THREADX

Training Information

2017

expresslogic

Training Overview

Express Logic offers a comprehensive training program at its state-of-the-art facility in San Diego, California. The training course features hands-on embedded systems development using ThreadX, the high-performance RTOS for deeply embedded applications.

Our electronic classroom contains eight modern workstations and multimedia projection capabilities. Each attendee has individual access to a workstation that is loaded with ThreadX and other software for embedded systems applications, so the class size is limited to eight.

The training course is intensive; it combines a rich mixture of RTOS concepts and hands-on embedded systems projects that reinforce these concepts. The projects are presented in a spiral order of depth as increasingly complex embedded systems concepts are investigated and explored.

Each attendee is provided with a detailed training workbook, documentation for the lab projects, a copy of the book titled *Real-Time Embedded Multithreading*, a copy of the ThreadX User Guide, a copy of the ThreadX Quick Reference Guide, and a Win32 ThreadX demonstration system.

Topics Covered

Day 1		Day 2	
Morning	RTOS and MultiTasking Fundamentals 2 Lab Projects Thread Design Mutual Exclusion with Mutexes	Morning	Inter-Thread Communication with Message Queues 2 Lab Projects Priority Inversion Preemption Threshold
Afternoon	Memory Management Using Timing Facilities 2 Lab Projects Using Counting Semaphores for Event Notification Synchronizing Threads with Event Flags Groups	Afternoon	Interrupts and I/O Designing a Multi-Threaded System 2 Lab Projects Tips, Hints, and Traps

Topics Day 1

RTOS and Multithreading Fundamentals
- ThreadX – Origin of Name
- Express Logic Products
- The First Embedded System
- Determinism
- Real-Time Kernel
- Thread Services
- Priorities
- Preemptive, Priority-Based Scheduling
- Real-Time Kernel Scheduling
- Potential Problems Caused by Preemption

Thread Design
- ThreadX Components
- User Guide Format
- Thread—Major Components
- Thread Stack
- Thread Control Block
- Thread Information Services
- Thread Management
- Thread Creation Overview
- Ready Thread List*
- Suspended Thread List
- Thread State Transition
- Quiz—Threads

Mutual Exclusion with Mutexes
- Mutex—Major Components
- Mutex Example
- Mutex Services
- Priority Inheritance
- Prioritize Example Suspended Thread List
- Mutex Management
- Example of Deadly Embrace

Memory Management
- Byte Pool – Major Components
- Memory Byte Pool Services
- Memory Byte Pool Management
- Multiple Pools
- Organization of Memory Byte Pool
- Quiz—Memory Byte Pools

Internal System Clock
Timer Ticks

ThreadX Project 1
Thread Creation and Coordination using a Mutex

Timing Facilities
- Timer—Major Components
- Application Timer Services
- Timer Management
- Quiz—Applications Timers

ThreadX Project 2
Analyzing the Behavior of Threads using Timing Facilities

Types of Program Execution
Initialization Process

Using Counting Semaphores for Event Notification
- Semaphore – Major Components
- Counting Semaphore Example
- Semaphore Services
- Semaphore Management
- Quiz—Semaphores

ThreadX Project 3
Using Wait Abort to Break Thread Suspension

Mutex v Counting Semaphore
Counting Semaphore with Ceiling

ThreadX Project 4
Using a Counting Semaphore for Event Notification

Synchronizing Threads with Event Flags Groups
- Event Flags—Major Components
- Example of an Event Flags Group
- Example of Set Operation
- Example of Get Operation
- Quiz—Event Flags Groups

ThreadX Project 5
Using an Event Flags Group to Synchronize Threads

Topics Day 2

Inter-Thread Communication with Message Queues

- Queue—Major Components
- Example of a Message Queue
- Message Queue Services
- Queue Management
- Queuing Model Classification
- Queuing Example
- Quiz—Queues

Memory Block Pool

- Major Components
- Memory Block Pool Services
- Memory Block Pool Management
- Calculate # of Memory Blocks
- Quiz—Memory Block Pools

ThreadX Project 7
Multiple Object Suspension System Using Event-Chaining

Context Switching
Typical Context Switch Actions

Time-Slicing
Round-Robin Scheduling
Round Robin Definition
Round-Robin Origin

Interrupts
ISR Template
Thread Interrupt Management
Where is Context Saved?

