Rapport Technique : Création d’un Système d’Exploitation Linux

Minimaliste avec Buildroot pour un Smartphone

Mini-Projet 2

Date : 25 Mai 2025

Préparé par : IBRAHIM ZARYOUH

**Table des matières**

1. [Présentation du projet](#_bookmark0)
2. [Explication des choix techniques (architecture, noyau, pilotes)](#_bookmark1)
3. [Procédure détaillée d’installation et de test](#_bookmark2)
4. [Problèmes rencontrés et solutions apportées](#_bookmark3)
5. [Scripts et fichiers de configuration](#_bookmark4)
   1. [Fichier de configuration de Buildroot](#_bookmark5) . . . . . . . . . . . . . . . . . . . .
   2. [Script d’automatisation pour compiler et flasher le système](#_bookmark6) . . . . . . . .

# Flasher Image sur Smartphone

1. [Conclusion](#_bookmark8)

# Présentation du projet

Ce projet vise à développer un système d’exploitation Linux minimaliste à l’aide de Buildroot, spécifiquement conçu pour un smartphone. Contrairement aux cartes Rasp- berry Pi, les smartphones nécessitent souvent un chargeur d’amorçage comme U-Boot ou Fastboot, ainsi que des pilotes adaptés à leur matériel, tels que l’écran tactile, le modem, les capteurs, etc. L’utilisation de QEMU pour tester le système avant de le flasher sur un smartphone réel permet d’éviter d’endommager l’appareil et d’accélérer les phases de développement et de débogage.

# Explication des choix techniques (architecture, noyau, pilotes)

Les choix techniques suivants ont été faits pour ce projet :

* **Architecture** : ARM Cortex-A53, compatible avec la plupart des smartphones modernes.
* **Chaîne d’outils** : Buildroot avec glibc pour une compatibilité et une performance optimales.
* **Chargeur d’amorçage** : U-Boot (version 2023.10), configuré pour la plateforme

vexpress\_ca9x4.

* **Noyau** : Linux version 6.6 avec prise en charge du Device Tree (DTS) personnalisé (my-smartphone.dts).
* **Pilotes** : Support pour l’écran tactile (Goodix), Wi-Fi, USB, stockage MMC, audio (SND\_SOC), et graphismes via Weston, Mesa3D et Libinput.
* **Système de fichiers** : Ext2 (512 Mo) avec initramfs pour une initialisation rapide.

# Procédure détaillée d’installation et de test

Les étapes suivantes décrivent l’installation et le test du système :

1. **Installation des dépendances** : Installer les outils nécessaires sur une distribution Ubuntu/Debian avec la commande suivante :

1

sudo apt - get update

sudo apt - get install -y build - essential git ncurses - dev python 3 unzip bc dtc

2

1. **Clonage de Buildroot** : Cloner le dépôt Buildroot si ce n’est pas déjà fait :

git clone https :// git. buildroot. org/ buildroot

1

1. **Création de la configuration personnalisée** : Créer un fichier de configuration

smartphone\_auto\_defconfig dans Buildroot pour un smartphone ARM générique.

1. **Configuration des pilotes** : Ajouter un fragment de configuration du noyau (common-drivers.config) pour les pilotes essentiels.
2. **Création du Device Tree** : Générer un fichier DTS minimal (my-smartphone.dts) adapté au smartphone cible.
3. **Compilation** : Charger la configuration et compiler avec :

1

make smartphone\_auto\_defconfig make

2

1. **Test avec QEMU** : Tester le système généré à l’aide de QEMU avec la commande suivante :

1

qemu - system - arm -M virt -m 512 - kernel output/ images/ zImage - initrd output/ images/ rootfs. cpio - nographic

1. **Flashage sur le smartphone** : Flasher les images générées (u-boot.bin et rootfs.ext2) sur le stockage du smartphone.

# Problèmes rencontrés et solutions apportées

Au cours du développement, les problèmes suivants ont été identifiés :

* **Problème** : Absence d’un fichier DTS spécifique au smartphone.
  + **Solution** : Utilisation d’un DTS générique basé sur vexpress-a15 comme solution temporaire, en attendant un fichier DTS spécifique.
* **Problème** : Conflits potentiels entre les pilotes.
  + **Solution** : Ajout de fragments de configuration (common-drivers.config) pour activer uniquement les pilotes nécessaires et éviter les conflits.

