

1 Buildroot

1.1 Répertoires

~/workspace	→ working space
/nano	→ working space for NanoPi
/buildroot	→ space for tools, kernel, rootfs generation
/board/friendlyarm/nanopi-neo-plus2	→ genimage.cfg, boot.cmd
/dl	→ downloaded « tared » packets: e.g. busybox-1.30.1.tar.bz2
/system/skeleton	→ Rootfs skeleton
/output	
/build	→ source codes and compiled packets, e.g.: linux-5.8.5
/images	→ Image, nanopi-neo-plus2.dtb, rootfs.ext4, u-boot.itb, boot.scr, sunxi-spl.bin
/target	→ rootfs not “tared”
/host/usr/bin	→ cross-compiler: aarch64-none-linux-gnu-gcc, ...

Files **u-boot.itb, sunxi-spl.bin, Image, nanopi-neo-plus2.dtb, rootfs.ext4, boot.scr** will be copied to the uSD card.

Ce qui est manquant dans le dossier output sera recompilé lorsque la commande **make** est lancée (ou alors en faisant la commande **make <package>-rebuild**).

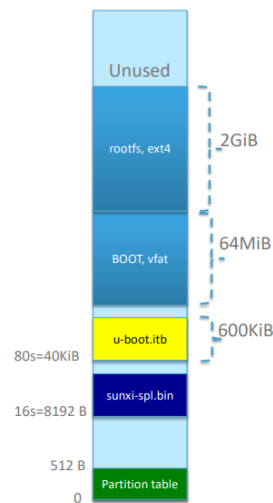
Le dossier **rootfs_overlay** permet d'ajouter des fichiers au **rootfs**
(/workspace/nano/buildrootboard/
friendlyarm/nanopi-neo-plus2/rootfs_overlay)

1.2 Compilation

Dans le répertoire **buildroot**, effectuer la commande **make menuconfig** puis **make**. **make clean** pour effacer tous les fichiers compilés.
La configuration permet notamment de

1. Modifier le rootfs

1.3 Carte SD



Sector size = 512 Bytes

Area Name	From (Sector #)	To (Sector #)	Size
rootfs, ext4			2GiB
BOOT, vfat			64MiB
U-boot-itb	80		600KiB
sunxi-spl.bin	16	79	32KiB
MBR (partition table)	0	15	512B

`genimage.cfg` → `genimage` → `sdcard.img` → `dd` → carte SD

Les fichiers pour l'initialisation sont

<code>rootfs.ext</code>	Root file system
<code>Image</code>	Noyau Linux
<code>nanopo-neo-plus2.dtb</code>	Flattened device tree
<code>boot.scr</code>	Commandes boot compilées utilisées par u-boot
<code>boot.vfat</code>	Partition boot
<code>u-boot.itb</code>	Boot loader
<code>sunxi-spl.bin</code>	Secondary Program Loader

`boot.vfat` contient `Image`, `nanopi-neo-plus2.dtb` et `boot.scr`

1.3.1 boot.scr

Le fichier `boot.scr` est utilisé par u-boot pour charger le kernel Linux. Il est créé avec la commande `mkimage`

1.3.2 boot.cmd

`boot.cmd` contient des informations de démarrage, notamment les emplacements des différents l'emplacement de `nanopi-neo-plus2.dtb`, du kernel et (si présent) de l'initramfs

1.4 Séquence de démarrage (6 phases)

1. Lorsque le μ P est mis sous tension, le code stocké dans son BROM va charger dans ses 32KiB de SRAM interne le firmware `sunxi-spl` stocké dans le secteur no 16 de la carte SD / eMMC et l'exécuter.
2. Le firmware `sunxi-spl` (Secondary Program Loader) initialise les couches basses du μ P, puis charge l'U-Boot dans la RAM du μ P avant de le lancer.
3. L'U-Boot va effectuer les initialisations hardware nécessaires (horloges, contrôleurs, ...) avant de charger l'image non compressées du noyau Linux dans la RAM, le fichier `Image`, ainsi que le fichier de configuration FDT (flattened device tree).
4. L'U-Boot lancera le noyau Linux en lui passant les arguments de boot (bootargs)
5. Le noyau Linux procédera à son initialisation sur la base des bootargs et des éléments de configuration contenus dans le fichier FDT (`sun50i-h5-nanopi-neo plus2.dtb`).
6. Le noyau Linux attachera les systèmes de fichiers (rootfs, tmpfs, usrfs, ...) et poursuivra son exécution.

