



Créer un projet avec BuildRoot



## Présentation

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement



## BuildRoot 2018 05

- Initialement pour le projet µClibc
- Existe depuis 2005
- Première version officielle en 2009
- Une version officielle tous les 3 mois
- Maintenu par Peter Korsgaard et Thomas Petazzoni
- Gestion sur GIT avec branches de version
  - Plus de 2000 paquets sources intégrés
  - Une vingtaine d'architectures supportées
  - Près de 180 cartes pré-configurées
  - 3 bibliothèques C disponibles
  - Gestion de la configuration jusqu'à la mise à jour de la carte



# Objectifs d'un projet

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Les contraintes d'un projet industriel

Point de vue d'un intégrateur

- ✚ Gestion et stockage des évolutions du projet uniquement.
- ✚ Reproductibilité du projet des années après la mise en service.
- ✚ Mise à jour rapide des évolutions de Buildroot pour une meilleure maintenance des corrections de sécurité.





Le projet selon BuildRoot



# Vue générale

- Présentation
- Objectifs d'un projet
- **Vue générale**
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Buildroot

arch  
board  
boot  
configs  
docs  
fs

linux  
package  
support  
system  
toolchain  
utils

## Projet

board  
configs  
overlays  
patches  
package

- **Le projet doit être indépendant du système Buildroot.**
- **Le projet ne doit contenir que :**
  - ❖ Les spécificités de la plate-forme (matériel spécifique, utilisateurs, configuration...).
  - ❖ Les outils de génération des applications propriétaires.



# Installation de BuildRoot

1/3

- Présentation
- Objectifs d'un projet
- Vue générale
- **Installation de BuildRoot**
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Buildroot

arch  
board  
boot  
configs  
docs  
fs      linux  
        package  
        support  
        system  
        toolchain  
        utils

### Téléchargement de BuildRoot

Depuis le site officiel

```
[~]$ wget https://buildroot.org/downloads/buildroot-2018.05.tar.bz2  
[~]$ tar -xjf buildroot-2018.05.tar.bz2
```

### Installer BuildRoot pour tous vos projets

Installation depuis le dépôt GIT

```
[~]$ su -  
[~]# cd /opt  
[opt]# git clone git://git.buildroot.net/buildroot  
[opt]# cd buildroot  
[buildroot]# mkdir dl  
[buildroot]# chmod a+w ./dl
```

Un répertoire pour tous vos projets, et un répertoire de téléchargement accessible en écriture.



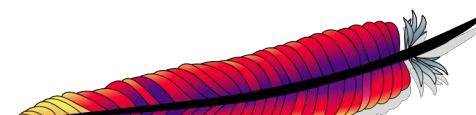
# Installation de BuildRoot

2/3

- Présentation
- Objectifs d'un projet
- Vue générale
- **Installation de BuildRoot**
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Proposer un site HTTP avec les paquets sources

Configurer un répertoire commun aux téléchargements et provisionner un serveur de fichiers pour les prochains projets.



**Apache Web Server**



**NGINX**



Il est bon d'avoir les données d'un projet sur un serveur même 5 voir 10 ans plus tard.

## Un répertoire par paquet

Toutes les versions dans un répertoire



# Installation de BuildRoot

3/3

- Présentation
- Objectifs d'un projet
- Vue générale
- **Installation de BuildRoot**
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Installer les outils nécessaires à BuildRoot

Selon le poste de travail il est nécessaire d'installer différents outils :

- ❖ make
- ❖ gcc
- ❖ g++
- ❖ openssl-dev
- ❖ ncurses-dev

Il est préférable que le répertoire de travail suive certaines règles :

- ❖ pas d'espace dans les chemins.
- ❖ le système de fichiers accepte les liens directs et symboliques.



# Création du projet

## 1/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- **Création du projet**
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Description du projet

dans le fichier *external.desc*

```
name: PROJECT
```

```
~  
~  
~
```

```
"external.desc"    1L,   14C      1,13
```



le nom du projet est repris dans des noms de variables.  
Il peut être en minuscule, mais c'est déconseillé.

Attention, à la syntaxe et  
particulièrement aux espaces



## Projet

board  
configs  
overlays  
patches  
package

**Config.in**  
**external.desc**  
**external.mk**

## Configuration du projet

dans le fichier *configs/project\_defconfig*

```
BR2_DEFCONFIG="$(BR2_EXTERNAL_PROJECT_PATH)/  
>>configs/project_defconfig"  
BR2_DL_DIR="/opt/buildroot/dl"  
BR2_ARCH="arm"  
BR2_arm=y  
BR2_cortex_a53=y
```

```
~
```

```
"configs/project_defconfig"    1L,   14C      1,13
```



Il est préférable d'utiliser la configuration d'une plate-forme en la copiant depuis l'environnement BuildRoot.



# Création du projet

## 2/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- **Création du projet**
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Préparation de l'environnement :

Création du répertoire de sortie avec toute l'arborescence du projet

```
[project]$ make O=${PWD}/output BR2_EXTERNAL=${PWD} -C /opt/buildroot project_defconfig
```



Il est préférable de définir la variable O avec le chemin d'un répertoire de sortie.



L'usage d'espace dans les noms de répertoire peut poser des problèmes.

### Une arborescence du répertoire de sortie

- ❖ **build** : construction des paquets
- ❖ **host** : outils de construction (toolchain)
- ❖ **images** : sortie de la distribution
- ❖ **staging** : sysroot
- ❖ **target** : rootfs (presque)



## Vérification du projet :

Vérification du fichier *output/.config*

```
BR2_DEFCONFIG="$HOME/project/configs/project_defconfig"
BR2_DL_DIR="/opt/buildroot/dl"
BR2_PRIMARY_SITE="http://intra.company.com/buildroot"
```

~  
~  
~

..config"

1L, 14C

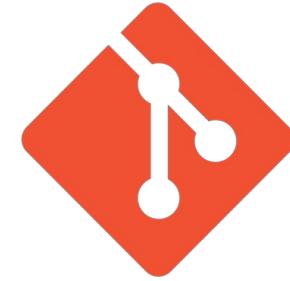
1,13



# Création du projet

3/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- **Création du projet**
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement



## Sauvegarder et partager son projet :

L'utilisation d'un gestionnaire de configuration tel que GIT permet dès cette étape de partager son travail avec les autres collaborateurs.

```
[project]$ git init  
[project]$ git add external.desc  
[project]$ git add external.mk  
[project]$ git add Config.in  
[project]$ git add configs/project_defconfig  
[project]$ git commit  
[project]$ git remote add origin https://git.company.com/project  
[project]$ git push origin master
```



Réaliser cette étape le plus tôt permet de bien identifier les étapes de construction.



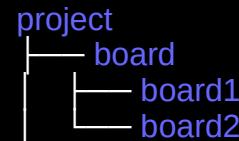
# Gestion du projet

1/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- **Gestion du projet**
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Rajouter une arborescence :

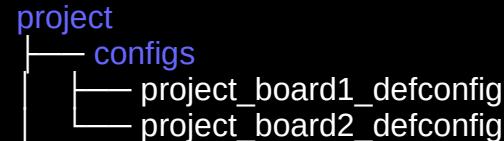
Chaque répertoire d'un projet a une fonction propre



**board** pour tous les scripts spécifiques à la carte



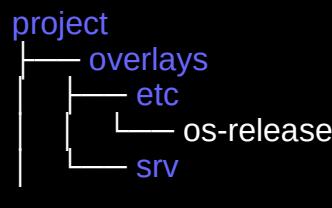
Penser à la génération de l'image, ajouter des partitions...