Thread Starvation
Priority Inversion
Example of Priority Inversion
Preventing Nondeterministic Priority Inversion Problems
Preemption-Threshold
Example of Preemption-Threshold
Mars Pathfinder Example

Event-Chaining

- Objectives
- Notification capabilities
- Registering an application function
- Example of Event-Chaining

ThreadX Project 8
Using Preemption-Threshold and Priority Inheritance to Eliminate Priority Inversion

ThreadX Demo System

Tips, Hints, and Traps

Summary and Wrap-Up

Training Benefits

Although ThreadX is an easy-to-use RTOS, this training course provides an ideal way to accelerate the learning process and to obtain valuable experience in applying ThreadX to the design and implementation of your embedded application.

As a result of taking this training course, attendees will acquire an in-depth knowledge of ThreadX, and will attain a focused approach to embedded systems development. This will improve productivity and significantly reduce the time-to-market for your development project.

The hands-on component of the training course is designed to reinforce and explore ThreadX and RTOS concepts. This approach encourages attendees to become active participants by engaging them in the learning process. It also enhances retention and understanding.

This training course will provide attendees the ability to get up to speed quickly and significantly enhance the prospects of success for your project. Training will also help you to optimize your investment in the premier RTOS for deeply embedded systems.

On-Site Course

We can offer a dedicated ThreadX training course at your site, or at any location of your choosing. There are several advantages of this type of course as follows:

- Less Time Away From Work
- No Travel Costs For Employees
- Class Size Not Limited to Eight
- Course Customized For Your Development Project

Contact us for more information.

Frequently Asked Questions

I thought ThreadX was easy to use. Why do I need training?

ThreadX is widely regarded as a fast, mature, and stable RTOS. We have carefully crafted ThreadX to provide the engineer with a powerful, elegant, and simple RTOS for project development. Although using ThreadX is intuitive and straightforward, our training course can help you achieve the most from your investment. We have used our extensive experience in embedded systems to design a training program tailored to your needs.

Training is too expensive and I can’t afford to take time off. Why can’t I learn the same concepts on my own?

Although it is possible that you can learn the same material on your own, it will probably take you much longer and you may miss many important features and techniques. Training will reduce the time required for you to optimize the features of ThreadX, thus reducing the cost of your project. We have committed significant resources in preparing a first-rate training experience for you. Thus, the training course will provide you with focused development insight in a short period of time. If you cannot attend the training course, consider hosting an on-site course.

I’m convinced that training is essential—how do I convince my boss?

Tell him/her that time is money! As a result of this training course, you will get up to speed quickly, significantly enhance the prospects of success for your project, and substantially reduce the time-to-market for your project. Training will also help you to optimize your investment in the premier RTOS for deeply embedded systems.

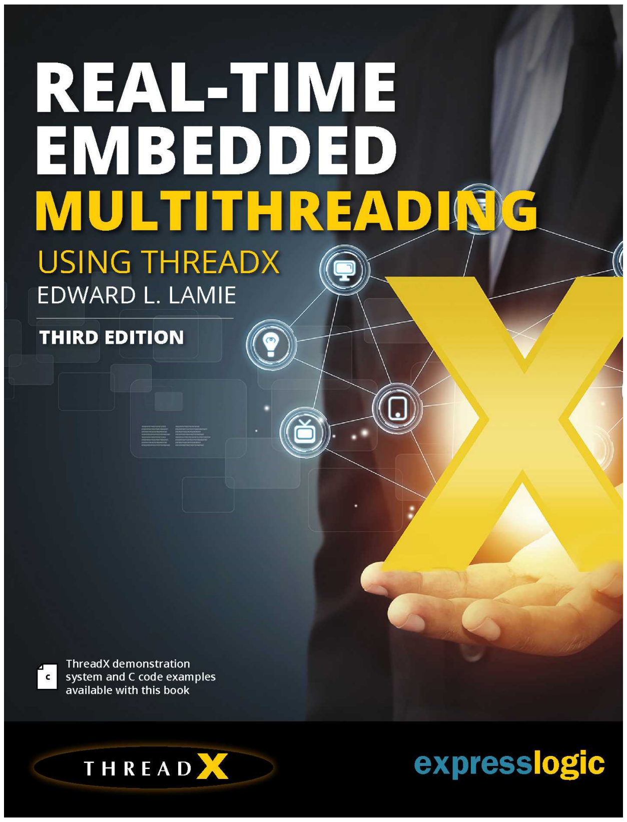
Are the training sessions strictly lecture-based, or is there a hands-on component?

