

DURÉE : 5 JOURS (35 HEURES) / INTRA ET INTER-ENTREPRISE

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

NATURE DE LA FORMATION : ADAPTATION ET DÉVELOPPEMENT DE COMPÉTENCES

Objectifs

Cette formation est réalisée autour de FSL Community BSP / LinRT BSP Oxygen, Mercury et Cobalt sous Yocto 2.4 Rocko, FSLC Linux kernel 4.9.67 sur le Kit SBC Mira phyCORE i.MX6 Quad :

- Maîtriser l'installation de l'environnement Yocto Rocko 2.4
- Maîtriser les commandes usuelles Bitbake
- Gérer la configuration de build et de la Machine i.MX 6
- Ajouter et Mettre en oeuvre ses propres layers Yocto
- Configurer et adapter le noyau linux low-latency (PREEMPT-RT) à sa plate-forme i.MX 6
- Construire et customisez son image Linux avec Yocto Temps Réel
- Adapter les drivers du noyau Linux pour i.MX6
- Configurer et adapter le noyau linux temps réel (Xenomai) à sa plate-forme i.MX6
- Développer et mettre au point des applications Temps Réel sur processeur NXP i.MX6

Pré-requis

Notre formation Yocto BSP pour processeurs NXP iMX 6 est adaptée aux développeurs en informatique et techniciens dans le domaine de Linux embarqué confrontés aux problèmes de portage d'applications sur système i.MX 6 sous Yocto.

Une connaissance des commandes utilisateur usuelles UNIX et des notions de programmation en langage sont souhaitables.

Projet Yocto

- Présentation du projet Yocto
- OpenEmbedded Core et poky
- Notion de Layers et de Recettes
- Les utilitaires GIT et repo
- Rôle de l'outil de construction bitbake

Yocto FSL Community

- Les layers spécifiques FSL
- Les différentes distributions FSL et FSLC
- Les images applicatives proposées

Travaux pratiques

- Mise en œuvre d'une configuration de BSP avec choix des layers, configuration de la distribution et de l'image fslc-framebuffer à générer pour une machine spécifique : Kit SBC-Mira phyCORE i.MX 6 Quad

1.

Yocto FSL Community BSP

- Fichier local.conf et réglages globaux
- Les différentes cibles NXP iMX supportées
- Notion de MACHINE et de PROVIDER
- Bootloader u-boot et barebox
- Spécificités et optimisation du noyau linux-fslc et linux-imx sous Yocto
- Customisation du kernel devicetree pour une plateforme Temps Réel Linux/Xenomai sous LinRT BSP Cobalt

Travaux pratiques

- Configuration du noyau linux PREEMPT-RT du kit Phytex SBC-Mira phyCORE i.MX 6 Quad
- Modification du devicetree et mise en œuvre d'un patch noyau avec l'utilitaire GIT sous Yocto

2.

Customisation d'une Image Yocto FSL Community

- Méthodologie
- Ajout de son propre Layer
- Recette minimale de construction de paquetage
- Customisation de recettes
- Ajout de d'étapes de compilation
- Notion de classes

3.

Software Development Kit Yocto RT

- Pluggin Eclipse Yocto SDK
- Ajout de package-dev et header kernel pour développement de device drivers et d'applications Xenomai
- Génération et déploiement d'un SDK complet LinRT Xenomai Cobalt

Travaux Pratiques :

- Mise en œuvre du SDK Yocto Xenomai LinRT Cobalt : configuration, compilation et débogage sur cible Mira phyCORE i.MX6 Quad d'une application Xenomai Temps Réel synchrone sur une interruption

Travaux pratiques (suite)

- Écriture d'une recette simple Yocto pour l'intégration de votre application à l'image filesystem finale
- Analyse d'exemples de paquetages
- Ajouter des fichiers de configuration à une recette
- Ecriture et ajout d'un layer à la compilation du système Linux embarqué LinRT Yocto pour le Kit SBC-Mira phyCORE i.MX 6 Quad
- Ajouter des fichiers de configuration à une recette
- Recette d'ajout de script de démarrage avec initd
- Recette d'ajout de service de démarrage sous Systemd

Introduction au développement de pilotes Linux / Pilotes de périphériques sous Linux

- Introduction à la programmation en mode noyau
- Architecture d'un module simple
- Programmation de pilote de périphérique
- Programmation de pilotes de périphériques simples : Structure de la File Operation
- API du noyau Linux et gestion mémoire en Kernel Mode

Travaux pratiques

- Compilation et déploiement d'un module driver linux externe/ propriétaire avec le SDK Yocto LinRT sous Eclipse
- Programmation de pilotes de périphériques Linux, Signaux et Timer en kernel mode sur plateforme i.MX6
- Driver bloquant, gestion d'interruption sur plateforme sur cible Phytex Mira phyCORE i.MX6 Quad

3.

Introduction aux applications faible latence (low-latency)

- Particularités du noyau PREEMPT-RT sous BSP Yocto LinRT Oxygen

Travaux pratiques

- Evaluation des performances avec et sans le support PREEMPT-RT sous BSP Yocto LinRT Oxygen sur cible Phytex Mira phyCORE i.MX6 Quad

- Création d'application multitâche faible latence POSIX pThread

Introduction aux applications embarquées temps réel en mode utilisateur

- API Skin Xenomai 3.0 pSOS/VxWorks sous BSP Yocto LinRT Mercury

Travaux pratiques

- Portage d'application VxWorks sous Linux avec le BSP Yocto LinRT Mercury sur cible Phytex Mira phyCORE i.MX6 Quad

Introduction aux applications embarquées temps réel dur en mode utilisateur et mode noyau

- Particularités du noyau Xenomai 3.0 et des extensions temps réel sous BSP Yocto LinRT Cobalt

- API Xenomai 3.0 et Modèle driver RTDM

- Recette Yocto LinRT BSP Cobalt Xenomai

4.

Introduction au support réseau RTnet

5.

- Particularités de la stack RTNet
- Support CAN Temps Réel dur et RT Socket
- Support Ethernet Temps Réel et champ d'application (PowerLink, Ethercat, Modbus TCP etc...)

Travaux pratiques

- Evaluation des performances du support Temps Réel sous BSP Yocto LinRT Xenomai Cobalt sur cible Phytex Mira phyCORE i.MX6 Quad

- Mise en œuvre d'une application CAN temps réel dur sous sous BSP Yocto LinRT Cobalt