Laboratoire 5
Introduction aux pilotes d'interface sous Linux

ELE4205 - Département de génie électrique
Polytechnique Montréal

15 octobre 2018

Table des matières

1 Mise en contexte

Le système d'exploitation Linux offre une abstraction du matériel pour être utilisable par les logiciels. Par exemple, lorsque vous connectez un dispositif compatible USB à votre poste, celui est automatiquement détecté par le système. C'est la tâche du pilote de cacher les détails matériels.

Dans ce laboratoire, vous allez utiliser le pilote de PWM. Vous aurez la tâche de créer un programme qui va s'interfacer avec le pilote.

2 Utilisation du PWM

Connectez le buzzer sur la pin `33` (`#108`) et sur le *ground* (`GND`), tel qu'illustré sur la figure 1.

Pour l'instant, connectez-vous comme `root` et allez dans le répertoire des pilotes :

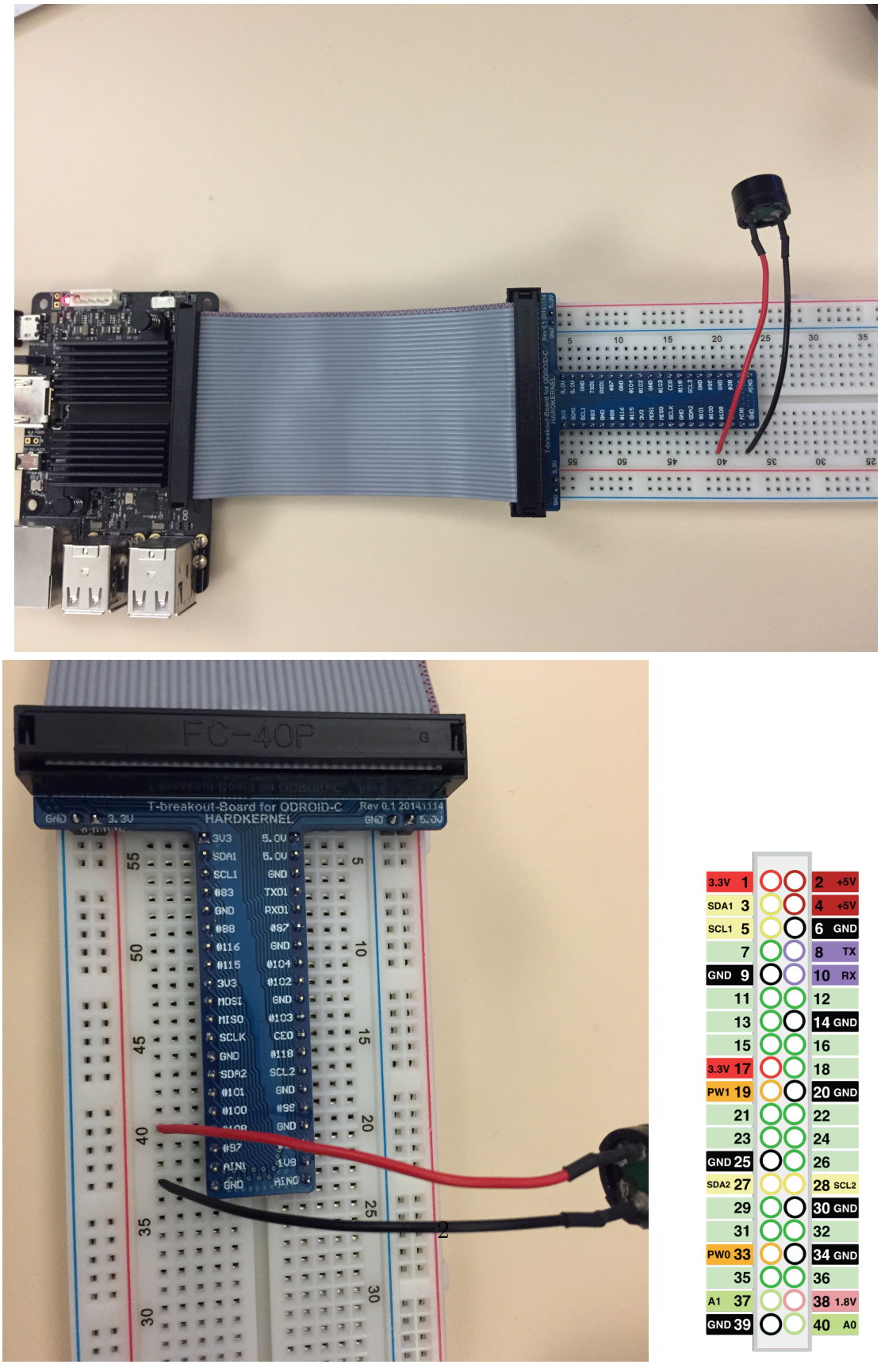

FIGURE 1: Branchement buzzer et Odroid-C2

```
$ ssh root@192.168.7.2
root@odroid-c2:~# cd /sys/devices/
root@odroid-c2:/sys/devices# ls pwm*
pwm-ctrl.42:
modalias   power      subsystem  uevent

pwm.41:
modalias   power      subsystem  uevent

duty       period     power      subsystem
```

