



**Université Paris 8**

**"Construction d'un noyau Linux pour l'embarqué"**

Réalisé par :

Rajmagan BALAKICHENIN  
Kevin TOLEON  
Noureddine DJERROUD

**Année universitaire: 2018-2019**

# SOMMAIRE

|   |   |
|---|---|
| I. Introduction.....                            | 2 |
| II. Étapes de construction du projet.....       | 3 |
| II.1 Acquisition des outils : .....             | 3 |
| II.2 Ajout d'un module.....                     | 3 |
| II.3 Manières de configurer le noyau.....       | 4 |
| II.4 Buildroot.....                             | 4 |
| II.5 Configuration noyau.....                   | 4 |
| III. Démarrer buildroot sous Raspberry Pi3..... | 5 |
| V. Travail non effectué : .....                 | 6 |



# I. Introduction

Notre projet consiste à pactiser un module de notre choix dans une distribution buildroot. Le choix du module et de la distribution buildroot est un n'est pas imposé, c'est à nous de choisir. Le projet doit est opération sur une Raspberry Pi3 dès son démarrage. La date limite du projet était pour le 06/12/2018 à 7 H, cependant nous avons eu le droit à une semaine de plus, pour terminer notre projet. Dans ce rapport, nous allons expliquer les étapes du projet, de l'installation d'un module jusqu'au démarrage du projet sur la raspberry Pi3. Mais aussi détailler sur les problèmes rencontrés, et pourquoi il y a eu ces problèmes.

## II. Étapes de construction du projet

### II.1 Acquisition des outils :

Pour réaliser ce projet nous devons tous d'abord télécharger un buildroot. Un buildroot est un outil qui permet d'automatiser la construction complète d'un système Linux embarqué. Nous avons donc téléchargé un buildroot qui contenait déjà tous les fichiers de base dans les dossiers boot, kernel, ... Pour utiliser le buildroot, il faut télécharger les outils et les dépendances qui le permettent de l'utiliser comme les dépendances de 'make', 'make config', que nous verrons plus tard.

Les commandes utilisées sont :

- apt-get install make
- apt-get install bison puis apt-get install flex. C'est deux dépendances sont utilisées pour la configuration du noyau.
- apt-get install libncurses5-dev. Pour utiliser make menuconfig que nous verrons aussi plus tard.
- Nous devons aussi acquérir une Raspberry Pi3 avec une carte SD, une Raspberry est un micro-ordinateur fonctionnant sur un processeur ARM.

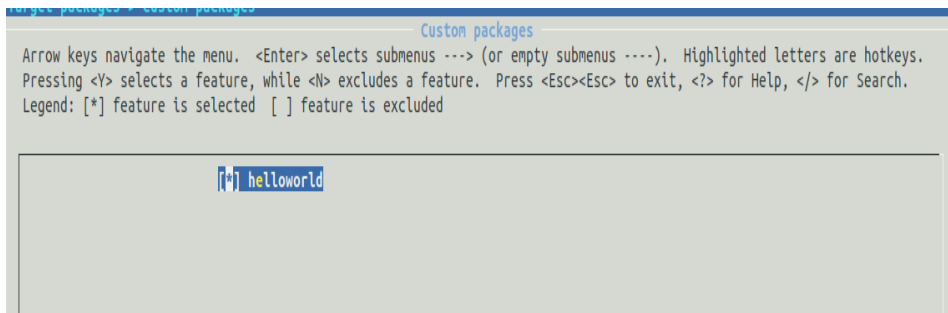
### II.2 Ajout d'un module

Notre module sera téléchargé sur internet, comme notre but était simplement de l'implémenter sur une Raspberry. Notre module sera un programme qui contiendra uniquement un hello world. Nous avons installé le module dossier helloworld que nous avons créé dans le dossier package. C'est ce dossier packagé que contiennent tous les modules que nous souhaitons. Pour connaître le chemin de nos modules, il faut les indiquer dans ce fichier: package/config.in, comme ceci :

```
menu "Custom packages"
    source "package/helloworld/Config.in"
endmenu
```

Pour vérifier que les modules soient reconnus lors du noyau, il faut l'activer dans la configuration du noyau.

