

# Prise en Main de FreeRTOS & File System FAT & Connectivité USB sur MCU ARM Cortex-M



2 jours sur FreeRTOS + ½ journée sur le FAT + ½ journée sur l'USB

## OBJECTIF

L'utilisation de microcontrôleurs 32 bit ne cesse de croître dans la réalisation d'applications électroniques embarquées. Lorsque des critères tels que la vitesse d'exécution, le besoin de connectivité, une taille mémoire limitée ou encore une consommation optimisée sont déterminants, l'utilisation de ces composants semble particulièrement adaptée. A cela s'ajoute une complexité logicielle croissante (multitâches, tâches prioritaires ou temps réel, utilisation de piles de communication) pouvant nécessiter la mise en place d'un noyau temps réel (RTOS) pour simplifier l'architecture logicielle. Tout cela implique de nombreux changements de méthodologie qu'il vaut mieux connaître avant de se lancer. Notre formation vise donc à accélérer la prise en main de de ces nouveaux concepts de programmation liés à l'utilisation d'un RTOS comme FreeRTOS et de finir par une mise en place concrète d'une gestionnaire de fichier FAT et d'un pile de protocole USB.

## VOUS APPRENDREZ COMMENT

- Prendre en main un Noyau Temps Réel (RTOS) avec comme exemple FreeRTOS
- Aborder concrètement une étude de cas d'une application sans OS à migrer vers une application avec RTOS
- Mettre en place une gestionnaire de fichier FAT et comment utiliser une communication USB HID ou encore comment gérer une clé USB
- Utiliser un environnement de développement et de débogue JTAG / SWD (une démonstration des possibilités débogue via la Trace ETM est possible sur demande)

## A QUI S'ADRESSE CE STAGE

Ce stage s'adresse aux ingénieurs et techniciens de développement qui souhaitent mettre en œuvre concrètement un RTOS et une stack TCP/IP. La pratique du langage C et de sa mise en œuvre dans des applications à base de microcontrôleur ARM Cortex-M est indispensable.

## EXERCICES PRATIQUES

Chaque participant sera doté pendant toute la durée du stage d'un PC muni d'un environnement de développement, d'une sonde de débogue USB - JTAG/SWD et d'une plateforme d'évaluation à base de composant à cœur ARM Cortex-M. Notre formation est essentiellement basée sur des exercices pratiques

## DOCUMENTS

L'ensemble des documents, comprenant les supports de cours, les notes d'application, les manuels d'utilisation, les articles techniques et les programmes étudiés pendant le stage vous sera remis au cours de la formation. Ce support vous apportera une aide précieuse pour exploiter avec succès FreeRTOS, un gestionnaire de fichier FAT et une stack USB dans vos applications futures.

## INSCRIPTIONS

Email : [info@cynetis-embedded.com](mailto:info@cynetis-embedded.com)

Téléphone : 01 85 08 70 69

Lieu de la formation :



Paris

ou sur site client

# Prise en Main de FreeRTOS & File System FAT & Connectivité USB sur MCU ARM Cortex-M



2 jours sur FreeRTOS + ½ journée sur le FAT + ½ journée sur l'USB

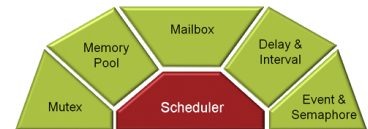
## CONTENU DU COURS : Prise en Main du Noyau Temps Réel FreeRTOS (2 jours)

### 1) Bases théoriques sur RTOS (durée : 2h)

- Notions de base sur les OS
- Le temps réel
- Les tâches et leurs états
- Gestion du temps dans l'OS
- Le Scheduler et les différents algorithmes
- La communication entre les tâches

### 2) Aperçu des APIs d'un RTOS (durée : 2h)

Exemple : FreeRTOS



### 3) Travaux pratiques sur les APIs (durée : 6h)

Les exercices sont réalisés sur des cartes d'évaluation à base de composant STM32 à cœur ARM Cortex-M4

- Mise en place de l'OS & création de tâches
  - Contrôle des tâches & Utilisation d'outils d'analyse
  - Comparaison des algorithmes Round Robin, Préemptif, Coopératif
- Les événements
  - Déclencher une tâche avec des événements
- Les Mutex
  - Leurs utilités, protection par Mutex
- Les sémaphores
  - Gestion d'une file d'attente par sémaphores
  - Synchronisation de plusieurs tâches
- Les boîtes aux lettres
  - Gestion d'une pile de messages d'erreurs
- Gestion mémoire et boîte aux lettres
  - Echange de données entre tâches asynchrones



### 4) Etude de cas RTOS (durée : 6h)

- Portage d'une application conçue initialement sans RTOS
- Discussion sur la mise en place des différentes APIs de l'OS

# Prise en Main de FreeRTOS & File System FAT & Connectivité USB sur MCU ARM Cortex-M



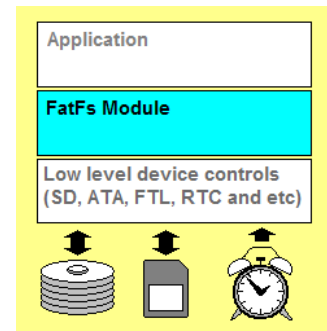
2 jours sur FreeRTOS + ½ journée sur le FAT + ½ journée sur l'USB

## CONTENU DU COURS : Prise en Main de FatFs (1/2 journée)

Les travaux pratiques sont réalisés autour d'un gestionnaire de fichiers embarqué open source (FatFs) sur carte à base de composant STM32 à cœur ARM Cortex-M4

### 1) Bases théoriques (durée : 1h)

- Aperçu sur le system FAT 12/16/32
- Les API de FatFs
- Aperçu sur les drivers et exemple sur un driver de carte SD
- Mise en place de FatFs et interaction avec les RTOS
- Configuration de FatFs



### 2) Travaux pratiques (durée : 3h)

- Exercice : Lecture, écriture sur une carte SD
- Mise en place d'un système de fichier FatFs
- Exercice : Utilisation du système de fichier FatFs (création, lecture/écriture de fichier)
- Utilisation du system de fichier avec un interpréteur de commandes

## CONTENU DU COURS : Prise en Main de l'USB (1/2 journée)

Les travaux pratiques sont réalisés autour de la pile de communication USB gratuite fournie par STMicroelectronics pour les composants STM32. La carte d'évaluation utilisée est à base de composant STM32 à cœur ARM Cortex-M4.

### 1) Bases théoriques (durée : 1h)

- Aperçu sur le system USB device/host
- Les descripteurs, les class, les protocoles.
- Enumération, système de transfert
- Report HID
- Les API des différentes class
- Mise en place et configuration d'un programme



### 2) Travaux pratiques (durée : 3h)

Selon le temps restant, il s'agira de choisir 1 ou 2 exercices dans la liste suivante :

- Exercice 1 : Etude d'un système descripteur HID
- Exercice 2 : Simulation d'un joystick sur PC (class HID device)
- Exercice 3 : Lecture sur SD carte à partir d'un PC (class MSC device)
- Exercice 4 : Lecture du contenu d'une clé USB (class MSC host)
- Exercice 5 : Etude d'un système de report HID avec un cas custom. Communication d'une carte avec un PC : commutation des LEDs sur la carte, relecture ADC de la carte