On constate des entrées pour le pilote PWM. Nous allons maintenant l'activer :

```
root@odroid-c2:/sys/devices# modprobe pwm-meson
root@odroid-c2:/sys/devices# modprobe pwm-ctrl
root@odroid-c2:/sys/devices# ls pwm*
pwm-ctrl.42:
driver     enable0    modalias   subsystem
duty0      freq0      power      uevent

pwm.41:
driver     modalias   power      pwm        subsystem  uevent
```

Le chargement du pilote a créé trois fichiers : `duty0`, `enable0` et `freq0` ; dans le répertoire `pwm-ctrl.42`. Le fichier `duty0` peut contenir une valeur $[0, 1023]$. Par exemple, une valeur de 256 correspond à un *duty cycle* d'environ 25%. Voyons comment générer la note LA à 440 hz.

```
root@odroid-c2:/sys/devices# cd pwm-ctrl.42/
root@odroid-c2:/sys/devices/pwm-ctrl.42# echo 512 > duty0
root@odroid-c2:/sys/devices/pwm-ctrl.42# echo 1 > enable0
root@odroid-c2:/sys/devices/pwm-ctrl.42# echo 440 > freq0
root@odroid-c2:/sys/devices/pwm-ctrl.42# cat duty0
512
root@odroid-c2:/sys/devices/pwm-ctrl.42# cat freq0
440
root@odroid-c2:/sys/devices/pwm-ctrl.42# cat enable0
PWM_0 : on
```

On peut arrêter le son avec

```
root@odroid-c2:/sys/devices/pwm-ctrl.42# echo 0 > enable0
```

3 Livrable

Modification de la fonction `tone` dans le programme livré avec le labo 5.

Indices

Voici les indices pour ce livrable :

— La fonction `tone` doit jouer une note pendant la durée donnée en ms. Un exemple est donné dans `SongParser` qui affiche la note à jouer en `stdout`.

— Le fichier `song.txt` doit être dans le même répertoire que l'exécutable sinon vous aurez des erreurs d'exécution.

— Les entrées `enable0` et `freq0` du pilote PWM sont des fichiers. Vous pouvez donc les ouvrir avec la librairie standard et en modifier leur contenu.

— Si tout fonctionne, à l'exécution du programme, le buzzer devrait jouer une musique d'un classique des jeux vidéo !

4 Compilation croisée pour le développement de modules

Pour le développement de modules, il nous faut un SDK qui comprend les *kernels headers* et les règles de compilation de modules.

Pour y arriver, il nous faut ajouter la ligne

```
IMAGE_INSTALL_append = " kernel-devsrc"
```

dans notre `local.conf` et regénérer le SDK et l'installer de nouveau (voir section 2, laboratoire 2).

Une fois le SDK installé, il faut générer des scripts de configuration noyau.

```
% cd /export/tmp/4205_nn/opt/poky/sysroots/aarch64-poky-linux/usr/src/kernel/
% bash
$ source /export/tmp/4205_nn/opt/poky/environment-setup-aarch64-poky-linux
$ make silentoldconfig scripts
```

Maintenant que notre environnement est prêt, télécharger les fichiers pour la partie module de laboratoire, faites-en l'extraction et changer pour le répertoire `test_driver` et ajuster la variable `ROOTDIR` dans le fichier `Makefile` pour refléter votre installation du SDK.

On peut batir et tester le module avec les commandes suivantes :

```
$ export LDFLAGS=" "
$ make
$ scp hellokernel.ko root@192.168.7.2:/home/root/
$ ssh root@192.168.7.2
root@odroid-c2:~# insmod hellokernel.ko
root@odroid-c2:~# dmesg
...
[ 1093.988543] Hello world!
root@odroid-c2:~# rmmod hellokernel.ko
root@odroid-c2:~# dmesg
...
[ 1093.988543] Hello world!
[ 1164.722701] Cleaning up module.
```

Le module minimal compilé est fonctionnel. Nous serions prêt pour le développement d'un module plus « utile ».