TP 1 RPI from scrath
Création du système

Objectifs :
Comprendre le processus de démarrage d'un système Linux embarqué.
Obtenir un système fonctionnel et à jour.
Créer des utilisateur.
Configurer l'accès réseau de la carte.
Installer des applications sur la carte.
Lancer des applications automatiquement au démarrage de la carte.

Voici une liste non exhaustive de notions et de commandes que vous devrez connaître à la fin du tp :
buildroot, bootloader , kernel,kernel commandline, root file system, partitions, point de montage, cross-compilation, make, makefile , /sbin/init et inittab,git, apt , scp , ssh , dd, wget, getty, pseudo système de fichiers /dev , /proc.

Le but de ce tp est de construire un système linux fonctionnel pour la carte Raspberry Pi à partir des sources des différents programmes.
Vous serez noté sur vos recherches, vos méthodes pour résoudre les différents problèmes qui ne vont pas manquer de se poser et sur la qualité de votre compte rendu. Le compte doit clairement présenter des notions abordées. Il doit être aussi un mode d'emploi très détaillé qui permettre de refaire rapidement les différentes manipulations.

Documents :
Le tp se base sur des articles de *Christophe Bleass* parus dans
Linux Magazine France (GLMF) :

http://www.blaess.fr/christophe/files/glmf/RPi-from-scratch-01.pdf
http://www.blaess.fr/christophe/files/glmf/RPi-from-scratch-02.pdf
http://www.blaess.fr/christophe/files/glmf/RPi-from-scratch-03.pdf
Les fichiers utilisés dans les articles sont accessibles ici :
http://www.blaess.fr/christophe/articles/files-glmf/

Attention comme l'article date de 2012, certains fichiers ne sont pas utilisable en l'état pour ce tp. De plus comme nous utiliserons la version buildroot 2014-11 et que notre carte est légèrement différente de celle de l'article (B+ versus B), les manipulations décrites ne peuvent pas être suivies à la lettre.

P.... de Proxy !!!!

Pour faire les manipulations depuis l'intérieur de l'université, il faut paramétrer le proxy de votre machine. Dans le réseau des salles GEII, la sortie se fait par une passerelle qui ne laisse passer que les protocoles http et https. De plus il s'agit d'un proxy avec authentification (login, mot de passe).

Paramètres du proxy : **adresse : 192.168.0.1 port : 3128**
Le problème principal est que les différents logiciels ne prennent pas tous la configuration de la même manière. De plus selon qu'ils s'exécutent depuis un terminal ou qu'ils sont lancés par un script la méthode d'accès aux paramètres du proxy est parfois différente.

Voici la méthode de configuration pour les principaux logiciels utilisés :

Pour changer l'environnement dans un terminal :

```
export http_proxy=http://utilisateur:motdepasse@ip_proxy:port
```
ou
```
export https_proxy=https://utilisateur:motdepasse@ip_proxy:port
```

(si votre mot de passe comporte des caractères spéciaux, il faut parfois mettre le contenu de la variable entre quotes **'....'**)

Pour voir la variable : **`$ echo $variable`**
Pour chercher dans l'environnement : **`$ env | grep "motif"`**

Utiliser WGET avec un proxy et authentification

Créer un fichier **.wgetrc** (n'oubliez pas le point devant le fichier) a la racine de votre répertoire personnel avec le contenu suivant :

```
http_proxy = http://votre_proxy:port_proxy/
proxy_user = votre_user_proxy
proxy_password = votre_mot_de_passe
use_proxy = on
wait = 15
```

Dans un terminal, la variable http_proxy suffit.

Utiliser GIT avec un proxy :

$ git config --global http.proxy http://user:pass@proxyhost:proxyport
$ git config --global https.proxy https://user:pass@proxyhost:proxyport

NE PAS OUBLIER D'EFFACER VOTRE MOT DE PASSE EN FIN DE SEANCE.

TP 1 RPI from scrath
Création du système

On n'utilisera pas la sortie HDMI mais plutôt l'UART de la carte. Il faut donc câbler un adaptateur UART/USB sur le port d'entrée/sorties de la carte.

ATTENTION il s'agit de port 3.3V non tolérant au 5V.

Il suffit de câbler TX,RX et GND.

Travail à effectuer :

Le but est de construire un système linux pour la carte Raspberry Pi B+. Le système doit être accessible par ssh et permettre l'exécution de scripts python3. Il faut également avoir accès au port série en python3 (module serial) et aux GPIO. Un serveur web minimaliste doit être lancé au démarrage de la carte.

- toolchain + kernel :
utiliser buildroot version 2014.11 :
http://buildroot.uclibc.org/downloads/buildroot-2014.11.tar.bz2
utiliser la configuration de départ : raspberrypi_defconfig
utiliser le protocole http pour télécharger le noyau : http://github.com/raspeberrypi/linux.git

- **root file system** : Essayer celui produit par buildroot puis créez le votre from scratch

- **tester la chaîne de compilation croisée** en compilant un programme C et en vérifiant son fonctionnement sur la carte.

- **paramétrer le réseau** (ip fixe)

- **configurer l'accès ssh**

- **permettre l'exécution de python3**

- **serveur web** (dropbear)

- **accès au GPIO**