# oRT3 - Real Time Programming with FreeRTOS

*Comprehensive FreeRTOS training: from Theory to Practice*

## Objectives

- Get an overview on Cortex-M4 architecture
- Discover the concepts of real time multitasking
- Understand Real Time constraints
  - Determinism
  - Preemption
  - Interrupts
- Understand the FreeRTOS architecture
- Discover the various FreeRTOS services and APIs
- Learn how to develop FreeRTOS applications
- Learn how to debug FreeRTOS applications

## Prerequisites

- Familiarity with embedded C concepts and programming
- Basic knowledge of embedded processors

## Course environment

- Theoretical course
  - PDF course material (in English)
  - Course dispensed using the Teams video-conferencing system
  - The trainer to answer trainees' questions during the training and provide technical and pedagogical assistance through the Teams video-conferencing system
- Practical activities
  - Practical activities represent from 40% to 50% of course duration
  - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
  - Example code, labs and solutions
- Virtual Machine with System Workbench for STM32 with GNU ARM Eclipse QEMU
- Emulated board STM32F4-Discovery

## Course duration

- Total: 20 hours
- 4 sessions, 5 hours each (excluding break time)
- From 40% to 50% of training time is devoted to practical activities
- Some Labs may be completed between sessions and are checked by the trainer on the next session

## Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

# Course Outline

## First Session

### Cortex-M Overview

- ARMv7-M Architecture
- Cortex-M4 Architecture
- Registers and Execution States
- Privileges, Mode and Stacks
- Reset Behavior
- Exception and Interrupts
- The System Timer
- Memory Model
- Power Management
- STM32F407x Implementation Example

**Exercise:** Create a new project
**Exercise:** Interrupt Management

### Real-Time Concepts

- Embedded system architectures
- Tasks and process
- Real-Time

**Exercise:** Context Switch

## Second Session

### Introduction to FreeRTOS

- The FreeRTOS Family
- FreeRTOS+Ecosystem
- Why use FreeRTOS
- FreeRTOS Code Structure

### Scheduling

- Scheduler
- Schedulability

### Task Management

- Creating Tasks
- Task Priorities
- Task States
- The idle task
- Delays
- Changing Task Priority
- Deleting Tasks
- Suspending Tasks
- Kernel Structures
- Thread Local Storage
- Kernel Interrupts on Cortex-M4
- Scheduling Traces

- Visual trace diagnostics using Tracealyzer

**Exercise:** Task Management
**Exercise:** Periodic Tasks
**Exercise:** Task Statistics

# Third Session

## Memory Management in FreeRTOS

- FreeRTOS Memory Managers
- Out of Memory management
- Stack overflow detection

**Exercise:** Context Switch Measurement

## Resource Management

- Mutual Exclusion
- Critical Sections
- Mutexes
- Gatekeeper Tasks
- Lock-Free Data Structures

**Exercise:** Resource Management

## Synchronization Primitives

- Queues
- Queues Sets
- Synchronization
- Events and Event Groups
- The Readers/writer problem
- Using Other Primitives within an ISR

**Exercise:** Queue Management
**Exercise:** Readers Writer Problem

# Fourth Session

## Interrupt Management

- Tasks and Interrupts
- FreeRTOS Binary and Counting Semaphores
- Task Notifications
- Stream Buffers
- Message Buffers
- Interrupt Nesting
- Low Power Support

**Exercise:** Interrupt Management
**Exercise:** Tickless Mode

## Software Timer

- Software Timers
- Deferred Interrupt Handling

**Exercise:** Implement Soft Timers
**Exercise:** Software Timer Management

## FreeRTOS-MPU

- The Cortex-M MPU
- The FreeRTOS-MPU Port
- Defining MPU Regions
- Creating User and System Tasks
- Practical Usage Tips

# Appendix

## Data Structures

- FIFO
- Linked list

## Memory Management and Real-Time

- Memory Management
- Memory Errors

## CMSIS-RTOS

- Overview
- Kernel Information and Control
- Threads Management
- Generic Wait Functions
- Communication and Resource Sharing
  - Semaphores
  - Mutex
  - Message Queue
  - Signal Events
  - Event Flags
  - Memory Pool
  - Mail Queue
- Timer Management
- Interrupt Service Routines