Armadeus

BTS IRIS

Lycée de
l'Hyrôme

Lycée de l'Hyrôme

Sommaire

- Présentation du projet
- Le matériel
- Le logiciel
- U-Boot
- TFTP
- NFS
- La chaine de développement

Lycée de l'Hyrôme

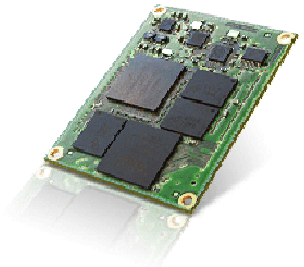
Présentation du projet Armadeus

- Le but du projet est de permettre à tous de développer facilement des systèmes embarqués à base de Logiciel/matériel Libres.
- "Armadeus" est la contraction de "ARM" et "Amadeus".
- Une communauté (association) participe aux développements (hardware/firmware/software) sous licence GPL/LGPL
  - La liberté d'exécuter le logiciel, pour n'importe quel usage ;
  - La liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources ;
  - La liberté de redistribuer des copies ;
  - La liberté d'améliorer le programme et de rendre publiques les modifications afin que l'ensemble de la communauté en bénéficie.
- Armadeus propose des cartes processeurs et de développement très performantes et à faible coût
- La documentation est contributive

Le matériel

APF27

- **L'APF27 est une carte à microprocesseur avec un rapport coût/performance extrêmement compétitif. (< 150 €)**



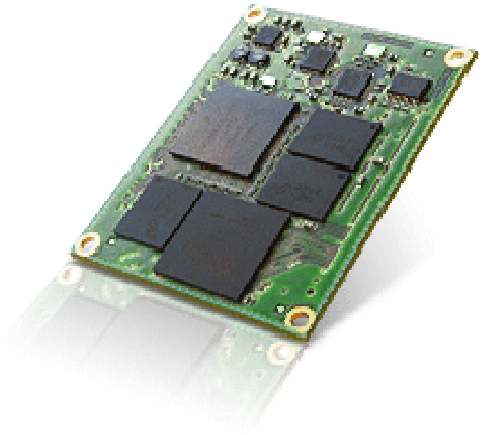
- **Microprocesseur**
  - i.MX27 (standard) 400MHz (Freescale)

- **RAM**
  - de 128 Mo (standard) à 256 DDR Mobile 32bits
    - DDR : « Double Data RateLa » fournit une meilleure bande passante que l'ordinaire SDRAM en transférant les données à la fois sur le front montant et sur le front descendant des impulsions d'horloge, ce qui a pour effet de doubler la vitesse d'accès à la mémoire, en lecture et en écriture.
- **FLASH**
  - de 256 Mo (standard) à 512 Mo NAND Mobile 16bits
    - La mémoire flash est une mémoire de masse à semi-conducteurs ré-inscriptible, elle stocke les bits de données dans des cellules de mémoire, mais les données sont conservées en mémoire lorsque l'alimentation électrique est coupée. La mémoire flash est un type d'EEPROM
    - La flash NOR : les temps d'effacement et d'écriture sont longs mais elle possède une interface d'adressage permettant un accès aléatoire et rapide à n'importe quelle position.
    - La flash NAND : Elle est plus rapide à l'effacement et à l'écriture, offre une plus grande densité et un coût moins important par bit. Toutefois son interface d'entrée / sortie n'autorise que l'accès séquentiel aux données.
    - Une cellule de mémoire flash ne peut être écrite 100 000 fois , 10 ans de rétention

