

# Mise en Œuvre du STM32 à cœur ARM Cortex-M & Prise en Main de FreeRTOS



(2 jours sur le STM32 + 2 jours sur FreeRTOS)

## OBJECTIF

L'utilisation de microcontrôleurs 32 bit à cœur ARM Cortex-M 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 composants comme le STM32 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 ces microcontrôleurs STM32 et des nouveaux concepts de programmation liés à l'utilisation d'un RTOS comme FreeRTOS.

## VOUS APPRENDREZ COMMENT

- Se décline la famille STMicroelectronics STM32
- Fonctionne le cœur ARM Cortex-M
- Utiliser la librairie logicielle CMSIS (ARM Cortex Microcontroller Software Interface Standard)
- Exploiter les périphériques proposés sur le STM32
- 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
- Utiliser un environnement de développement et de débogage JTAG / SWD (une démonstration des possibilités débogage 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 microcontrôleur à cœur ARM Cortex-M de la famille STM32 conjointement avec un RTOS. La pratique du langage C et de sa mise en œuvre dans des applications à base de microcontrôleur 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 débogage USB - JTAG/SWD et d'une plateforme d'évaluation à base de composant STM32. 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 cette famille de composants STM32 et FreeRTOS 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

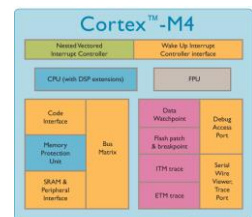
ou sur site client

(2 jours sur le STM32 + 2 jours sur FreeRTOS)

## CONTENU DU COURS : Mise en Œuvre du STM32 à cœur ARM Cortex-M (2 jours)

### 1) Présentation de la famille STM32 et du cœur ARM Cortex-M4 (durée : 4 heures)

- La famille STM32
- Schéma fonctionnel
- Format de la mémoire
- Les modes de Boot
- ST ART Accelerator
- DMA & Multi-Bus Matrix
- Modes Processeur & Privilèges
- Registres du cœur
- Les Exceptions et les Interruptions (NVIC, SCB)
- Le SysTick au cœur des OS
- Débogue (JTAG, SWD, Trace)
- Les bibliothèques CMSIS (ARM, DSP, SVD)
- Comparaison Cortex-M3 / Cortex-M4
- Revue des périphériques du STM32



### 2) Travaux pratiques avec la librairie CMSIS de ST sur carte d'évaluation à base de composant STM32F4 à cœur Cortex-M4, avec des outils de développement & débogue avec sonde JTAG/SWD (durée : 12h)

*Les participants pourront choisir leurs TP d'un commun accord. Quatre à six TP est une moyenne raisonnable. D'autres périphériques peuvent être abordés selon les besoins.*

- 1 - Programme de base
  - Mise en place projet : LED, LCD, System, Startup
  - Découverte de la librairie CMSIS de ST
- 2 - Systick
  - Mise en place et utilisation du Systick
- 3 - Exercice sur les horloges RCC et la gestion du NVIC
  - Mise en place HSE HSI et PLL
  - Mise en place Flag CSS pour déclencher le NMI
  - Mise en place IT RCC pour gestion HSE+PLL
- 4 - Exercice sur les GPIO et EXTI
  - Configuration des ports pour gérer les LEDs et le clavier
  - Déclencher une IT à partir du clavier
- 5 - Exercice sur la RTC
  - Mise en place d'un chronomètre
- 6 - Exercice sur l'ADC
  - Mesure d'une tension sur le potentiomètre
  - Mise en place du DMA
- 7 - Exercice sur le DAC
  - Génération d'un signal triangulaire
  - Génération d'un signal sinus avec utilisation du DMA
- 8 - Exercice sur les modes basse consommation (PWR)
  - Mise en mode Stop, Sleep ou Standby
- 9 - Exercice sur le Timer
  - Génération d'un signal avec Output Compare
  - Capture d'un signal / Génération d'un signal PWM



**Remarque :** Notre formation STM32 est reconnue officiellement par STMicroelectronics

La société Cynetis Embedded est enregistrée en tant que prestataire de formation sous le numéro **11 75 53750 75**.  
Cet enregistrement ne vaut pas agrément de l'Etat (en application de l'article L6352-12 du code du travail).

(2 jours sur le STM32 + 2 jours sur FreeRTOS)

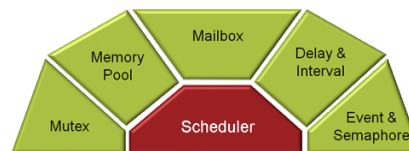
## CONTENU DU COURS : Prise en Main du Noyau Temps Réel FreeRTOS (2 jours)

### 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

- Mise en place de l'OS & Création de tâches
  - Contrôle des tâches & Aperçu des outils [Perceprio](#) pour FreeRTOS
  - Comparaison des algorithmes Roundrobin, 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