**configs** pour les fichiers de description du projet



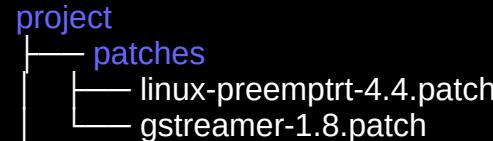
Penser à auto-référencer ce fichier dans lui-même



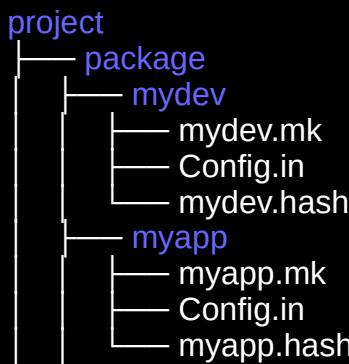
**overlays** pour les fichiers statiques à installer par défaut.



Il est possible d'installer des fichiers dans overlays depuis un package, si le produit de l'image doit être réutilisable.



**patches** pour les fichiers à appliquer sur les sources d'un paquet.



**package** pour les scripts de génération de vos développements.



Il est possible de créer un paquet pour des applications déjà prêtes depuis BuildRoot mais dans une autre version.



# Gestion du projet

## 2/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- **Gestion du projet**
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Transformer son projet :

### Utilisation des outils de configuration de BuildRoot

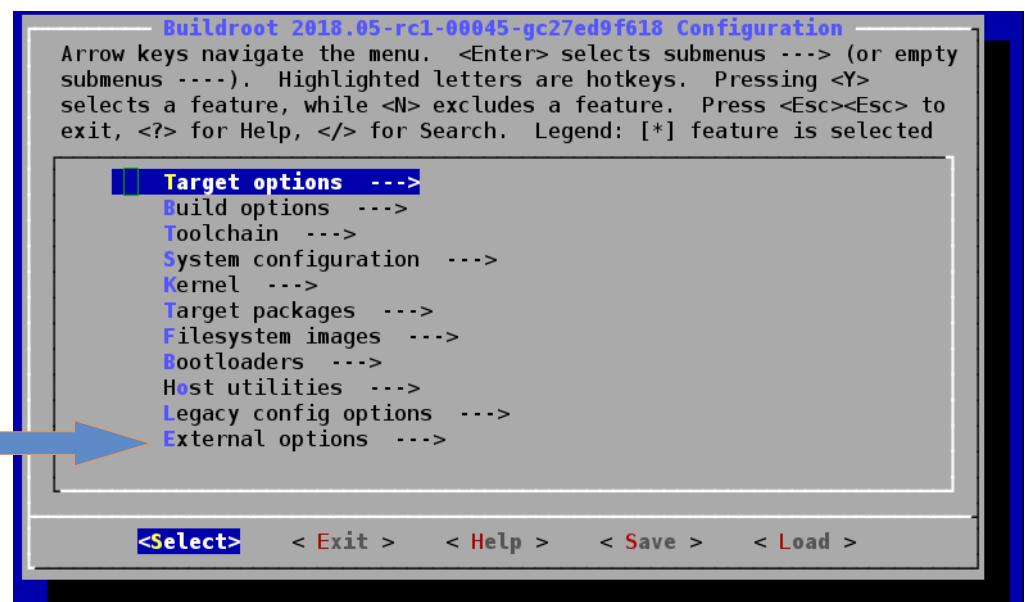
*external.mk*

```
include $(sort $(wildcard ${BR2_EXTERNAL_PROJECT_PATH}  
>>/package/*/*.mk))  
~  
"external.mk" 1L, 14C 1,13
```

```
source "${BR2_EXTERNAL_PROJECT_PATH}/package/my_app/Config.in"  
source "${BR2_EXTERNAL_PROJECT_PATH}/package/my_dev/Config.in"  
~  
"Config.in" 1L, 14C 1,13
```

*Config.in*

```
[project]$ cd output  
[output]$ make menuconfig
```

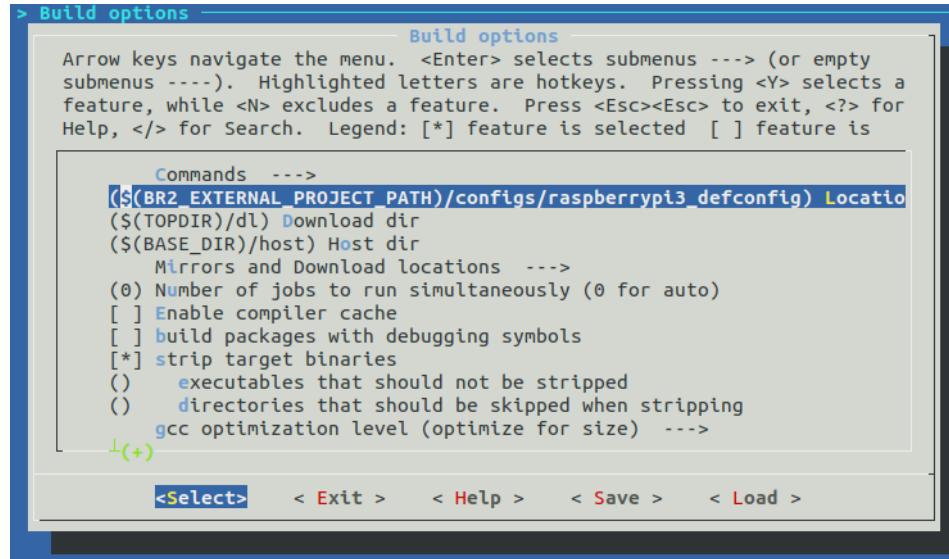


# Gestion du projet

3/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- **Gestion du projet**
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Sauvegarder son projet :



- .config contient toute la configuration.
- project\_defconfig contient le minimum pour reconstruire .config



Permet de réutiliser une configuration avec une version postérieure de Buildroot

### Régénération de config/project\_defconfig

```
[output]$ make savedefconfig  
[output]$ cd ..  
[project]$ git add -p configs/project_defconfigs  
[project]$ git commit
```