- **EEPROM**
  - de 2 kbits (standard) à 512 kbits
    - Stockage adresse MAC

Le matériel

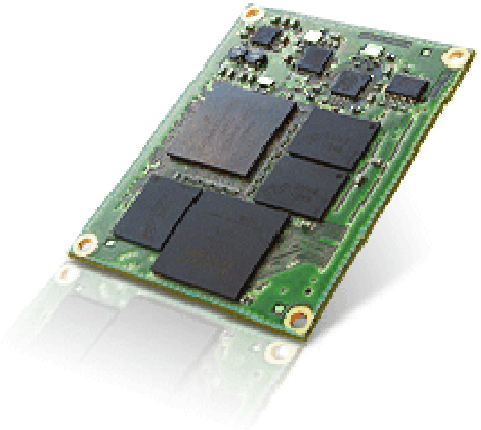
- APF27
  - Périphériques
    - 6 x RS232
    - 2 x I2C
    - 3 x SPI et 2 x SSI (Port série synchrone rapide)
    - 1 x USB USB OTG Hi-Speed host 400 Mbps
    - 1 x USB Host (Hi-Speed) 12Mbps
    - 1 x USB Host (Full-speed) 400Mbps
    - 1 x Ethernet 10/100 Mbits
    - 2 x SD/MMC
    - 1 x RTC sans batterie
    - 1 x PWM résolution 16bits
    - 6 x Timer 32 bits
    - 1 x Watchdog ajustable
    - 1 x port Keypad (jusqu'à 8 x 8 touches)
    - 1 x Contrôleur vidéo LCD STN/TFT (résolution max. 800 x 600 px, 18 bits)
    - 1 x MPEG/H.264 codec video MPEG4
    - 1 x CSI (interface capteur vidéo CMOS)
    - jusqu'à 107 Entrés/Sorties (GPIO)

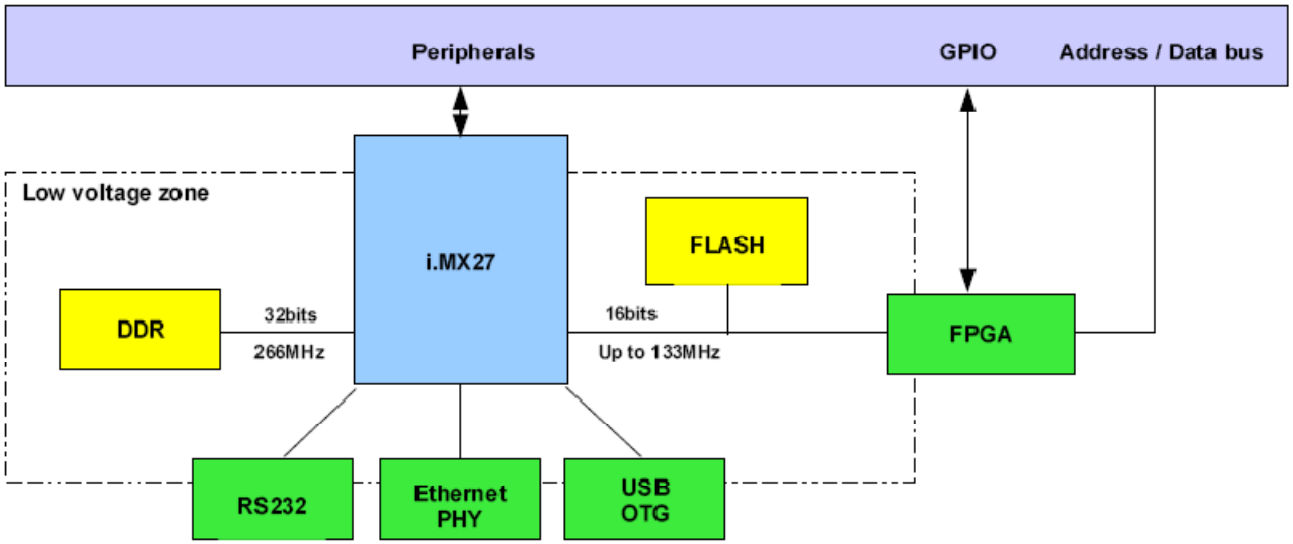


Le matériel

APF27

- Tranceivers
  - USB OTG
  - RS232 (2 ports, RX/TX)
  - Ethernet 10/100Mbits autoMDX





