I - Initialisation de la BeagleBone

Le projet commence par l’installation de debian sur la carte SD qu’utilisera la beagleBone pour booter. Après avoir mis à jour de nombreux paquets tels que le compilateur C des processeurs ARM sur la clé USB, on crée des partitions sur la carte SD l’une contenant une petite petite initialisation permettant de charger la seconde partition plus complète. On utilise un port série et minicom afin de communiquer avec la beagle bone. Il faut enfin connecter au réseau la BeagleBone pour quelle soit accessible par plusieurs PC et non seulement celui connecté en port série.

**Installation de la distribution Debian sur le beaglebone :**

* on boot sur la clef :
  + set root=(hd1,msdos1)
  + linux /vmlinuz
  + boot
* On ajoute un utilisateur :
  + adduser mireille
    - mdp : blanc
* on mets à jour les paquets et on installe pas mal de chose :
  + dpkg --add-architecture armhf
  + apt-get update
  + apt-get install gcc-5-arm-linux-gnueabihf (compilateur GCC pour processeurs arm)
* On branche la carte SD sur laquelle on va installer debian :
  + Création de 2 partitions en utilisant l’utilitaire fdisk (fdisk /dev/sdb pour modifier la cle usb)
    - Création d’une partition de 75Mo pour le boot, flagé “Bootable”, partition W95 Fat32 (code 0b)
    - Création d’une 2nde partition avec le reste de la carte pour le système
  + Formatage des 2 partitions
    - mkfs.vfat /dev/${dev}1 -F 32 -n BBB\_Boot  
      mkfs.ext4 /dev/${dev}2 -L BBB\_System
  + Copie des fichiers du système sur la Partition 2, dédié au système
    - mount /dev/${dev}2 /mnt/usb/ && cd /mnt/usb/ && tar xf /opt/BeagleBone/debian-7.2-armhf-2013-12-11\_CPE.tar.xz
  + Copie du noyau sur la Partition 1 pour le boot :
    - mount /dev/${dev}1 /mnt/usb/ && cp /opt/BeagleBone/Build/results/\* /mnt/usb/
  + Branchement du port série en usb sur l’ordi pour controler la carte.
  + Dans un terminal
    - Installer Minicom (sudo apt-get install minicom lrzsz)
    - Config (minicom -a) -> changer /dev/\_\_\_ par /dev/ttyUSB0
  + Mettre la carte sous tension -> Les commandes s’affichent sur l’ordi connecté au port série.
  + Pour connecter au réseau :
    - Sur l’ordinateur :
      * Rajouter une interface sur eth0 :
        + ifconfig eth0:0 @IP\_ordi
      * Transfert de la connexion internet vers la beaglebone:
        + echo 1 > /proc/sys/net/ipv4/ip\_forward
        + iptables -P FORWARD ACCEPT
        + itpables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
    - Configuration de l’ethernet sur la beaglebone
      * Configuration de eth0 :
        + ifconfig eth0 @IP\_carte
      * echo “nameserver 208.67.222.222” > /etc/resolv.conf
      * Pour obligé le système à toujours mettre eth0 au démarrage
      * d
      * Ajouter une route par défaut
        + route add default gw @IP\_pc

Réponses questions :

### Questions générales

* Qu'est-ce que Linux ?

Linux est un système d’exploitation

* Qu'est-ce qu'un système embarqué ?

Un système embarqué est un système électronique et informatique **automne** spécialisé dans une tache précise.

* Qu'est-ce qu'une bibliothèque C ?

Une bibliothèque C est un fichier contenant un ensemble de fonction.

* Qu'est-ce que Busybox ?

BusyBox est un logicielqui implémente un grand nombre des commandes standard sous Unix,

### La carte BeagleBone Black

* Recherchez les caractéristiques techniques de la carte BeagleBone Black intéressantes pour notre application (des sondes d'exploration) (Vous pouvez commencer par la page des specs sur le site de BeagleBoard, mais ce n'est pas ce qui est attendu)

Note : Une présentation assez intéressante de la carte est faite [ici](http://www.blaess.fr/christophe/2013/05/13/un-nouvel-os-a-ronger/) par Christophe Blaess.

### Système par défaut

* Quels ont été les différents systèmes proposés par défaut sur les cartes "en sortie d'usine" ?
* L'utilisation de ces systèmes est-elle obligatoire ?
* Quels sont les avantages et les inconvénients liés à l'utilisation de ces systèmes ?
* Quelles sont les alternatives ?

**Apt-file** permet de retrouver à quel *paquet* appartient un *fichier*. Cette commande peux s'avérer pratique dans bien des cas, notamment lors d'erreurs dans le terminal.

Boot de la carte ⇒ Chargement d’un fichier léger (OLM) qui va instancier la RAM afin de charger le véritable boot (u-boot.img) plus lourd dans la RAM. C’est ce deuxième fichier qui chargera l’image du système.

Rapport :

* Récupérer les sources de la carte RF sur le GIT <http://gitclone.techno-innov.fr/modules>
* Avec git gui, changer la remote passer le la master à la RF\_sub1G
* Dans