Software Marketing, Embedded Real-Time Solutions
The current economic climate is adding to the increasing pressure being experienced in new application development today. This article will highlight the challenges companies face in the embedded market, and stress the importance of more integrated solutions, industry standards and collaboration with partners as a means to survive through these rocky times. The article will focus on how solutions, such as the Freescale CodeWarrior™ Development Systems, are providing integrated solutions which include embedded RTOSes such as OSEK/VDX and low-level drivers. These solutions go some way to lighten this burden for application developers and considerably reduce time to market.
Breaking Down the Barriers for Embedded Software Developers with More Tightly Integrated Solutions

The recent economic climate has added to the increasing pressures experienced by application developers today. Companies have tightened their budgets whilst confronting the growing challenges in the embedded market. Suppliers have shifted up a gear to respond to growing demands and higher expectations. The automotive market, which is experiencing rapid software growth, is an example in which the creation of industry standards, and collaboration through joint ventures or partnerships, has helped alleviate some of this pressure and reduce the amount of effort required by the developer.

The embedded market today has technical and commercial requirements that are subject to ever changing market demands and consumer preferences. Across the board, market conditions and intensifying competition are driving companies to produce products with increased integration for a wider range of features at lower cost margins. Silicon manufacturers have responded with smaller packaging, higher performance, multi-core architectures, deeper integration and increased memory size. These more powerful processors with multi-core capability require a higher level of sophistication in application development tools. Reliance on software to differentiate features for products is becoming standard practice. These trends are creating an even greater need for closer cooperation between suppliers and for standardized development processes.

Embedded developers face the inevitable dilemma of cost versus time to market. Budget constraints may require the use of older hardware or a mix of different computer systems. Meanwhile, market pressure holds developers to aggressive schedules. Another hurdle might be that the project’s specification could require engineers to master a new processor architecture. An existing body of debugged, time-tested code must be ported or rendered obsolete. Worse still, the engineers may have to spend time learning a new suite of development tools. Choices need to be made but changes are time-consuming and expensive. Changing a processor platform, real-time operating system (RTOS) or tool chain takes time and significant investment in evaluation, integration, training and procurement that may sometimes outweigh and prevent the benefits they bring.

New technologies and evolving application needs call for the creation of open standards and encourage cooperation between suppliers. Industry consortiums have been formed, allowing customers, suppliers and competitors to come together to define common standards that will produce the shared benefits of reducing costs and risks. They also help ensure interoperability between related products. The automotive market has been quite successful at encouraging and adopting global standards. The CAN1 hardware protocol and OSEK/VDX2 RTOS are just two examples of standards that have been widely adopted not only for the automotive market, but also for embedded applications in other markets, such as industrial control. Reliability and quality processes are increasingly critical as manufacturers cannot afford failures in the field and are passing these requirements onto their suppliers to uphold. Focus on quality processes will become an increasing requirement for suppliers to demonstrate compliance.

The existence of a stable, open, embedded RTOS standard has implications for the development of embedded software. Those with a cohesive software development strategy based on a standard application environment should reap the benefits of shorter development cycle time, improved application quality and stability, portability across applications and platforms, increased flexibility, modularity and more. The extent to which these objectives can be achieved depends on more than just “plugging in” an RTOS. Use of a standard RTOS enables the user to achieve separation of the application from the hardware, especially if standard communications protocol and peripheral driver interfaces are used in conjunction. In the case of the OSEK/VDX standard, the definition for the communication portion (OSEK/VDX COM3) is expected to be completed by the middle of this year. Software designers will then have a standard set of system services, API, configuration method and well defined, predictable behavior on which to build applications. The ability to reconfigure developed software is significant and should not be underestimated as an advantage for the development process.

The traditional method of code development and requirements capture is changing through the use of graphical model-based Simulation (MBS) and unified modeling language (UML) tools for system design.

