INTRODUCTION

À

YOCTO



Écrit par

Anirudh THABJUL

Consultant Électronique

TABLE DES MATIÈRES

[Introduction À YOCTO 3](#_Toc525117808)

[QU’EST-CE QUE YOCTO ? 3](#_Toc525117809)

[POURQUOI L’YOCTO A ETE MIS EN PLACE ? 3](#_Toc525117810)

[LES AVANTAGES DU YOCTO 4](#_Toc525117811)

[LE HARWARE UTILISÉ 4](#_Toc525117812)

[LA PROCEDURE POUR CONSTRUIRE UNE IMAGE UTILISANT LE YOCTO 5](#_Toc525117813)

[CONFIGURER LE SYSTÈME HÔTE 5](#_Toc525117814)

[INSTALLATION DE L'UTILITÉ DE REPO 6](#_Toc525117815)

[ENLEVEZ LES PAQUETS INUTILES DE LA CONSTRUCTION D'IMAGE 6](#_Toc525117816)

# Introduction À YOCTO

## QU’EST-CE QUE YOCTO ?

Le Projet d'Yocto est un projet de collaboration open source qui aide des développeurs à créer un noyau personnalisé Linux pour des produits embarqués, indépendamment de l'architecture de Hardware (Par Exemple ARM, MIPS, PowerPC, and x86 etc ).

## POURQUOI L’YOCTO A ETE MIS EN PLACE ?

Expliquez en termes simples, Vous pouvez inclure ou exclure des composants créant un système qui répond à vos besoins précisément. Le système de fin peut démarrer juste votre application. Il peut prendre en charge vos composants hardware ou votre environnement de communication et puisque tout est open source, vous avez le contrôle total.

Expliquez en termes techniques, L'Yocto consiste en modèles, aussi appelés des « recettes », qui décrivent comment construire des composants, comme des bibliothèques et des utilités, qui sont incluses dans le système Linux embarqué. Ces recettes peuvent être groupées dans des « Couches » qui promeut la gestion de sortie dans des unités seules plutôt que maintenir qu'une série complexe des révisions de « build » pour chaque bibliothèque et outil. Les recettes et les couches sont regroupés en « images » qui sont des instantanés du noyau « Root File System » pour flasher sur un périphérique.

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

## LES AVANTAGES DU YOCTO

Voici quelques points saillants pour le projet Yocto

* Fournit un noyau Linux récent avec un ensemble de commandes de système et de bibliothèques appropriées pour l'environnement embarqué.
* Rend les composants du système tels que
  1. [X11](https://fr.wikipedia.org/wiki/X_Window_System)
  2. [GTK +](https://www.gtk.org/)
  3. [Qt](https://www.qt.io/)
  4. [Clutter](https://en.wikipedia.org/wiki/Clutter_(software)) etc ...

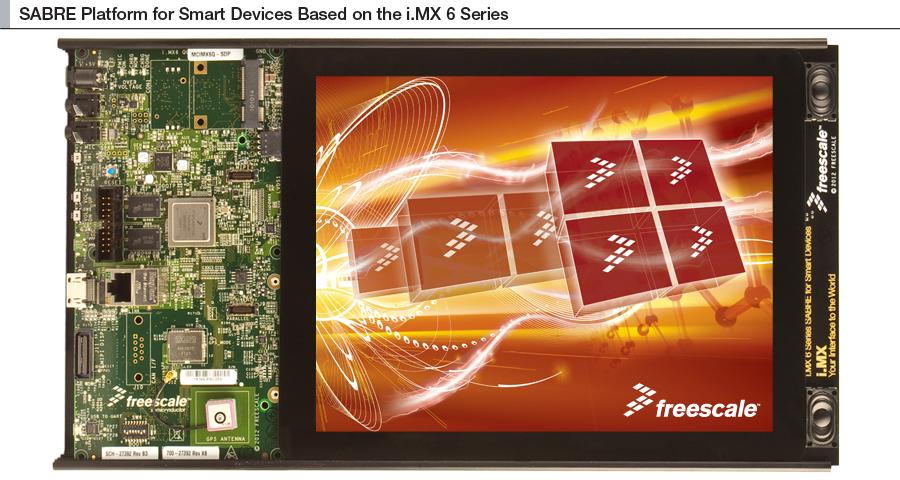
pour vous permettre de créer une expérience utilisateur enrichie sur les appareils dotés d'affichage. Pour les périphériques qui n'ont pas d'affichage ou où vous voulez utiliser l'alternative « UI framework », ces composants n'a pas besoin d'être installée.

* Crée un noyau concentré et stable compatible avec le projet "OpenEmbedded" avec lequel vous pouvez créer et développer facilement et de manière fiable.
* Prend entièrement en charge une large gamme d'émulation de hardware et de périphérique via « [Quick EMUlator](https://www.qemu.org/) (QEMU) ».
* Fournit un mécanisme de couche qui vous permet de facilement prolonger le système, faire des personnalisations et les garder organisés.

