

Module d'approfondissement SEEE

Systèmes d'exploitation et environnements d'exécution embarqués
Prof. Daniel Rossier / Magali Fröhlich

Introduction REPTAR

Labo reptar1 (v2017.1.8)

Ce laboratoire vous permettra de vous familiariser avec l'environnement REPTAR, aussi bien dans sa version **émulée** que **réelle**. Il permet également de découvrir les nombreux utilitaires logiciels permettant de gérer des environnements complexes destinés aux systèmes d'embarqués.

Le gestionnaire de version utilisé dans le cadre de ces laboratoires est **git**.

Consignes de laboratoire

- Les manipulations 1 à 4 de ce laboratoire sont individuelles; les manipulations 5 et 6 s'effectueront en binôme.
- Vous disposez d'un guide d'utilisation de l'infrastructure pour les laboratoires SEEE. Vous trouverez l'aide nécessaire à l'utilisation des différents outils (*git*, *eclipse*, etc.).

Manipulation no. 1 - Mise en place de l'environnement, utilisation de *git*

L'environnement de développement utilisé dans ce cours est l'IDE **Eclipse**. Un *workspace* Eclipse a été préconfiguré dans le dépôt *git* (fichiers dans le répertoire caché `.metadata/` à la racine).

a) Il faut tout d'abord récupérer le dépôt étudiant pour les laboratoires SEEE à l'aide de la commande suivante (via une fenêtre de *terminal*) :

```
$ git clone firstname.lastname@eigit.heig-vd.ch:/home2/reds/seee/seee_student
```

Le nom du *workspace* est `seee_student` et correspond au répertoire cloné.

b) Lancez Eclipse et ouvrez le *workspace* `seee_student`. Vous devriez obtenir la liste des projets (à gauche). Chaque projet a un lien symbolique dans la racine du *workspace*.

c) Compilez maintenant l'émulateur Qemu. Dans une fenêtre de terminal, lancez la commande suivante à partir de votre répertoire `seee_student` :

```
$ make qemu
```

Manipulation no. 2 – Démarrage de *Qemu* (environnement émulé)

a) Depuis Eclipse, lancez le *debugger* avec la configuration de *debug* « *qemu-reds Debug* ». Dans la fenêtre *Console*, vous pourrez entrer directement des commandes de U-boot (tapez *help* par exemple).

b) Interrompez l'exécution du programme en cliquant sur l'icône *pause*. Identifiez la ligne en cours d'exécution dans le code source.

c) Stoppez l'exécution, et dans une fenêtre de commande, démarrez *qemu* à l'aide du script *stf* (en tapant `./stf`) dans le répertoire racine. Vous arrivez dans *U-boot*.

Manipulation no. 3 – Tests avec U-boot

- a) Dans U-boot, listez les variables d'environnement avec la commande **printenv**. Observez les variables prédéfinies « *tftp1*, *tftp2* et *goapp* ». Ces variables définissent des commandes U-boot qui peuvent être exécutées à l'aide de la commande *run* (par exemple *run tftp1*). La commande **go <addr>** permet de lancer l'exécution à l'adresse physique <addr>. Vous pouvez **définir/modifier vos propres variables** et les sauvegarder dans la *flash* émulée avec la commande **saveenv** (seulement avec le lancement via *stf*).
- b) La production de l'exécutable *helloworld_u-boot* s'effectue en tapant la commande *make* dans le répertoire contenant les sources du programme. Ensuite, vous pouvez transférer le fichier (extension *.bin*) dans U-boot et exécuter le binaire (aidez-vous des variables d'environnement prédéfinies).
- c) Testez le *debugger* dans Eclipse avec le projet *helloworld_u-boot*. Mettez un **breakpoint** dans le code source au démarrage du programme, et lancez le *debugger* avec la configuration de *debug* « *helloworld_u-boot Debug* ».

