



Application Note

AN1203: ThreadX[®] Kernel Awareness

Contents

Introduction.....	4
Intended Readers.....	4
Usage.....	5
Requirements.....	5
Finding the Views.....	5
Thread List.....	6
Semaphores.....	7
Mutexes.....	8
Message Queues.....	8
Event Flags.....	9
Timers.....	10
Memory Block Pools.....	11
Memory Byte Pools.....	12

INTRODUCTION

This Application Note provides information regarding the debug support and kernel awareness features for the Express Logic ThreadX® real-time operating system, available in the **Atollic® TrueSTUDIO®** product.

The kernel awareness features for Express Logic ThreadX in **Atollic TrueSTUDIO** provide the developer with a detailed insight into the internal data structures of the ThreadX kernel. During a debug session, the current state of the ThreadX kernel and the various ThreadX kernel objects such as tasks, mailboxes, semaphores and software timers, can be easily inspected in a set of dedicated views, in the **Atollic TrueSTUDIO** debug perspective.

INTENDED READERS

This document is primarily intended for software developers using the Express Logic ThreadX operating system in **Atollic TrueSTUDIO** projects.

USAGE

This section outlines the information provided in the ThreadX-specific debugger views in the *Atollic TrueSTUDIO* debugger. After reading this section, you should be able to use this information in your own project.

REQUIREMENTS

The kernel awareness features described in this document is based on ThreadX Cortex-M4/GNU Version G5.5.5.0.

FINDING THE VIEWS

A number of debugger views are available in the *Atollic TrueSTUDIO Debug* perspective when debugging an application containing the ThreadX real-time operating system.

These views are available from either the **View** top level menu or the **Show View** toolbar dropdown list button.

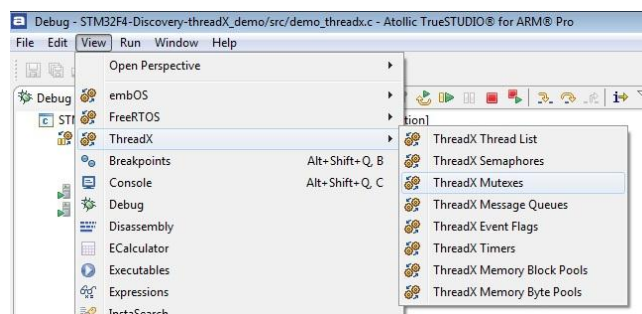


Figure 1 - View Top Level Menu

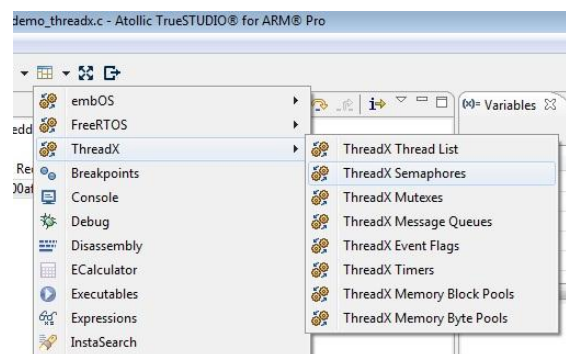
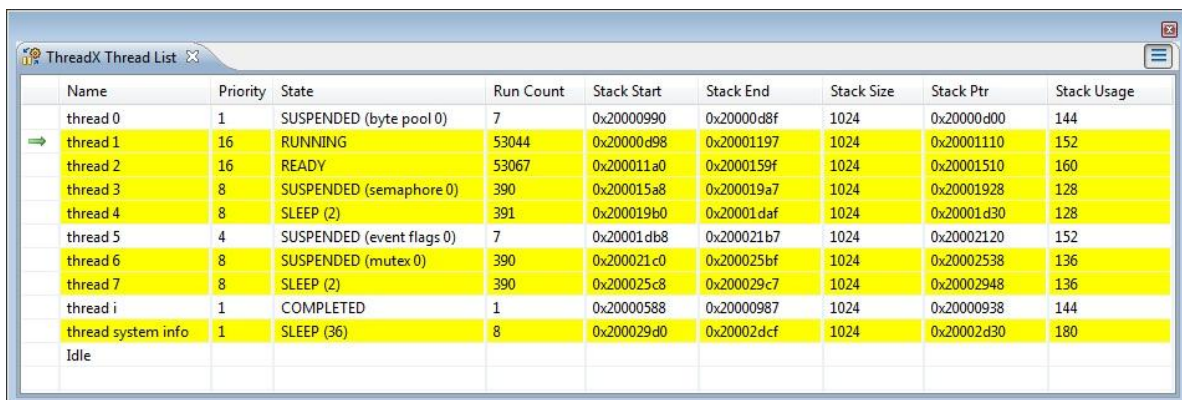


Figure 2 - Show View Toolbar Button

THREAD LIST

The **ThreadX Thread List** view displays detailed information regarding all available threads in the target system. The thread list is updated automatically each time the target execution is suspended.