The range in specialized, high-level design tools for rapid application development, auto code generation, verification and simulation is increasing. These high-level design tools, or visual programming languages (VPL), make it easier to build an application by providing a higher abstraction layer than was previously available. By applying a graphical and pictorial approach to their task, the developer can concentrate on the application being built, rather than the actual code being written. This approach allows more consistency and a better overview of the application. Using VPL can simulate the application before the hardware is needed in an effort to find errors in the code as early as possible. This can lead to an application that is scaleable, reliable, maintainable and reusable. The use of these high-level tools assumes or forces a discipline of defining a common software approach. The focus on design and systemic ways to improve design is growing and creating competitive and financial advantages for those who successfully implement it. The challenge remains for developers to integrate and optimize these capabilities with their already existing tool chain and processes.

As an example of increasing sophistication and a next generation Integrated Development Environment (IDE), the Freescale CodeWarrior™ tool suite helps software developers clear many of these hurdles. By combining editor, compiler,
linker and debugger into a single application (all controlled by an easy to use GUI), CodeWarrior accelerates the development process. The CodeWarrior tool suite runs on many popular platforms, including Windows®, Mac®, Linux® and Solaris™ operating systems. Virtually the same GUI is presented to the engineer regardless of the host. In addition, the CodeWarrior tool suite’s source and project files can be exchanged amongst host systems without any conversion. This host-neutral capability helps engineers clear the cost hurdle mentioned earlier, because it allows them to use a mix of different computers to tackle a time-critical project. The CodeWarrior tool suite also offers a choice of high-level programming languages (C, C++, Object Pascal and Java™), as well as full support for in-line assembly.

It is becoming increasingly important that tools not limit the choice of processor for the next “hot” product and that they generate code for the most popular processors. This allows for better optimization of the overall system cost since developers can port software to a new processor without having to learn new tools or lose an existing code base in the process. Understanding the full development process and the challenges discussed above is critical to being able to deliver more compelling solutions that will add value. There is a trend forming in the embedded market toward customer solutions. Gone are the days when components needed to be purchased from multiple sources with the hope that everything will work together.

No one company can hope to solve and provide this whole tool chain for the development cycle on their own and be successful. There are niche players who specialize in specific tools which, when combined with third party products, create a very powerful combination. Freescale has created development solutions that are architecture- and industry-specific, and which take care of the integration issues for a range of RTOSes and third party high-level design tools. Furthermore, Freescale’s tools are created during the early stages of silicon design to help ensure an optimized result. These solutions provide excellent tools which allow developers to make better use of resources and focus more on innovation and time to market.

Freescale’s CodeWarrior Development System is an example of an industry solution that provides a combination of necessary hardware and software, making the start-up of development work as smoothly and efficiently as possible, since it has taken care of the integration issues already (see Figure 1). This comprehensive, ‘out-of-the-box’ development environment tightly integrates hardware development platforms with CodeWarrior for Embedded Systems, Freescale’s high-performance and highly stable OSEKturbo RTOS (developed using SEI CMM4 level 5 process), low-level driver software as well as benchmarks, demo software and complete documentation. The goal is to have developers spend less time on tool chain optimization and spend more time focused on their application.

Embedded developers are now realizing the benefits for a cohesive software development strategy; in some cases it is a matter of survival. Integrated development tool solutions play an important part in this process. Those with a cohesive software development strategy based on a standard application environment should reap the benefits of shorter cycle times, improved quality, portability across applications and platforms, increased flexibility, modularity and more.
References:


2. OSEK/VDX Consortium—OSEK is an abbreviation for the German term “Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug” (English “Open Systems and the Corresponding Interfaces for Automotive Electronics”)—for more information, please visit: http://www.osek-vdx.org/.


4. SEI CMM—Software Engineering Institute’s Capability Maturity Model is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes—for further reference please visit: http://www.sei.cmu.edu/cmm/cmm.html.
Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright license granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.