



Green Hills Software's INTEGRITY® RTOS Exploits Atmel's AT91RM9200 MCU Architecture to Optimize Performance, Responsiveness and Security

*Peter Bishop, Communications Manager, Atmel Corporation
Amy Brock, Systems Software Engineer, Green Hills Software, Inc.*

Summary

Real-time applications for embedded microcontrollers are complex, requiring high code and data bandwidths running under prioritized, multi-tasking regimes. Mission-critical tasks require a deterministic response from both hardware and software resources. Green Hills Software's INTEGRITY® real-time operating system (RTOS) exploits the advanced architecture of Atmel's AT91RM9200 microcontroller (MCU) based on the ARM920T™ processor in order to optimize system performance, responsiveness and security. INTEGRITY fully utilizes the ARM920T Memory Management Unit (MMU) to guarantee memory availability and protection. Its advanced scheduler ensures processor availability for the kernel and user tasks. It fully exploits the Advanced Interrupt Controller (AIC) to provide rapid, prioritized and secure interrupt handling and context switching.

Table of Contents

Background.....	3
Typical Embedded Microcontroller System Architecture	4
INTEGRITY Overview.....	5
AT91RM9200 Architecture Overview	6
INTEGRITY Key Features	8
Virtual Memory Management	8
Processor Scheduling	9
Interrupt Handling and Context Switching.....	10
AT91RM9200 Software Development with INTEGRITY.....	10
Conclusion	12
References	13
Editor's Notes	13
About Atmel Corporation.....	13
About Green Hills Software, Inc.	13

Background

Real-time applications based on embedded microcontrollers multiplex a number of tasks with greatly differing priorities. At the one extreme are mission-critical tasks such as computing a positional fix or decrypting an incoming signal stream. At the other extreme are housekeeping tasks such as the accumulation and reporting of statistics. At all times a deterministic response is required to input from the user interface, and at the same time a user display requires real-time update. Data integrity is paramount, and a security mechanism must be maintained against internal errors in application code as well as against external intruders.

These requirements demand appropriate levels of performance and responsiveness from all elements of the application system: hardware, operating system and application software. Specifically these include the need for:

- **Always-on systems.** Downtime as a consequence of system failures or for upgrades or maintenance must be kept to a minimum. For stand-alone, battery-powered systems this also mandates a minimal power consumption to limit the outages due to battery exchange or re-charge.
- **High performance from the embedded processor core, interfaces and peripherals.** The RTOS overhead needs to be kept to a minimum, both in execution time and memory footprint.
- **Deterministic response to high-priority real-time events.** This can only be achieved by a combination of high processor performance together with memory allocation and scheduling algorithms that guarantee the immediate availability of all the system resources required by the high-priority task.
- **Rapid context switching** in order to minimize the overhead from frequent interrupts and fine-grain multi-tasking.
- **Fail-soft or limited consequences of errors** in application software modules. This requires robust protection mechanisms in memory allocation and time slicing.