## La configuration BuildRoot

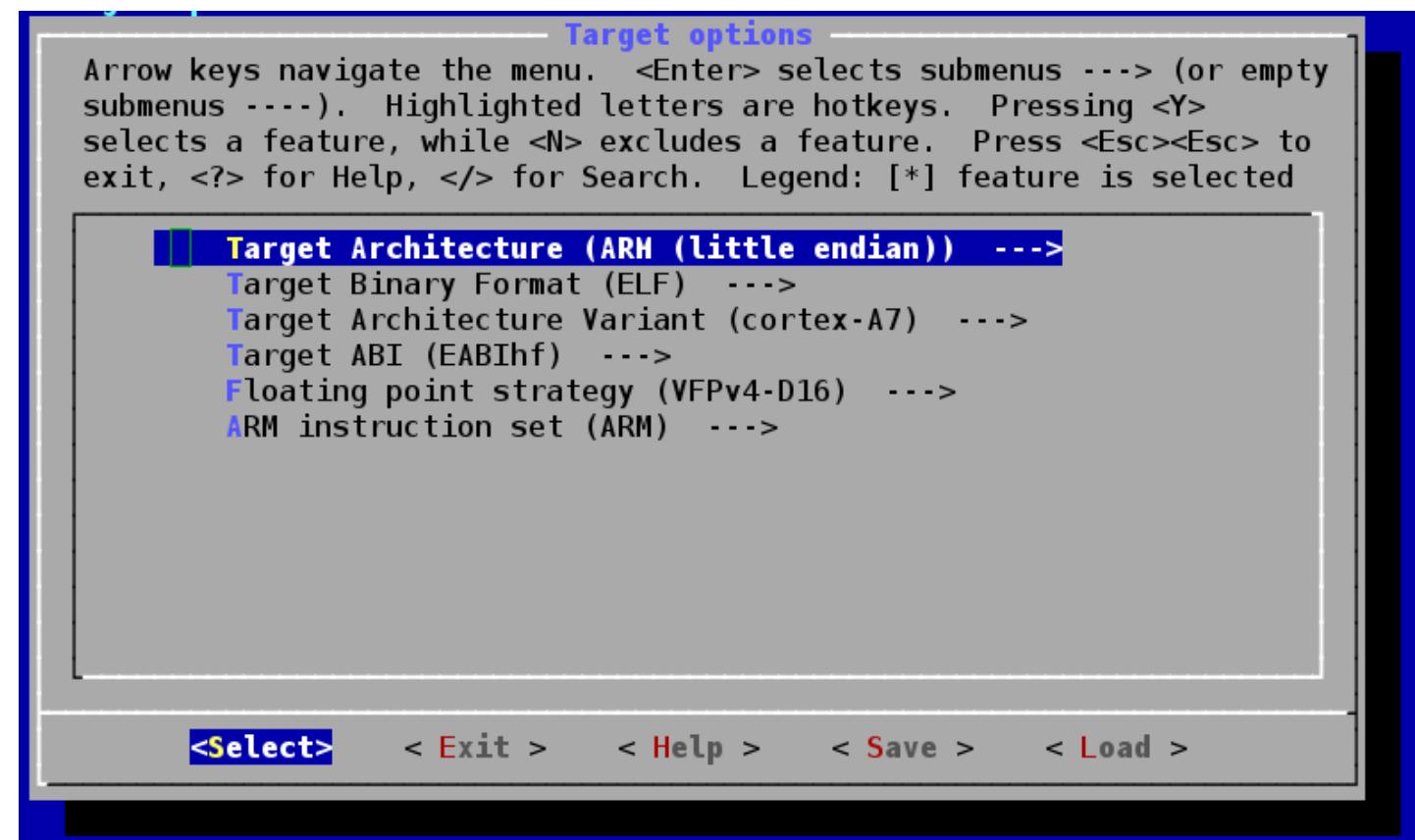


# Inspection des menus

1/10

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- **Inspection des menus**
- Projet global
- Intégration des applications
- Développement

## Target options :

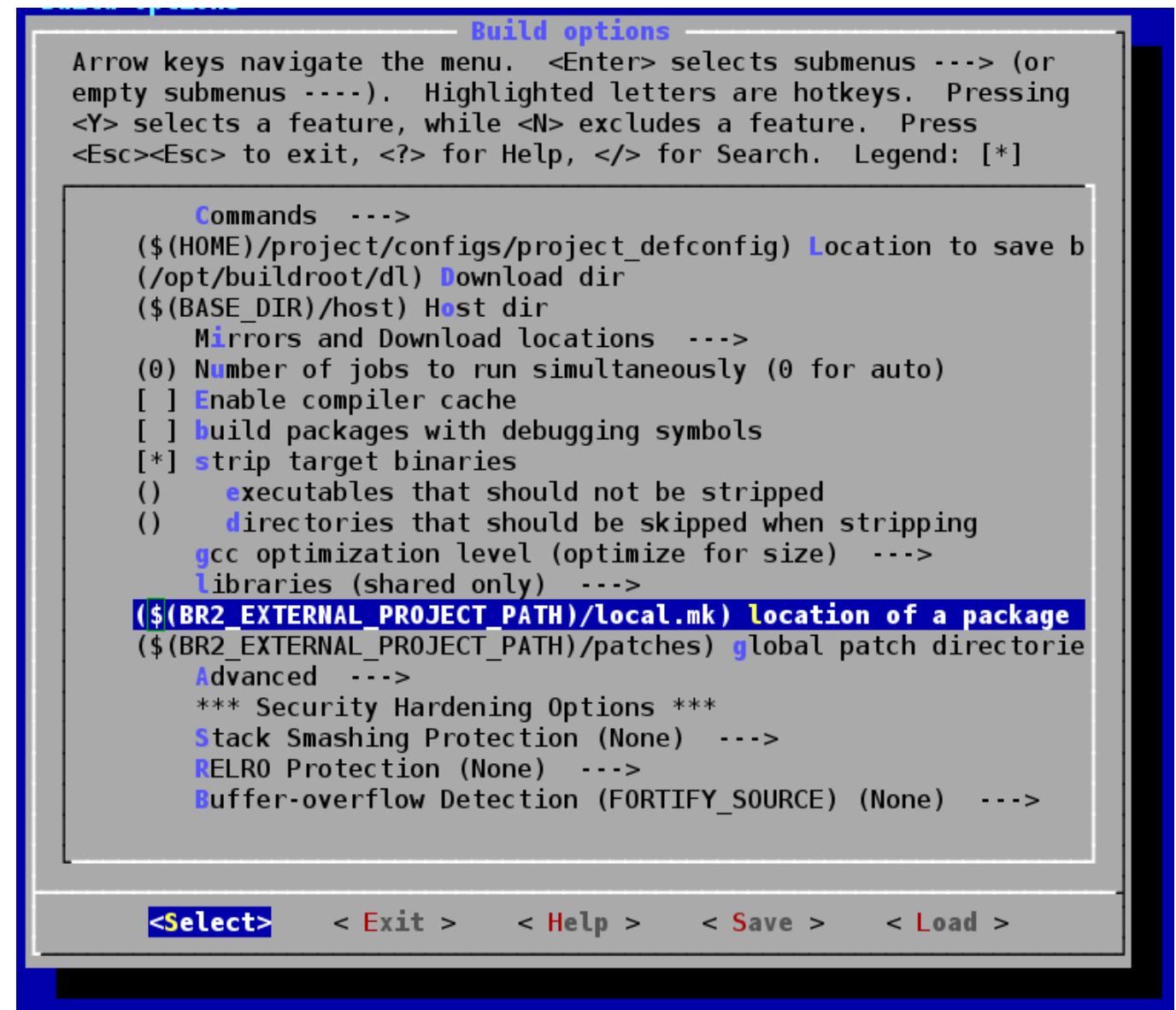


# Inspection des menus

## 2/10

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- **Inspection des menus**
- Projet global
- Intégration des applications
- Développement

### Build options :



Utilisation de la variable **BR2\_EXTERNAL\_<project name>\_PATH** contenant le chemin réel du fichier *external.desc*