Consulter le lien suivant pour plus d'informations 🡪 [Yocto Project](https://www.yoctoproject.org/docs/2.2.2/mega-manual/mega-manual.html)

# LE HARWARE UTILISÉ

Nous avez pris prendre le kit d'évaluation [i.MX6DLSABRESD](https://www.nxp.com/support/developer-resources/evaluation-and-development-boards/sabre-development-system/sabre-platform-for-smart-devices-based-on-the-i.mx-6-series:RDIMX6SABREPLAT) qui utilise le processeur i.MX6 dual-lite comme la référence.



# LA PROCEDURE POUR CONSTRUIRE UNE IMAGE UTILISANT LE YOCTO

Par exemple, les exigences d'une application

|  |  |
| --- | --- |
| 1 | Le support Qt5 |
| 2 | Démarrage automatique de l'application basée sur Qt5,  Par exemple « l’application qui affiche ‘Hello World’ » |
| 3 | Le support nécessaire : USB, Touch, Ethernet, Jeu de vidéo sans son |
| 4 | Démarrez dès que possible |

Ceux-ci sont les étapes principaux pour construire l'image

1. Tout d'abord, configurez le système hôte
2. Installation de l'utilité de repo
3. Installation de git
4. Construire une image minimale par exemple « core-image-minimal »
5. Ajouter le support Qt5
6. Ecrire et intégrer une application Qt dans une construction d'image
7. Optimiser le temps de démarrage
   1. Enlevez les paquets inutiles de la construction d'image
   2. Supprimer les fonctionnalités inutiles de la construction d'image du noyau
   3. Faites des pilotes de hardware inutiles comme "Modules"
8. Construire l'image optimisée

## CONFIGURER LE SYSTÈME HÔTE

* On le recommande que fournissez au moins 120Gb la mémoire, qui est assez pour compiler n'importe quel Back-end (par exemple x11).
* Ceux-là sont les Distributions Linux supportées
  1. Ubuntu 14.10
  2. Ubuntu 15.04
  3. Ubuntu 15.10
  4. Ubuntu 16.04 (LTS)
  5. Fedora release 22
  6. Fedora release 23
  7. CentOS release 7.x
  8. Debian GNU/Linux 8.x (Jessie)
  9. Debian GNU/Linux 9.x (Stretch)
  10. OpenSUSE 13.2
  11. OpenSUSE 42.1
  12. Linux Mint
* Ci-dessous sont les paquets hôtes essentiels d'yocto prévoient que vous devez installer avant que vous ne commenciez à construire les images

$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev xterm sed cvs subversion coreutils texi2html docbook-utils python-pysqlite2 help2man make gcc g++ desktop-file-utils libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf automake groff curl lzop asciidoc uboot-mkimage u-boot-tools

## INSTALLATION DE L'UTILITÉ DE REPO

Repo est un outil construit sur « Git » qui fait plus facile de gérer les projets qui contiennent des dépôts multiples, qui ne doivent pas être sur le même serveur. Repo complète très bien la nature en couches du projet Yocto, facilitant ainsi l'ajout de ses propres couches au BSP.

Pour installer l'utilité « repo », exécutez ces étapes

1. Créez un dossier « bin » dans votre dossier « home »

$ cd ~ (This takes you to home directory)

$ mkdir ~/bin (this step may not be needed if the bin folder already exists)

$ curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo

$ chmod a+x ~/bin/repo

1. Ajoutez la ligne suivante au fichier « .bashrc » pour vous assurer que le dossier « ~/bin » se trouve dans votre variable PATH

$ export PATH=~/bin:$PATH

## INSTALLATION DE GIT

Exécutez ces étapes pour l'installation

$ sudo apt-get install git

$ git config --global user.name "Your Name"

$ git config --global user.email "Your Email"

$ git config –list

Consulter le lien suivant pour plus d'informations 🡪 [Git Installation](https://www.youtube.com/watch?v=xh32DIHApZA)

## CONSTRUIRE UNE IMAGE MINIMALE

Tout d'abord, nous avons besoin de construire une image avec des paquets minimum installés pour démarrer c'est-à-dire l’image « core-image-minimal ». exécutez ces étapes

1. Les couches du « Yocto Project" sont téléchargées dans le répertoire des sources. Cela configure les recettes utilisées pour construire le projet.

$ cd ~ (Si vous voulez télécharger dans le dossier « home »)

$ mkdir imx-yocto-bsp (Le nom du dossier peut également être modifié)

$ cd imx-yocto-bsp

$ repo init -u https://source.codeaurora.org/external/imx/fsl-arm-yocto-bsp -b imx-morty (consulter le lien hypertexte pour les nouvelles versions)

$ repo sync

1. Configurez le dossier « build » en entrant la commande suivante

$ MACHINE=<machine name> [DISTRO](https://en.wikipedia.org/wiki/Linux_distribution)=<distro name> source fsl-setup-release.sh -b <build dir>

Pour notre hardware c'est-à-dire « imx6dlsabresd »

$ MACHINE=imx6dlsabresd DISTRO=fsl-imx-x11 source ./fsl-setup-release.sh -b build

## ENLEVEZ LES PAQUETS INUTILES DE LA CONSTRUCTION D'IMAGE

Quand vous construisez l'image fournie par yocto, nous obtenons tant de paquets installés par défaut mais dans beaucoup d'applications, nous n'avons pas besoin de tous ces paquets. Ainsi dans ces cas, Nous pouvons enlever certains de paquets inutiles pour réduire la taille de « Rootfs ».