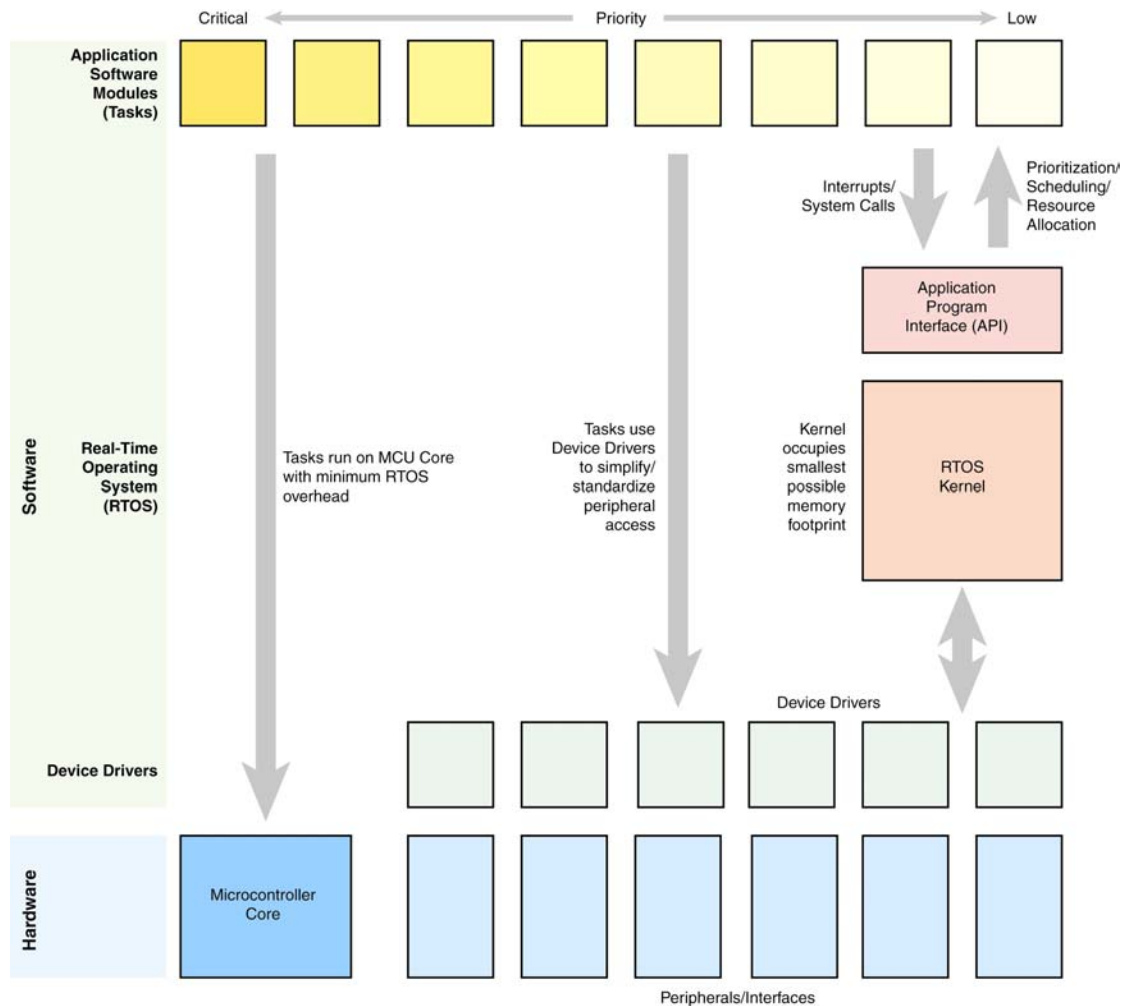
From the point of view of developing applications, a clean virtual machine interface is mandatory, as is tight integration into the application development environment. From the business point of view, a royalty-free model allows application developers to predict and control their expenditures.

INTEGRITY[®], the state-of-the-art RTOS from Green Hills Software Inc. has been fine-tuned to take advantage of the advanced architectural features of Atmel's AT91RM9200 ARM920T-based microcontroller in order to meet all the challenges outlined above. The remaining sections of this White Paper explain how this is achieved.

Typical Embedded Microcontroller System Architecture

Figure 1 shows the hardware/software architecture and interactions of a typical embedded microcontroller system.

Figure 1. Typical Embedded Microcontroller Hardware/Software Architecture



The RTOS kernel resides between the application software modules (tasks) and the device drivers. It handles interrupts, system calls, and provides resource allocation, scheduling and prioritization services. The device drivers provide standardized access mechanisms to the underlying register structure of the devices.

The RTOS kernel should occupy a minimum memory footprint and take up a minimum of processor time, thus enabling the tasks to run on the MCU core with minimum RTOS overhead.

INTEGRITY Overview

INTEGRITY was designed to take full advantage of the architectural features of state-of-the-art 32- and 64-bit embedded microcontrollers. Its streamlined kernel delivers minimum interrupt latency and predictable real-time response under the most demanding conditions of use, as the kernel never disables the processor's interrupt handling mechanism.

