

# New Methods of Debugging and Testing Improve the Software Quality of AUTOSAR ECUs

Richard E. Lotoczky  
Vector CANtech Inc.

Mark Schwager  
Vector Informatik GmbH

## ABSTRACT

The development of ECU software has been the focus of improvement since its widespread inception in the late 1970's. Since that time, we have seen the transition from assembly language to higher level languages, like C, to higher levels of abstraction with the use of MATLAB® to build C language constructs. One of the areas that had not changed significantly is the methods used to evaluate, validate and verify performance.

In many applications, software has been validated on the bench using the generated code in conjunction with the target ECU and some sort of I/O simulation technique. In other applications, such as AUTOSAR, software components may be developed and tested in a virtual environment on a PC and later transferred to the target ECU for functional testing in a simulated environment on the bench.

In this paper, the concepts for two new tools are presented. The first concept describes an emulation platform that automatically configures the AUTOSAR Run Time Environment (RTE) and Basic Software (BSW) to test the performance of AUTOSAR software components using traditional techniques or cycle by cycle in the virtual environment. Because software is usually available for a target ECU long before the production hardware is available, the use of such an emulator to execute target ECU binaries provides an understanding of system behavior and aids in identification and resolution of performance issues before they are encountered in the vehicle.

The second concept is a monitoring and debugging platform for AUTOSAR software components that reside on a target ECU. It will leverage the information contained in the AUTOSAR Run Time Environment (RTE) and Basic Software (BSW) to gauge the performance of a Software Component (SWC) and assist in debugging any anomalies discovered at this level.

**CITATION:** Lotoczky, R. and Schwager, M., "New Methods of Debugging and Testing Improve the Software Quality of AUTOSAR ECUs," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-0427.

## INTRODUCTION

Software development methods are continuously evolving as the level of software complexity increases. Traditionally, software verification and testing has been accomplished on the test bench using an emulator, some form of software debugger, a target ECU and some sort of plant simulator to provide the necessary hardware support. This approach has always been somewhat problematic and in some cases has hampered development of new applications due to the inherent lag in hardware availability.

With the advent of AUTOSAR, an improved series of techniques are now available to the software developer and tester that will improve the overall process. New ECUs that

use AUTOSAR Basic Software (BSW) and Run Time Environment (RTE) have the inherent advantage of a formal operating system (OS) as well as defined interfaces between standardized software layers. These interface definitions allow the developer to test and validate software completely in a virtual environment.

## VIP-VIRTUAL INTEGRATION PLATFORM

The first of the new concepts available to developers is called VIP, the Virtual Integration Platform. As its name suggests, VIP is a virtual environment for SWCs that allows the developer to observe the function of software components (SWCs) running in native mode on a PC. In this environment,

the developer can verify software operability as it evolves, perform automated and interactive testing and engage in the integration and testing of components on a RTE and BSW simulation early in the ECU development cycle. The challenge is to provide a simulation that is realistic enough for the scope of the test.

The primary goal of such a platform is to provide an environment that supports both interactive and fully automated testing. It should also allow for the reuse of tests across multiple phases of the development process and permit developers to get an “early start” with integration and validation. Since the platform for this environment is a PC, multiple installations are easily implemented to further enhance the development process.

In order to fully test SWCs, some method is needed to exercise the simulation running on the virtual platform. This requires the use of an external test tool. Software execution is handled by the VIP emulator. Control, visualization, logging, environment simulation (plant model) and data reporting are functions of the external test tool. Both of these tools may be hosted on the same PC.

## VIP - Components

There are three main components associated with Virtual Integration Platform (VIP): the VIP emulator, the BSW configurator and the VIP API.

### VIP Emulator

A PC based emulator should ideally model the execution of SWCs with the required level of detail necessary for the test. It should support the execution of actual SWC code without “instrumentation” (a facility to monitor execution and I/O behavior). The level of detail used in the emulation is defined by the test. Two emulation modes are available in VIP:

- AUTOSAR MCAL/OS emulation mode
- Microcontroller emulation mode

The AUTOSAR MCAL/OS emulation mode uses a BSW, OS and MCAL (MicroController Abstraction Layer) adaption compiled for the PC. It emulates realistic pre-emptive OS behavior. To use this version of the emulator, the SWC must be compiled with a compiler compatible with the platform operating system (e.g. Microsoft Visual C/C++® compiler). This mode is helpful to start-up the virtual ECU rather quickly and to evaluate its overall functionality.

Pre-emptive OS behavioral emulation can only be an abstraction of the real ECU behavior as long as the specifics of the microcontroller are not completely modeled. This mode might not be sufficient for accurate timing analysis.

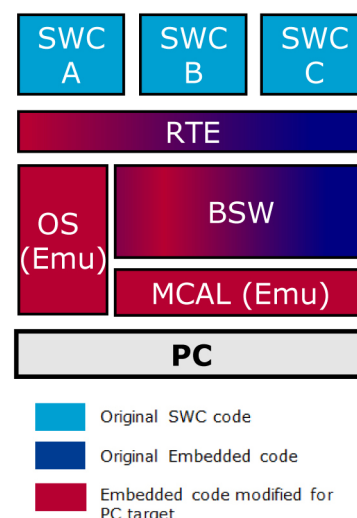


Figure 1. AUTOSAR MCAL/OS Emulation Mode.

The Microcontroller emulation mode, on the other hand, can emulate SWC and BSW code that has been compiled by the native compiler for the target MCU by using a third party instruction set simulator (ISS). In this case, behavior and timing down to the assembly language level can be realistically modeled. Pre-emption effects, like race-conditions and task durations, can be studied in a very accurate way. This requires a complete configuration for the target ECU and generation of this configuration can be a very time consuming process. However, the resulting embedded code can not only be used in the Microcontroller emulation mode but can also be flashed onto a real ECU. A seamless migration from the AUTOSAR MCAL/OS emulation mode to the microcontroller mode is also beneficial. Seamless means that test cases can be reused and extended with more detailed tests for timing analysis.

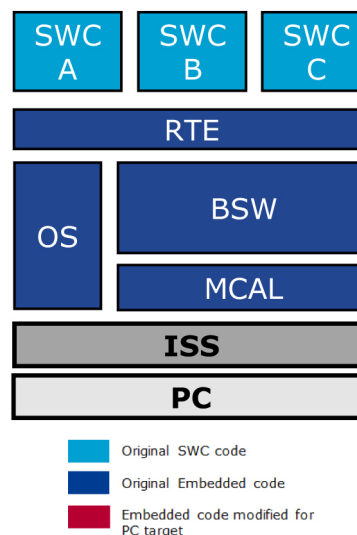


Figure 2. AUTOSAR Microcontroller Emulation Mode.