As the mobile and consumer industry continues to grow, manufacturing organizations will need to address concerns with increasing device complexity, time-to-market, and bill of material costs containment. Technical challenges such as deployment of Linux to mobile platforms and integration of functionality need to be addressed with timely and efficient solutions. Challenges such as increased device complexity that affects productivity, access to debug ports, and additional costs and validation that affect time-to-market are all potential obstacles for mobile and consumer device engineers.

Freescale Semiconductor and Wind River have a long-standing partnership designed to provide joint solutions that enable their mutual customers to succeed. This white paper discusses how features of the Freescale i.MX applications processor and a complete Wind River development tool ecosystem can help mobile device manufacturers get to market faster and at lower cost.

**Market and Technology Trends**

The consumer industry is adopting mobile devices rapidly. It is estimated that there will be more than 1.4 billion mobile handsets in use by the year 2010. There is also increasing consolidation of features in mobile devices. In addition to supporting voice calls, modern handsets must contain integrated solutions for MP3 audio playback, Bluetooth, Wi-Fi, GPS, fully functional Web browsers, and more.

The technology of mobile devices is changing as well. Smartphones are rapidly adopting Linux as the operating system of choice. Linux growth in these types of devices is occurring at a 47 percent compound annual growth rate (CAGR), according to IMS Research, a market researcher in the global electronics industry. Additionally, alliances for standardizing Linux on mobile devices, such as the LiMo Foundation and the Open Handset Alliance (OHA), are also appearing.

**Challenges in Developing and Deploying Mobile and Consumer Devices**

**Business Challenges**

Operators, handset original equipment manufacturers (OEMs), and silicon providers face common challenges in bringing new smartphone and single-core handset models to market. They must carefully balance developing differentiating features with escalating research and development costs and ever-growing time-to-profit pressures. Yet for all that effort, the typical handset is only on the market four months before an OEM must introduce a new version to remain competitive. The average end consumer searches for a new handset that offers leapfrog functionality less than 18 months after buying their original handset. This hyper innovation rate, coupled with low margins, makes the mobile handset market challenging for providers of smartphones and single-core handsets.
Technical Challenges
The rapid evolution of the mobile market, along with shrinking time-to-market windows and low-cost requirements, puts significant pressure on technical teams developing mobile and consumer devices. The move to Linux for mobile platforms means that developers need to be ready to design mobile devices specifically for Linux and to ramp up development and test of various Linux OS components for mobile devices.

The movement to Linux introduces new challenges in debugging mobile devices. Traditionally developers have relied on agent-based solutions, and advances in agent-based debugging have helped software developers troubleshoot specific user-space problems. However, for mobile devices, agent-based solutions are not enough and the use of JTAG technology, optimized for Linux debugging, is required. JTAG is particularly useful in the following scenarios:

- When an agent is not available
- When a communication port is not available for agent-based debugging (typical in consumer devices to save bill of material costs)
- When security of host-based debugging is a concern

Another challenge that mobile device developers will face is the containment of development and product cost. This means the mobile device will not have wired ports that may be common in other devices. For example, mobile devices typically will have only mini-USB ports and they will lack Ethernet and serial ports. Given the high performance and lower power consumption requirements of mobile devices, it is imperative that device software is efficient and consumes the least amount of memory. Increased processor use and memory overhead will not only take away from the user experience of the mobile device but it can also add to the power consumption and can shorten battery life.

An Introduction to the i.MX Applications Processor Family
Freescale’s i.MX family is based on ARM core technology and designed for use in portable media players, smartphones, wireless PDAs, mobile infotainment, and many other mobile wireless applications. Board support packages are available for a broad range of popular open operating systems, such as Linux and key real-time operating systems (RTOSes).

Freescale’s i.MX family is engineered with Smart Speed technology. Smart Speed is an intelligent integrative approach that employs hardware accelerators to offload the CPU and a crossbar switch to bring parallelism to the system. The result is a processor that performs like a much higher megahertz device but conserves power for long battery life.

Wind River’s tools support the i.MX31, i.MX27, and i.MX21 applications processors.

i.MX31 and i.MX31L Applications Processors
Freescale’s applications processors feature an advanced and power-efficient implementation of the ARM1136JF-S core, which operates at speeds starting at 532MHz. With a vector floating point coprocessor and L2 cache, they are designed for any wireless device running computationally intensive multimedia applications such as portable media players and portable navigation devices. They are also ideal for mobile devices favored by the power user who keeps several applications running at once. Target devices include feature-rich smartphones, digital video recorders, digital cameras, mobile gaming consoles, mobile multimedia players, and many other mobile wireless applications. The i.MX31 processor adds a graphics accelerator to the functions of the i.MX31L.
i.MX27 Applications Processor

Derived from the popular i.MX21 processor and based on the ARM926EJ-S core, the i.MX27 processor adds an H.264 D1 hardware codec for high-resolution video processing, an Ethernet 10/100 MAC, security, plug-and-play connectivity, and more power-management features. The i.MX27 processor is an excellent choice for video and voice over Internet Protocol (V2IP), surveillance systems, intelligent remote controls, point-of-sale terminals, and many other wireless and embedded applications.

i.MX21 Applications Processor

Operating at speeds up to 266MHz, the ARM-based i.MX21 processor is designed to deliver the ultimate performance in multimedia to handheld devices. The processor can drive CIF 30fps video encode/decode, or videoconference at 15fps while sending an email, all at low power. The i.MX21 processor provides an exceptional video experience via special video encode/decode features and the ultimate 2-D/3-D experience, thanks to a bus master interface (BMI) to external graphics chips. USB On-the-Go (OTG) offers plug-and-play connectivity.