Virtual memory management in INTEGRITY allows the system designer to guarantee run-time security and resource availability. INTEGRITY utilizes the Memory Management Unit (MMU) of the ARM920T to create multiple partitioned and protected virtual address spaces. Tasks are prevented from corrupting memory in other address spaces, either inadvertently or maliciously. Each address space receives its own pool of memory upon creation, which protects it from being starved by the memory needs of the rest of the system. Using the INTEGRITY allocation model, a developer can prove that a complex program will execute as expected.

INTEGRITY is fully compatible with the MULTI[®] Integrated Development Environment, also from Green Hills Software. MULTI includes Green Hills Software's high performance optimizing compilers for the ARM920T, and has a number of unique features that have been implemented to support application development with INTEGRITY. Together these tools facilitate and accelerate the development and debugging of application code modules. For operational systems, INTEGRITY supports non-intrusive field debugging and software upgrades.

From a business perspective, no run-time royalties are payable per deployed copy of INTEGRITY. This enables system integrators to plan and control their system development costs.

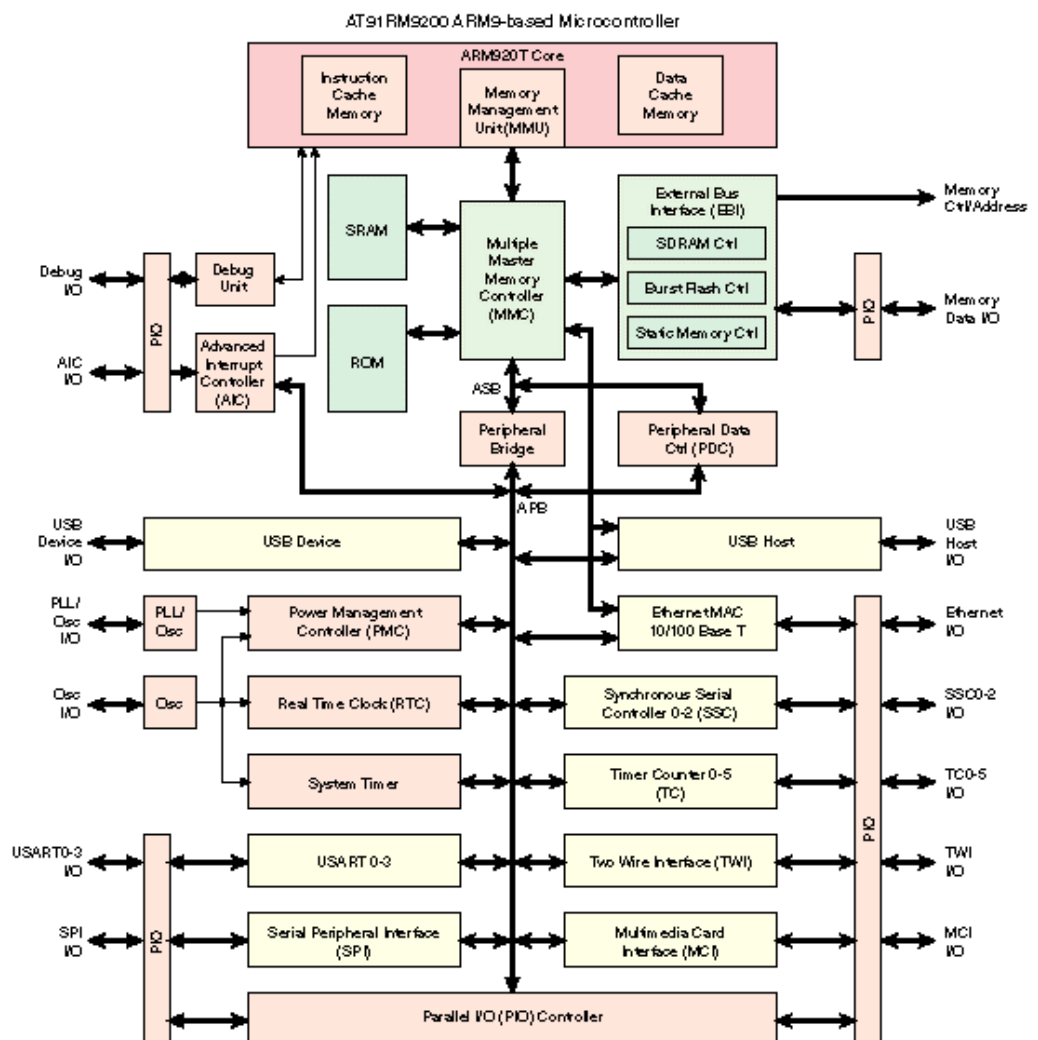
Figure 2. The INTEGRITY Package



AT91RM9200 Architecture Overview

Figure 3 shows a simplified block diagram of the architecture of the AT91RM9200. It is built around an ARM920T processor core that attains more than 200 MIPS thanks to its Harvard architecture with separate instruction and data cache memories and buses, and pipelined instruction processing flow. Its main feature of interest to an RTOS is its Memory Management Unit (MMU). The MMU supports a 4-gigabyte virtual address space common to the core and peripherals. It provides address translation and access permission at a physical level. These facilities are exploited by INTEGRITY at the application level.

Figure 3. Simplified View of AT91RM9200 Architecture



The Multiple-master Memory Controller (MMC) enables devices other than the ARM920T core to become memory masters. This makes most efficient use of the throughput of the memory address and data buses and reduces the processor overhead during memory

