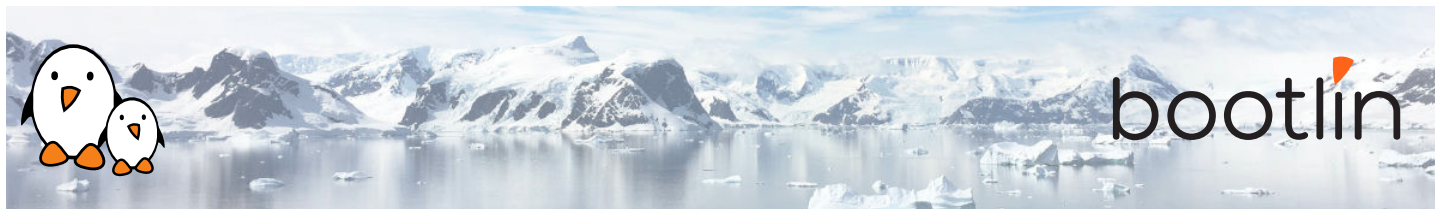


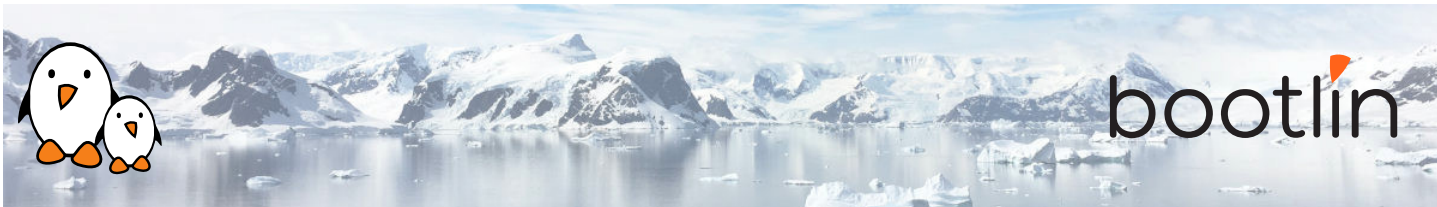
# Formation optimisation du temps de démarrage de Linux embarqué

Formation sur site, 3 jours  
Dernière mise à jour : 29 April 2024

<b>Titre</b>	<b>Formation optimisation du temps de démarrage de Linux embarqué</b>
<b>Objectifs opérationnels</b>	<ul style="list-style-type: none"><li>• Être capable d'utiliser les outils et techniques pour mesurer le temps de démarrage d'un système embarqué.</li><li>• Être capable de réduire le temps de démarrage au niveau de l'initialisation de l'espace utilisateur Linux.</li><li>• Être capable de réduire le temps de démarrage au niveau de l'initialisation du noyau Linux.</li><li>• Être capable de réduire le temps de démarrage au niveau de l'initialisation du chargeur de démarrage.</li><li>• Être capable de mettre en oeuvre d'autres techniques avancées et alternatives d'optimisation du temps de démarrage.</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/boot-time">https://bootlin.com/doc/training/boot-time</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 développeurs de systèmes Linux embarqués. Personnes offrant de l'assistance à de tels développeurs.



<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="https://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="https://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="https://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>	<p>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.</p>



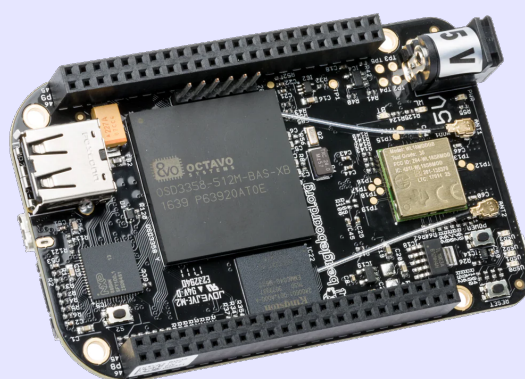
## Handicap

Les participants en situation de handicap qui ont des besoins spécifiques sont invités à nous contacter à l'adresse [training@bootlin.com](mailto:training@bootlin.com) afin de discuter des adaptations nécessaires à la formation.

## 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.



## Démonstrations

Les démos de cette formation utiliseront les périphériques matériels suivants :

- Une webcam USB
- Une carte d'extension d'écran tactile LCD connectée à la carte BeagleBone Black, pour afficher la vidéo capturée par la webcam.



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

### Cours - Méthodes

- Comment mesurer le temps de démarrage
- Principales approches

### TP - Construction du système

- Téléchargement du code source du chargeur de démarrage, du noyau et de Buildroot
- Prise en main de la carte, mise en place de la communication série
- Configuration de Buildroot et génération du système
- Configuration et compilation du chargeur de démarrage U-Boot. Préparation d'une carte SD pour démarrer le système.
- Configuration et compilation du noyau. Démarrage du système.

