

DURÉE ET TARIF : 5 jours (35 heures) -

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 du BSP Yocto Phytex PD\_20.1.0, Linux kernel sur le Kit SBC Sargas phyCORE-STM32MP1.

- Maîtriser l'installation de l'environnement Yocto
- Maîtriser les commandes usuelles Bitbake
- Gérer la configuration de build et de la Machine phyCORE-STM32MP1
- Ajouter et Mettre en oeuvre ses propres layers Yocto
- Configurer et adapter le noyau linux à sa plate-formeSTM32MP1
- Construire et customisez son image Linux openLinuxST avec Yocto
- Générer ses propres devicetree à l'aide de CubeMX pour STM32 MPU
- Adapter les drivers du noyau Linux pour phyCORE-STM32MP1
- Développer et mettre au point des applications FreeRTOS avec le SDK Yocto phyCORE-STM32MP1

## Pré-requis

Notre formation Yocto BSP pour processeurs STM32MP1 est adaptée aux développeurs en informatique et techniciens dans le domaine de Linux embarqué confrontés aux problèmes de portage d'applications sur système STM32MP1 sous Yocto, en exploitant les capacités temps réel du Cortex-M4 et les mécanismes de sécurité du Cortex-A7.

### Introduction à la technologie Qt5

#### 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 STM32 MPU openSTLinux

- Les layers spécifiques openSTLinux
- 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 Phytex SBC Sargas phyCORE-STM32MP1

#### Phytex Yocto STM32 MPU BSP

- Fichier local.conf et réglages globaux
- Notion de MACHINE et de PROVIDER

1.

### Customisation d'une Image Yocto openSTLinux (suite)

- Customisation de recettes
- Ajout de d'étapes de compilation
- Notion de classes

#### Travaux pratiques

- Ajouter des fichiers de configuration à une recette
- Recette d'ajout de script de démarrage avec systemd, passage initrd pour distribution minimale Tiny openSTLinux
- Ajout de package-dev et header kernel pour développement de device driver
- Génération et déploiement d'un SDK Yocto complet
- Notion d'Extended-SDK

#### Travaux pratiques

- Mise en œuvre du SDK GNU GCC/GDB Yocto sous Eclipse: configuration, compilation et débogage sur cible SBC Sargas phyCORE-STM32MP1

3.

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

- Introduction à la programmation en mode noyau
- Architecture d'un module simple

#### Travaux pratiques

- Compilation et déploiement d'un module linux externe/ propriétaire avec le SDK Yocto
- Programmation de pilotes de périphériques simples : Structure de la File Operation
- API du noyau Linux et gestion mémoire en Kernel Mode

#### Travaux pratiques

- Programmation de pilotes de périphériques Linux, Signaux et Timer en kernel mode sur plateforme STM32MP1
- Driver bloquant, gestion d'interruption sur plateforme sur SBC Sargas phyCORE-STM32MP1

4.

### Phytex Yocto STM32 MPU BSP (suite)

- Spécificités et optimisation du noyau linux-stm32mp sous Yocto
- Customisation du kernel devicetree à l'aide de CubeMX

#### Travaux pratiques

- Ajout de son propre Layer YOCTO BSP et creation de sa propre MACHINE STM32MP1
- Configuration du noyau linux du kit Phytex SBC Sargas phyCORE-STM32MP1
- Modification du devicetree et mise en œuvre d'un patch noyau avec l'utilitaire GIT sous Yocto
- Génération d'une image Linux pour SOM phyCORE-STM32MP1

#### Customisation d'une Image Yocto openSTLinux

- Méthodologie
- Recette minimale de construction de paquetage

#### Travaux pratiques

- 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 Yocto pour MACHINE STM32MP1

2.

### Mise en oeuvre de boot secure

- Context Cortex-A Secure et Non-Secure
- Basic boot chain et trusted boot chain
- Trusted Firmware et Trusted Devices
- Notion de Trusted Application et de secure Monitor
- Crypto support et Signed Bootloader

#### Travaux pratiques

- Mise en œuvre de Trusted device et limitations STM32MP1
- Exemple de Trusted Application Cortex-A (OP-TEE Monitor)

#### Concept de Communication Inter Processeurs

- Notion de Domain ID et Ressources
- API RPMSG FreeRTOS M4
- Drivers RPMSG Linux

#### Travaux pratiques

- Echanges de données entre application sous Linux et Services FreeRTOS via RPMSG

5.