transfers. The External Bus Interface (EBI) provides direct communication to a variety of off-chip memories and memory cards including Flash, SDRAM, CompactFlash® and SmartMedia™. This flexible combination allows the memory architecture of the system to be optimized for the application.

All AT91RM9200 peripherals are configured and controlled by a set of registers visible within the ARM920T address space. The various peripherals have a nearly-identical register structure for ease of programming. Software drivers are provided for all peripherals enabling most operations to be handled at an abstract level (and in a high-level language) by application code. Data transfer to and from the serial peripherals is via the 20-channel Peripheral Data Controller (PDC) that provides a DMA function to keep processor overhead to a minimum during bulk data transfers.

One system peripheral of interest to an RTOS is Atmel's proprietary Advanced Interrupt Controller (AIC). This transforms the basic two-level interrupt structure of the ARM920T core into an eight-level priority, individually maskable, vectored interrupt handling regime. An interrupt source is allocated to each AT91RM9200 peripheral device, as well as to seven external interrupt lines and to software-sourced exceptions. INTEGRITY takes full advantage of these features and never over-rides them: under no circumstances does INTEGRITY disable interrupts at the processor level.

The rich set of register-configurable, high-performance peripherals includes remote connectivity (USB Host and Device), network connectivity (Ethernet 10/100 Base-T MAC), local peripheral interfacing (Serial Peripheral Interface and Two-wire Interface), storage (Multimedia Card Interface) and control (3-channel Synchronous Serial Controller, 6-channel Timer/Counter). They enable the AT91RM9200 to be integrated seamlessly into a wide variety of high-performance applications. All peripherals are programmable in the address space of the ARM920T core, facilitating application software development.

Power consumption by the ARM920T core and the entire peripheral set is regulated by the Power Management Controller that provides independent peripheral clock control (including a real-time clock) and implements an idle mode in which the processor clock is stopped until an interrupt is received.

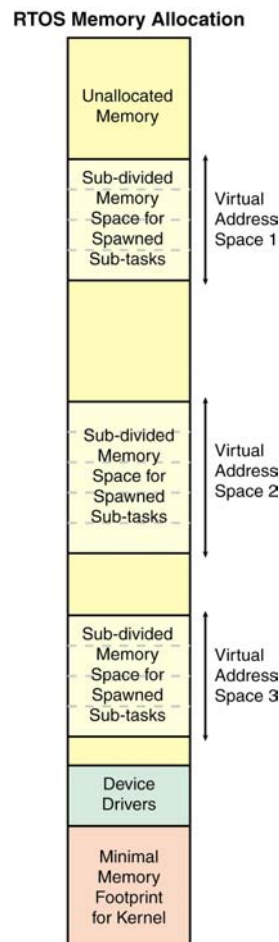
INTEGRITY – Key Features

The key INTEGRITY features needed by developers of AT91RM9200 applications are virtual memory management, processor scheduling and interrupt handling/context switching. These are discussed in the three following sections.

