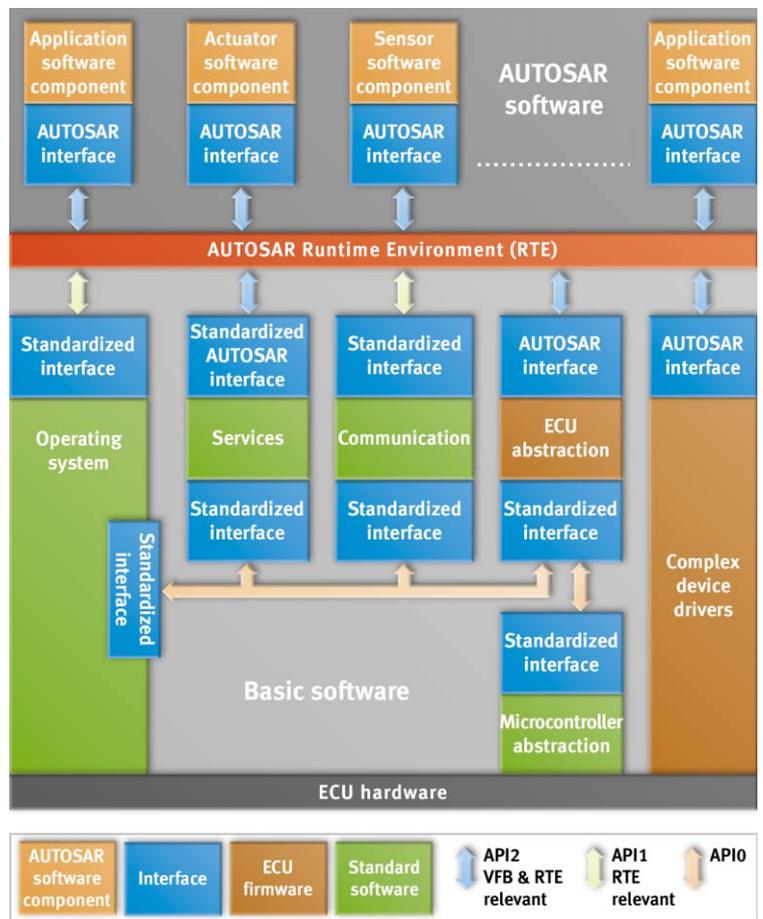
The goal is to have a standardized tool chain that will guide and assist the design engineer through the complete process. Introducing the AUTOSAR process would be a breakthrough in automotive E/E design. It would be a radical step and its repercussions would change the industry forever.

## AUTOSAR software architecture

The basic concept underlying this architecture is that abstraction of the hardware will be done in layers. Closest to the hardware is the microcontroller abstraction layer; as the name suggests, this layer abstracts the microcontroller. The goal is to have a hardware-independent API but a hardware-dependant implementation. The next layer is the ECU abstraction layer.

The purpose of this layer is to abstract the other components on the ECU printed circuit board. The third layer is the services layer. This layer is almost hardware independent (the operating system needs a timer) and its task is to handle the different types of background services needed. Examples are network services, NVRAM handling, operating system. The fourth and last layer is the runtime environment from the basic software. All software components running above the RTE are hardware independent components. Within the partnership, the APIs and the features of these modules, except for the complex drivers, are specified. How to realize this API functionality is up to each implementation.

Let's assume that we have a standard API. The question is how do we adapt this API to fit different microcontroller platforms and system needs? For example: one SPI channel for one application may need to run at 1 MHz with no chip select, but another SPI channel for a different application needs to run at 250 kHz with chip select. The AUTOSAR approach is to have a very flexible configuration concept. This means that the basic software should be configurable, so that it can be adapted to fit both the ECU hardware and the application needs. Inside AUTOSAR, this configuration concept will be standardized. The tools used for the configuration will, however, not be standardized. The tool feature set will be up to each tool vendor, but the input format to all



Software architecture



**all-electronics.de**  
ENTWICKLUNG. FERTIGUNG. AUTOMATISIERUNG



Entdecken Sie weitere interessante Artikel und News zum Thema auf all-electronics.de!

**Hier klicken & informieren!**



tools will be same. Different tools will have different graphical user interfaces and feature sets. Some tools may be able not only to configure the software but also to generate the source code for the software implementation.

To make possible seamless usage of different tools, it is essential that configuration is handled in a uniform way, ie, based on a common file format. For this purpose, AUTOSAR will standardize a configuration template based on the XML file format. This configuration template will describe the parameters that should be configured and the dependencies that exist between them.

In order to keep the software as portable as possible, the AUTOSAR partnership will specify as many of the configuration parameters as possible. It will not, however, be possible to standardize all configuration parameters. There will always be hardware- and implementation-specific parameters that are not covered by the standard specification. The template being worked on must therefore be flexible and allow for adding those specific configuration parameters.