Donc il faut faire : `make menuconfig`, aller dans `target` et activer les modules.



L'image ci-dessus prouve que notre module `helloworld` s'est bien attaché à notre noyau.

## II.3 Manières de configurer le noyau

Pour pouvoir utiliser notre buildroot afin de la booter sur notre Raspberry Pi3, nous avons choisi de la configurer par défaut avec cette commande : `make raspberrypi_defconfig`.

“\_defconfig” qui permet de laisser la configuration par défaut de l'ensemble des makefiles. Ensuite il faut configurer notre noyau et il y a plusieurs commandes pour configurer notre noyau, `make config`, `make menuconfig`, ...

Nous avons choisi la commande : `make menuconfig`, car nous voulions activer une option du noyau. De plus, `make menuconfig`, nous offre une interface graphique, qui rend plus agréable les configurations. Mais surtout elle permettra aussi de pouvoir compiler le dossier buildroot pour une architecture destinée à la Raspberry Pi3 (ARM).

## II.4 Buildroot

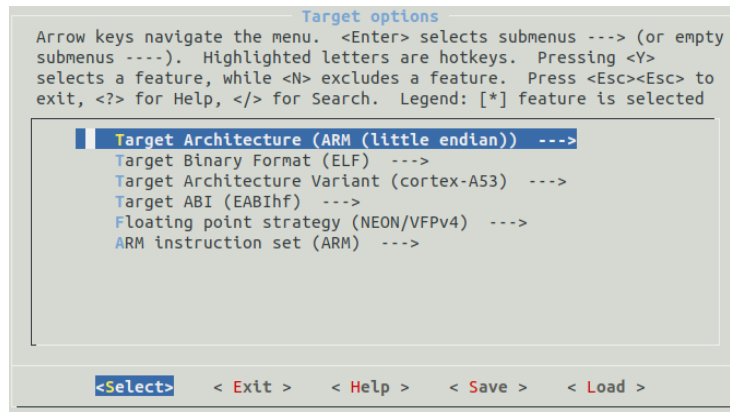
Buildroot est techniquement un ensemble de Makefiles définissant, en fonction des options paramétrées par l'utilisateur, la manière de compiler chaque paquet sélectionné avec des options particulières. Il construit finalement une distribution complète et cohérente dont chaque composant a été compilé. Il possède un outil confortable de configuration, basé et très similaire à celui du noyau Linux: `menuconfig` et qui peut être utilisé dans tout projet.

## II.5 Configuration noyau

Tout ceci sera compilé avec une cross-compilation avec la configuration adaptée à notre système. Grâce à une toolchain qui est un ensemble de paquet utilisé lors d'une compilation du noyau que nous avons configuré au préalable avec le `make menuconfig`, cela a pour effet de créer un fichier `.config` dans le répertoire actuel. Ce fichier répertorie la configuration de la cible. Il peut être transmis à un tiers qui pourra ainsi construire une toolchain identique. La compilation est lancée avec la commande `build`.

L'ajout d'un module se fait avant la configuration du noyau. Il peut être chargé dynamiquement sans avoir besoin de recompiler le noyau (avec la commande `insmod` ou `modprobe`) ou de redémarrer le système.

Cependant, il ne faut de dire au noyau sur quel architecture devra démarrer notre système :



Une fois que tout est configuré, l'outil de configuration génère un fichier .config qui contient l'ensemble de la configuration. Effectue un make qui démarrera la compilation

La commande make effectuée généralement les étapes suivantes :