Virtual Memory Management

INTEGRITY's unique memory quota system guarantees memory availability to the kernel, application and system tasks, and device drivers. It assigns memory for these objects at system initialization time, according to a static design model, using INTEGRITY's Integrate utility. During operation, if a parent task spawns child tasks, the corresponding memory is taken from the memory allocated to the parent's virtual address space rather than drawing memory from a common pool, as is done in most traditional RTOSes. This prevents the finite memory space from being overwhelmed by pitfalls such as intentional or unintentional recursive task creation. Figure 4 illustrates this memory management scheme.

Figure 4. INTEGRITY Memory Management



For virtual partitions, INTEGRITY detects memory access violations, preventing an errant task from deliberately or inadvertently reading, writing or executing memory outside its own partition. This isolates errors within an address space, and allows the kernel and other virtual partitions to continue execution unaffected. By signaling a violation, INTEGRITY enables the system to respond in a fail-safe or fail-soft manner to a wide range of application software errors.

Within the virtual address space is the set of virtual device drivers that make the device-dependent aspects of peripheral control and data transfer transparent to application software tasks. This greatly simplifies the writing of application software code.

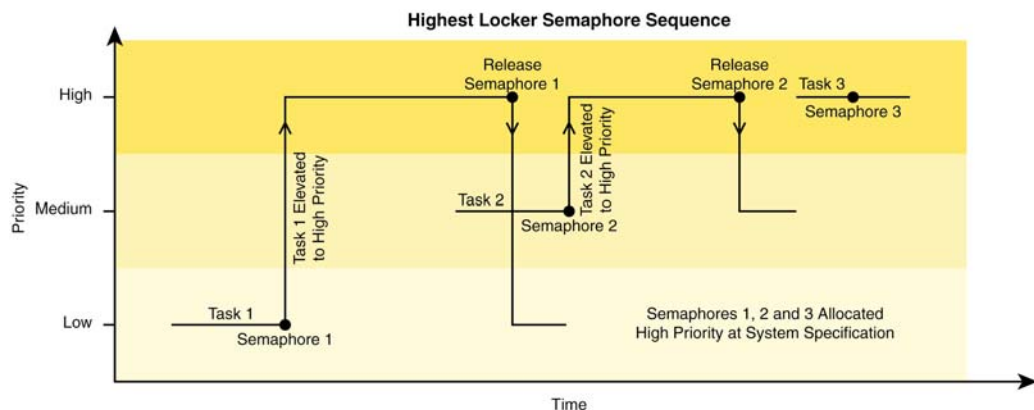
Processor Scheduling

INTEGRITY uses a two-level priority based scheduler. The task with the highest priority that is runnable will get control of the processor. Between ready tasks with equal priority, a weighted round-robin time-slice scheme applies. A task's weight dictates how many times per scheduling round the task gets to run; a task with weight 3 gets to run three times as often as a task with weight 1. In order to prevent tasks from monopolizing processor time, if a task spawns sub-tasks, the scheduling weight is sub-divided.

One potential problem of prioritized task scheduling is priority inversion, or deadlock due to improperly handled communication between tasks of different priorities. The classic situation is where a high priority task and a low priority task share a semaphore for synchronization. The low priority task owns the semaphore and when the high priority task becomes runnable, it immediately blocks on the semaphore. Suppose a third task with medium priority is running. This task can prevent the high priority task from running, because it prevents the low priority task from releasing the desired semaphore. This inversion can jeopardize correct run-time behavior of the system because it violates the priority assumptions for timing analysis.

The solution implemented by INTEGRITY is to use a special semaphore construct to raise temporarily the priority of the semaphore owner to the highest priority of any task that could block on the semaphore. This highest locker semaphore technique prevents priority inversion and chained blocking, as shown in Figure 5.

