Formation développement Linux embarqué avec Yocto Project et OpenEmbedded

Session de 3 jours

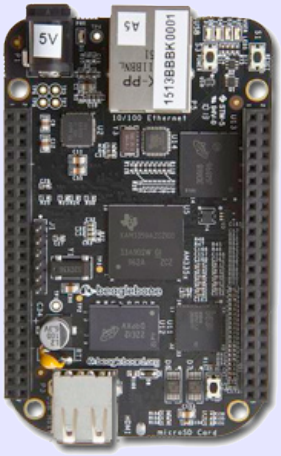
Titre	**Formation développement Linux embarqué avec Yocto Project et OpenEmbedded**
Aperçu	Comprendre l'architecture de Yocto Project Utilisation pour compiler un système de fichiers et exécuter celui-ci sur votre plate-forme matérielle Étendre des recettes (*recipes*) existantes et en écrire de nouvelles Création de *layers* Intégration de votre matériel dans un *BSP* Création d'images sur mesure Développement applicatif à l'aide du SDK de Yocto Project
Supports	Vérifiez que le contenu de la formation correspond à vos besoins : `https://bootlin.com/doc/training/yocto`.
Durée	**Trois** jours - 24 h (8 h par jour) 40% de présentations et 60% de travaux pratiques.
Formateur	Un des ingénieurs mentionnés sur : `https://bootlin.com/training/trainers/`
Langue	Présentations : Français Supports : Anglais
Public visé	Sociétés et ingénieurs intéressés par l'utilisation de Yocto Project pour construire leur système Linux embarqué.
Pré-requis	**Connaissance de Linux embarqué**, sujet couvert par notre formation Linux embarqué (`https://bootlin.com/training/embedded-linux/`) **Connaissance et pratique des commandes UNIX ou GNU/Linux** Les personnes n'ayant pas ces connaissances doivent s'autoformer, par exemple en utilisant nos supports de formation disponibles en ligne : (`https://bootlin.com/blog/command-line/`

Équipement nécessaire	**Pour les sessions sur site uniquement** Le matériel est fourni par Bootlin durant les sessions inter-entreprises • Projecteur vidéo • Un ordinateur sur chaque bureau (pour une ou deux personnes), avec au moins 8 Go de RAM, un processeur au moins équivalent à un Intel Core i5, et Ubuntu Linux installé dans une **partition dédiée d'au moins 40 Go. L'utilisation de Linux dans une machine virtuelle n'est pas supportée**, en raison de problèmes avec la connexion au matériel. • Nous avons besoin d'Ubuntu Desktop 18.04 (Xubuntu et autres variantes fonctionnent également). Nous ne supportons pas d'autres distributions, car nous ne pouvons tester toutes les versions des paquets. • **Connexion à Internet** (directe ou par le proxy de l'entreprise). • **Les ordinateurs contenant des données importantes doivent être sauvegardés** avant d'être utilisés dans nos sessions. Certains participants ont déjà commis des erreurs lors de travaux pratiques avec pour conséquence des pertes de données.
Supports	Copie électronique des présentations et travaux pratiques. Version électronique des données pour les travaux pratiques..

Matériel, première option

Carte BeagleBone Black

- Un processeur ARM AM335x de Texas Instruments (à base de Cortex-A8), avec accélération 3D, etc.
- 512 Mo de RAM
- 2 Go de stockage eMMC embarqué sur la carte
  (4 Go avec la révision C)
- USB hôte et device
- Sortie HDMI
- Connecteurs à 2 x 46 broches, pour accéder aux UARTs, aux bus SPI, aux bus I2C, et à d'autres entrées/sorties du processeur.



Matériel, deuxième option

Carte STMicroelectronics STM32MP157A-DK1 Discovery

- Processeur STM32MP157A (double Cortex-A7) de STMicroelectronics
- Alimentée par USB
- 512 Mo DDR3L RAM
- Port Gigabit Ethernet port
- 4 ports hôte USB 2.0
- 1 port USB-C OTG
- 1 connecteur Micro SD
- Debugger ST-LINK/V2-1 sur la carte
- Connecteurs compatibles Arduino Uno v3
- Codec audio
- Divers: boutons, LEDs

1er jour - Matin

Cours - Introduction aux outils de compilation de systèmes Linux embarqué

- Vue d'ensemble de l'architecture d'un système Linux embarqué
- Méthodes pour compiler un système de fichiers
- Utilité des outils de compilation

Cours - Vue d'ensemble de Yocto Project et du système de référence Poky

- Organisation des sources du projet
- Création d'un système de fichiers avec Yocto Project

TP - 1ère compilation avec Yocto Project

- Téléchargement du système de référence Poky
- Compilation d'une image système

1er Jour - Après-midi

Cours - Utilisation de Yocto Project - Notions de base

- Structure des fichiers générés
- Flasher et installer l'image du système

TP - Flasher et booter

- Flasher et booter l'image du système sur la carte

Cours - Utilisation de Yocto Project - Utilisation avancée

- Configuration de la compilation
- Personnalisation de la sélection de paquetages

TP - Utilisation de NFS et configuration de la compilation

- Configurer la carte pour démarrer via NFS
- Apprendre à utiliser le mécanisme `PREFERRED_PROVIDER`

2ème jour - Matin

Cours - Écriture de recettes - Fonctionnalités de base

- Écriture d'une recette minimale
- Ajout de dépendances
- Organisation du développement avec *bitbake*

TP - Ajouter la compilation d'une application

- Création d'une recette pour *nInvaders*
- Ajout d'*nInvaders* à l'image finale

Cours - Écriture de recettes - Fonctionnalités avancées

- Extension et redéfinition de recettes
- Rajouter des étapes au processus de compilation
- Familiarisation avec les classes
- Analyse d'exemples
- Logs
- Mise au point des dépendances

2ème jour - Après-midi

TP - Apprendre à configurer les paquetages

- Extension d'une recette pour ajouter des fichiers de configuration
- Utilisation de `ROOTFS_POSTPROCESS_COMMAND` pour modifier le système de fichier final
- Étude des dépendances entre paquetages

Cours - Layers

- Ce que sont les *layers*
- Où trouver les *layers*
- Création d'un *layer*

TP - Écriture d'un layer

- Apprendre à écrire un *layer*
- Ajouter le *layer* à la compilation
- Inclure *nInvaders* dans le nouveau *layer*

3ème jour - Matin

Cours - Écriture d'un BSP	TP - Mise en oeuvre de modifications du noyau
• Extension d'un BSP existant • Ajout d'une nouvelle machine • Chargeurs de démarrage • Linux et la recette linux-yocto • Ajouter un type d'image personnalisé	• Extension de la recette pour le noyau pour ajouter le pilote pour le Nunchuk • Configurer le noyau pour compiler le pilote du Nunchuk • Jouer à *nInvaders*

3ème jour - Après-midi

Cours - Création d'une image sur mesure	TP - Création d'une image sur mesure
• Écriture d'une recette d'image • Ajouter des utilisateurs et des groupes • Ajouter une configuration personnalisée • Écrire et utiliser des groupes de recettes de paquetages	• Écrire une recette d'image personnalisée • Ajouter *nInvaders* à l'image sur mesure
Cours - Création et utilisation d'un SDK	**TP - Expérimentations avec le SDK**
• Comprendre l'utilité d'un SDK pour le développeur d'applications • Construire un SDK pour l'image sur mesure	• Construction d'un SDK • Utilisation du SDK de Yocto Project