There is one column for each type of thread parameter, and one row for each thread. If the value of any parameter for a particular thread has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Priority	State	Run Count	Stack Start	Stack End	Stack Size	Stack Ptr	Stack Usage
thread 0	1	SUSPENDED (byte pool 0)	7	0x20000990	0x20000d8f	1024	0x20000d00	144
→ thread 1	16	RUNNING	53044	0x20000d98	0x20001197	1024	0x20001110	152
thread 2	16	READY	53067	0x200011a0	0x2000159f	1024	0x20001510	160
thread 3	8	SUSPENDED (semaphore 0)	390	0x200015a8	0x200019a7	1024	0x20001928	128
thread 4	8	SLEEP (2)	391	0x200019b0	0x20001daf	1024	0x20001d30	128
thread 5	4	SUSPENDED (event flags 0)	7	0x20001db8	0x200021b7	1024	0x20002120	152
thread 6	8	SUSPENDED (mutex 0)	390	0x200021c0	0x200025bf	1024	0x20002538	136
thread 7	8	SLEEP (2)	390	0x200025c8	0x200029c7	1024	0x20002948	136
thread i	1	COMPLETED	1	0x20000588	0x20000987	1024	0x20000938	144
thread system info	1	SLEEP (36)	8	0x200029d0	0x20002dcf	1024	0x20002d30	180
Idle								

Figure 3 - ThreadX Thread List View

Please note that due to performance reasons, stack analysis (the **Stack Usage** column) is disabled by default. To enable stack analysis, use the **Stack analysis** toggle toolbar button in the **View** toolbar:



The available parameters are described in the table below:

Name	Description
N/A	Indicates the currently running thread. The currently running thread is indicated by a green arrow symbol.
Name	The thread name.
Priority	The thread priority.
State	The state of the current thread. The name of the object that currently suspends a thread is presented in parenthesis. For sleeping threads, the remaining sleep time (ticks) is presented.

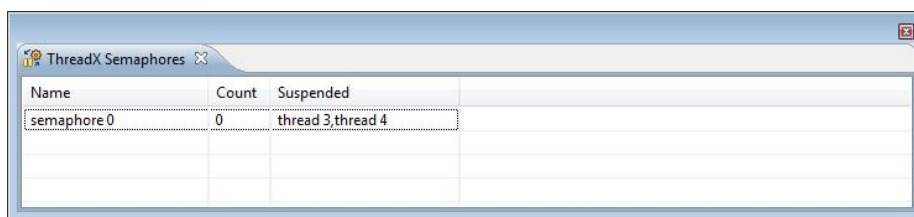
Name	Description
Run Count	The threads run counter.
Stack Start	The start address of the stack area.
Stack End	The end address of the stack area.
Stack Size	The size of the stack area (bytes).
Stack Ptr	The address of the thread stack pointer.
Stack Usage	The maximum stack usage (bytes).

Table 1 – ThreadX Thread Parameters

SEMAPHORES

The **ThreadX Semaphores** view displays detailed information regarding all available resource semaphores in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of semaphore parameter, and one row for each semaphore. If the value of any parameter for a particular semaphore has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Count	Suspended
semaphore 0	0	thread 3,thread 4

Figure 4 - ThreadX Semaphores View

The available parameters are described in the table below:

Column	Description
Name	The name of the semaphore.
Count	The current semaphore count.
Suspended	The threads currently suspended because of the

Column	Description
	semaphore state.

Table 2 – ThreadX Semaphore Parameters

MUTEXES

The **ThreadX Mutexes** view displays detailed information regarding all available mutexes in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of mutex parameter, and one row for each mutex. If the value of any parameter for a particular mutex has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Owner	Owner Count	Suspended
mutex 0	thread 6	2	thread 7

Figure 5 - ThreadX Mutexes View

The available parameters are described in the table below:

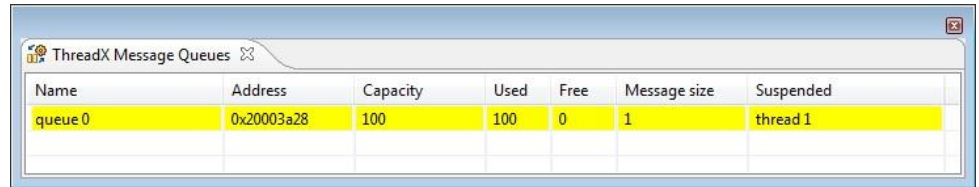
Column	Description
Name	The name of the mutex.
Owner	The thread that currently owns the mutex.
Owner Count	The mutex owner count (number of get operations performed by the owner thread).
Suspended	The threads currently suspended because of the mutex state.