Manipulation no. 4 – Tests avec Linux

Le démarrage de Linux dans l'environnement émulé peut s'effectuer à l'aide des scripts *stf*.

- a) Lancez le script *./deploy* qui permettra de déployer le noyau Linux dans la *sdcard* virtuelle (ignorez l'erreur due à l'absence de certains fichiers).
- b) Poursuivez ensuite en **cross-compilant** l'application *helloworld* pour Linux (via *make*).
- c) Copiez l'exécutable dans le **rootfs**. Pour cela, il faut:
1. Monter l'image de la *sdcard* à l'aide du script **mount-sd.sh** (mot de passe: **reds**)
 2. Transférer l'application dans le répertoire */root* du **rootfs** (monté sur *filesystem_tmp*)
 3. Démonter l'image avec le script **umount-sd.sh**

```
$ ./mount-sd.sh
$ sudo cp helloworld_linux/helloworld filesystem_tmp/root
$ ./umount-sd.sh
```

- d) Lancez le script *stg* suivi de la commande *boot* dans U-boot pour amorcer le démarrage de *Linux*.
Login : root Password : root
- e) Lancez votre application
- f) Dans *Linux*, tapez la commande suivante :

```
$ /usr/share/qt/examples/effects/lighting/lighting -qws &
```

Manipulation no. 5 – Tests sur la plate-forme réelle

a) Dans un premier temps, il est nécessaire d'installer le serveur *tftp* qui vous servira à déployer les images sur la plateforme.

```
$ ./install_tftp_server
```

Indications

1. Pour se connecter à l'interface série de la carte *REPTAR*, il faut au préalable rediriger le convertisseur USB-série du *host* vers la machine virtuelle. Référez-vous au chapitre « Activation du port série sous Virtualbox » du document « *Guide d'utilisation de l'infrastructure des salles A07/A09* ».
2. Le répertoire racine du serveur TFTP tournant sur la VM est */home/reds/tftpboot*
3. L'adresse IP de la carte *REPTAR* est 192.168.1.254

a) Déployez l'application *helloworld* dans U-boot sur la plate-forme *REPTAR* avec l'interface réseau. Le transfert peut s'effectuer avec la commande *tftp*.

Il est nécessaire d'exécuter la commande suivante pour mettre à jour les adresses IP et MAC de la plate-forme *REPTAR* :

```
# run setmac setip
```

b) Déployez l'application *helloworld* dans Linux à l'aide du réseau et de la commande *scp*. La commande *run boot mmc* démarre Linux. Login : root Password : root

Manipulation no. 6 – Accès aux périphériques REPTAR

Le projet à utiliser dans cette manipulation est le projet *gpio_u-boot*. Les *GPIOs* ont déjà été préconfigurées par U-boot.

Sur la base de l'exemple ci-dessus, développez une application permettant d'interagir avec les *LEDs* et les *switchs* présents sur la carte *CPU* de la plate-forme *REPTAR*.

Le but de l'application est d'allumer une *LED* lorsqu'on appuie sur un *switch*.

- La *LED 0* doit s'allumer lorsqu'on appuie sur le *SWITCH 0*.
- La *LED 1* s'allume si l'on appuie sur le *SWITCH 1*.
- Et ainsi de suite pour les *LEDs* et *switchs* 0 - 3 de la carte *CPU*.

Le *switch* numéro 4 sert à quitter l'application. Aidez-vous des fichiers d'en-tête (*#include*) déjà présents dans le chablon fourni.

L'application *gpio_u-boot* est à déployer dans U-boot via la commande *tftp*.

Indications

- Les fonctions *gpio_set_value()* et *gpio_get_value()* permettent d'écrire et de lire une valeur sur une *GPIO* donnée.
- Les constantes concernant les *LED* et *switches* sont définies dans le fichier *board/ti/reptar/reptar.h*
- A titre d'exemple, la *LED 0* est associée à la constante *GPIO_LED_0*, le bouton *switch 0* à la constante *GPIO_SW_0*.