Le matériel

APF27 Dev

- **La carte électronique APF27_Dev est une plateforme de développement pour l'expérimentation d'applications Linux embarquées.**

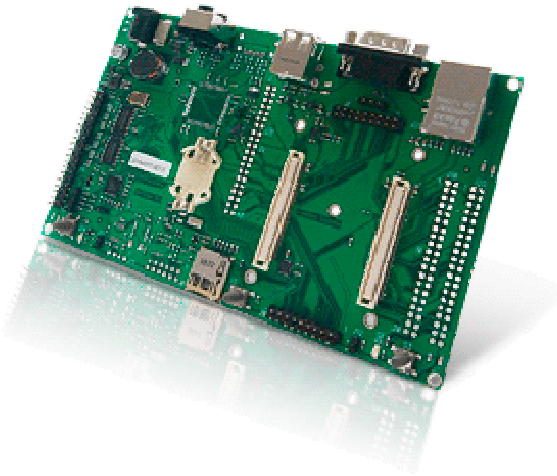
- **Elle permet d'accéder à toutes les fonctionnalités de la carte à microprocesseur APF27 tout en offrant un ensemble de solutions additionnelles :**

  - Un USB High speed et un USB 2.0 full speed
  - Contrôleur dalle tactile
  - Lecteur micro-SD
  - Sortie audio stéréo + entrée micro
  - 2 Leds de visualisation
  - 2 boutons poussoirs
  - Contrôleur bus CAN
  - Convertisseur analogique / numérique (7 voies, 10 bits) modèle MAX1027
  - Convertisseur numérique / analogique (2 voies, 10 bits) modèle MAX5821
  - Horloge avec batterie de maintien
  - Contrôleur HDMI/DVI.
  - Bouton de reset

Le matériel

APF27 Dev

Connecteurs

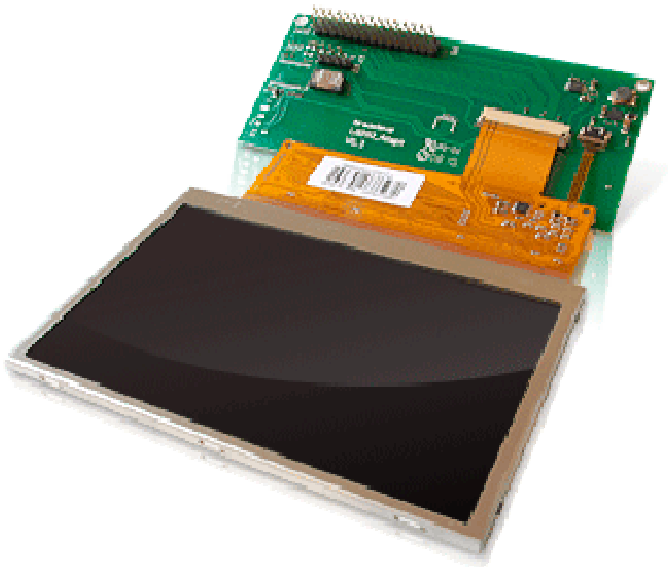


- Jack 2.5mm pour l'alimentation
- Ethernet (RJ45) avec transformateur d'isolement intégré et leds d'état
- 2 USB maîtres (type A)
- Prise HDMI pour la sortie TV numérique
- Prise jack 3,5mm stereo pour la sortie audio et 3.5mm mono pour l'entrée micro
- DB9 Male (console)
- 2 connecteurs Hirose pour la carte APF27
- Cavalier de sélection du mode de démarrage
- Slot pour carte micro-SD (port SD2)
- Bus LCD pour la connexion rapide d'un écran
- Connecteur pour écran tactile (pins au pas de 2.54mm)
- Connecteur pour RS232 (UART3

LQ043 Adapt

- La carte LQ043_Adapt est une carte d'adaptation mécanique et électrique permettant d'utiliser les écrans LCD de la série LQ043 de la société Sharp avec les cartes Armadeus.

