



oRT3 - Programmation temps réel avec FreeRTOS

Usage complet de FreeRTOS: De la théorie à la pratique

Objectifs

- Obtenir une vue d'ensemble de l'architecture Cortex-M4
- Découvrir les concepts du multitâche temps réel
- Comprendre les contraintes du temps réel
 - Déterminisme
 - Préemption
 - Interruptions
- Comprendre l'architecture de FreeRTOS
- Découvrir les différents services et APIs de FreeRTOS
- Apprendre à développer des applications FreeRTOS
- Apprendre à déboguer les applications FreeRTOS

Pré-requis

- Familiarité avec les concepts et la programmation du C embarqué
- Connaissance de base des processeurs embarqués

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais).
 - Cours dispensé via le système de visioconférence Teams.
 - 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.
 - Un PC Linux en ligne par stagiaire pour les activités pratiques.
 - 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.
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session 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

Déroulé pédagogique

- Durée totale : 20 heures en 4 sessions de 5 heures chacune (hors temps de pause).
- De 40% à 50% du temps de formation est consacré aux activités pratiques qui servent à valider la bonne compréhension des concepts enseignés.
- Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.

- En début de chaque session une interaction avec les stagiaires permet d'adapter si nécessaire le contenu du cours à leurs besoins.

Modalités d'évaluation

- Les prérequis indiqués ci-dessus sont évalués avant la formation par l'encadrement technique du stagiaire dans son entreprise, ou par le stagiaire lui-même dans le cas exceptionnel d'un stagiaire individuel.
- Les progrès des stagiaires sont évalués de deux façons différentes, suivant le cours:
 - Pour les cours se prêtant à des exercices pratiques, les résultats des exercices sont vérifiés par le formateur, qui aide si nécessaire les stagiaires à les réaliser en apportant des précisions supplémentaires.
 - Des quizz sont proposés en fin des sections ne comportant pas d'exercices pratiques pour vérifier que les stagiaires ont assimilé les points présentés
- En fin de formation, chaque stagiaire reçoit une attestation et un certificat attestant qu'il a suivi le cours avec succès.
 - En cas de problème dû à un manque de prérequis de la part du stagiaire, constaté lors de la formation, une formation différente ou complémentaire lui est proposée, en général pour conforter ses prérequis, en accord avec son responsable en entreprise le cas échéant.

Plan

Première 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

Deuxième 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

Troisième 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

Quatrième 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

Annexes

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

Renseignements pratiques

Renseignements : 24 heures