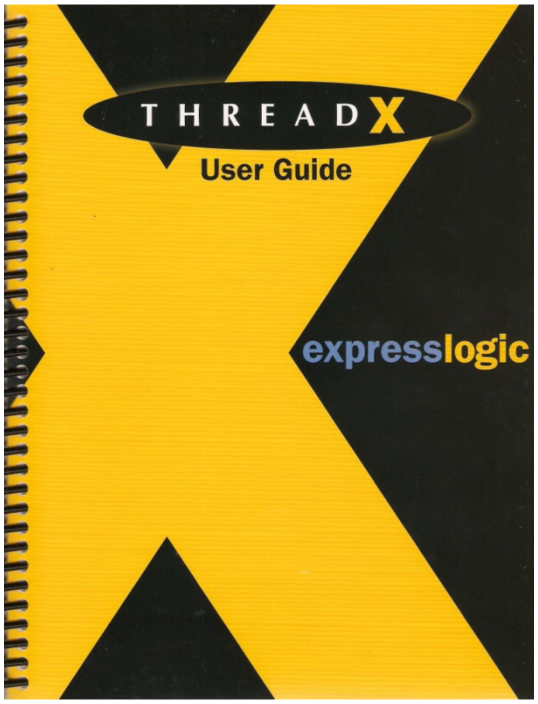
The training sessions consist of a pedagogically sound combination of lectures and hands-on embedded systems lab projects. The lab projects complement and reinforce the ThreadX and RTOS concepts discussed in the lecture. A multimedia approach is used, and the material is presented in an engaging, but fast-moving format.

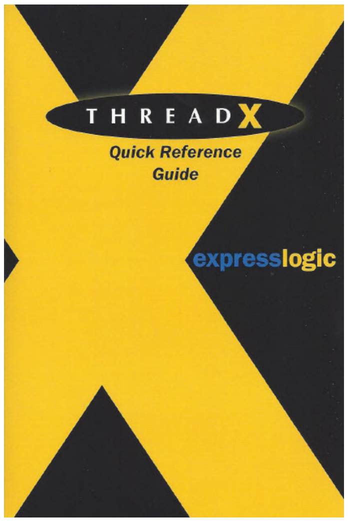
What background should I have in order to take the ThreadX training course?

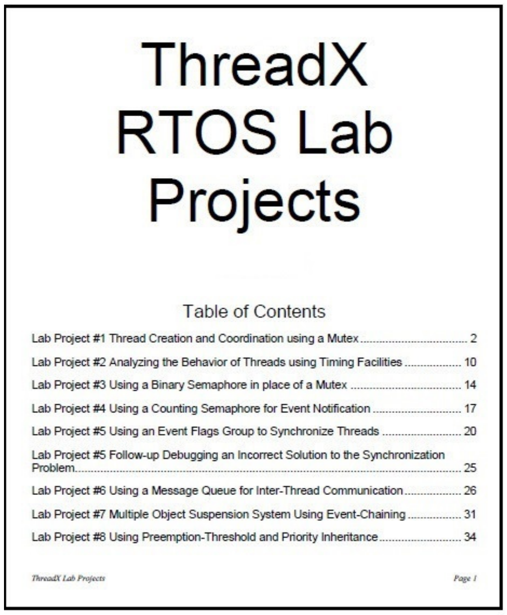
You should have several years of C/C++ programming experience and some exposure to embedded systems development.

Sample Training Materials









ThreadX RTOS Lab Projects

Table of Contents

ThreadX Lab Projects *Page 1*

For More Information

Contact Ed Lamie at
ELamie@ExpressLogic.com