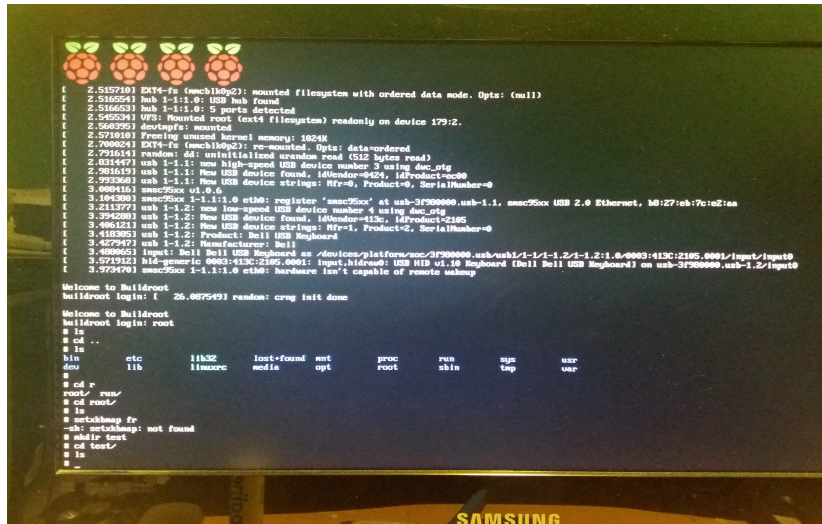
- téléchargement des fichiers sources
- configuration de la cross-compilation
- correction et installation des paquets cibles
- construction d'une image du noyau
- construction d'une image du bootloader
- création d'un système de fichier racine nommé

Le résultat de la construction est stocké dans un répertoire qui contiendra plusieurs sous répertoires :

- images/: toutes les images (bootloader, noyau, système de fichier racine)
- build/ : tous les composants sont construits ici avec leurs sous répertoires respectifs.
- Staging/ : contient la hiérarchie des fichiers système de la racine : les entêtes et les bibliothèques de la chaîne de compilation.
- target/ : contient le système de fichiers rootfs
- host/ : contient l'installation des outils compilés pour l'hôte qui sont nécessaires pour la bonne exécution de Buildroot, y compris la chaîne d'outils de compilation croisée

### III. Démarrer buildroot sous Raspberry Pi3

Après la compilation, il y a un dossier important qui aura été créé qui est output, qui contiendra le dossier image, ou il y aura les fichiers les plus importants. Mais surtout il contiendra le noyau « zImage » et un autre fichier « sdcard.img » qui faudra le copier dans la carte Raspberry. La commande utilisée : `sudo cp output/images/sdcard.img /media/finalubu/sdb.`



L'image ci-dessous, nous montre le résultat de notre projet. Comme nous pouvons le voir, il y a c'est bien le buildroot qui démarre.

#### IV. Problème survenue

Cependant, il y a un problème liée à l'installation du noyau.

Puisque quand lsmod est lancé, cette commande devrait afficher la liste des modules installé sur notre buildroot. Mais rien ne s'affiche, cela veut dire qu'il n'y a pas de module. Donc cela veut dire, qu'il fallait utilisé make\_modules, puis make\_modules\_install, cependant nous avons pu le voir sur le make menuconfig, que le module helloworld est associé au noyau.

Après la compilation de la commande make, il y a un dossier module qui s'y trouve et à l'intérieur de celui-ci devrait avoir un autre dossier qui se nomme module, cependant il est absent. C'est dans ce dossier que devait contenir tous les modules installé.

#### V. Travail non effectué :

- Impossible de vérifier la présence du module lors du démarrage du buildroot

# Références Bibliographiques

<http://langevin.univ-tln.fr/CDE/LEXYACC/Lex-Yacc.html>

<http://www.linuxembedded.fr/2011/03/ajouter-un-package-dans-buildroot-en-5-minutes/>

<http://silanus.fr/sin/wp-content/uploads/2016/03/TP3.pdf>

<https://buildroot.org/downloads/manual/manual.html>

<https://www.blaess.fr/christophe/2014/07/22/ajouter-un-module-noyau-personnel-dans-buildroot/>

<https://www.youtube.com/watch?v=84T-P5ye864>

[https://doc.ubuntu-fr.org/tutoriel/tout\\_savoir\\_sur\\_les\\_modules\\_linux](https://doc.ubuntu-fr.org/tutoriel/tout_savoir_sur_les_modules_linux)