- Ce TFT est un modèle 4.3 pouces avec une résolution de 480x272 pixels. Il est équipé d'un rétroéclairage par led et d'un écran tactile



Le logiciel

Le projet Armadeus repose sur des briques de logiciels libre :

- U-Boot pour le bootloader
- Linux 2.6 pour le système d'exploitation
- Buildroot pour la génération du « rootfs »
- Toolchain la chaine d'outils (compilateur, éditeur de lien, debugger, etc...)
- Qt/Embedded OpenSource ou SDL pour l'interface graphique

U-Boot

- U-Boot (Universal Boot Loader) est le "BIOS" présent sur l'APF27
  - U-Boot est open source (GPL), sans royalties et bénéficie d'une communauté de développeurs importante et extrêmement active.
- Les tâches principales d'U-Boot
  - Initialisation du matériel et plus particulièrement du contrôleur mémoire
  - Passage des paramètres de démarrage au noyau Linux
  - Lancement du noyau Linux
- U-Boot permet aussi de :
  - Lire et écrire dans différentes zones mémoire
  - Charger des images binaires dans la RAM par câble série, Ethernet
  - Copier des images binaires de la RAM vers la FLASH
  - Programmer le FPGA

U-Boot

- La communication avec U-Boot se fait à travers une liaison série RS232
  - 115200 bit/s
  - 8 bits
  - Pas de parité
  - 1bit de stop

- Au boot de la carte différents messages s'affichent sur le terminal série

```
U-Boot 1.3.4 (Mar  7 2009 - 14:58:12) apf27 patch 2.0

I2C:   ready
DRAM:  64 MB
NAND:  256 MiB
*** Warning - bad CRC or NAND, using default environment

In:    serial
Out:   serial
Err:   serial
nand_unlock: start: 000a0000, length: 267780096!
got MAC address from EEPROM: 00:1e:ac:00:00:03
Hit any key to stop autoboot:  0

Loading from NAND 256MiB 1,8V 16-bit, offset 0x180000
Automatic boot of image at addr 0xa0000000 ...
## Booting kernel from Legacy Image at a0000000 ...
   Image Name:   Linux-2.6.27.13
   Created:      2009-03-07  14:08:53 UTC
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1833524 Bytes =  1.7 MB
   Load Address: a0008000
   Entry Point:  a0008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux.....................................................................
Linux version 2.6.27.13 (eric@shuttle) (gcc version 4.2.1) #1 PREEMPT Sat Mar 7 15:08:45 CET 2009
```

U-Boot

Les variables d'U-Boot

- **U-Boot peut être configuré à travers des variables d'environnements et des scripts :**
  - printenv : affichage des variables et des scripts
  - setenv : affectation d'une variable
  - saveenv : sauvegarde en flash des variables
  - run : permet d'exécuter un script

- **La configuration réseau d'U-Boot**
  - On pourra à travers le réseau
    - Mettre à jour U-Boot
    - Mettre à jour le noyau linux
    - Mettre à jour le système de fichier linux

```
BIOS> setenv netmask 255.255.255.0
BIOS> setenv ipaddr 192.168.0.10
BIOS> setenv serverip 192.168.0.2
```

```
BIOS> printenv
bootcmd=run jffsboot
bootdelay=20
baudrate=115200
ethaddr=
autoload=no
...
```

```
BIOS> setenv ipaddr 192.168.0.10
```

```
BIOS> saveenv
Saving Environment to Flash...
. done
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash... done
. done
Protected 1 sectors
```

```
BIOS> run flash_reset_env
```

U-Boot

- Table des partitions d'U-Boot:

NAND flash address range	Type
0x00000000 - 0x0009FFFF (640KB, including NAND SPL and 384KB spare memory for bad blocks)	U-Boot
0x000A0000 - 0x000FFFFF (384KB)	U-Boot environment variables
0x00100000 - 0x0017FFFF (512KB)	FPGA bitfile
0x00180000 - 0x0067FFFF (5MB)	Linux kernel image
0x00680000 - End of FLASH (>~250MB)	Root filesystem

- Le contenu de chaque partition peut être mis à jour, en transférant au préalable en RAM des fichiers générés sur le PC de développement (toolchain)
  - Par liaison série
  - Par réseau en utilisant le protocole TFTP
- Dans des phases de test il est possible de déporter le système de fichier grâce à NFS
- Le système de fichier peut également être placé sur une carte mémoire MMC/SD

U-Boot

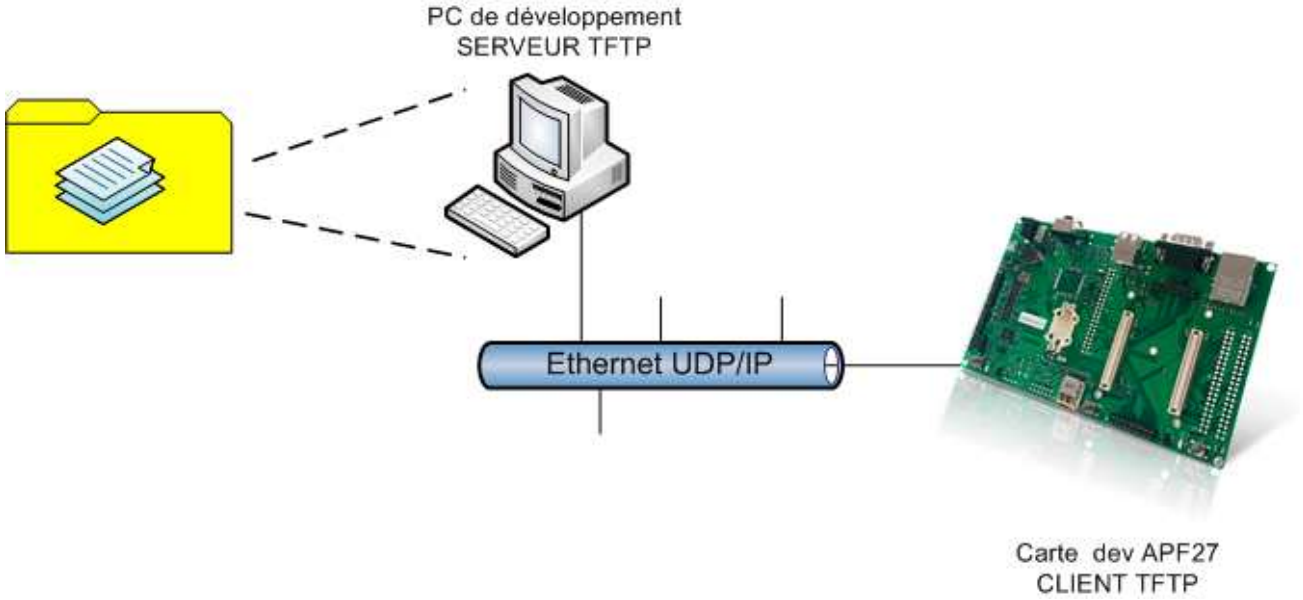
Les fichiers générés par le buildroot et le toolchain utilisés par U-Boot :

- *apf27-u-boot.bin*: Le fichier image d'U-Boot, pour le mettre à jour.
- *apf27-linux.bin*: Le fichier image du noyau Linux.
- *apf27-rootfs.arm.jffs2*: Le fichier image du système de fichier (rootfs) linux au format JFFS2 (Journaling Flash File System version 2)
- *apf27-rootfs.arm.tar*: Le fichier image du système de fichier (rootfs) linux pour un boot via NFS/MMC

