

## ThreadX Highlights

- Complete ARC600, ARC700, ARC EM, ARC HS Support
- Reasonable Pricing
- **No Royalties**
- Complete ANSI C Source Code
- MISRA-C:2004 and MISRA C:2014 Compliant
- Safety-Certification for IEC 61508 SIL 4, IEC 62304 Class C, ISO 26262 ASIL D, UL/IEC 60730, UL/IEC 60335, UL 1998, and EN 50128 SW-SIL 4
- Easy to use API
- Small Size & Scales Automatically
- Fast Real-Time Response
- Picokernel™ Architecture
- Preemption-Threshold™ Technology
- Event-Chaining™ Technology
- Run-Time Stack Analysis
- Run-Time Performance Metrics
- Event-Trace Capability
- Unlimited Threads, Queues, Event Flags, Semaphores, Mutexes, Timers, Block and Byte Memory Pools
- Deterministic Processing
- Optimized Application Timers
- Fully Integrated with FileX, GUIX, NetX, NetX Duo, and USBX
- Extensive Development Tool Choices
  - MetaWare
  - GNU
- ThreadX Kernel Aware Debugging
  - MetaWare
  - Lauterbach
- Optimized Interrupt Handling
  - Most ThreadX services are allowed from ISRs
  - System stack used in ISRs
  - Fast Interrupt with Hardware Register Bank Support
  - Nested Interrupt Support
- ARC HS Hardware Register Bank Per-Thread Context Support

## Small Footprint (all sizes in bytes)

Instruction Area Sizes	
<b>Core Services:</b>	<b>3,886</b>
Queue Services:	1,158
Event Flag Services:	946
Semaphore Services:	472
Mutex Services:	1,436
Block Memory Services:	602
Byte Memory Services:	1,102

RAM Sizes	
Global Data Area:	0.5-2KB
Thread Control Block:	188
Timer Control Block:	44
Queue Control Block:	60
Semaphore Control Block:	32
Mutex Control Block:	52
Event Flag Control Block:	40
Block Memory Control Block:	48
Byte Memory Control Block:	52

## Fast Execution (ARC HS Processor @400MHz)

	IR	TS	TR	TRCS
tx_thread_suspend	0.23µs	0.40µs	-	-
tx_thread_resume	-	-	0.17µs	0.54µs
tx_thread_relinquish	0.06µs	-	-	0.46µs
tx_queue_send	0.21µs	0.39µs	0.30µs	0.58µs
tx_queue_receive	0.20µs	0.41µs	0.47µs	0.68µs
tx_semaphore_get	0.13µs	0.45µs	-	-
tx_semaphore_put	0.11µs	-	0.25µs	0.46µs
tx_mutex_get	0.19µs	0.43µs	-	-
tx_mutex_put	0.19µs	-	0.49µs	0.62µs
tx_event_flags_set	0.13µs	-	0.38µs	0.56µs
tx_event_flags_get	0.13µs	0.40µs	-	-
tx_block_allocate	0.15µs	0.40µs	-	-
tx_block_release	0.13µs	-	0.26µs	0.48µs
tx_byte_allocate	0.26µs	0.52µs	-	-
tx_byte_release	0.15µs	-	0.66µs	0.80µs
Context Switch (CS):		0.35µs		
Boot Time (BT):		5.0µs		
Interrupt Latency Range (ILR):		0.0µs - 0.50µs		

**Immediate Response (IR):** Time required to process the request immediately, i.e., no thread suspension or thread resumption.

**Thread Suspend (TS):** Time required to process the request when the calling thread is suspended due to unavailability of the resource.

**Thread Resumed (TR):** Time required to process the request when a previously suspended thread (of the same or lower priority) is resumed as a result of the request.

**Thread Resumed and Context Switched (TRCS):** Time required to process the request when a previously suspended higher-priority thread is resumed as a result of the request. Since the resumed thread is higher-priority, a context switch to the resumed thread is also performed from within the request.

**Context Switch (CS):** Time required to save current thread's context, find highest priority ready thread, and restore its context.

**Boot Time (BT):** Time required from tx\_kernel\_enter to the dispatch of the first thread.

**Interrupt Latency Range (ILR):** Amount of time interrupts are disabled.