



# Formation développement Linux embarqué avec Buildroot

Formation sur site, 3 jours

Dernière mise à jour : 20 May 2024

<b>Titre</b>	<b>Formation développement Linux embarqué avec Buildroot</b>
<b>Objectifs opérationnels</b>	<ul style="list-style-type: none"><li>• Être capable de comprendre le principe d'un build system Linux embarqué, et comparer Buildroot aux autres outils offrant des fonctionnalités similaires.</li><li>• Être capable de créer un système Linux embarqué simple avec Buildroot : créer une configuration, lancer la compilation, installer le résultat sur une plateforme embarquée.</li><li>• Être capable d'ajuster la configuration de Buildroot pour construire un système Linux embarqué adapté à des besoins spécifiques : choix de la chaîne de compilation croisée, gestion de la configuration du noyau Linux, personnalisation du système de fichiers racine.</li><li>• Être capable de créer de nouveaux paquets dans Buildroot pour intégrer des applications et bibliothèques supplémentaires dans le système Linux embarqué.</li><li>• Être capable d'utiliser les outils proposés par Buildroot pour gérer et analyser le build : suivi des vulnérabilités, conformité aux licences open-source, etc.</li><li>• Être capable de développer et déboguer des applications user-space Linux dans un contexte où Buildroot est utilisé.</li><li>• Être capable d'interagir avec la communauté open-source du projet Buildroot et de comprendre le fonctionnement interne de Buildroot.</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/buildroot">https://bootlin.com/doc/training/buildroot</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 qui utilisent déjà Buildroot ou qui sont intéressées par l'utiliser pour construire leurs systèmes 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 8 Go de RAM et Ubuntu Linux 22.04 installé dans une <b>partition dédiée d'au moins 30 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>

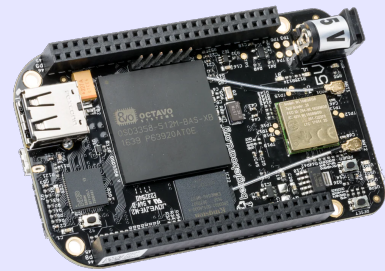


<b>Modalités d'évaluation</b>	Seuls les participants qui auront assisté à l'intégralité des journées de formation, et qui auront obtenu plus de 50% de réponses correctes à l'évaluation finale recevront une attestation individuelle de formation de la part de Bootlin.
<b>Handicap</b>	Les participants en 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.

## Plateforme matérielle pour les travaux pratiques, option #1

### Carte **BeagleBone Black**

- Processeur ARM AM335x (single Cortex-A8) de Texas Instruments
- Alimentation par USB
- 512 MB de RAM
- 2 ou 4 GB de stockage eMMC
- USB hôte et périphérique
- 1 slot Micro SD
- Sortie HDMI
- Connecteur 2 x 46 broches, pour accéder aux UARTs, bus SPIs, I2Cs, etc.





## Plateforme matérielle pour les travaux pratiques, option #2

### Carte STMicroelectronics STM32MP157D Discovery Kit 1

- Processeur STM32MP157D (dual Cortex-A7) de STMicroelectronics
- Alimentation par USB
- 512 MB DDR3L RAM
- Ethernet Gigabit
- 4 ports USB 2.0 hôte
- 1 port USB-C OTG
- 1 slot Micro SD
- Debugger ST-LINK/V2-1
- Connecteurs compatibles Arduino
- Codec audio, boutons, LEDs



## 1<sup>er</sup> jour - Matin

### Cours - Introduction à Buildroot et aux systèmes de build

- Architecture générale d'une système Linux embarqué
- Choix entre systèmes de build et distributions binaires
- Rôle d'un système de build
- Comparaison des systèmes de build existants

### Cours - Présentation de Buildroot

- Points clés autour du projet
- Téléchargement des sources de Buildroot
- Configuration simple de Buildroot
- Exécution d'une première compilation



### TP - Utilisation simple de Buildroot

- Téléchargement et configuration de Buildroot
- Configurer et compiler un système simple avec Buildroot pour un système embarqué
- Flasher et tester le système généré par Buildroot

### Cours - Gestion de la compilation et de la configuration

- Compilation en dehors des sources
- Utiliser et créer des fichiers *defconfigs*
- Fragments de *defconfigs*
- Autres astuces pour la compilation

## 1<sup>er</sup> jour - Après-midi

### Cours - Sources de Buildroot et arborescence des fichiers générés

- Détails sur l'organisation du code source de Buildroot
- Détails sur l'arborescence des fichiers générés

### Cours - Chaînes de compilation *toolchains* dans Buildroot

- Les différents possibilités d'usage de chaînes de compilation dans Buildroot.
- Tour d'horizon des options liées aux chaînes de compilation.
- Utilisation de chaînes de compilation binaires, comme celles de Bootlin. Détails sur les fonctionnalités *multilib* et l'intégration des *toolchains* dans Buildroot.
- Génération de *toolchains* sur mesure avec *Crosstool-NG*, et leur utilisation comme chaînes externes.