Le protocole TFTP

- Le protocole TFTP (Trivial File Transfer Protocol) est utilisé pour :
  - mettre à jour U-Boot
  - mettre à jour le noyau linux
  - mettre à jour le rootfs de linux
  - télécharger un fichier (programme)

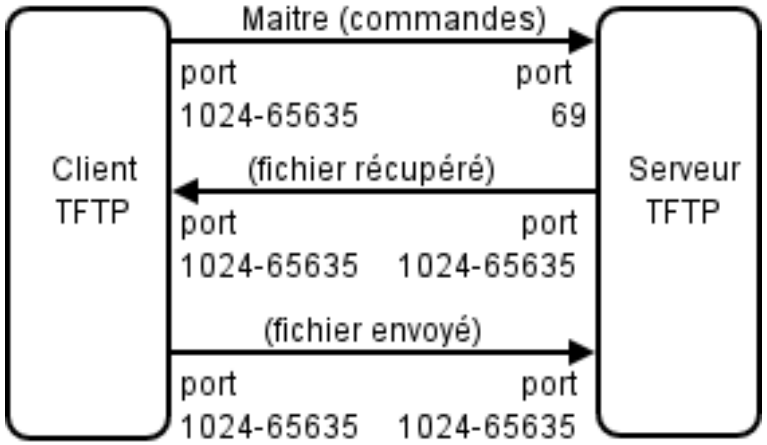
- Le serveur TFTP installé sur le PC de développement partagera un répertoire dans lequel le client TFTP (U-Boot ou un client Linux) récupérera les fichiers images avant de flasher la mémoire.



Le protocole TFTP

Description du protocole (RFC1350) :

- fonctionne en UDP sur le port 69
  - implique que le client et le serveur doivent gérer eux-mêmes une éventuelle perte de paquets. On réserve généralement l'usage du TFTP à un réseau local.
- ne gère pas le listage de fichiers
- ne dispose pas de mécanismes d'authentification, ni de chiffrement.
- Il faut connaître à l'avance le nom du fichier que l'on veut récupérer.
- Aucune notion de droits de lecture/écriture en standard.
- très utilisé pour la mise à jour des logiciels embarqués sur les équipements réseaux (routeurs, pare-feu, …)



Client/serveur TFTP

U-Boot intègre un client TFTP qui utilise les variables d'environnement et des scripts (ou commandes) pour aider aux téléchargements et aux mises à jour des images :

- **Variable serverip : adresse du serveur TFTP**
- **Commande tftpboot : permet de charger en RAM un fichier image (il faut flasher la mémoire avec le script « run flash-* »)**
- **Le script « run update_* » permet de télécharger une image et de flasher en une seule opération**
  - run update_kernel
    - Récupération via TFTP du fichier a*pf27-linux.bin et flash la partition du noyau linux*
  - run update_rootfs
    - Récupération via TFTP du fichier a*pf27-rootfs.arm.jffs2 et flash la partition du système de fichier linux*
- **Les fichiers images doivent être placés dans le répertoire de partage du serveur TFTP**

