TOLEON kevin

BALAKICHENIN Ramagan

DJERROUD Nourredine

**Rapport Linux Embarquée**

L’objectif du projet et de packetiser des moules dans une distribution buildroot qui doit tourner sur un Raspberry Pi3.

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 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” permet de laisser la configuration par défauts de l’ensemble des makefiles.   
Ensuite il faut configurer notre noyau, pour cela nous avons utilisé : make menuconfig.  
Il y a plusieurs commandes pour configurer notre noyau, make config, make menuconfig, make xconfig, …

Configuration noyau

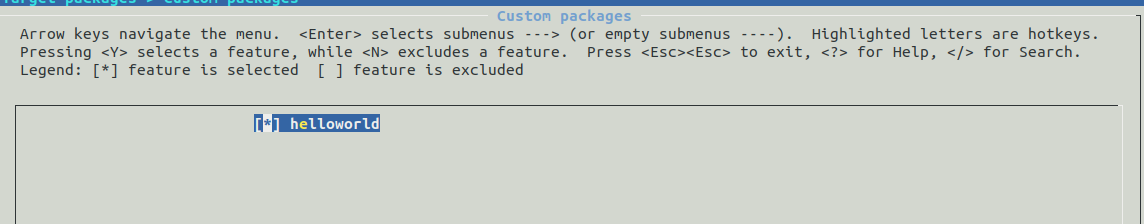
Tous ceci sera compiler avec une cross-compilation avec la configuration adaptée à notre système. Grace a une toolchain que nous avons configurer au préalable avec le make menuconfig, cela a pour effet de créer un fichier **.config** dans le répertoire actuel. Ce fichier répertorie la configuration de 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. Il peut parfois être utile d'empêcher le noyau de charger un module automatiquement avec la commande blacklist nom-du-module.

Partie module

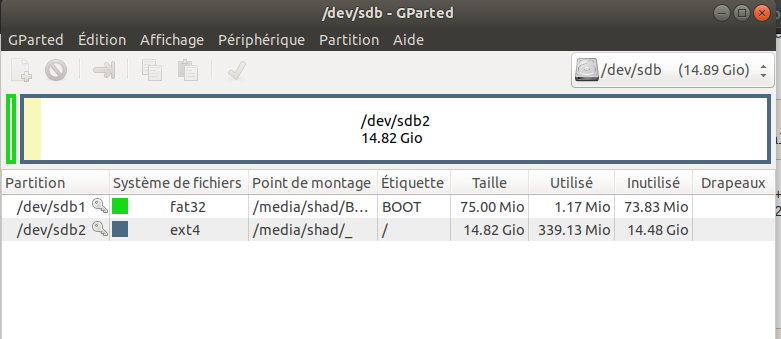
Nous avons installé le module souhaité dans un dossier helloword dans le dossier package.  
 Dans ce dossier package qui contient les modules, il a tous les modules que nous souhaitons et pour faire reconnaitre notre module, il faut faire connaitre le chemin du module avec dans le fichier package/Config.in.

Pour vérifier que les modules soit reconnu lors du noyau, il faut activer ce module dans la configuration du noyau. Donc il faut faire make menuconfig, aller dans target et activer les modules.



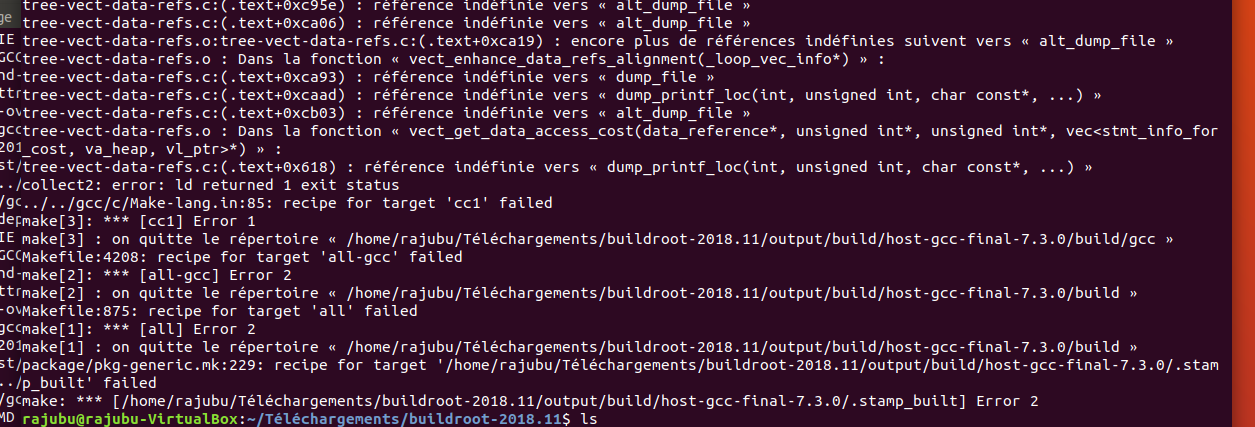
Partitionnement de la carte SD

Nous avons partitionné la carte come ceci en respectant l’espace conseille indiquer sur les tutos du web. La partition fat 32 devais recevoir les fichiers ZImage qui est l’image du noyau sur lequel la raspberry devra booter et recevoir aussi le dossier rpi-firmware qui contient tous les fichier nécessaire au boot de la carte. La partition ext4 contient les dossier rootfs.tar qui contient la racine et tous les fichier de bases.



Problème survenue

Les problèmes rencontrés : le noyau ses greffé à coté de mon noyau. Nous avons téléchargé toutes les dépendances nécessaires.



Nous avons rencontré des problèmes avec la compilation du noyau les chemins spécifier aboutissent à des erreurs de compilation car certaines références sont indéfinies ce qui produit des erreurs comme Dump-files.

Nous avons pensé qu’il fallait mettre à jours les dépendances cependant le problème n’est pas lié à la version.