## How will AUTOSAR change life for microcontroller vendors? Impact on E/E architectures

Normally there is a trade-off between standardization and optimization. Introducing this standardized concept is likely to lead to software and perhaps runtime overhead. This overhead may make it necessary to increase microcontroller resources, such as RAM, ROM and CPU performance. Normally this would lead to an increase in system cost, which no one is prepared to pay. One solution to the problem could be to alter the vehicle E/E architecture, by moving away from the current one-ECU-one-function concept to a more centralized concept where several functions are bundled in one ECU.

Changing to this style of E/E architecture with fewer ECUs but more functionality would save a lot of overhead cost such as:

- ▶ Housings
- ▶ Printed circuit boards
- ▶ Voltage regulator
- ▶ Transceivers

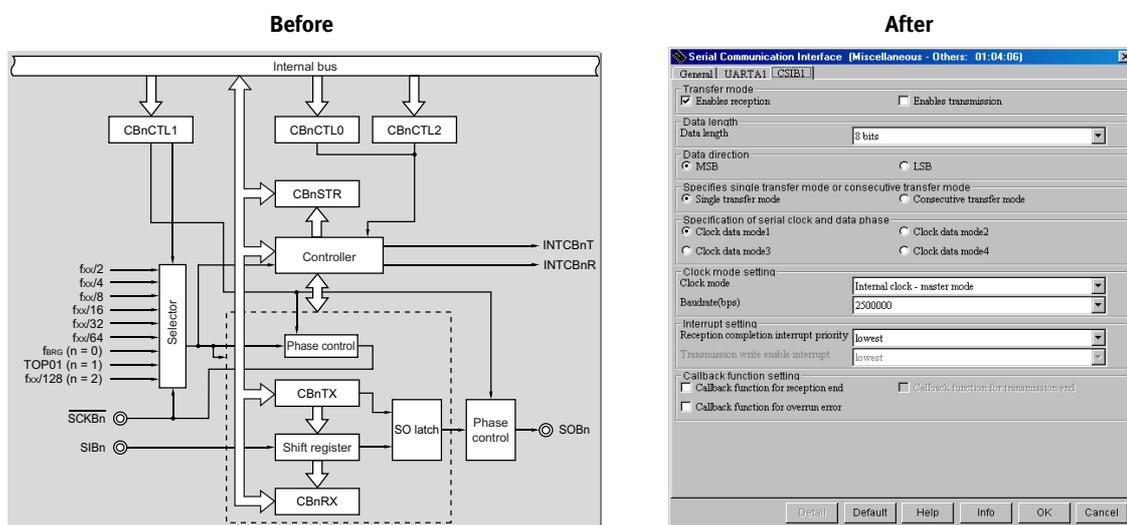
This is a topic that is already being discussed in the automotive industry. Work on a more centralized E/E architecture has already started and will continue steadily. One of the main hurdles is how to integrate software components from different suppliers onto one ECU. This is not only technically challenging, but there are also issues with liability. Who is responsible for a system failure when the software components are supplied by different companies?

Introducing AUTOSAR software will ease integration work significantly. In that sense the success of the AUTOSAR standardization could be a decisive factor for further growth in vehicle software functionality. Automotive OEMs know it is not feasible to continue building on the current network architecture. Today there are up to 75 ECUs in one vehicle. In view of the expected growth in software functionality, radical change is essential. If no solution is found and the number of ECUs continues to grow, the OEMs will quickly run up against some insoluble problems, including:

- ▶ Cost
- ▶ Current consumption
- ▶ Space
- ▶ Cabling

Introducing new network architectures with fewer ECUs is one possible solution. For the microcontroller vendors this means that several mid-range and low-end microcontrollers will be replaced by one high-end microcontroller possibly controlling smart sensors and actuators directly or via a bus.

## Changing the user interface



Interface for software developer

Working with the AUTOSAR software architecture, the main focus for microcontroller vendors will be the software modules interacting with the ECU hardware:

- ▶ Microcontroller abstraction
- ▶ Operating system
- ▶ Complex device drivers

Implicit in the new software architecture and configuration concept is the view that the methods used by software developers will change. When introducing a new microcontroller, they will no longer call on a traditional user manual that explains a number of registers and physical signals; instead they will switch to using the AUTOSAR configuration tool to configure standard drivers the way they want them. In consequence, the microcontroller vendor will no longer be selling a “chip + user manual” but a “chip + software + configuration set”. To be successful, this “new” product will have to be supported by a user-friendly configuration tool.

