

(1 jour sur l'ARM Cortex-M + 2 jours sur FreeRTOS + 1 jour sur le TCP/IP)

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 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 ces microcontrôleurs ARM Cortex-M et des nouveaux concepts de programmation liés à l'utilisation d'un RTOS comme FreeRTOS et de finir par une mise en place concrète d'une pile de protocole TCP/IP

VOUS APPRENDEZ COMMENT

- Fonctionne le cœur ARM Cortex-M
- Utiliser la librairie logicielle CMSIS (ARM Cortex Microcontroller Software Interface Standard)
- 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 stack TCP/IP, utiliser les sockets UDP et TCP et configurer un serveur Web dynamique
- 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 microcontrôleur à cœur ARM Cortex-M conjointement avec 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 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 et une connectivité TCP/IP dans vos applications futures.

INSCRIPTIONS

Email : info@cynetis-embedded.com

Téléphone : 01 85 08 70 69

Lieu de la formation :



Paris

ou sur site client

(1 jour sur l'ARM Cortex-M + 2 jours sur FreeRTOS + 1 jour sur le TCP/IP)

CONTENU DU COURS : ARM CORTEX-M (1 JOUR) ET FREERTOS (2 JOURS)

1) Introduction sur le cœur ARM Cortex-M sur base d'exemple du STM32 (durée : 8h)

- Aperçu sur le cœur ARM Cortex-M4
- Modes Processeur & Privilèges
- Registre du cœur (NVIC, SysTick...)
- Mise en place des outils de développement et de débogue JTAG/SWD
- Découverte de la librairie CMSIS
- Travaux Pratiques (configuration d'I/O, mise en place du SysTick)

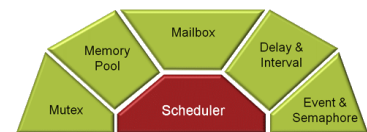


2) Bases théoriques sur RTOS (durée : 2h)

- 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

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

Exemple : FreeRTOS



4) 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 & Aperçu des outils [Perceprio](#) pour FreeRTOS
 - 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



5) Etude de cas RTOS (durée : 6h)

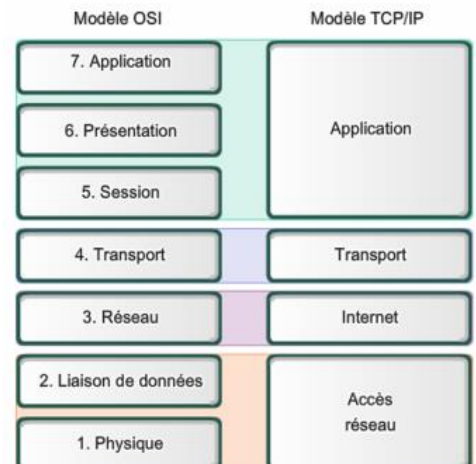
- Portage d'une application conçue initialement sans RTOS
- Discussion sur la mise en place des différentes APIs de l'OS

(1 jour sur l'ARM Cortex-M + 2 jours sur FreeRTOS + 1 jour sur le TCP/IP)

CONTENU DU COURS : CONNECTIVITE TCP/IP (1 JOUR)

1) Bases théoriques (durée : 2h)

- Présentation du modèle OSI
- Présentation des couches protocolaires TCP/IP
- Ethernet et la gestion de collisions
- Adresse MAC et adresse IP
- Notion de routage
- Protocole ARP
- Utilisation des sockets en mode connecté (TCP)
- Utilisation des sockets en mode non connecté (UDP)
- Notions de Ports et notions de Sockets
- Connexion Client – Serveur



2) Travaux pratiques (durée : 6h)

Les travaux pratiques sont réalisés autour d'une pile de communication TCP/IP embarquée open source (CycloneTCP) sur carte à base de composant STM32 à cœur ARM Cortex-M4

- Prise en main d'une stack TCP/IP avec FreeRTOS sur STM32
- Fonctions de base d'un contrôleur Ethernet
- Mise en place d'un DHCP
- Interaction de la stack TCP/IP avec le RTOS FreeRTOS
- Découverte des sockets UDP et TCP au travers d'une application de tchat
- Mise en place d'un serveur Web avec contenu dynamique (CGI et Ajax)
- Mise en place de sockets et établissement d'une communication avec une application PC