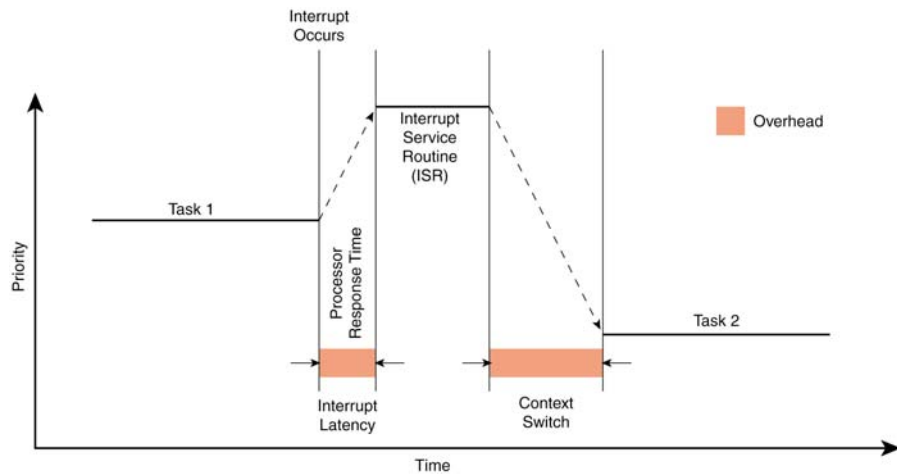
Figure 5. Highest Locker Semaphore



Interrupt Handling and Context Switching

INTEGRITY fully exploits the Advanced Interrupt Controller (AIC) of the AT91RM9200 to provide rapid, prioritized, secure interrupt handling. The kernel never masks or blocks interrupts, and avoids instructions such as division that have long latencies and could temporarily block interrupts. By providing a vectored, prioritized interrupt handling mechanism implemented in hardware, the AIC enables transfers to interrupt handlers to be achieved in a minimum number of instruction cycles. Similarly, context switching is achieved with minimum delay. See Figure 6 for details.

