TP3 MASTER2LSE- Part 3 – COMPILATION DU NOYAU

Nous allons dans ce TP travailler sur la compilation du noyau Linux à partir de Buildroot.

Voici comment configurer les différents outils utilisés par Buildroot :

- Le noyau Linux : **`$ make linux-menuconfig`**
- U-Boot : **`$ make uboot-menuconfig`**
- Busybox : **`$ make busybox-menuconfig`**
- Uclibc : **`$ make uclibc-menuconfig`**

Explorez rapidement le contenu de chaque menu de configuration.

UTILISATION DES LEDS

Nous souhaitons utiliser les LEDs dans ce TP. Lors du premier TP, nous avons téléchargé une distribution Debian sur la carte. Après démarrage, les 4 LEDs se situant à côté de l'interface Micro-USB et du connecteur pour le câble réseau se mettaient à clignoter chacune à un rythme particulier. Il s'agit des LED USR0, USR1, USR2, et USR3 (D2, D3, D4, D5). Par défaut, ces LEDs clignotent comme suit :

- USR0 est utilisé pour clignoter au rythme du battement de cœur
- USR1 est utilisé pour clignoter lors d'accès à la carte SD
- USR2 est utilisé pour clignoter lors d'activité du CPU
- USR3 est utilisé pour clignoter lors d'accès à la mémoire flash eMMC

Nous allons tenter dans cette partie de commander ces LEDs.

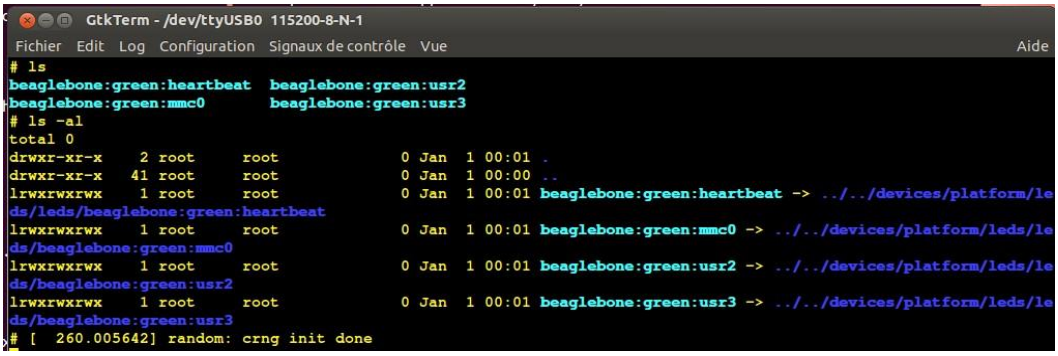
Le fichier permettant d'interagir avec les LEDs se trouve dans /sys/class/leds. Comme vous pourrez le constater, ce fichier est absent de /sys (lorsque vous démarrez avec la version de Linux présente sur la carte micro SD).

L'une des façons de l'inclure est de rajouter les « bonnes » options lors de la configuration du noyau.

Allez sur le menu de configuration du noyau et cherchez les options permettant d'exporter les LEDs dans le /sys. Explorez les options dans le menu « Device Driver/Led Support» et « Device Driver/Input Device Support »

Attention la compilation du noyau peut prendre quelques minutes. Je vous conseille très fortement de créer un script permettant de supprimer le contenu de la carte micro SD et d'y copier les nouveaux fichiers ainsi que le système de fichier. Notez qu'il n'est pas nécessaire de re formater et recréer le système de fichiers à chaque fois que vous mettez une nouvelle version du noyau sur la carte.

Une fois que vous avez compilé le noyau avec les bonnes options, vous devriez avoir un affichage ressemblant au suivant dans « /sys/class/leds » **lorsque vous démarrez la carte**.



On peut aller sur le répertoire de la première LED (usr0 ou celle indiquée beaglebon:green:heartbeat). Vous y trouverez des fichiers permettant de contrôler la LED en question. Le fichier « **trigger** » spécifie ce qui enclenche la LED. En regardant le contenu du fichier, vous remarquerez que le paramètre « [heartbeat] » est sélectionné. Le fichier « brightness » permet de contrôler l'allumage de la LED.

On peut par exemple modifier le « trigger » :

```
$ echo none > trigger
```

Vous remarquerez que la LED s'est éteinte. On peut la rallumer avec :

```
$ echo 1 > brightness
```

Ou sinon, on peut définir un mode, par exemple la LED qui s'allume en fonction de l'activité du CPU :

```
$ echo cpu > trigger
```

Ou sinon remettre la configuration de battement de cœur :

```
$ echo heartbeat > trigger
```

On peut aussi utiliser un timer, faites un :

```
$ echo timer > trigger
```

Une fois cela réalisé, vous remarquerez que 2 fichiers sont créés dans le répertoire : « delay_on » et « delay_off ». Positionnez ces délais à 200 et à 800 respectivement. Que se passe-t-il ? Inverser les valeurs pour voir ce qui se passe.

Nous souhaitons utiliser les 4 LEDS pour reproduire le mouvement de va et vient gauche →droite puis droite → gauche en continu (si l'image ci-dessous ne vous dit rien, allez voir la vidéo suivante sur youtube : https://youtu.be/jNQKHzXP14Y?t=8)

Reproduisez le mouvement des LEDs sur la calandre du véhicule grâce aux LEDs de la Beaglebone. Faites cela à partir d'un programme C. Rendez la fréquence de va-et-vient configurable.

Où est le compilateur C pour le système d'exploitation construit par buildroot ?

Vous trouverez ci-dessous un exemple de contrôle des LEDs avec un programme C

```c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv)
{
    FILE *LEDHandle = NULL;
    char *LEDBrightness = "/sys/class/leds/beaglebone:green:usr3/brightness";
    printf("\nLED USR3 blink program\n");

    while (1)
    {
        if ((LEDHandle = fopen(LEDBrightness, "r+")) != NULL)
        {
            fwrite("1", sizeof(char), 1, LEDHandle);
            fclose(LEDHandle);
        }
        sleep(1);
        if ((LEDHandle = fopen(LEDBrightness, "r+")) != NULL)
        {
            fwrite("0", sizeof(char), 1, LEDHandle);
            fclose(LEDHandle);
        }
        sleep(1);
    }
    return 0;
}
```