Developing and Debugging Operating Systems and Applications on Mobile Devices

The increased complexity of mobile products that are in development impacts the overall productivity of development teams. Tools that improve productivity by giving developers deep visibility into the embedded mobile system will help speed development. The rich functionality of i.MX applications processors results in the interaction of numerous peripherals that are integrated within the mobile consumer device—for example, i.MX21-based devices that interface to external graphics, USB OTG, and Wi-Fi. This increased integration will require visibility into the hardware and software interaction. Linux increases debug complexity as well with boot loader bring-up, kernel and kernel module stabilization, and application development.

Tools that easily support rapid innovation and commercialization are a critical must-have during the design and test phases of development. The development environments need to be open and easy to use. Support for development and debug of Linux devices is very important. Additionally, development and debug tools for mobile devices must be mindful of cost implications and therefore should not rely on the need for wired ports. Finally, development and debug should not require increased power and memory consumption in the final product.

Requirements for Developing Linux on Mobile Devices

The first step in troubleshooting Linux is to understand the Linux operating system and the areas in which a debugger needs to analyze interactions. Unlike traditional device operating systems, Linux comprises multiple interacting components and employs complex memory mapping and management schemes.

The components of the Linux operating system are the kernel, kernel modules, and application software. The Linux kernel is the core of the operating system and has ultimate authority. Kernel modules are elements of the operating system that are dynamically loaded and unloaded and are typically used for device drivers. The Linux kernel partitions all software to run in either user mode or kernel mode. Applications that run in user mode on a Linux system have reduced privileges. Debugging Linux embedded projects can be complex because development spans the boot loader, kernel, kernel module, and applications.

Debug Challenges in Linux

- Hardware initialization
- Debugging the boot loader
- Access to instruction and data cache
- Access to initialization code in the kernel
- Debugging interactions between applications and kernel

While developing a mobile consumer device on the i.MX processor, the Linux developer will be required to establish a target configuration file for the boot loader, traverse Linux virtual addressing from user mode to kernel mode and back, map Linux kernel symbol information, and troubleshoot across the user and kernel space. These tasks are not easily solved with agent-based debugging solutions such as KGDB or GDB. Therefore, debugging a Linux-based embedded system requires powerful tools that expose the operation and the interaction with the hardware.
Cost-Sensitive Mobile Development Requirements

System access in mobile devices is challenging. Size, cost, and power constraints restrict adding Ethernet ports on these types of devices. Additionally, depending on the development cycle, a working IP stack or communication port may not be available. This means that traditional software-based development and debug solutions may not be viable or cost-effective for mobile devices. An excellent alternative is to use the JTAG interface of the i.MX family of applications processors. Wind River and Freescale have worked closely to test and validate the JTAG interface in i.MX applications processors to enable developers to perform extensive debugging and diagnostics of mobile consumer devices.

The JTAG interface on all i.MX applications processors enables a programmer to access an on-chip debug module that is integrated in the CPU and offers powerful and direct control of the processor. For embedded device developers, the advantage of using JTAG to debug is that it offers direct control of the processor and eliminates the need for communication ports, instrumentation, and agent-based software. The added benefit is that using JTAG-based development tools eliminates the need to add agent software that can consume CPU cycles and memory, thereby improving performance and enhancing battery life.

Improving Test and Validation

The increased complexities of mobile devices require a lengthy test and validation phase that comprises many edit-debug-compile cycles. During development, there may be limited access to target systems and these systems may be scattered across the globe. The use of multiple development tools, with the need for training and expertise of developers, affects productivity. With remote development teams, inefficiencies can become an issue if the development infrastructure does not support remote development. All of this can impact delivery of the final product within a prescribed time-to-market window. Enabling remote development was a key requirement for the joint development solution that Freescale and Wind River offer. It was critical that the offered solution be complete and support remote development.

Development Ecosystem for i.MX Applications Processors

Freescale Semiconductor and Wind River have a strategic alliance that helps customers enhance system performance, reduce cost, and accelerate time-to-market. The alliance unites the breadth and depth of each company’s products and technologies to enable optimized solutions designed to offer an easy, productive “out-of-the-box” experience for embedded system and device software developers.

Workbench On-Chip Debugging Edition Benefits

- Bring up new hardware quickly—even before an operating system is available
- Ensure that the operating system and middleware run effectively on the hardware early in the development cycle
- Enhance product quality and speed time-to-market
- Improve overall development productivity by standardizing on a common Eclipse-based debug platform
- Debug complex problems that require visibility between Linux system domains: hardware, OS, kernel modules, and applications

A cornerstone of the alliance is to provide extensive support for Freescale’s i.MX applications processors. This support is delivered across Wind River’s broad portfolio of Device Software Optimization (DSO) solutions, including Wind River Workbench, On-Chip Debugging Edition, which is an award-winning Eclipse-based development tools suite that provides advanced JTAG-based debugging of mobile devices, including support for Linux-based devices.