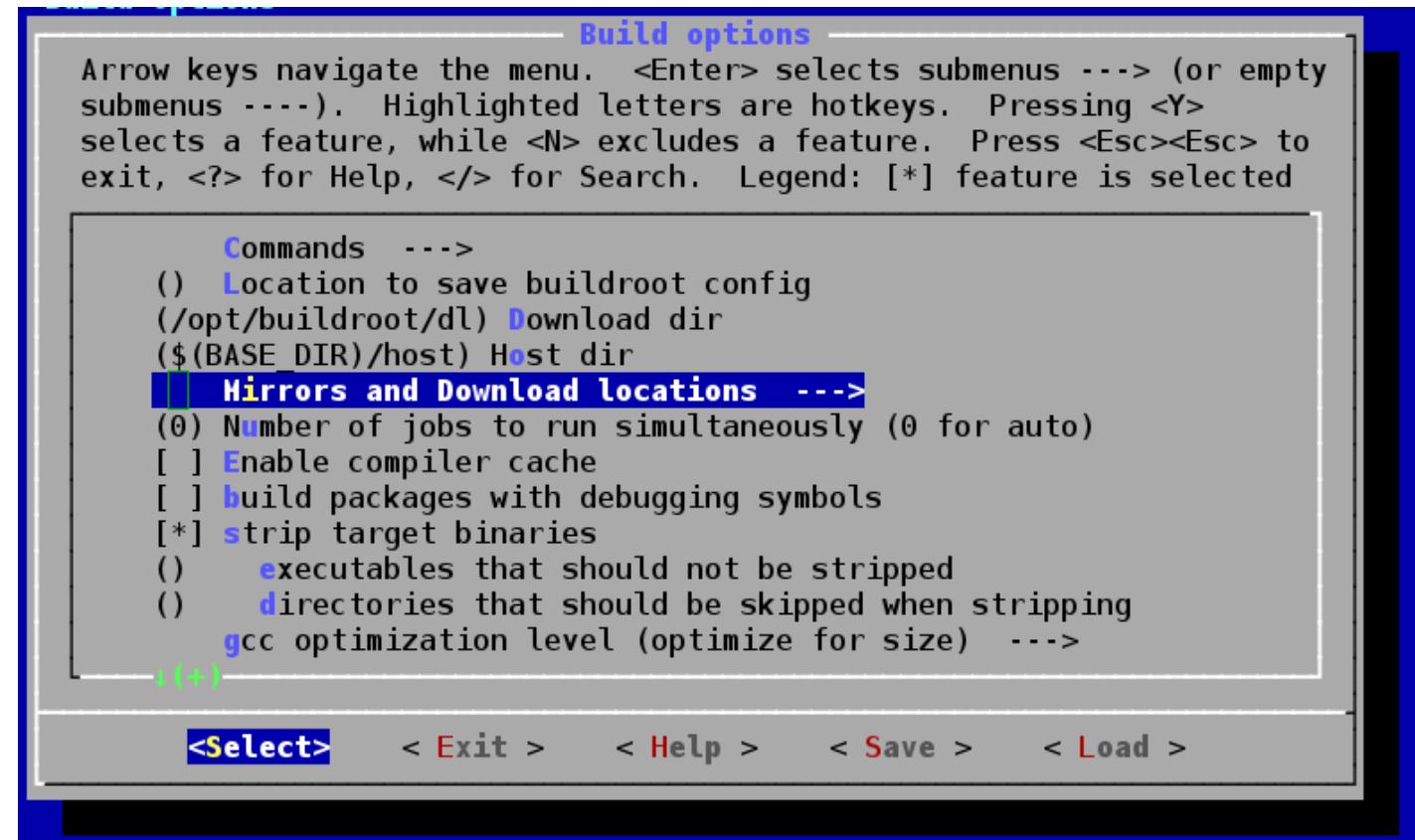


# Inspection des menus

3/10

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- **Inspection des menus**
- Projet global
- Intégration des applications
- Développement

Chercher les sources sur un serveur local.

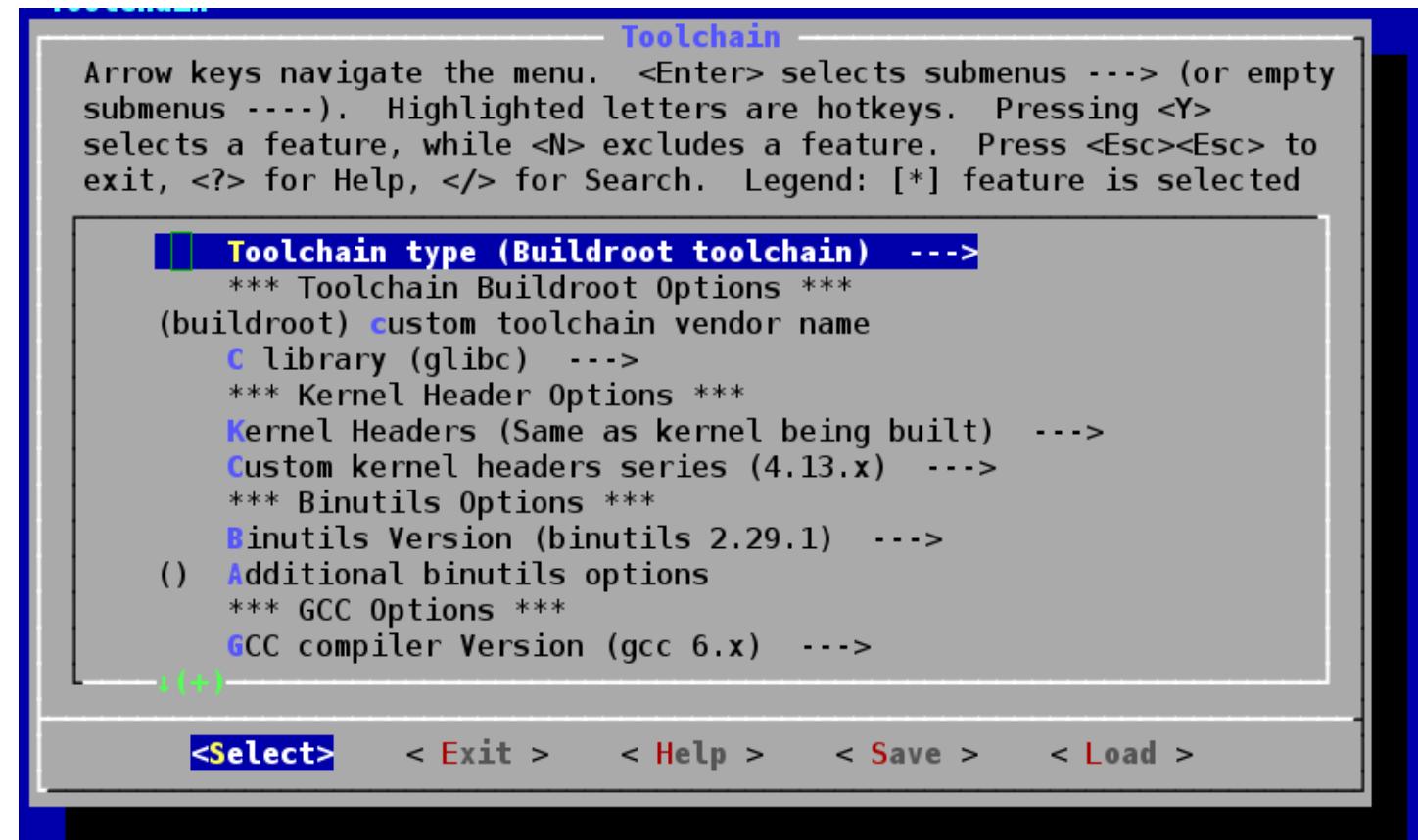


# Inspection des menus

3/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Toolchain :