## Challenges for microcontroller vendors

As previously explained, the main user interface for the software programmer will change from microcontroller registers to a standard software API. Since this interface is the same for all microcontrollers, it will be very important for the microcontroller vendors to work out how to differentiate their products from the products of the competitors. Many traditional factors like:

- ▶ Price
- ▶ Quality
- ▶ Current consumption

will continue to play a major role. But from a purely software and functional perspective the product from vendor A will look the same as the product from vendor B. How will it be possible to differentiate one product from another at all? The answer to that question is that one solution for a specific software API may be better or worse than another. The next question is how to arrive at the best solution? In order to do so we need some measurable parameters such as:

- ▶ Run time
- ▶ RAM needs
- ▶ ROM needs
- ▶ Timing constraints
- ▶ Robustness

These parameters are nothing new to a person familiar with microcontroller benchmarks. But the benchmark concept is new. Previously a benchmark might have been based on reference values like whetstone, dryhstone, assessed by an independent organization, or a company proprietary benchmark. Results were sometimes difficult to compare – or to apply to an actual project. It was not always clear whether benchmark results meant that the microcontroller could handle the project requirements or not. With AUTOSAR this will all change; all benchmarks will be based on AUTOSAR software. This means the results will be easy to compare and easier to apply to an actual project. Microcontroller products will no longer be compared mainly on their CPU cores, which is a relatively simple benchmark. Now the criterion for performance comparison will be the CPU core together with the AUTOSAR software implementation. This has huge implications for microcontroller vendors.

A specific solution optimized for one application may score best on one benchmark. But this solution may not be suited for a slightly different application. This is the dilemma that microcontroller vendors face. They must always aim for a product with the best balance among performance, cost and flexibility. Now the main criterion for judging performance will be the handling of AUTOSAR software and the measure of flexibility will be the configuration possibilities.

Microcontroller vendors will have to acquire an extra level of competence. Assuming that the benchmark scenario described above is right, then AUTOSAR will compel them to shift the focus away from hardware to a more system-oriented view. When new products are being developed, they will have to consider not only hardware but the combination of hardware and AUTOSAR software.

The current trend is for microcontroller vendors to outsource software development. The question is whether this is the best approach in the long term. It is very important for microcontroller vendors to have control of their own products – in this case the combination of hardware and software. In order to have the control of the complete product it is necessary to have control over the software development. This can only be achieved by being a part of and taking an active role in the software development process. At the start of an implementation, it is essential for the microcontroller vendor to work with the implementation concept and specification. After implementation, being able to review, test and verify the software is an important factor in assuring the performance and reliability of the complete system. Managing the software development process will become a key skill for success in the automotive microcontroller market.

## Layered hardware architecture

It is widely accepted in the automotive industry that the AUTOSAR standard will be one of the most important factors for future development in automotive electronics. A change is certainly needed. But is a change needed at all costs? Current AUTOSAR software architecture has some critical issues that need to be considered.

One example is that modern high-end microcontroller architectures aim to reduce the volume of interrupts issued to the

main CPU. Since the vast majority of the interrupts are issued by peripheral units on the microcontroller, the obvious solution is to try to handle interrupt events from the peripherals by other means than interrupting the CPU. This is normally achieved by adding intelligence in the peripheral (eg, message buffer handling), using a DMA or a programmable co-processor.

One of the problems with the current AUTOSAR software architecture is that inadequate consideration has been given to this approach of implementing buffering or data handling functionality in the hardware. Under AUTOSAR, a microcontroller abstraction layer is specified. This is a driver layer that will abstract the peripherals on the CPU. In the above example, some of the functionality specified as software in an AUTOSAR API will, in fact, be implemented in hardware, making the software API superfluous. The question is whether the software without this API is AUTOSAR compliant or not?

The implication is that there is a need to define some sort of scalability or different levels of conformity. If support for something that is specified in software is actually built in to the hardware, there must be a way to exclude those software APIs and still be AUTOSAR compliant. If a solution is not found, then the standard will limit the innovation potential for automotive microcontroller vendors. Ultimately this will benefit no-one and will pose a real threat to the whole standard.

## What action is NEC ELECTRONICS taking on AUTOSAR?

NEC ELECTRONICS joined the AUTOSAR partnership as premium member in March 2004. NEC ELECTRONICS' main area of expertise is microcontrollers and the main focus inside the AUTOSAR partnership is on the software modules that are very close to or directly interacting with the microcontroller hardware. That is the area where NEC ELECTRONICS as a major microcontroller supplier for the automotive business worldwide can contribute the most. NEC ELECTRONICS employees are currently members in the following workgroups:

- ▶ SPAL (special peripheral abstraction layers) (active)
- ▶ Memory management (active)
- ▶ Configuration (active)
- ▶ FlexRay
- ▶ CAN/LIN
- ▶ Mode management
- ▶ Gateways
- ▶ Operating system

Internally NEC has begun working systematically on bringing AUTOSAR standards and methodology to bear on the product development process. With a clear goal in view:

NEC intends to provide system solutions that both support and enhance the implementation of AUTOSAR-compatible software. System solutions are software and hardware working together. Providing system solutions does not necessarily mean that NEC ELECTRONICS will produce the system solution in house. It means that NEC Electronics will make sure that there are AUTOSAR-compatible system solutions on the market for NEC ELECTRONICS microcontrollers. These system solutions will be implemented either internally or in close cooperation with the primary software developer. NEC ELECTRONICS will play a major role in the software development process to ensure the optimum implementation for that specific microcontroller.

*Peter Jakobsson*  
*Senior Engineer AUTOSAR & Standardisation*  
*Automotive Business Unit*