



# EMBEDDED SOFTWARE DEVELOPMENT TOOLS

# THREADX® FOR MIPS

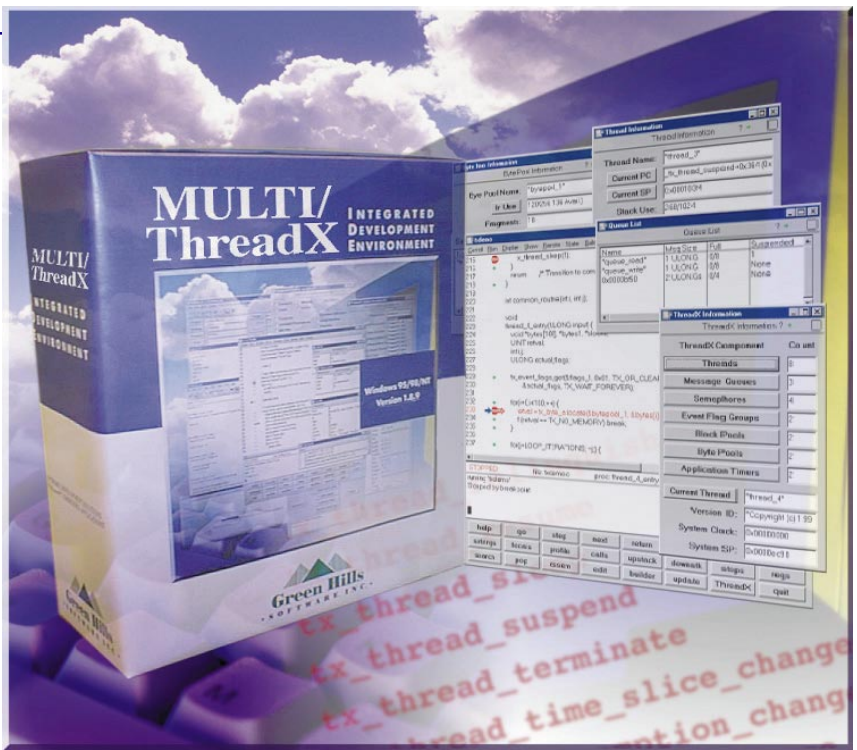
## THREADX RTOS

The ThreadX Real-Time Operating System is a highly efficient, robust, royalty-free kernel designed for deeply embedded applications requiring a small footprint and rapid real-time response. ThreadX provides super-fast context switching times while giving developers a rich set of services to use for task control and communication. ThreadX is fully integrated with Green Hills Software's MULTI® Integrated Development Environment, providing Optimizing Compilers, Source Debugger, Graphical Project Builder and many other powerful tools for development of embedded applications.

### HIGHLIGHTS

- Complete MIPS family support
- Reasonable pricing
- No Royalties
- Complete ANSI C source code
- Easy to use and powerful services
- Responsive Technical Support
- Unlimited Threads, Queues, Event Flags, Timers, Semaphores, Block Pools, and Byte Pools
- Flexible memory usage
- Timeout available on all thread suspension
- Advanced preemption-threshold technique
- Low-overhead Application Timers
- Size scales automatically
- picokernel architecture for size and speed
- Small footprint (sizes in bytes):
 

Instruction area size:	5-25K
Global data area:	696
Thread control block:	144
Timer control block:	44
Semaphore control block:	28
Queue control block:	56
Event Flag control block:	32
Block Memory control block:	48
Byte Memory control block:	52



- Fast Execution (33Mhz, kseg0, R3000):
 

context switch	1.4us
tx_thread_suspend	3.6us
tx_thread_resume	3.2us
tx_thread_relinquish	0.9us
tx_semaphore_get	1.3us
tx_semaphore_put	1.0us
tx_queue_send	2.3us
tx_queue_receive	2.3us
tx_event_flags_set	2.5us
tx_event_flags_get	2.5us
tx_block_allocate	2.1us
tx_block_release	1.4us
tx_byte_allocate	4.8us
tx_byte_release	2.2us

v1, a0-a3, and t0-t9). The full register set is saved only if thread preemption is required.

## MIPS OPTIMIZATIONS

ThreadX optimizes context switching on the MIPS. When context switching occurs inside of a ThreadX service call, only the registers preserved across function calls are saved as part of the thread's context, i.e. registers s0-s8.

A similar technique is used in interrupt handling. On the front end of interrupt service routines, only the compiler's scratch registers are saved initially (registers at v0-

## IMPROVE YOUR MIPS DEVELOPMENT

Let our extensive experience with the MIPS family of microprocessors help your product development. The ThreadX high-performance real-time kernel, helps improve your product's quality and its time-to-market. In addition, using ThreadX makes it easier to enhance your product in the future.

## EASY TO USE

ThreadX is designed for ease of use. The API is designed to be easy to understand, powerful, and consistent. The same is true with our reference manual and other supporting documentation.



THREADX FOR MIPS