## 1<sup>er</sup> Jour - Après-midi

### Cours - Mesure du temps

- Techniques génériques par logiciel
- Techniques matérielles
- Solutions spécifiques à chaque étage du démarrage

### TP - Mesure du temps - Solution logicielle

- Modification du système pour mesurer le temps au niveau des différentes étapes.
- Chronométrer les messages sur la console série
- Chronométrer le démarrage de l'application

### Cours - Optimisations des chaînes de compilation

- Introduction aux chaînes de compilation
- Bibliothèques C
- Informations de taille
- Mesure de la performance d'un exécutable avec la commande `time`



### TP - Optimisations des chaînes de compilation

- Mesure du temps d'exécution de l'application
- Passage à une chaîne Thumb2
- Génération d'un SDK Buildroot pour recompiler plus vite

## 2<sup>ème</sup> Jour - Matin

### Cours - Optimisation de l'application

- Utilisation de `strace` et `ltrace`
- Autres techniques de profiling

### TP - Optimisation de l'application

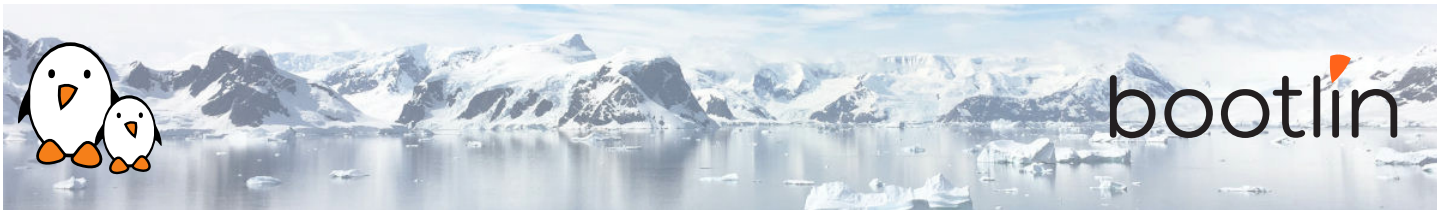
- Rechercher d'options de configuration inutiles dans des applications
- Modification de ces options à travers Buildroot
- Expériences avec `strace` pour suivre l'exécution d'un programme

### Cours - Optimisation du démarrage du système

- Utilisation de BusyBox `bootchartd`
- Optimisation des scripts d'init
- Possibilité de démarrer directement votre application

### TP - Optimisation du démarrage du système

- Utilisation de Buildroot pour supprimer scripts et commandes non nécessaires
- Une méthode pour identifier tous les fichiers inutilisés
- Simplification de BusyBox
- Démarrage de l'application en tant que programme init.



## 2<sup>ème</sup> Jour - Après-midi

### Cours - Optimisations de systèmes de fichiers

- Systèmes de fichiers disponibles, aspects de performance et de temps de démarrage
- Comment accélérer UBIFS
- Paramètres pour réduire le temps de démarrage
- Démarrer depuis un initramfs
- Utilisation d'exécutables statiques : contraintes de licence

### TP - Optimisations de systèmes de fichiers

- Essayer et mesurer deux systèmes de fichiers bloc : ext4 et SquashFS.
- Essai et benchmark de la solution initramfs. Contraintes en rapport avec cette solution.

### Cours - Optimisations du noyau

- Utilisation d'*Initcall debug* to générer un *boot graph*
- Options de compression et liées à la taille
- Réduction ou suppression de la sortie console
- Plusieurs réglages pour réduire le temps de démarrage

### TP - Optimisations du noyau

- Génération et analyse d'un *boot graph* pour le noyau
- Identifier et éliminer les fonctionnalités du noyau non nécessaires
- Trouver la meilleure option de compression pour votre système

## 3<sup>ème</sup> Jour - Matin

### TP - Optimisations du noyau

- Poursuite du TP





## 3<sup>ème</sup> Jour - Après-midi

### Cours - Optimisations du chargeur de démarrage

- Conseils génériques pour réduire la taille et le temps de démarrage d'U-Boot.
- Optimisation des scripts d'U-Boot et du chargement du noyau
- Sauter le chargeur de démarrage - Comment modifier U-Boot pour activer son *Falcon mode*

### Cours - Le *Falcon mode* d'U-Boot

- Principes et objectifs
- Prétraitement effectué par U-Boot pour préparer le démarrage de Linux
- Utilisation de la commande `spl export` pour faire ce traitement à l'avance.
- Modification du code source d'U-Boot et configuration pour démarrer directement Linux et sauter le deuxième étage d'U-Boot.
- Exemples and instructions de mise en oeuvre sur MMC et flash NAND
- Comment débbugger le Falcon mode
- Comment revenir à U-Boot
- Limitations

### TP - Optimisations du chargeur de démarrage

- Utilisation des techniques ci-dessus pour rendre le chargeur de démarrage le plus rapide possible
- Passer à un stockage plus rapide
- Sauter le chargeur de démarrage avec le *Falcon mode* d'U-Boot

### Conclusion - Résultats obtenus

- Partage et comparaison des résultats obtenus par les différents groupes
- Questions / réponses, partage d'expérience avec le formateur