# Scripts et fichiers de configuration

## Fichier de configuration de Buildroot

Le fichier smartphone\_auto\_defconfig configure Buildroot pour inclure U-Boot, le noyau Linux, et les pilotes nécessaires. Voici son contenu :

# Architecture BR 2 \_arm =y BR 2 \_cortex\_a 53 =y

# Chaîne d’ outils BR 2 \_TOOLCHAIN\_BUILDROOT\_GLIBC =y

# Configuration du système BR 2 \_SYSTEM\_DHCP =" eth0 "

BR 2 \_ROOTFS\_DEVICE\_TABLE =" system / device\_table . txt"

# Chargeur d’ amorçage (U- Boot) BR 2 \_TARGET\_UBOOT =y BR 2 \_TARGET\_UBOOT\_CUSTOM\_VERSION =y BR 2 \_TARGET\_UBOOT\_CUSTOM\_VERSION\_VALUE ="2023 .10 "

BR 2 \_TARGET\_UBOOT\_BOARD\_DEFCONFIG =" vexpress\_ca 9 x 4 "

# Noyau BR 2 \_LINUX\_KERNEL =y

BR 2 \_LINUX\_KERNEL\_CUSTOM\_VERSION =y

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21 BR 2 \_LINUX\_KERNEL\_CUSTOM\_VERSION\_VALUE ="6 .6 "

22 BR 2 \_LINUX\_KERNEL\_DEFCONFIG =" vexpress"

23 BR 2 \_LINUX\_KERNEL\_DTS\_SUPPORT =y

24 BR 2 \_LINUX\_KERNEL\_CUSTOM\_DTS\_PATH =" board / my - smartphone / my - smartphone . dts"

25

26 # Paquets essentiels

27 BR 2 \_PACKAGE\_BUSYBOX =y

28 BR 2 \_PACKAGE\_DROPBEAR =y

29 BR 2 \_PACKAGE\_WPA\_SUPPLICANT =y

30 BR 2 \_PACKAGE\_WPA\_SUPPLICANT\_WIFI =y

31

32 # Support graphique et tactile

33 BR 2 \_PACKAGE\_WESTON =y

34 BR 2 \_PACKAGE\_WESTON\_DEFAULT\_COMPOSITOR =" drm - backend . so"

35 BR 2 \_PACKAGE\_LIBDRM =y

36 BR 2 \_PACKAGE\_LIBINPUT =y

37 BR 2 \_PACKAGE\_LIBGTK 3 =y

38

39 # Système de fichiers

40 BR 2 \_TARGET\_ROOTFS\_EXT 2 =y

41 BR 2 \_TARGET\_ROOTFS\_EXT 2 \_SIZE ="512 M"

42 BR 2 \_TARGET\_ROOTFS\_INITRAMFS =y

43

44 # Pilotes communs pour smartphones

45 BR 2 \_LINUX\_KERNEL\_CONFIG\_FRAGMENT\_FILES =" board / my - smartphone / common - drivers. config "

## Script d’automatisation pour compiler et flasher le système

Le script Bash suivant automatise le processus de configuration et de compilation :

#!/ bin / bash

# Script pour automatiser la configuration de Buildroot avec U-

Boot et écran tactile pour un OS minimal Linux sur smartphone

# Installation des dépendances ( Ubuntu / Debian ) sudo apt - get update

sudo apt - get install -y build - essential git ncurses - dev python 3 unzip bc dtc

# Clonage de Buildroot

if [ ! -d " buildroot" ]; then

git clone https :// git. buildroot. org/ buildroot

fi

cd buildroot

# Chargement de la configuration make smartphone\_auto\_defconfig

# Compilation du système

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

make

# Instructions de sortie

echo " Compilation terminée ! Les images sont dans output/ images /" echo " Binaire U- Boot : output/ images/u- boot. bin "