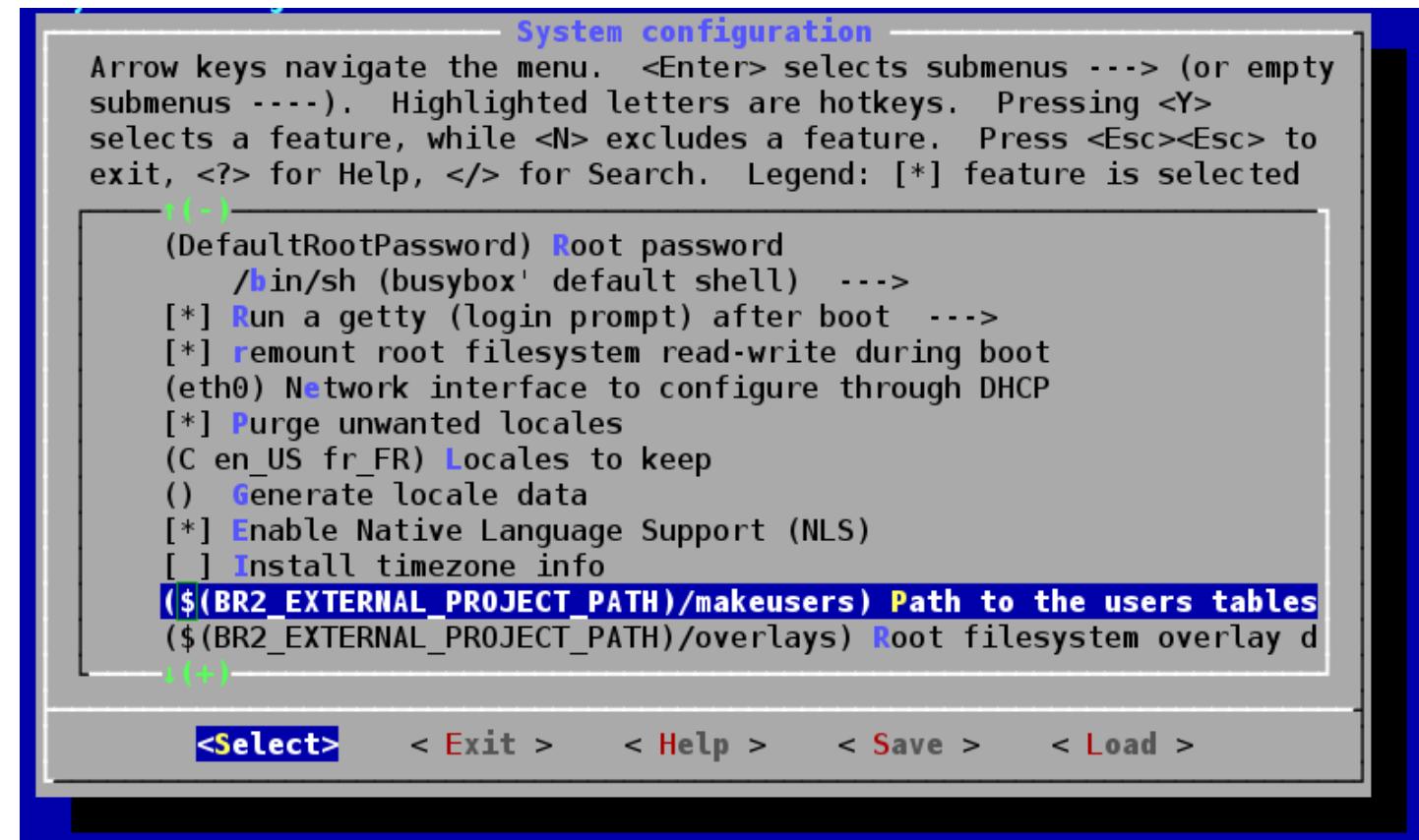


# Inspection des menus

4/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## System configuration :



Configurer le mot de passe de l'utilisateur *root* du système.

Overlay : un squelette de l'arborescence du système.



# Inspection des menus

5/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Kernel :

```
Kernel
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N> excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected
r (-)
(4.13.7) Kernel version
() Custom kernel patches
    Kernel configuration (Using a custom (def)config file) ...
    ($(BR2_EXTERNAL_PROJECT_PATH)/board/kernel_defconfig) Configuration
    ($(BR2_EXTERNAL_PROJECT_PATH)/kernel.fragment.config) Additional
        Kernel binary format (zImage) --->
        Kernel compression format (gzip compression) --->
[*] Build a Device Tree Blob (DTB)
() In-tree Device Tree Source file names
[$(BR2_EXTERNAL_PROJECT_PATH)/board/kernel.dts) Out-of-tree Device Tree Source file names
[ ] Install kernel image to /boot in target
[ ] Needs host OpenSSL
s (+)

<Select> < Exit > < Help > < Save > < Load >
```

Configuration des *patches* et extensions du kernel tel que *preempt-rt* et *xenomai*

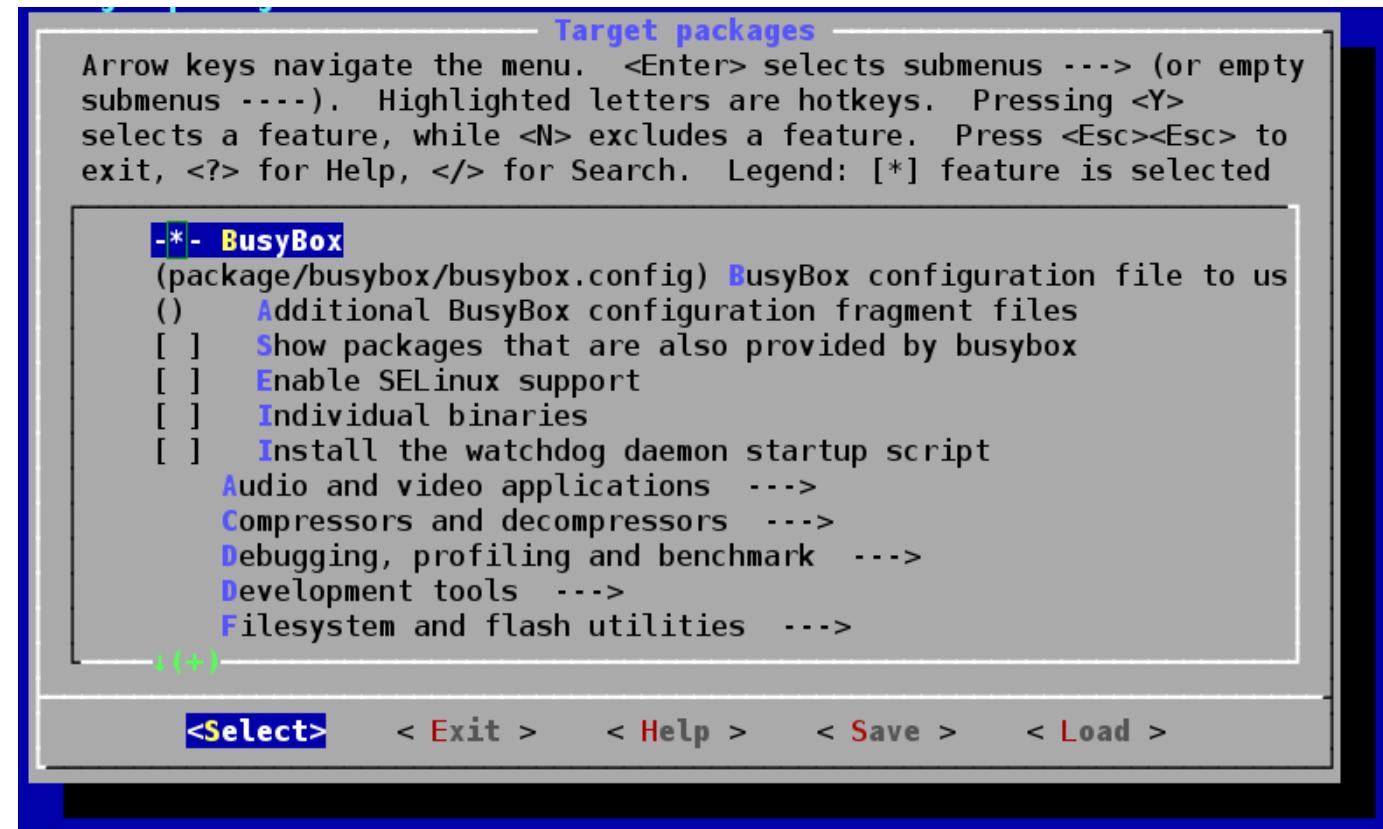


# Inspection des menus

6/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Target packages :



Des milliers (plus de 2000) paquets pouvant être intégrés, mais aussi un système fonctionnel avec une dizaine d'entre eux.

