



Advanced RTOS Techniques

Course Overview

Understanding basic RTOS principles is easy but applying those techniques and reaching an advanced understanding of the concepts is difficult. This workshop propels developers into advanced RTOS design techniques that are rarely covered in traditional RTOS courses but are critical to the practicing engineer. In this course we will examine practical examples and techniques that will accelerate development activities and improve the robustness of RTOS-based designs. We will be using Percepio Tracealyzer to not only visualize our applications but also spot common RTOS issues such as priority inversions and deadlock.

Who Should Attend?

Attendees are engineers who are interested in or will be designing applications using a Real-time Operating System. The course covers advanced techniques 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 a basic understanding 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 identify and resolve common RTOS application issues
- The techniques available to developers designing applications using an RTOS
- Design patterns and when to use them
- How to use the microcontroller hardware to debug in real-time
- Advanced debugging techniques such as Deep Insight Analysis
- How to optimize an RTOS application
- Techniques for securing an RTOS based application and kernel
- Example code and test hardware platform
- 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 – Synchronization Patterns

- Unilateral rendezvous
- Credit Tracking
- Bilateral Rendezvous
- Client-Server
- Synchronizing multiple tasks
- Selecting the right synchronization method
- Best Practices

Session 2 – RTOS Pitfalls and Issues

- Complexity
- Issues – Task Jitter
- Issues – Thread Starvation
- Issues – Priority Inversion
- Issues – Deadlock
- Solutions
- Best Practices

Session 3 – Debugging RTOS Applications

- Printf
- ITM
- ASSERT
- SWD
- OS-Aware Debugging
- Real-time Tracing
- Debugging tools
- Deep Insight Analysis
- Error Trapping
- Best Practices

Session 4 – RTOS Application Design Patterns

- Designing an IoT Sensor Node Overview
- UART command processing
- Sensor Sampling Strategies
- Command Processing
- User Interactions
- TCP/IP Integration
- System Testing
- Best Practices

Session 5 – Optimizing RTOS Applications

- RTOS Configuration
- Memory
- Performance
- Energy
- Trade-offs
- Best Practices

Session 6 – Securing an RTOS Application

- Using the MPU with Tasks
- Defining trusted and untrusted code
- Using Arm Trustzone with an RTOS
- Thinking like a hacker
- Best Practices

Session 7 – RTOS Application Best Practices

- 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/>