19

20

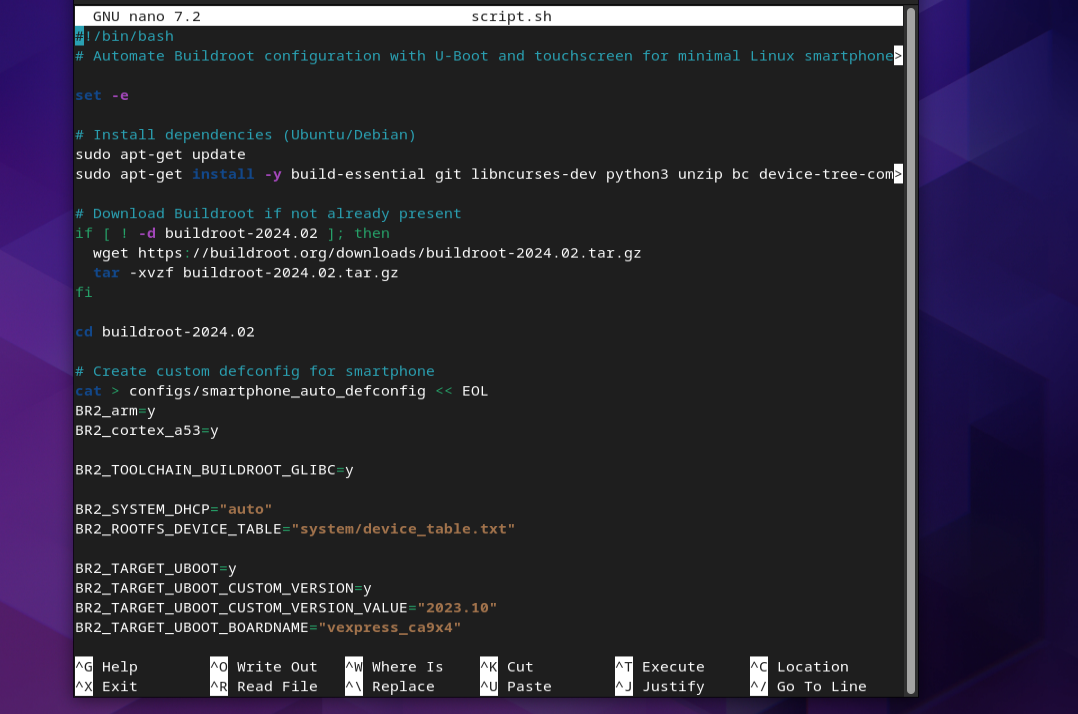
21

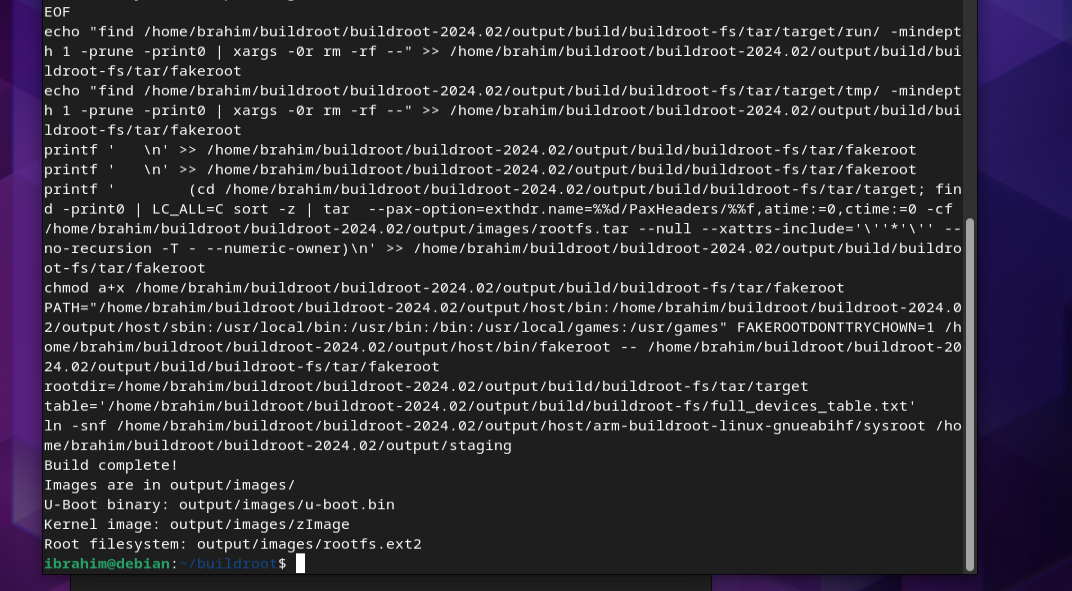
22

23

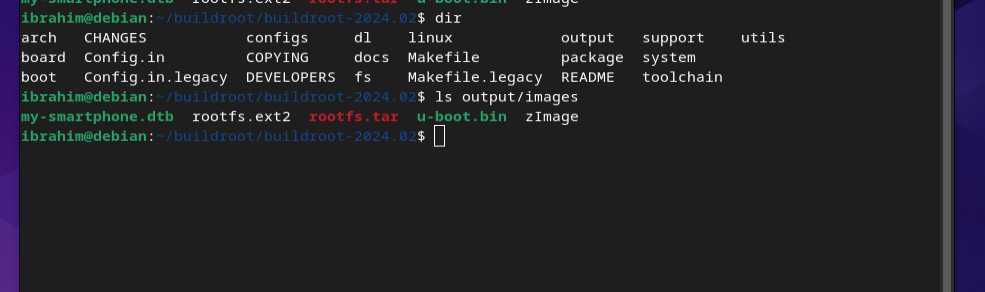
24

25

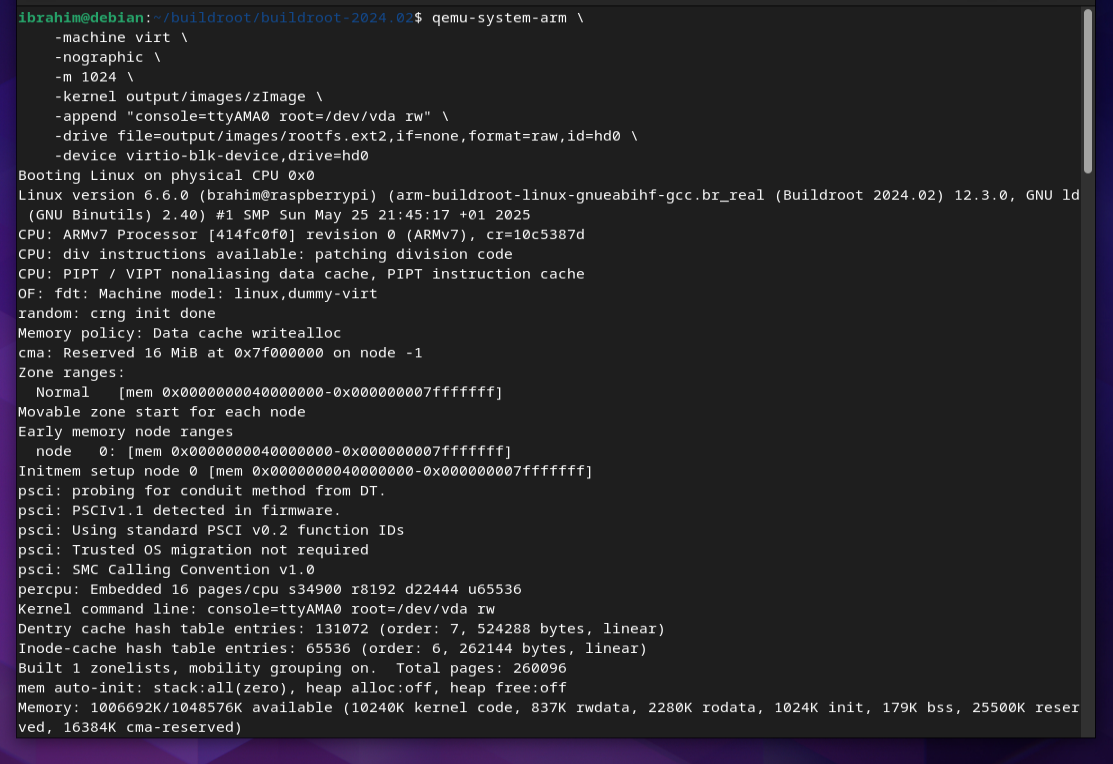


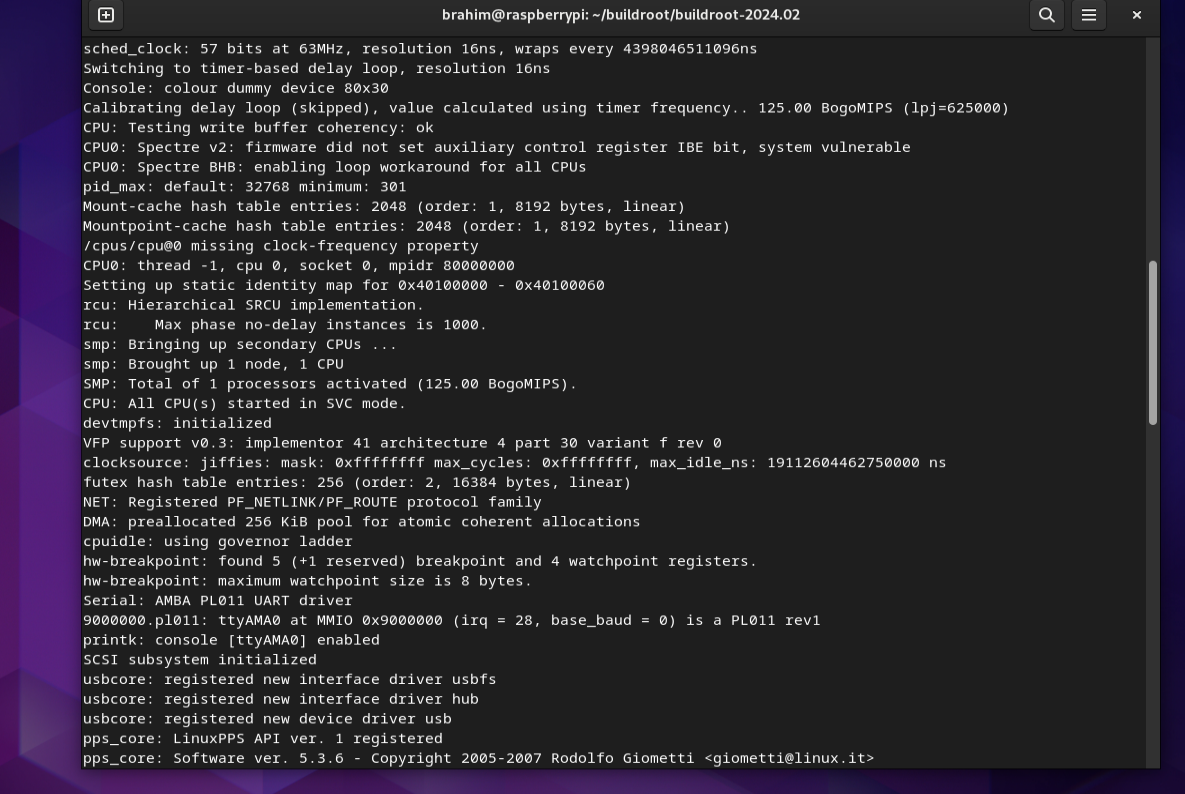


**Les images :**



**Tester sur qemu :**

****

****

# Flash the Image to the Smartphone

# Apres la compilation réussie, en flasher les images générées sur le smartphone à l’aide de Fastboot. Pour automatiser ce processus :

# #!/bin/bash

# fastboot flash boot output/images/zImage

# fastboot flash rootfs output/images/rootfs.ext4

# Conclusion

Ce projet a permis de développer un système Linux minimaliste pour un smartphone en utilisant Buildroot. Les fichiers générés incluent boot.img (noyau Linux), rootfs.ext4 (système de fichiers), et u-boot.bin (chargeur d’amorçage). Pour plus de détails, les journaux de compilation sont disponibles dans le dossier output/.