Des paquets en Open-Source mais aussi des paquets propriétaires comme les bibliothèques OpenGL des certains constructeurs.

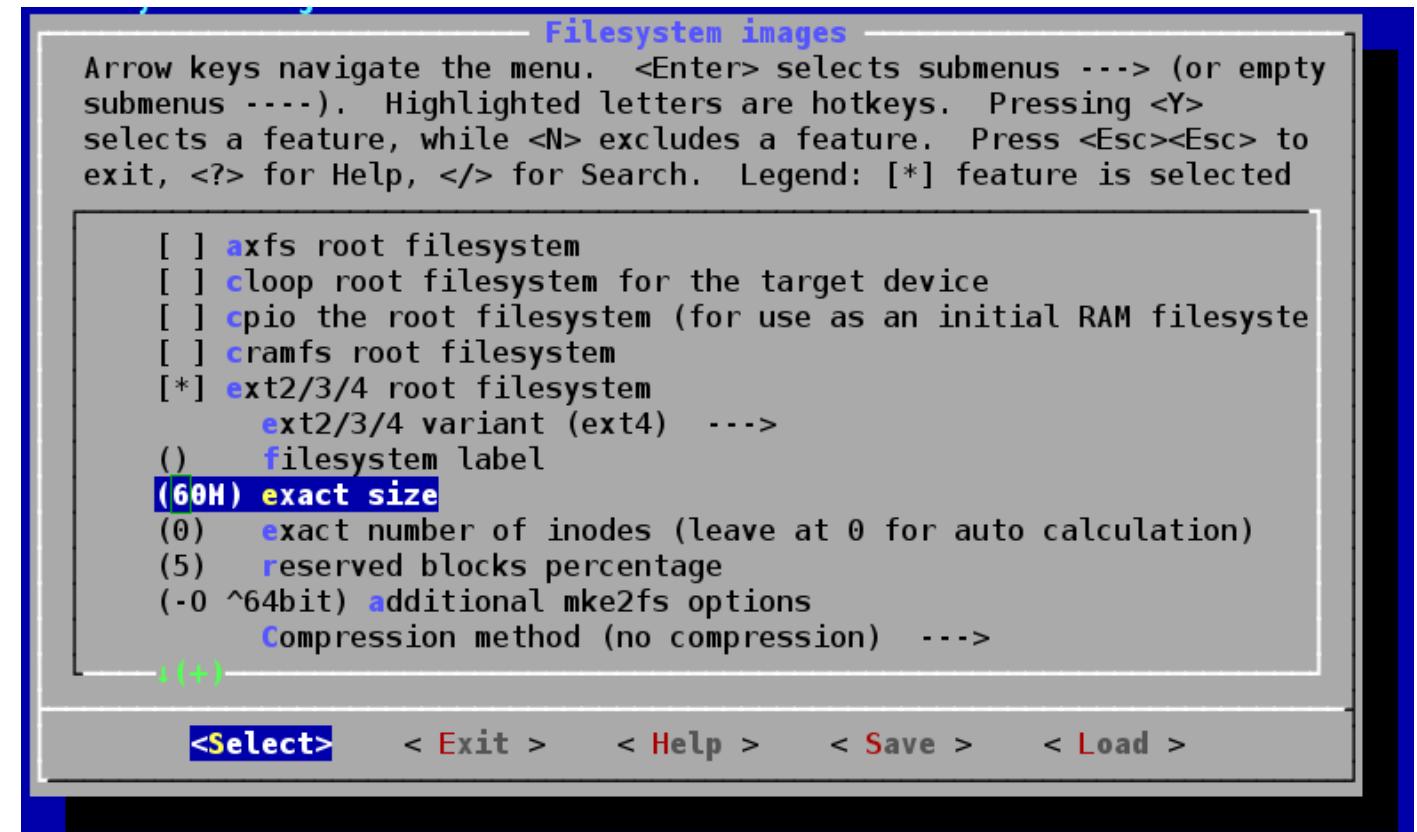


# Inspection des menus

7/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Filesystemimages :



## Différents types de Rootfs :

- sur carte SD ou disque avec un *filesystem* tel que **ext4**.
- sur mémoire flash avec un *filesystem* tel que **ubifs**.
- en lecture seule avec un *filesystem* tel que **squashfs**.

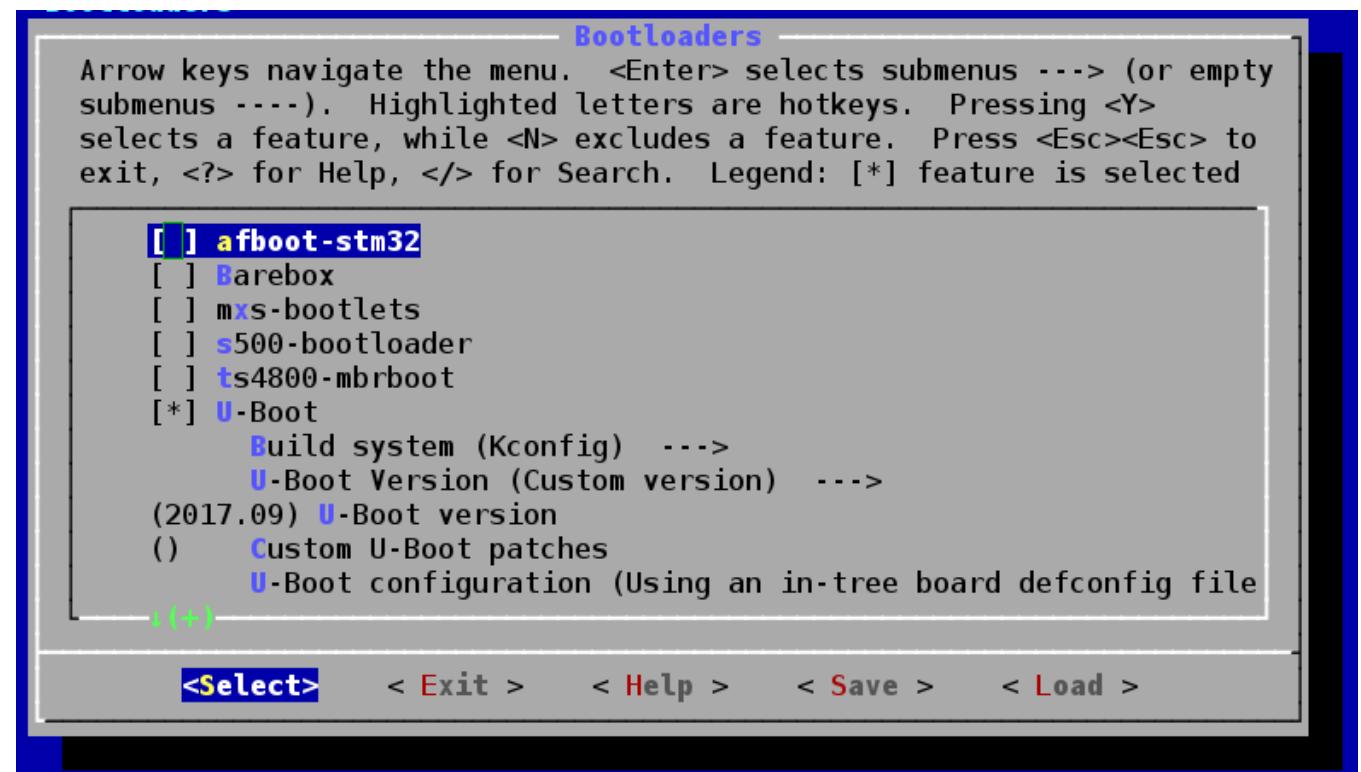


# Inspection des menus

8/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Bootloader :



U-Boot mais pas que...