Figure 6. INTEGRITY Interrupt Handling and Context Switching

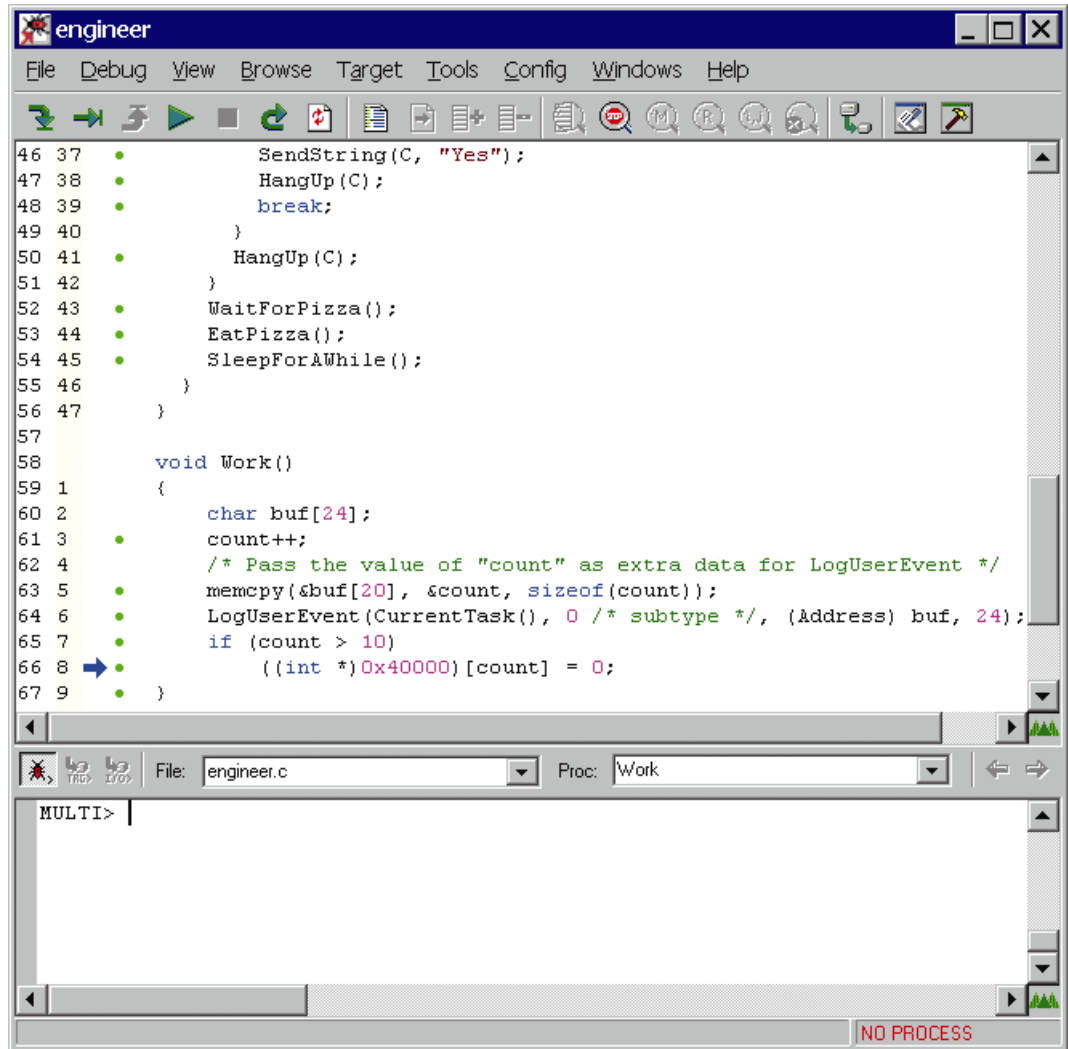


AT91RM9200 Application Software Development with INTEGRITY

The AT91RM9200 Board Support Package (BSP) includes Green Hills Software's MULTI[®] Integrated Development Environment (IDE) as well as the INTEGRITY port for the AT91RM9200. MULTI provides a complete set of mutually-aware tools, including everything that is needed to complete a major programming project where time-to-market is of the essence. These tools include some of the fastest C/C++ compilers in the industry, graphical project builder, source level debugger, run-time error checking, performance profiler, class browser and editor.

The MULTI Source-level Debugger provides INTEGRITY-aware debugging and has facilities for program loading, execution, run control and monitoring. Its graphical user interface is shown in Figure 7, featuring icon buttons for common actions, together with a source code window and a command window. Its intuitive user interface puts its advanced features at the disposal of application developers with a minimum of training.

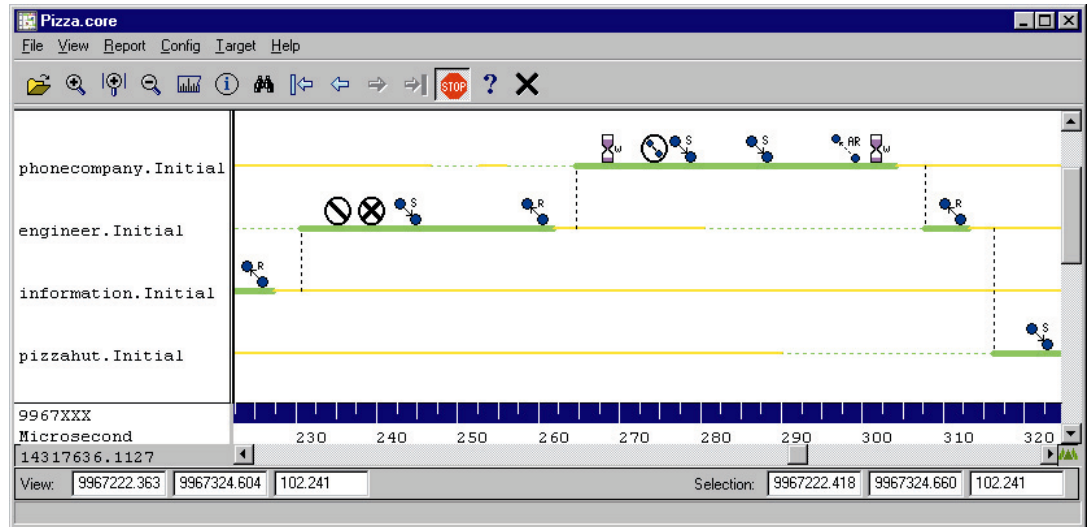
Figure 7. MULTI Source-level Debugger Window



MULTI also provides tools unique to INTEGRITY including EventAnalyzer™, ResourceAnalyzer™, Tracepoints and the ISIM simulator.

