



# Formation développement Linux embarqué avec Yocto Project et OpenEmbedded

Formation sur site, 3 jours

Dernière mise à jour : 20 May 2024

<b>Titre</b>	<b>Formation développement Linux embarqué avec Yocto Project et OpenEmbedded</b>
<b>Objectifs opérationnels</b>	<ul style="list-style-type: none"><li>• Être capable de comprendre le rôle et le principe d'un build system Linux embarqué, et comparer Yocto Project/OpenEmbedded aux autres outils offrant des fonctionnalités similaires.</li><li>• Être capable de configurer et de réaliser la compilation d'un système Linux embarqué simple avec Yocto, et d'installer le résultat sur une plateforme embarquée.</li><li>• Être capable d'écrire ou d'étendre des recettes de paquets, pour vos propres paquets ou personnalisations.</li><li>• Être capable d'utiliser des <i>layers</i> de recettes existants, et de créer votre propre nouveau <i>layer</i>.</li><li>• Être capable d'intégrer le support pour votre carte embarqué dans un <i>layer BSP</i>.</li><li>• Être capable de créer des images personnalisées.</li><li>• Être capable d'utiliser le SDK du Yocto Project pour développer des applications.</li><li>• Être capable d'utiliser devtool pour développer une recette.</li></ul>
<b>Durée</b>	<b>Trois jours - 24 h (8 h par jour)</b>
<b>Méthodes pédagogiques</b>	<ul style="list-style-type: none"><li>• Présentations animées par le formateur : 40% de la durée de formation</li><li>• Travaux pratiques réalisés par les participants : 60% de la durée de formation</li><li>• Version électroniques de supports de présentation, des instructions et des données de travaux pratiques. Les supports sont librement disponibles sur <a href="https://bootlin.com/doc/training/yocto">https://bootlin.com/doc/training/yocto</a>.</li></ul>
<b>Formateur</b>	Un des ingénieurs mentionnés sur : <a href="https://bootlin.com/training/trainers/">https://bootlin.com/training/trainers/</a>



<b>Langue</b>	Présentations : Français Supports : Anglais
<b>Public visé</b>	Sociétés et ingénieurs intéressés par l'utilisation de Yocto Project pour construire leur système Linux embarqué.
<b>Pré-requis</b>	<ul style="list-style-type: none"><li>• <b>Connaissance et pratique des commandes UNIX ou GNU/Linux</b> : les participants doivent être à l'aise avec l'utilisation de la ligne de commande Linux. Les participants manquant d'expérience sur ce sujet doivent se former par eux-mêmes, par exemple en utilisant nos supports de formation disponible à l'adresse <a href="http://bootlin.com/blog/command-line/">bootlin.com/blog/command-line/</a>.</li><li>• <b>Expérience minimale en développement Linux embarqué</b> : les participants doivent avoir une compréhension minimale de l'architecture d'un système Linux embarqué : rôle du noyau Linux par rapport à l'espace utilisateur, développement d'applications espace utilisateur en C. Suivre la formation <i>Linux embarqué</i> de Bootlin, disponible sur <a href="http://bootlin.com/training/embedded-linux/">bootlin.com/training/embedded-linux/</a>, permet de remplir ce pré-requis.</li><li>• <b>Niveau minimal requis en anglais : B1</b>, d'après le <i>Common European Framework of References for Languages</i>, pour nos sessions animées en anglais. Voir <a href="http://bootlin.com/pub/training/cefr-grid.pdf">bootlin.com/pub/training/cefr-grid.pdf</a> pour une auto-évaluation.</li></ul>
<b>Équipement nécessaire</b>	<p><b>Pour les sessions en présentiel dans les locaux de nos clients, notre client doit fournir :</b></p> <ul style="list-style-type: none"><li>• Projecteur vidéo</li><li>• Un ordinateur sur chaque bureau (pour une ou deux personnes), avec au moins un processeur Intel Core i5, 8 Go de RAM et Ubuntu Linux 22.04 installé dans une <b>partition dédiée d'au moins 100 Go</b>.</li><li>• Les distributions autres que Ubuntu Linux 22.04 ne sont pas supportées, et l'utilisation de Linux dans une machine virtuelle n'est également pas supportée.</li><li>• <b>Connexion à Internet rapide et sans filtrage</b> : au moins 50 Mbit/s de bande passante en téléchargement, et pas de filtrage des sites Web et protocoles.</li><li>• <b>Les ordinateurs contenant des données importantes doivent être sauvegardés</b> avant d'être utilisés dans nos sessions.</li></ul>

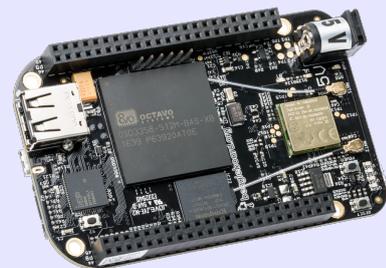


<b>Modalités d'évaluation</b>	Seuls les participants qui auront assisté à l'intégralité des journées de formation, et qui auront obtenu plus de 50% de réponses correctes à l'évaluation finale recevront une attestation individuelle de formation de la part de Bootlin.
<b>Handicap</b>	Les participants en situation de handicap qui ont des besoins spécifiques sont invités à nous contacter à l'adresse <a href="mailto:training@bootlin.com">training@bootlin.com</a> afin de discuter des adaptations nécessaires à la formation.