Les microcodes pour certaines cibles.  
Les bootloader pour toutes les architectures



# Inspection des menus

9/9

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Intégration des applications
- Développement

## Host utilities :

The screenshot shows a terminal window titled "Host utilities". The menu describes how to navigate using arrow keys, enter for submenus, and other keyboard shortcuts. It lists several host tools, with "host imx-mkimage" highlighted with a green border. The menu includes a legend for selected items and footer options for selection, exit, help, save, and load.

```
Host utilities
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N> excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected

[ ] host gptfdisk
[*] host imx-mkimage
[ ] host imx-usb-loader
[ ] host jq
[ ] host jsmin
[ ] host lpc3250loader
[ ] host lttng-babeltrace
[ ] host mfgtools
-*-
[ ] host mkpasswd
[ ] host mtd, jffs2 and ubi/ubifs tools
[*] host mtools

<Select> < Exit > < Help > < Save > < Load >
```

Les outils génériques de construction du système  
mais aussi les outils mise à jour de la carte





L'environnement de développement



# Projet global

1/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- **Projet global**
- Configuration d'un paquet
- Construction d'un paquet
- Développement

## Arborescence de travail :

```
/opt/buildroot
├─ CHANGES
├─ COPYING
├─ Config.in
├─ Config.in.legacy
├─ DEVELOPERS
├─ Makefile
├─ Makefile.legacy
├─ README
└─ arch
```

```
  └─ board
```

```
  └─ boot
```

```
  └─ configs
```

```
  └─ dl
```

```
  └─ docs
```

```
  └─ fs
```

```
  └─ linux
```

```
  └─ package
```

```
  └─ support
```

```
  └─ system
```

```
  └─ toolchain
```

```
  └─ utils
```

```
project
├─ .git
├─ Config.in
└─ board
  └─ board1
    └─ board2
    └─ busybox.config
  └─ configs
    └─ project_board1_defconfig
    └─ project_board2_defconfig
  └─ devsrc *
    └─ ouistiti *
  └─ external.desc
  └─ external.mk
  └─ kernel.fragment.config
  └─ makedev
  └─ makeusers
  └─ overlays
    └─ etc
    └─ srv
  └─ package
    └─ my_app
    └─ my_dev
  └─ patches
```

### Index of /dl

- Parent Directory
- acl/
- alsa-lib/
- alsa-utils/
- atk/
- attr/
- audiofile/
- autoconf/
- automake/
- binutils/

```
project
└─ output
  └─ Makefile
  └─ build
  └─ host
  └─ images
  └─ staging
  └─ target
```



# Projet global

## 2/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- **Projet global**
- Configuration d'un paquet
- Construction d'un paquet
- Développement

## Informations du projet dans le fichier project/external.desc

```
name : PROJECT
desc :
~
~
"Config.in" 1L, 14C      1,13
```

## Informations de configuration dans le fichier project/Config.in

```
source "$BR2_EXTERNAL_PROJECT_PATH/package/my_app/Config.in"
~
~
"Config.in" 1L, 14C      1,13
```

## Informations de compilation dans le fichier project/external.mk

```
include $(sort $(wildcard $(BR2_EXTERNAL_PROJECT_PATH)/package/*/*.mk))
~
~
"external.mk"      1L, 14C      1,13
```



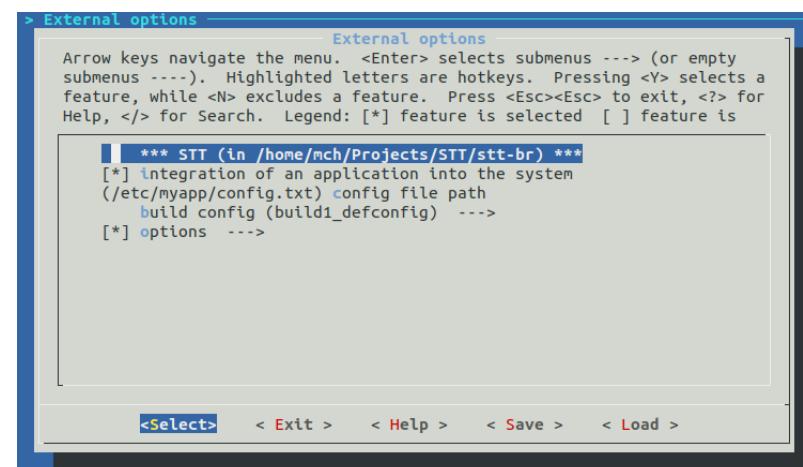
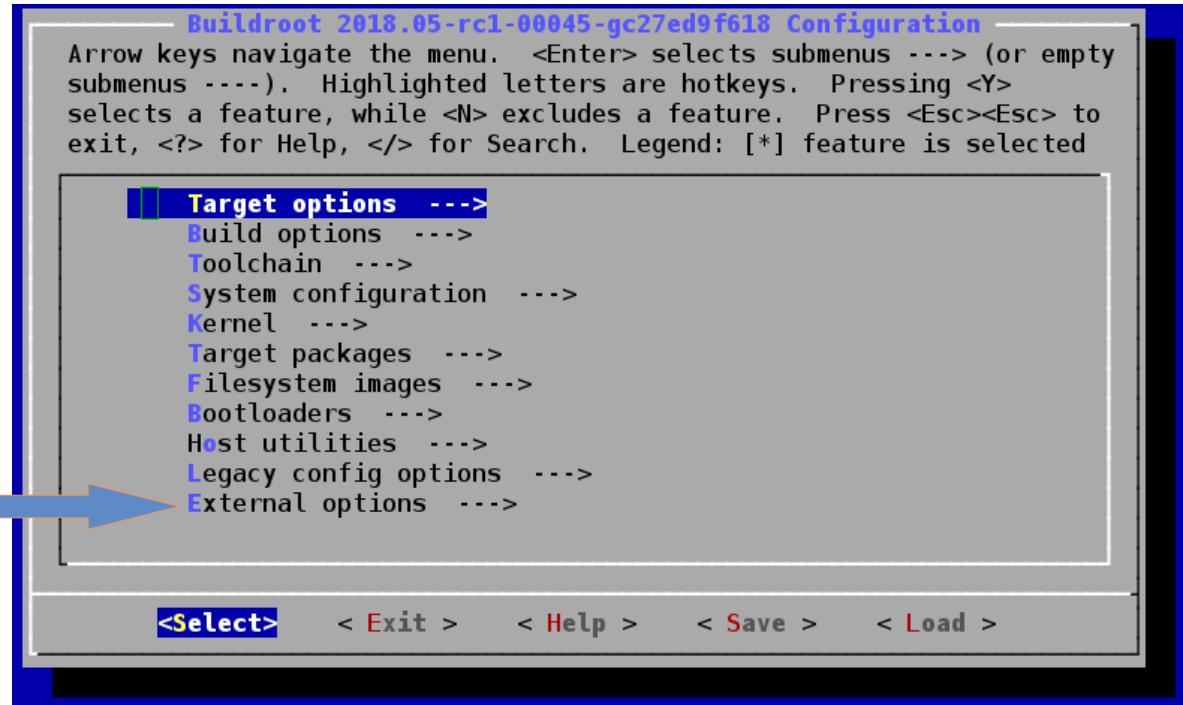
# Projet global

## 3/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- **Projet global**
- Configuration d'un paquet
- Construction d'un paquet
- Développement

