



# OS2 - MQX Programming on Kinetis Microcontroller

*Programming applications using the MQX operating system*

## Objectives

- Becoming familiar with the NXP IDE, Kinetis SDK or CodeWarrior
- Get an overview of Kinetis and Cortex-M4 core architecture
- Revise the concepts of real time multitasking
- Understand the MQX architecture
- Discover the various MQX services and APIs
- Learn how to develop MQX applications
- Learn how to debug MQX applications
- Learn how to use MQX Library (USB, TCP/IP, File System, Embedded GUI)

## Course environment

- Convenient course material with space for taking notes
- Documentation, labs and solutions
- A PC under Windows 7 for two trainees
- A NXP Kinetis K70 or K60 (Cortex/M4) with SDK or CodeWarrior IDE

## Prerequisites

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

## Environnement du cours

- Cours théorique
  - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
  - Cours dispensé via le système de visioconférence Teams (si à distance)
  - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
  - Les activités pratiques représentent de 40% à 50% de la durée du cours
  - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
  - Exemples de code, exercices et solutions
  - Pour les formations à distance:
    - ▶ Un PC Linux en ligne par stagiaire pour les activités pratiques, avec tous les logiciels nécessaires préinstallés.
    - ▶ Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
    - ▶ Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
  - Pour les formations en présentiel:
    - ▶ Un PC (Linux ou Windows) pour les activités pratiques avec, si approprié, une carte cible embarquée.
    - ▶ Un PC par binôme de stagiaires s'il y a plus de 6 stagiaires.
  - Pour les formations sur site:
    - ▶ Un manuel d'installation est fourni pour permettre de préinstaller les logiciels nécessaires.
    - ▶ Le formateur vient avec les cartes cible nécessaires (et les ramène à la fin de la formation).
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session (demi-journée en présentiel) une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

## Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

# Plan du cours

## First day

### Cortex-M4 Overview

- Registers, Mode and Stacks
- Exception Management

### MQX at a Glance

- MQX overview
- Organization of MQX
- MQX directory structure
  - RTOS directory
  - PSP, BSP, I/O and others source subdirectories
- Initializing and starting MQX
- Developing with NXP CodeWarrior Development Studio
  - Build projects
  - PSP build-project
  - BSP build-project
  - Post-build processing
  - Processor expert

**Exercise :** Creating a simple MQX project using SDK or NXP CodeWarrior Development Studio and the Processor Expert Tool

## Second day

### Managing and Scheduling tasks with MQX

- Managing Tasks
  - Creating tasks
  - Managing task errors
  - Terminating tasks
- Scheduling
  - FIFO scheduling
  - Round Robin scheduling
- Context Switch

**Exercise :** Use the MQX API to manage tasks

### Memory Management

- Memory with variable-size blocks
- Lightweight memory with variable-size blocks
- Memory with fixed-size blocks
  - Creating partitions
  - Allocating and freeing partition blocks

**Exercise :** Managing memory

## Synchronizing Tasks

- Synchronizing tasks through MQX RTOS
  - Events
  - Lightweight events
  - Lightweight semaphores
  - Semaphores
  - Lightweight Messages queue
- Mutual Exclusion through MQX RTOS
  - Create critical sections
  - Mutexes
  - Avoiding Priority inversion

**Exercise :** Synchronizing tasks using MQX semaphores API

**Exercise :** Create mutual exclusion using MQX semaphores API

## Third day

### Interrupt Management

- Handling Interrupt with Cortex-M4 core
- Handling Interrupt and Exceptions in MQX RTOS
  - Initializing interrupt handling
  - Restrictions on ISRs
  - Handling exceptions
  - Handling ISR exceptions
  - Handling task exceptions

**Exercise :** Use the MQX API to handle interrupt

### Timing with MQX RTOS

- Time components
- Timers
- Lightweight timers
- Watchdogs
- Hardware Timer on Cortex-M4

**Exercise :** Using software timers

### Debugging the application

- Instrumentation
  - Logs
  - Lightweight logs
  - Kernel logs
  - Stack usage utilities

**Exercise :** Debug an application with the log component

### IO Drivers at a glance

- Drivers architecture
- Installing Drivers
- Using Drivers

**Exercise :** Hands-on: Working with the ADC Driver

### Getting Started with the MQX Libraries

- How to start with the different libraries (MFS, Shell, RTCS, USB) using the providing examples

- RTCS and Shell Libraries at a glance

**Exercise :** Running a simple TCP echo Server / HTTP server application