## 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

Une de ces cartes de STMicroelectronics :  
**STM32MP157A-DK1**, **STM32MP157D-DK1**,  
**STM32MP157C-DK2** ou **STM32MP157F-DK2**

- Processeur STM32MP157, 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
- Écran LCD tactile (uniquement sur cartes DK2)



## 1<sup>er</sup> 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

- Présentation de l'outil de compilation Yocto / OpenEmbedded et de sa terminologie.
- Vue d'ensemble du système de référence Poky



### **Cours - Bases de l'utilisation de Yocto Project**

- Mise en place du répertoire de travail et de l'environnement
- Configuration de l'outil de compilation
- Compilation de l'image d'un système de fichiers racine

### **TP 1 - 1<sup>ère</sup> compilation avec Yocto Project**

- Téléchargement du système de référence Poky
- Configuration de l'outil de compilation
- Compilation d'une image système

## **1<sup>er</sup> Jour - Après-midi**

---

### **Cours - Utilisation de Yocto Project - Notions de base**

- Structure des fichiers générés

### **TP 1- Flasher et booter**

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

### **Cours - Utilisation de Yocto Project - Utilisation avancée**

- Assignation des variables, opérateurs et surcharge
- Paquetages : variantes de paquetages
- Options de la commande bitbake

### **TP 2- Utilisation de NFS et configuration de la compilation**

- Configurer la carte pour démarrer via NFS
- Rajouter un paquetage au système de fichiers racine
- Apprendre à utiliser le mécanisme PREFERRED\_PROVIDER
- Se familiariser avec les options de la commande bitbake



## 2<sup>ème</sup> jour - Matin

### Cours - Écriture de recettes - Fonctionnalités de base

- Recettes : vue d'ensemble
- Organisation d'un fichier de recette
- Application de patches
- Exemples de recettes

### TP 3 - Ajouter la compilation d'une application

- Création d'une recette pour *nInvaders*
- Résolution de problèmes liés à la recette
- Résolution de problèmes liés à la compilation croisée
- Ajout d'*nInvaders* à l'image finale

### Cours - Écriture de recettes - Fonctionnalités avancées

- Extension et redéfinition de recettes
- Paquetages virtuels
- Familiarisation avec les classes
- Inclusion d'exemples avec BitBake
- Mise au point des recettes
- Configuration de l'utilisation du réseau par BitBake

## 2<sup>ème</sup> jour - Après-midi

### Cours - Layers

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

### TP 4- Écriture d'un layer

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



## 3<sup>ème</sup> jour - Matin

---

### TP 5 - Étendre une recette

- Étendre la recette pour le noyau pour rajouter des patches
- Configurer le noyau pour compiler le pilote du nunchuk
- Modifier la recette ninvaders pour rajouter des patches
- Jouer avec *nInvaders*

### Cours - Écriture d'un BSP

- Introduction aux layers BSP
- Ajout d'une nouvelle machine
- Configuration du chargeur de démarrage
- Linux : la classe kernel.bbclass et la recette linux-yocto

### TP 6 - Création d'une configuration spécifique pour une machine

- Créer un nouvelle configuration de machine
- Compiler une image pour la machine

### Cours - Layers de distribution

- Configuration d'une distribution
- Layers de distribution



## Cours - Images

- Écriture d'une recette d'image
- Types d'images
- Écriture et utilisation de groupes de paquets

## TP 7- Création d'une image sur mesure

- Rajouter une recette de base pour une image
- Sélectionner les fonctionnalités et les paquets de l'image
- Ajouter un groupe de paquets sur mesure
- Ajouter une variante d'image pour le débogage

## 3<sup>ème</sup> jour - Après-midi

### Cours - Écriture de recettes - Pour aller plus loin

- Le sysroot de chaque recette
- Utilisation de code Python code dans les méta-données
- Drapeaux de variables
- Fonctionnalités de paquets et PACKAGESCONFIG
- Fonctionnalités conditionnelles
- Découpage de paquets
- Détails sur les dépendances

### Cours - Licences

- Gestion de licences open source

### Cours - SDK pour le projet Yocto

- Objectifs du SDK
- Compilation et personnalisation d'un SDK
- Utilisation d'un SDK pour le projet Yocto

### TP 8 - Développement d'une application au moyen du SDK de Poky

- Compilation d'un SDK
- Utilisation du SDK



### **Cours - Devtool**

- Présentation devtool
- Cas d'utilisation de devtool

### **TP 9 - Utilisation de devtool**

- Création d'une nouvelle recette
- Modification d'une recette pour ajouter un nouveau patch
- Mettre à jour une recette pour prendre en charge une nouvelle version

### **Cours - Gestion automatique de layers**

- Gestion automatique de layers

### **Cours - Gestion de paquetages à l'exécution**

- Introduction à la gestion de paquetages à l'exécution
- Configuration de la compilation
- Configuration d'un serveur de paquetages
- Configuration de la machine cible