## Retour à la configuration du projet

```
[project]$ cd output  
[output]$ make menuconfig
```

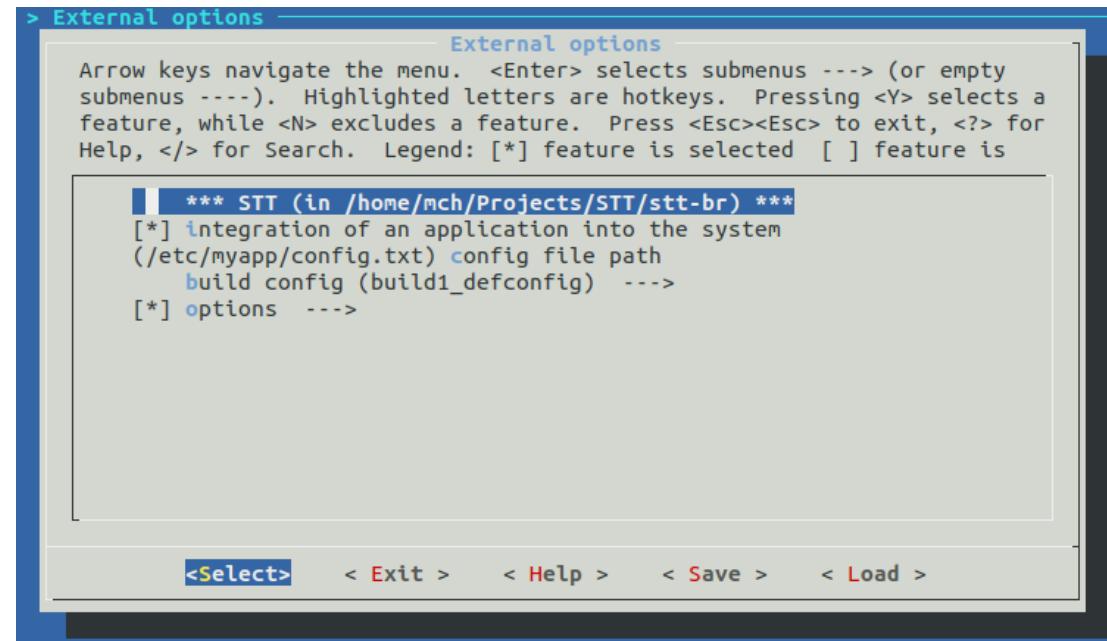


# Configuration d'un paquet

1/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- **Configuration d'un paquet**
- Construction d'un paquet
- Développement

## Informations de configuration dans le fichier package/<package name>/Config.in



Le nom d'un paquet est défini par le nom du répertoire qui contient un fichier Config.in

Le nom d'un paquet est remplacé



# Configuration d'un paquet

2/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Configuration d'un paquet
- Construction d'un paquet
- Développement

## Création du menu

### Entrée booléenne

```
config BR2_PACKAGE_MY_APP
    bool
    prompt "integration of an application into the system"
    help
        This is an application developped by my company
    default y
    ~
    ~
"package/my_app/Config.in"  6L,  14C      1,13
```

[\*] integration of an application into the system (NEW)

### Choix multiple

```
choice
    prompt "build config"
    config BR2_PACKAGE_MY_CONFIG1
        bool "build1_defconfig"
    config BR2_PACKAGE_MY_CONFIG2
        bool "build2_defconfig"
    endchoice
    ~
"package/my_app/Config.in"  6L,  14C      1,13
```

build config (build1\_defconfig) --->

(x) build1\_defconfig  
( ) build2\_defconfig



# Configuration d'un paquet

3/3

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- **Configuration d'un paquet**
- Construction d'un paquet
- Développement

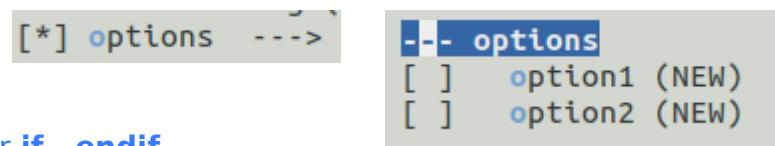
## Entrée champs

```
config BR2_PACKAGE_MY_CONFIGFILE
    string
    prompt "config file path"
    default "/etc/myapp/config.txt"
    ~
    ~
"package/my_app/Config.in" 6L, 14C 1,13
```

(/etc/myapp/config.txt) config file path (NEW)

## Menu sélectionnable

```
menuconfig BR2_PACKAGE_MY_OPTIONS
    bool "options"
if BR2_PACKAGE_MY_OPTIONS
config BR2_PACKAGE_MY_OPTION1
    bool "option1"
config BR2_PACKAGE_MY_OPTION2
    bool "option2"
endif
~
"package/my_app/Config.in" 6L, 14C 1,13
```



- Condition par **if...endif**
- Menu par **menu...endmenu**
- Gestion des dépendances **depends on BR2\_PACKAGE\_<nom du paquet>**
- Gestion de dépendance inverse **select BR2\_PACKAGE\_<nom du paquet>**

Toute la documentation :

<http://kernel.org/doc/Documentation/kbuild/kconfig-language.txt>



# Construction d'un paquet

1/2

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Configuration d'un paquet
- **Construction d'un paquet**
- Développement

## Règles de compilation dans le fichier package/<package name>/<name>.mk

```
MY_APP_VERSION = my_app-1.0
MY_APP_SITE = git://git.company.com/project_a/my_app.git
MY_APP_SITE_METHOD = git

MY_APP_CONF_OPS = \
    --prefix=/usr

$(eval $(autotools-package))
~  
~  
~  
"package/my_app/my_app.mk" 6L, 14C 1,13
```



# Construction d'un paquet

## 2/2

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- **Configuration d'un paquet**
- Construction d'un paquet
- Développement

## Règles complémentaires :

- Projet "./configure ; make ; make install" : **`$(autotools-package)`**
- Projet "make defconfig ; make ; make install" : **`$(kconfig-package)`**
- Projet "cmake ; make ; make install" : **`$(cmake-package)`**
- Projet module kernel : **`$(kernel-module)`**
- Projet Python : **`$(python-package)`**
- Projet non standard : **`$(generic-package)`**
  
- Gestion des licences : **`<pkg>_LICENCE`** et **`<pkg>_LICENCE_FILE`**
- Gestion des dépendances : **`<pkg>_DEPENDENCIES += <pkg2>`**
- Options de configuration : **`<pkg>_CONF_OPTS += --host=`**
- Options de compilation : **`<pkg>_MAKE_OPTS += -DDEBUG`**
- Fichier de configuration : **`<pkg>_KCONFIG_FILE += my_defconfig`**
- Ajout d'un utilisateur : **`<pkg>_USERS += <name>`**
- ...



# Développement

- Présentation
- Objectifs d'un projet
- Vue générale
- Installation de BuildRoot
- Création du projet
- Gestion du projet
- Inspection des menus
- Projet global
- Configuration d'un paquet
- Construction d'un paquet
- Développement

## Surcharge des sources :

depuis le fichier *local.mk* de toutes les sources nécessitant une synchronisation des sources avec l'environnement de build

```
MY_APP_OVERRIDE_SRCDIR=$(BR2_EXTERNAL_PROJECT_PATH)/devsrc/my_app  
APP2_OVERRIDE_SRCDIR=$(BR2_EXTERNAL_PROJECT_PATH)/devsrc/app2  
APP2_OVERRIDE_SRCDIR_RSYNC_EXCLUSIONS=--excludes docs  
~  
~  
~  
"local.mk"    2L,   14C      1,13
```

## Configuration du projet dans le menu "Build options"

```
BR2_PACKAGE_OVERRIDE_FILE="$(BR2_EXTERNAL_PROJECT_PATH)/local.mk"  
~  
~  
~  
"configs/project_defconfig"    1L,   14C      1,13
```

## Construction depuis l'environnement de build

```
[output]$ make my_app-rebuild  
[output]$ make my_app-reconfigure  
[output]$ make my_app-rebuild all
```