Debugging embedded Linux products traditionally required a mixture of hardware and software tools—JTAG tools for hardware bring-up and agent-based solutions for software development. These tools often solved point problems but were not designed for integrated Linux development. Wind River has changed the way developers can debug Linux by merging traditional JTAG hardware debugging with kernel and user-space analysis in an Eclipse-based development environment. The extensive collaboration between Freescale and Wind River allows developers to use the JTAG connection when traditional agent-based solutions are not technically or economically feasible.

With advanced JTAG tools from Wind River and sophisticated mobile consumer device applications processors from Freescale, developers can debug the hardware, boot loader, kernel, and user space. This enables them to spot system crash events and other problems that occur between the kernel, user space, and target. Moreover, developers using the Wind River Linux operating system can move seamlessly between the agent and JTAG debugging. These innovations in on-chip debugging provide powerful alternatives for projects where agent-based debugging is either not available or isn’t the most cost-effective solution.
An example that highlights the benefits of the combined Wind River-Freescale solution is a handheld satellite receiver developed by Mobile Broadcasting Corporation. This mobile consumer device is built using an i.MX processor running Wind River Linux and was developed using Wind River Workbench, On-Chip Debugging Edition. The Freescale i.MX processor improved development efficiency by 20 percent and accelerated time-to-market by two months. This customer success shows how Freescale and Wind River are working together to help engineers design and develop cutting-edge consumer and mobile products.

Productivity-Enhancing Debug Solution

Based on the industry-standard Eclipse framework and technology-leading hardware diagnostic capabilities, Workbench On-Chip Debugging software combines hardware debugging with a project-based, integrated development environment to reduce costs and complexity. It provides a centralized, standards-based development environment that automates the debugging and analysis process. Its project-oriented methodology simplifies the handoff between various phases of development, from initial hardware bring-up to firmware design, OS implementations, and platform and application development.

Features for Debugging Linux on Mobile Devices

- Boot loader development, enabling the ability to debug issues when bringing up Linux on i.MX applications processors
- Kernel-mode debugging to troubleshoot OS and device driver problems
- User-mode debugging without kernel instrumentation, communications channel, or IP stack
- Advanced features in user-mode debugging that allow developers to see the system and application context, including how the kernel was entered, by which thread (or threads), parameters used, and thread variables
- Comprehensive system-mode debugging: breakpoints that halt the entire processor and all threads

Workbench On-Chip Debugging software is based on the Eclipse framework, providing the DSO industry’s integrated JTAG debugging environment that can be easily extended through in-house, third-party, open source, or commercially available Eclipse plug-ins. Wind River on-chip debugging solutions enable you to take advantage of the growing Eclipse ecosystem to improve collaboration between hardware, firmware, and software developers, ensuring a smooth handoff in the debug process.

In a global environment, efficiency and productivity gains are realized when development teams adopt solutions that are easy to use and minimize training and ramp-up time. Coupled with Wind River ICE 2, an advanced network-based emulator, fully functional remote development and collaboration features enable efficient development across global development teams.

Wind River Workbench is a comprehensive solution that also includes powerful code analysis capabilities as well as integrated agent debugging capabilities. The code analysis capabilities provide insight into memory usage and run-time performance. Powerful system viewer features help to visualize OS events to improve run-time efficiency. Profile and data scop ing analysis show vital information about running programs to better test and validate the system.

Support for Linux-Based Mobile Devices

Running on the industry-standard Eclipse platform, Workbench On-Chip Debugging offers advanced JTAG-based debugging features for mobile devices such as Freescale’s i.MX applications processors and full support for debugging these devices for Linux. Wind River Workbench, On-Chip Debugging Edition supports a variety of Linux distributions including Wind River Linux. For other Linux distributions such as roll-your-own Linux or semiconductor Linux that are based on kernel.org Linux, Wind River has extended the on-chip debugging capabilities of Workbench to enable debugging of these distributions as well. The
JTAG-based debug capabilities for Linux devices help reduce bill of materials cost by eliminating the need for expensive ports on the mobile device. Workbench On-Chip Debugging uses the JTAG interface without the use of agent software, IP stack, and communications port or kernel instrumentation for all device development.

Freescale and Wind River Solutions for Consumer and Mobile Devices

- Validated and tested JTAG interface for on-chip debugging
- Comprehensive Linux development and debug environment
- End-to-end development and platform solutions
- World-class support and professional services for customized solutions

With the challenges of time-to-market pressure and containment of development and product costs, today’s consumer and mobile device manufacturers require comprehensive solutions that encompass embedded processors and development environments. Freescale develops best-in-class applications processors, and Wind River provides optimized development and platform solutions for embedded devices. The combined solution ensures success for sophisticated consumer and mobile products.

This paper is cosponsored with Freescale.