

Mehdi Fracso  
Dayet Félix

## Compte rendu TP3 LO52

### 1) Merger la branche SnakeTeacher dans la vôtre afin de récupérer le répertoire Rsc-libusb

Tout d'abord, on se place dans notre branche avec la commande :

```
$ git checkout FelixDayet_MehdiFracso
```

Nous n'avons pas eu le besoin de réaliser un Git merge SnakeTeacher, car les deux dossiers étaient déjà présents.

On trouve deux répertoires :

- Libusb-1.0.3 : pour l'implémentation du composant libusb
- Device : pour l'ajout du produit

### 2) Implémentation de la libusb

Déterminez les fichiers sources et les headers pour la compilation d'une libusb sur Android

En analysant les différents fichiers, on trouve les fichiers sources (extension .c) et headers (extension .h) sous le dossier libusb :

- core.c
- descriptor.c
- io.c
- libusb.h
- libusb.h
- sync.c
- os/linux\_usbfs.c
- os/linux\_usbfs.h

Ecrire le fichier Android.mk

Pour pouvoir écrire ce fichier, nous avons dû visiter la documentation le concernant en visitant le lien suivant :

<https://android.googlesource.com/platform/ndk/+4e159d95ebf23b5f72bb707b0cb1518ef96b3d03/docs/ANDROID-MK.TXT>

Et en étudiant cette dernière, on a pu relever des points importants dont on peut citer :

- LOCAL\_PATH : indique le chemin des fichiers sources dans la branche de développement
- include \$(CLEAR\_VARS): réinitialise toutes les autres variables sauf le LOCAL\_PATH
- LOCAL\_MODULE : contient le nom du module compilé
- LOCAL\_SRC\_FILES : Liste les noms des fichiers sources à compiler par le système de build, ces fichiers seront assemblés en un module.

- `include $(BUILD_SHARED_LIBRARY)`: Fait appel à une macro qui collecte tout le paramétrage défini par les variables définies avant et que leurs noms commencent avec un `LOCAL_*`.

Voici le fichier `android.mk` que nous avons écrit :

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := libusb
LOCAL_SRC_FILES := core.c descriptor.c io.c sync.c os/linux_usbfs.c os/linux_usbfs.h
include $(BUILD_SHARED_LIBRARY)
```

Pour la compilation, nous avons utilisé la commande suivante en lui donnant les droits nécessaires :

```
$ sudo chmod 0755 configure
$ sudo ./configure
$ sudo make
```

#### Erreur sur la macro `TIMESPEC_TO_TIMEVAL`

Cette erreur est due à une absence de définition dans le fichier `io.c`, dans le repertoire `libusb`. Pour la corriger : on ajoute les lignes :

```
# define TIMESPEC_TO_TIMEVAL(tv, ts) {
    (tv)->tv_sec = (ts)->tv_sec;
    (tv)->tv_usec = (ts)->tv_nsec / 1000;
}
#endif
```

Source : <https://code.woboq.org/userspace/glibc/time/sys/time.h.html>  
Lignes 42-46

#### Erreur : `build/tools/apriori/prelinkmap.c(137) : library "libusb.so" not in prelink map`

On corrige cette erreur en ajoutant l'adresse de la librairie `libusb` dans le fichier `build/tools/apriori/prelinkmap`.

```
# extended system libraries
```

```
libusb.so 0xAE600000
```

### 3) Implémentation d'un nouveau produit Android

Produit : Lo52\_FelixDayet\_MehdiFracso

Propriété personnalisées :

ro.hw=lo52 net.dns1 = 8.8.8.8 net.dns2 = 4.4.4.4

Voici le fichier que nous avons créé:

```
$(call inherit-product, device/linaro/hikey/hikey-common.mk)
```

```
PRODUCT_PACKAGES += libusb
```

```
PRODUCT_PROPERTY_OVERRIDES := /
```

```
ro.hw=lo52 net.dns1 = 8.8.8.8 net.dns2 = 4.4.4.4
```

```
DEVICE_PACKAGE_OVERLAYS := device/utbm/Lo52_FelixDayet_MehdiFracso/overlay
```

```
PRODUCT_NAME := Lo52_FelixDayet_MehdiFracso
```

```
PRODUCT_BRAND := Lo52_FelixDayet_MehdiFracso
```

```
PRODUCT_DEVICE := Lo52_FelixDayet_MehdiFracso
```

```
PRODUCT_MODEL := Lo52_FelixDayet_MehdiFracso
```

Lien utilisé :

<https://android.googlesource.com/device/linaro/hikey/+/refs/heads/master/hikey.mk>