

Formation développement Linux embarqué avec Buildroot

Session de 3 jours

Titre	Formation développement Linux embarqué avec Buildroot
Thématiques	<p>Introduction à Buildroot Gérer et compiler une configuration Arborescence des sources et des fichiers générés Gestion de la configuration du noyau Linux avec Buildroot Système de fichiers racine Infrastructure de téléchargement Introduction à GNU Make Ajout de nouveaux paquets Aspects avancés sur les paquets Buildroot Analyse du build Sujets avancés Développement d'application avec Buildroot Comprendre le fonctionnement interne de Buildroot Communauté Buildroot: obtenir du support et contribuer Quoi de neuf dans Buildroot ?</p>
Supports	<p>Vérifiez que le contenu de la formation correspond à vos besoins : https://bootlin.com/doc/training/buildroot.</p>
Durée	<p>Trois jours - 24 h (8 h par jour) 40% de présentations et 60% de travaux pratiques.</p>
Formateur	<p>Thomas Petazzoni. Thomas est un des principaux développeurs de Buildroot depuis 2009, avec plus de 2700 patches intégrés et une participation active au processus de développement.</p>
Langue	<p>Présentations : Français Supports : Anglais</p>
Public ciblé	<p>Sociétés qui utilisent déjà Buildroot ou qui sont intéressées par l'utiliser pour construire leurs systèmes Linux embarqué.</p>
Pré-requis	<p>Connaissance de Linux embarqué, sujet couvert par notre formation Linux embarqué : https://bootlin.com/training/embedded-linux/</p> <p>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/)</p>

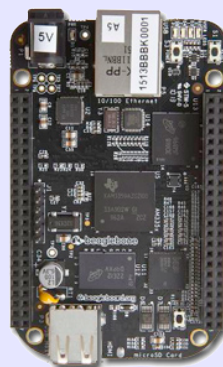


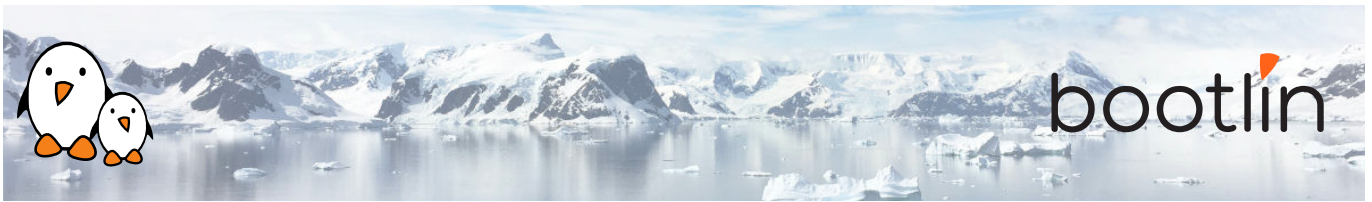
Équipement nécessaire	<p>Pour les sessions sur site uniquement</p> <p>Le matériel est fourni par Bootlin durant les sessions inter-entreprises</p> <ul style="list-style-type: none">• Projecteur vidéo• Un ordinateur sur chaque bureau (pour une ou deux personnes), avec au moins 8 Go de RAM, et Ubuntu Linux installé dans une partition dédiée d'au moins 30 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 16.04 (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

La plateforme matérielle utilisée pendant les travaux pratiques de cette formation est la carte **BeagleBone Black**, dont voici les caractéristiques :

- 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.





1^{er} jour - Matin

Cours - Introduction à Buildroot et aux systèmes de build

- Architecture générale d'un 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 la carte BeagleBone Black
- Flasher et tester le système généré sur la carte BeagleBone Black.

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^{er} 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.

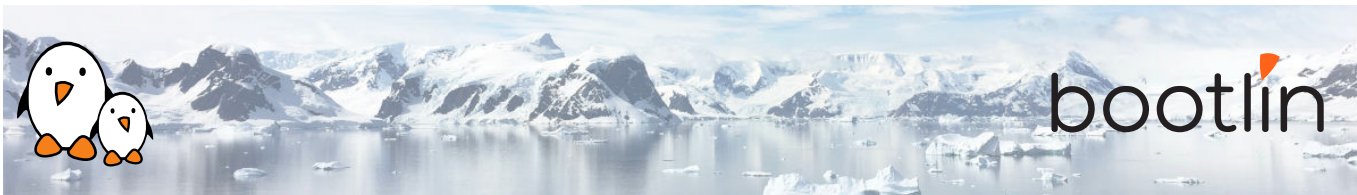
2^{ème} jour - Matin

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

TP - Nouveaux composants dans Buildroot

- Comment rajouter de nouveaux paquets au système de configuration de Buildroot
 - Comprendre les différentes infrastructures de paquets : pour des composants *generic*, *autotools*, *CMake*, *Python* et autres
 - Écriture d'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.
- 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^{ème} jour - Après-midi

Cours - Notions avancées sur les paquets

TP - Paquetages avancés

- 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
- 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^{ème} 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^{ème} 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, nouveautés dans Buildroot

- Obtenir de l'assistance technique: *Bugzilla*, *liste de discussion*, *IRC*
- Contribuer: comprendre le processus de développement, comment soumettre des correctifs
- Nouveautés dans Buildroot: résumé des principaux changements depuis les deux dernières années