Le protocole NFS

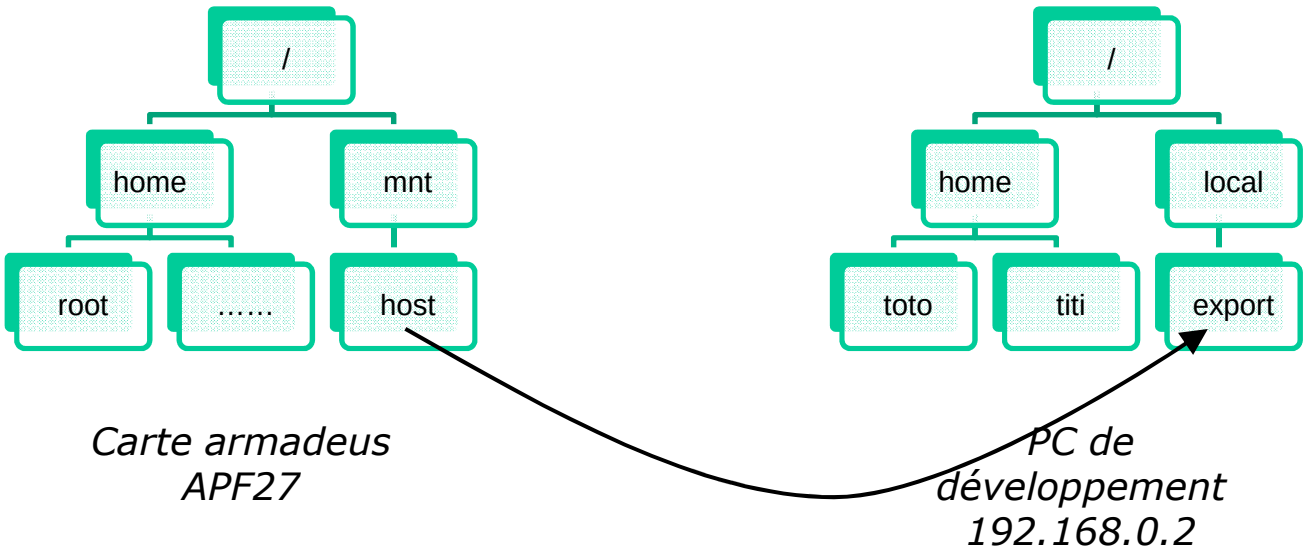
- Le système de fichiers en réseau (Network File System ou NFS) est un protocole développé par Sun Microsystems qui permet à un ordinateur d'accéder à des fichiers via un réseau.

  - Un serveur NFS donne l'autorisation d'accès à certaines arborescences du système de fichiers, on parle alors de partage ou d'exportation

  - Le client NFS accède à la ressource partagées par un montage dans son arborescence.

  - L'accès aux fichiers est totalement transparent pour les utilisateurs et les application

```
# mount -t nfs 192.168.0.2:/local/export /mnt/host
```



Toolchain

- Description
  - La chaine d'outils (compilateur, éditeur de lien, debugger, etc...) doit être compilée sous Linux

- Buildroot
  - Buildroot est un ensemble de scripts et de patchs (makefiles) facilitant la génération d'une chaine de compilation croisée ainsi que la création du système de fichiers des cartes APF. Il utilise la librairie C embarquée: uClibc.

- BusyBox: le couteau suisse pour Linux embarqué
  - BusyBox combine de nombreux utilitaires Linux, optimisés en taille, dans un seul exécutable.

La chaine de développement

Avant de pouvoir écrire et de compiler un programme à exécuter sur la carte APF27, il faut installer sur le PC de développement, la chaine de développement :

- Il faut construire les outils de compilation (cross compilation) gcc pour processeur ARM

- Il faut compiler et construire l'ensemble des fichiers et exécutables pour la carte

  - Le noyau linux
    - kernel
    - drivers
  - Le rootfs
    - Les outils GNU
    - L'arborescence

La chaine de développement

- **Les différentes étapes sont :**

  - **Vérification des paquets nécessaire (Debian)**
    - Installation des paquets
  - **Récupération du logiciel Armadeus**
    - Drivers
    - Programmes de test
    - Makefile
  - **Configuration du buildroot**
    - Configuration des fichiers générés
    - Configuration du noyau
    - Configuration des applications à embarquer
  - **Construction du buildroot**
    - Récupération des sources
      - Noyau
      - Drivers
      - Outils (gcc…)
      - Application
    - Préparation des compilation
      - Génération des makefiles
    - Compilation
    - Génération des fichiers images
      - U-Boot
      - Noyau linux
      - rootfs