



RTOS Fundamentals

Course Overview

Embedded systems are becoming complex and even resource constrained devices are requiring a Real-Time Operating System (RTOS). In this workshop, attendees will be walked through RTOS fundamentals starting with simple bare-metal scheduling techniques through the intricacy required to design a RTOS based application. Attendees will examine practical examples and techniques that will decrease their learning curve and help them avoid the common pitfalls many developers fall into when starting to use an RTOS. This course will also leverage Percepio Tracealyzer to further enhance attendees RTOS application understanding.

Who Should Attend?

Attendees are engineers who are interested in or will be designing applications using a Real-time Operating System. The course covers the fundamentals that are rarely discussed in courses or online literature, making the course immediately applicable to the real world. The course is appropriate for engineers with little experience in the area or experienced engineers looking for new techniques and skills. Attendees should understand the fundamentals of the C programming language. Advanced concepts are reviewed as they apply.

Key Take Aways

- Hands-on experience developing RTOS based applications
- How to develop your own real-time scheduler
- The steps necessary to decompose an application into tasks
- Analysis techniques for setting task priorities
- Developing a software architecture
- How to synchronize tasks with semaphores, mutexes and more
- Memory management strategies for real-time systems
- Example code and test hardware platform
- Key concepts related to robustness and efficiency
- Troubleshooting and debugging techniques
- Best practices for RTOS based applications

Course Format

This course is offered online, as a self-paced download. Other formats, such as onsite classes or live online sessions, may be available from time to time; please contact course lecturer Jacob Beningo directly for information about these alternatives.

Cost: USD 995 / EUR 895, including a 90-day education license for Percepio Tracealyzer.

Sessions Overview

Session 1 – Real-time Embedded Systems Concepts

- Hard versus soft real-time
- Bare metal scheduling techniques
- Designing a cooperative scheduler
- Best Practices

Session 2 – Real-time Operating Systems

- RTOS Characteristics
- Bare metal versus RTOS
- Real-time Kernels Overview
- Preemptive Scheduling
- RTOS terminology
- Selecting the right RTOS
- Setting up an RTOS
- Tools Overview
- Best Practices

Session 3 – Managing Tasks and Threads Part 1

- Introduction to tasks
- Task versus Thread
- Handling Critical Sections
- Reentrant Functions
- Task Control Blocks
- Preemption and context switches
- Creating tasks
- Identifying and partitioning tasks
- RTOS Application software architecture

Session 4 – Managing Tasks and Threads Part 2

- Setting task priorities using rate monotonic analysis (RMA)
- RTOS based Round Robin scheduling
- Time slicing
- Stack sizing and issues
 - Stack overflow detection
 - Worst case stack analysis techniques
- Best Practices

Session 5 – Task Synchronization Fundamentals Part 1

- Task Synchronization and Coordination
- Semaphores
- Binary versus counting
- Mutexes
- Synchronizing from ISRs
- Semaphores versus mutexes
- Best Practices

Session 6 – Task Synchronization Fundamentals Part 2

- Message Queues
- Event Flags
- Selecting the right synchronization
- Best Practices

Session 7 – Managing Memory

- Singly linked lists
- Doubly linked lists
- Dynamic versus static memory
- Understanding the heap and Malloc
- Memory byte pools
- Memory block pools
- Memory Management Strategies
- Best Practices

Note: Lectures 1 through 6 have associated labs.

Frequently Asked Questions (FAQ)

What hardware and toolchain does the course use?

FreeRTOS is supported by almost every silicon vendor. The lecture material in this course should be applicable to most supported hardware but the labs are designed for the STM32L475 IoT Discovery Node and its associated configuration utility STM32CubeMX. You can order the board and software from your favorite electronics distributor.

How do I access the course materials?

After ordering at the Percepio [web store](#), you will receive download instructions to access the materials.

How long do I have access to the course materials for?

Attendees have access to the materials indefinitely. Attendees not only get to keep the materials and recordings but may also attend any online live runs of the class at any point in the future and will also get access to any updates to the course.

Lecturer Background



Jacob Beningo is a Certified Software Development Professional (CSDP), chair of the IEEE South Eastern Michigan Consultants Affinity Group, an independent consultant and lecturer who specializes in the design of embedded software for resource constrained and low energy mobile devices. He has successfully completed projects across a number of industries including automotive, defense, medical and space. He enjoys developing and teaching real-time and reusable software development techniques using the latest methods and tools. He blogs for EDN.com about embedded system design

techniques and challenges. Jacob holds bachelor's degrees in Electrical Engineering, Physics and Mathematics from Central Michigan University and a Master's degree in Space Systems Engineering from the University of Michigan.

Additional resources, templates and Jacob's monthly embedded software newsletter can be found at www.beningo.com. Check out his other workshops at <http://www.beningo.com/services/workshops/>

