

# Plateformes de génération d'un Linux embarqué

## I. Introduction

Il existe plusieurs plateformes qui permettent de construire une version linux embarqué adaptée aux besoins du système embarqué, parmi lesquels :OpenEmbedded, les outils du projet Yocto, Buildroot.

## II. Open Embedded [1]

OpenEmbedded est un cadre d'automatisation de construction et un environnement de compilation croisée utilisé pour créer des distributions Linux pour les périphériques embarqués. [2] Le framework OpenEmbedded est développé par la communauté OpenEmbedded, qui a été formellement établie en 2003. OpenEmbedded est le système de construction recommandé du Yocto Project , qui est un groupe de travail Linux Foundation qui assiste les entreprises commerciales dans le développement de systèmes basés sur Linux pour produits embarqués.

Le système de construction est basé sur des «recettes» BitBake , qui spécifient comment un paquet particulier est construit, mais comprend également des listes de dépendances et d'emplacements de code source, ainsi que des instructions sur la façon d'installer et de supprimer un paquet compilé. Les outils OpenEmbedded utilisent ces recettes pour récupérer et corriger le code source, compiler et lier des binaires, produire des packages binaires ( ipk , deb , rpm ) et créer des images amorçables.

Historiquement, la collection de recettes d'OpenEmbedded était stockée dans un référentiel unique, et les métadonnées étaient structurées sous une forme maintenant appelée "OpenEmbedded-Classic". En 2010, il était devenu de plus en plus difficile de gérer le nombre toujours croissant de recettes. Pour résoudre ce problème, les métadonnées de recette ont été divisées en plusieurs couches. La couche la plus basse, qui comprend des métadonnées indépendantes de la plate-forme et de la distribution, est appelée "OpenEmbedded-Core". Des instructions spécifiques à l'architecture, spécifiques à l'application et dépendantes de la distribution sont appliquées dans les couches de support cible appropriées qui peuvent remplacer ou compléter les instructions des couches inférieures. De plus, les modifications apportées aux recettes au niveau de la couche principale sont désormais gérées avec un modèle d'extraction : au lieu de valider leurs modifications directement dans le référentiel (comme c'était le cas auparavant), les développeurs envoient désormais leurs correctifs à la liste de diffusion. Les correctifs, s'ils sont approuvés, sont ensuite fusionnés (extraits) par un responsable.

Le framework OpenEmbedded peut être installé et mis à jour automatiquement via Git .

## III.Le projet Yocto [2]

Le projet Yocto est un projet open source collaboratif de la Linux Foundation dont le but est de produire des outils et des processus permettant la création de distributions Linux pour les logiciels embarqués et IoT indépendants de l'architecture sous-jacente du matériel embarqué. Le projet a été annoncé par la Linux Foundation en 2010 et lancé en mars 2011, en collaboration avec 22 organisations (Intel, TI, Xilinx , AMD, Mentor Graphics, Wind River, etc.) et OpenEmbedded .

L'objectif du projet Yocto est d'améliorer le processus de développement logiciel pour les distributions Linux embarquées . Le projet Yocto fournit des outils, des métadonnées et des processus interopérables qui permettent le développement rapide et reproductible de systèmes embarqués basés sur Linux dans lesquels chaque aspect du processus de développement peut être personnalisé. En octobre 2018, Arm Holdings s'est associé à Intel afin de partager du code pour les systèmes embarqués via le projet Yocto.

## IV. Buildroot

### IV.1. Qu'est ce que buildroot ? [1]

Buildroot est un ensemble de Makefiles et de correctifs qui simplifie et automatise le processus de création d'un environnement Linux complet et amorçable pour un système embarqué, tout en utilisant la compilation croisée pour permettre la construction de plusieurs plates-formes cibles sur un seul système de développement basé sur Linux. Buildroot peut créer automatiquement la chaîne d'outils de compilation croisée requise, créer un système de fichiers racine, compiler une image du noyau Linux et générer un démarrage pour le système embarqué ciblé, ou il peut effectuer toute combinaison indépendante de ces étapes. Par exemple, une chaîne d'outils de compilation croisée déjà installée peut être utilisée indépendamment, tandis que Buildroot crée uniquement le système de fichiers racine.

### IV.2. Principe de fonctionnement [4]

Buildroot est techniquement un ensemble de Makefiles définissant, en fonction des options paramétrées par l'utilisateur, la manière de compiler chaque paquet sélectionné avec des options particulières. Il construit au final une distribution complète et cohérente dont chaque composant a été compilé.

### IV.3. Compilation et configuration des «briques» principales [4]

Buildroot permet de compiler toutes les briques de base nécessaires pour obtenir une version Linux embarqué. Cela concerne quatre composants principaux :

- La toolchain
- Le bootloader
- Le noyau Linux
- Le rootfs

### IV.4. La toolchain [4]

La toolchain désigne l'ensemble des outils à compiler qui nous permettra ensuite d'avoir un environnement capable de cross compiler depuis notre architecture host (par exemple de l'x86\_64) vers l'architecture cible (par exemple ARM).

La toolchain regroupe un certain nombre de composants obligatoires comme :

- un compilateur
- un éditeur de lien (linker)
- un assembleur

Buildroot propose plusieurs mécanismes pour utiliser une toolchain. Le plus direct est d'utiliser la toolchain interne, et dans ce cas c'est Buildroot qui gèrera la création et l'utilisation de la toolchain.

Il est possible aussi d'utiliser un backend vers crosstool-ng qui est un outil spécifique pour faire des toolchains, disposant en particulier d'un nombre d'options de configuration plus important. En contrepartie certaines options ne sont pas directement prises en compte par buildroot, et il peut être nécessaire d'accéder à la configuration de crosstool-ng via la commande : `make ctng-menuconfig`

Notons qu'il est aussi possible d'utiliser une toolchain externe et donc déjà compilée, mais il n'y a du coup pas de possibilité pour buildroot de vérifier que les options de compilation sont cohérentes avec les composants qui seront ensuite sélectionnés.

## Références

[1] <https://en.wikipedia.org/wiki/OpenEmbedded>

[2] [https://en.wikipedia.org/wiki/Yocto\\_Project](https://en.wikipedia.org/wiki/Yocto_Project)

[3] <https://en.wikipedia.org/wiki/Buildroot>

[4] <https://www.linuxembedded.fr/2013/02/tutorial-un-systeme-linux-embarque-operationnel-avec-buildroot>

[5] [https://fr.wikipedia.org/wiki/Chargeur\\_d%27amor%C3%A7age](https://fr.wikipedia.org/wiki/Chargeur_d%27amor%C3%A7age)