

I.MX7 FreeRTOS - YOCTO SDK LINUX KERNEL RPMsg DRIVER

DURÉE : 3 JOURS (21 HEURES)

SANCTION DE LA FORMATION : ATTESTATION DE FIN DE STAGE ET DE PRÉSENCE, ÉVALUATION DES ACQUIS

NATURE DE LA FORMATION : ADAPTATION ET DÉVELOPPEMENT DE COMPÉTENCES

OBJECTIFS

Cette formation est réalisée autour du BSP FSL Community / BSP LinRT Helium Yocto 2.4 Morty, FSLC Linux kernel 4.9.67 sur le Kit SBC Zeta phyCORE-i.MX 7d

- Développer et mettre au point des applications et de drivers Linux avec le SDK Yocto i.MX 7
- Configurer les ressources d'E/S et des périphériques sous Linux (Cortex A7) et sous FreeRTOS (Cortex M4) de la plateforme i.MX 7
- Configurer le mapping memory Cortex M4/Cortex A7 sur CPU i.MX 7
- Cross compiler, exécuter et déboguer une application Linux sur Cortex A7 avec l'IDE Eclipse et le SDK Yocto
- Cross compiler, exécuter et déboguer une application FreeRTOS sur Cortex M4 avec l'IDE Kinetis
- Concevoir des drivers RPMsg inter-processeurs Cortex M4/Cortex A7 sur CPU i.MX 7

PUBLIC ET PRE-REQUIS

Notre formation Yocto SDK Linux Kernel i.MX7 API / FreeRTOS API pour processeurs NXP iMX 7 est adaptée aux développeurs en informatique et techniciens dans le domaine de Linux embarqué confrontés aux problèmes de portage d'applications mettant en oeuvre des mécanismes de communication inter-processeurs entre les cœurs Cortex A7 sous Linux et le Cortex M4 sous FreeRTOS. Cette formation s'adresse aux développeurs ayant déjà une bonne maîtrise des contenus abordés dans la formation Yocto BSP pour processeurs NXP iMX 7.

Une connaissance des commandes UNIX usuelles et des notions de programmation en langage C est nécessaire.

1ère Journée

Yocto LinRT BSP pour i.MX 7

- Fichier local.conf et réglages globaux
- Notion de MACHINE et de PROVIDER
- Spécificités et optimisation du noyau linux-fslc et linux-imx sous Yocto pour i.MX 7
- Customisation du kernel devicetree spécifiques i.MX 7

Travaux Pratiques

- Configuration du noyau linux du kit Phytex Zeta phyCORE-i.MX 7d
- Modification du devicetree et mise en oeuvre d'un patch noyau avec l'utilitaire GIT sous Yocto de configuration des ressources et périphériques Cortex M4/Cortex A7
- Génération d'une image Linux LinRT BSP pour Kit Phytex Zeta iMX 7d avec support FreeRTOS en RAM TCM avec chargement automatique au lancement de la plate-forme Phytex Zeta phyCORE-i.MX 7d

Software Development Kit Yocto LinRT BSP Helium

- Pluggin Eclipse Yocto SDK LinRT
- Génération et déploiement du SDK LinRT BSP Helium complet pour i.MX 7

Travaux pratiques

- Cross compilation et débogage d'application Linux sur kit Phytex Zeta phyCORE-i.MX 7d

2ème Journée

FreeRTOS sous i.MX 7

- Présentation de FreeRTOS
- Zone mémoire d'exécution de FreeRTOS sous Cortex M4 i.MX 7
- LinRT BSP FreeRTOS Kinetis support pour i.MX 7

Travaux pratiques

- Cross-compilation de FreeRTOS sous Kinetis

FreeRTOS API

- GPIO Driver
- General Purpose Timer
- UART, I2C et eCSPI
- Local Memory Controller

Travaux pratiques

- Génération d'une image Linux LinRT BSP pour Kit Phytex Zeta iMX 7d avec support FreeRTOS «RAM DDR cachable» sur la SBC Phytex Zeta phyCORE-i.MX 7d
- Déploiement et débogage d'applications FreeRTOS sur Cortex M4 via JTAG et IDE Kinetis

3ème Journée

FreeRTOS API (suite)

- Messaging Unit
- Ressource Domain Controller
- Hardware Controller
- Hardware Semaphore (SEMA4)

Introduction au développement inter-processeur cortex M4 / cortex A7

- Rappels structure d'un module driver simple sous Linux
- Drivers RPMsg Linux
- API RPMsg FreeRTOS

Travaux pratiques

- Programmation de pilotes de périphériques simples RPMsg communicante avec une tâche FreeRTOS sur Kit Phytex Zeta phyCORE-i.MX 7d

Moyens pédagogiques

- Vidéo-projecteur. Tableau blanc ou paper-board.
- Un PC par binôme
- Une connexion internet

Supports :

- Kit de développement
- support de cours et travaux pratiques (version papier ou PDF)