- EventAnalyzer enables the application developer to view and track software events in a full-featured graphical viewer, as illustrated in Figure 8. These events include context switches, interrupts, semaphore give/take, message send/receive and other user-defined events. This gives a level of understanding of the complex real-time interactions within the system that is difficult to achieve by other means.

Figure 8. MULTI EventAnalyzer Window



- ResourceAnalyzer™ is a development and run-time analysis tool that provides extensive visibility into CPU execution at the task and address space level. The ResourceAnalyzer is a graphical interface to critical system information including task stack usage, address space memory use, CPU time use history, and memory use history.
- INTEGRITY's Tracepoints enable non-intrusive data collection in running systems. When a task hits a tracepoint, the specified registers and memory are probed and their values stored for later view. The task continues with only brief interruption, making tracepoints ideal for troubleshooting or monitoring a fielded system.
- ISIM enables programmers to develop and test embedded INTEGRITY-based applications when target hardware is not yet manufactured or the supply is too limited to accommodate the entire programming team. ISIM completely simulates INTEGRITY's API, including virtual memory, peripherals, and TCP/IP.

Conclusion

The synergy between the AT91RM9200 and INTEGRITY gives application developers the benefits of maximum system performance due to minimum OS overhead, in particular rapid, secure interrupt handling and context switching. Critical tasks run with absolute priority thanks to guaranteed system resources, both memory and processor availability. Secured memory partitioning ensures reliability and protection against inadvertent or deliberate errors in application software modules.

Application software developers benefit from a Board Support Package that provides a complete and highly integrated development environment including compilers, source code debug, event analysis, RTOS and Green Hills Probe™ hardware debug. The royalty-free licensing package keeps application development costs under control.

References

Green Hills Software Inc., Web: www.ghs.com

ARM Ltd., Web: www.arm.com

Editor's Notes

About Atmel Corporation

Founded in 1984, Atmel Corporation is headquartered in San Jose, California with manufacturing facilities in North America and Europe. Atmel designs, manufactures and markets worldwide, advanced logic, mixed-signal, nonvolatile memory and RF semiconductors. Atmel is also a leading provider of system-level integration semiconductor solutions using CMOS, BiCMOS, SiGe, and high-voltage BCDMOS process technologies.

Further information can be obtained from Atmel's Web site at www.atmel.com.

Contact: Peter Bishop, Communications Manager, Atmel Rousset, France, Tel: (+33) (0) 4 42 53 61 50, e-mail: pbishop@atmel.com.

About Green Hills Software Inc.

Founded in 1982, Green Hills Software Inc. is the technology leader for real-time operating systems and software development tools for 32- and 64-bit embedded systems. Green Hills Software's royalty-free INTEGRITY® real-time operating system, fully integrated with its market leading compilers and MULTI® Integrated Development Environment, provides a total development and run-time solution that addresses both deeply embedded and maximum reliability applications.

Green Hills Software is headquartered in Santa Barbara, California, with European headquarters in the United Kingdom.

For more information on Green Hills Software products, call 805-965-6044, e-mail sales@ghs.com or consult the Green Hills Web site at www.ghs.com.

© Atmel Corporation 2003. All rights reserved.

Atmel, DataFlash, FingerChip, the Atmel logo and combinations thereof are registered trademarks, and others contained herein are trademarks, of Atmel.

ARM, ARM920T, ARM7TDMI, MultiTrace and Multi-ICE are trademarks and registered trademarks of ARM Ltd. Green Hills Software, the Green Hills Software logo, INTEGRITY, MULTI, Probe, Slingshot, ResourceAnalyzer and EventAnalyzer are registered trademarks and trademarks of Green Hills Software Inc. CompactFlash is a registered trademark of the CompactFlash Association. SmartMedia is a trademark of the Solid State Floppy Disk Card Forum. SD Card is a trademark of the SD Card Association.

Other terms and product names in this document may be the trademarks of others.