Table 3 – ThreadX Mutex Parameters

MESSAGE QUEUES

The **ThreadX Message Queues** view displays detailed information regarding all available message queues in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of message queue parameter, and one row for each message queue. If the value of any parameter for a particular message queue has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Address	Capacity	Used	Free	Message size	Suspended
queue 0	0x20003a28	100	100	0	1	thread 1

Figure 6 - ThreadX Message Queues View

The available parameters are described in the table below:

Column	Description
Name	The name of the message queue.
Address	The address of the message queue.
Capacity	The maximum number of entries allowed in the queue.
Used	The current number of used entries in the queue.
Free	The current number of free entries in the queue.
Message size	The size (in 32-bit words) of each message entry.
Suspended	The threads currently suspended because of the message queue state.

Table 4 – ThreadX Message Queue Parameters

EVENT FLAGS

The **ThreadX Event Flags** view displays detailed information regarding all available event flag groups in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of parameter, and one row for each event flag group. If the value of any parameter for a particular event flag group has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Flags	Suspended
event flags 0	0x0	thread 5

Figure 7 - ThreadX Event Flags View

The available parameters are described in the table below:

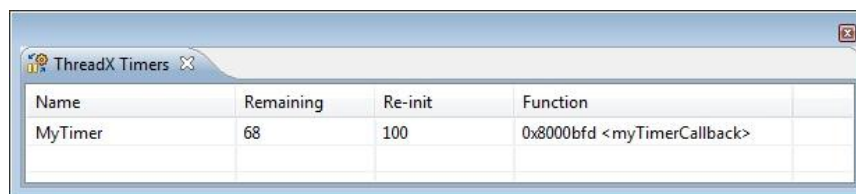
Column	Description
Name	The name of the event flag group.
Flags	The current value of the event flag group.
Suspended	The threads currently suspended because of the state of the event flag group.

Table 5 – ThreadX Event Flag Parameters

TIMERS

The **ThreadX Timers** view displays detailed information regarding all available software timers in the target system. The timers view is updated automatically each time the target execution is suspended.

There is one column for each type of timer parameter, and one row for each timer. If the value of any parameter for a particular timer has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Remaining	Re-init	Function
MyTimer	68	100	0x8000bfd <myTimerCallback>

Figure 8 - ThreadX Timers View

The available parameters are described in the table below:

Name	Description
Name	The name of the software timer.
Remaining	The remaining number of ticks before the timer expires.

Name	Description
Re-init	The timer re-initialization value (ticks) after expiration. Contains value 0 for One-Shot timers.
Functions	The address and name of the function that will be called when the timer expires.

Table 6 – ThreadX Timer Parameters

MEMORY BLOCK POOLS

The **ThreadX Memory Block Pools** view displays detailed information regarding all available memory block pools in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of parameter, and one row for each memory block pool. If the value of any parameter for a particular memory block pool has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Address	Used	Free	Total	Block size	Pool size	Suspended
block pool 0	0x20002f70	0	12	12	4	100	

Figure 9 - ThreadX Memory Block Pools View

The available parameters are described in the table below:

Column	Description
Name	The name of the block pool.
Address	The block pool starting address.
Used	The current number of allocated blocks.
Free	The current number of free blocks.
Size	The total number of blocks available.
Block size	The size (bytes) of each block.

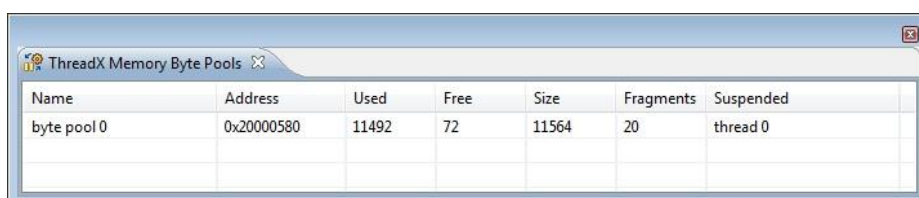
Column	Description
Pool size	The total pool size (bytes).
Suspended	The threads currently suspended because of the state of the memory block pool.

Table 7 – ThreadX Memory Block Pool Parameters

MEMORY BYTE POOLS

The **ThreadX Memory Byte Pools** view displays detailed information regarding all available memory byte pools in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of parameter, and one row for each memory byte pool. If the value of any parameter for a particular memory byte pool has changed since the last time the debugger was suspended, the corresponding row will be highlighted in yellow.



Name	Address	Used	Free	Size	Fragments	Suspended
byte pool 0	0x20000580	11492	72	11564	20	thread 0

Figure 10 - ThreadX Memory Byte Pools View

The available parameters are described in the table below:

Column	Description
Name	The name of the byte pool.
Address	The byte pool starting address.
Used	The current number of allocated bytes.
Free	The current number of free bytes.
Size	The total number of bytes available.
Fragments	The number of fragments.
Suspended	The threads currently suspended because of the state of the memory byte pool.

Table 8 – ThreadX Memory Byte Pool Parameters