## Cours - Gestion de la configuration du noyau Linux

- Charger, modifier et sauvegarder la configuration du noyau.

## Cours - Construction du système de fichier racine dans Buildroot

- Comprendre comment Buildroot construit le système de fichiers racine : *skeleton*, installation de composants, *overlays*, scripts *post-build* et *post-image*.
- Personnalisation du contenu du système de fichiers
- Configuration du système : sélection de la *console*, plusieurs méthode de gestion de */dev*, les différentes implémentations d'*init*, etc.
- Comprendre comment Buildroot génère les images de systèmes de fichiers.

## TP - Personnalisation du système de fichiers

- Exploration des fichiers générés
- Personnalisation du système de fichiers racine en utilisant un *rootfs overlay*
- Personnaliser le noyau avec des correctifs et des options de configuration supplémentaires
- Rajout de nouveaux composants
- Utilisation de fichiers *defconfig* et compilation en dehors des sources.





## 2<sup>ème</sup> jour - Matin

### Cours - Infrastructure de téléchargement dans Buildroot

- Méthodologie de téléchargement
- Site primaire et sites de backup, compilation en mode déconnecté
- Téléchargement via systèmes de contrôle de versions, vérification d'intégrité
- Cibles *make* en rapport avec les téléchargements

### Cours - Introduction à GNU Make

- Éléments de base des règles de make
- Définition et utilisation de variables
- Conditions et fonctions
- Écriture de recettes

### Cours - Intégration de nouveaux composants dans Buildroot

- Comment rajouter de nouveaux paquetages au système de configuration de Buildroot
- Comprendre les différentes infrastructures de paquetages : pour des composants *generic*, *autotools*, *CMake*, *Python* et autres
- Écriture un fichier `Config.in` pour un composant : comment exprimer des dépendances vers d'autres composants, vers des options de toolchains, etc.
- Détails sur l'écriture d'une recette pour un composant : description de l'emplacement du code source, de la méthode de téléchargement, de configuration, de compilation et d'installation, gestion des dépendances, etc.

### TP - Nouveaux composants dans Buildroot

- Création d'un nouveau paquetage pour *nInvaders*
- Comprendre comment rajouter des dépendances
- Ajouter des correctifs pour *nInvaders* pour prendre en charge le contrôle via un *Nunchuk*



## 2<sup>ème</sup> jour - Après-midi

### Cours - Notions avancées sur les paquetages

- Rapport de licences
- Prise en charge des correctifs : ordre d'application et format, répertoire global pour les correctifs, etc.
- Utilisateur, droit d'accès, tables de fichiers devices
- Script d'init et fichiers unitaires pour systemd
- Scripts de configuration
- Compréhension des *hooks*
- Surcharger des commandes
- Gestion des paquetages legacy
- Paquetages virtuels

### TP - Paquetages avancés

- Packager une application avec une dépendance obligatoire et une dépendance optionnelle
- Packager une bibliothèque, hébergée sur GitHub
- Utilisation de *hooks* pour ajuster les paquetages
- Rajouter un correctif à un paquetage

## 3<sup>ème</sup> jour - Matin

### Cours - Analyse d'une compilation : licences, dépendances, temps de construction

- Utilisation de l'infrastructure de gestion des informations légales
- Représentation graphique des dépendances entre paquetages
- Collecte d'informations et représentation du temps de compilation

### Cours - Sujets avancés

- BR2\_EXTERNAL pour stocker des personnalisations à l'extérieur des sources de Buildroot
- Cibles make spécifiques pour les paquetages
- Comprendre les recompilations
- Astuces pour compiler plus vite





## TP - Sujets avancés

- Utilisation des capacités de génération de graphes de temps de compilation
- Génération de graphes de dépendances
- Utilisation du rapport sur les licences, et ajout d'informations légales à vos propres paquetages
- Utilisation de BR2\_EXTERNAL

## 3<sup>ème</sup> jour - Après-midi

### Cours - Développement applicatif avec Buildroot

- Utilisation de Buildroot pendant le développement d'applications
- Utilisation de l'environnement de Buildroot pour compiler des applications en dehors de Buildroot
- Générer un SDK pour d'autres développeurs
- Débug à distance avec Buildroot

### TP - Développement applicatif avec Buildroot

- Compiler et exécuter votre propre application
- Débug à distance de votre application
- Utilisation de `<pkg>_OVERRIDE_SRCDIR`

### Cours - Comprendre les mécanismes internes de Buildroot

- Description détaillée du processus de compilation de Buildroot : toolchain, paquetages, construction du système de fichiers race, fichiers *stamp*, etc.
- Comprendre les paquetages virtuels.

### Cours - Obtenir de l'aide et s'impliquer

- Obtenir de l'assistance technique : *Bugzilla*, *liste de discussion*, *IRC*
- Contribuer : comprendre le processus de développement, comment soumettre des correctifs