S. Lavirotte
Polytech'Nice – Sophia
SI4
2018-2019

TD n° 11
Architectures mixtes

Pour ce TD, nous allons mettre en place une architecture mixte qui utilise à la fois un microprocesseur et un micro-contrôleur.

Pour information, pour éviter les problèmes de copier-coller, vous trouverez les codes présents dans ce document à l'adresse :

```
http://trolen.polytech.unice.fr/cours/isle/td11/code.txt
```

1 Architecture mixte

Pour tirer le meilleur parti des deux types de processeurs, nous allons mettre en place une architecture mixte à savoir une architecture utilisant à la fois un microprocesseur et un microcontrôleur. Ce type de système est utilisé quand on a besoin d'avoir accès à des capteurs qui nécessitent de temps de mesure constants et connus et à la fois un traitement de certaines tâches lourdes et/ou complexes. Pour ce sujet de TD, nous allons donc mettre en place une architecture mixte basée sur :

- une carte Raspberry Pi équipée d'un microprocesseur et donc d'un système d'exploitation
- une carte fille GrovePi+ équipée d'un microcontrôleur et d'un ensemble de connecteurs pour y connecter des capteurs.

Nous allons tout d'abord installer un système d'exploitation sur la carte à base de microprocesseur avant l'installer le programme sur microcontrôleur pour réaliser la lecture des données de capteurs et envoyer ces informations aux programmes fonctionnant sur le microprocesseur.

2 Carte à base de microprocesseur

2.1 Installation du système Raspbian

Nous avons vu lors des séances précédentes comme créer un système sur mesure pour votre plate-forme embarquée. Pour gagner du temps, nous allons cette fois-ci récupérer un système déjà prêt à l'emploi pour la Raspberry Pi : la distribution Raspbian.

Commencer par récupérer la distribution Raspbian Lite (la plus compacte qui n'inclut ni interface graphique, ni applicatifs de bureautique comme c'est le cas de l'autre distribution Raspbian) à l'adresse suivante :

```
https://www.raspberrypi.org/downloads/raspbian/
```

Cette distribution est basée sur la Debian et permet donc le déploiement sur la plateforme de nouveaux applicatifs à l'aide de paquetages.

Une fois l'image récupérée et décompressée, utilisez l'application adaptée pour inscrire le système sur la carte SD.

2.2 Configuration minimale du système de base

Sur cette distribution, il y a un serveur SSH qui est installé et est fonctionnel. Toutefois, celui n'est pas activé par défaut. Pour l'activer, il est nécessaire de créer un fichier `ssh` (sans extension) sur la première partition de la carte SD, une fois celle-ci flashée.

De plus, il y a un serveur `avahi` qui permet d'adresser la plateforme via son nom de machine (`hostname`) qui par défaut est `raspberrypi`. Vous pouvez donc vous connecter à distance via la commande (il vous faut avoir installé le service Bonjour sur votre machine cliente) :

```
ssh pi@raspberrypi.local
```

Le mot de passe de l'utilisateur `pi` par défaut est `raspberry`.

TD n° 11
Architectures mixtes

En tant que super-utilisateur, lancez l'application `raspi-config` pour étendre la partition sur l'ensemble de la carte SD (*Expand Filesystem*). Modifiez le nom de votre plateforme (`hostname`) pour avoir un nom unique sur le réseau local.

Un redémarrage de la plateforme est nécessaire pour prendre en compte le redimensionnement de la partition racine du système de fichiers. Vous pouvez vous connectez à la plateforme via son nouveau nom : `ssh pi@new_name.local`

Il ne vous reste plus qu'à mettre à jour la liste des paquetages disponibles (mais nous ne le ferons pas durant ce TD pour éviter de perdre du temps) :

```
sudo apt-get update
```

3 Carte fille à base de microcontrôleur

3.1 Installation des applicatifs GrovePi

Il est tout d'abord nécessaire de récupérer et d'installer un ensemble de programmes et outils pour communiquer avec la carte GrovePi.

```
cd
mkdir Desktop
sudo curl -kL dexterindustries.com/update_grovepi | bash

sudo apt-get install git
cd Dexter
git clone https://github.com/DexterInd/GrovePi
cd GrovePi/Script
sudo chmod +x install.sh
sudo ./install.sh
sudo reboot
```

Après le redémarrage de la plateforme, pour vérifier que celle-ci est bien détectée, il suffit de réaliser la commande suivante :

```
sudo i2cdetect –y 1
```

Nous voici maintenant avec un système pouvant communiquer avec la plateforme GrovePi. Il faut maintenant mettre un programme sur le microcontrôleur pour pouvoir accéder aux valeurs des capteurs que l'on connectera.

3.2 Flasher un programme pour le traitement des entrées/sorties

Pour tester que le programme reconnaît bien le port `gpio` pour faire le flashage de ATmega, lancez la commande suivante.

```
sudo avrdude -c gpio -p m328p -v –t
```

Vous vous retrouvez avec le prompt de commande `avrdure`. Pour quitter proprement ce programme, il suffit de taper la commande `quit`.

3.2.1 Installation de l'environnement de développement pour microcontrôleur

Nous devons maintenant installer l'environnement de développement (compilateur et bibliothèques) pour le micro-contrôleur.

```
sudo apt-get install arduino-core
```

3.2.2 Compilation du firmware

L'étape suivante est de pouvoir produire son propre programme pour l'ATmega328 de la plate-forme GrovePi. Ceci pourra être utile pour produire un firmware qui soit adapté à nos propres besoins ou tout simplement pour produire le tout dernier firmware à partir des sources de GrovePi. Pour la production du firmware, il est recommandé d'utiliser `ino` qui est un constructeur automatique de projet. Celui-ci définit automatiquement les dépendances en fonction des `#include` dans les sources.

Pour installer `ino`, il suffit de lancer la commande suivante :

```
sudo pip2 install ino
```

La création d'un nouveau projet et la compilation de celui-ci est réalisé de la manière suivante :

```
mkdir firmware && cd firmware
ino init ; # utilisé l'option -t comme dans la documentation GrovePi provoque une
erreur car le template n'est pas connu
rm src/sketch.ino
cp -a ~/Dexter/GrovePi/Firmware/Source/grove_pi/* src/
ino list-models
ino clean
ino build -m atmega328
```

3.2.3 Mise à jour du firmware sur le microcontrôleur

Une fois le fichier généré, il ne reste plus qu'à le flasher sur la plate-forme GrovePi en utilisant le script fournit dans les sources de GrovePi/Firmware :

```
cp .build/atmega328/firmware.hex ~/Dexter/GrovePi/Firmware/grove_pi_firmware.hex
cd ~/Dexter/GrovePi/Firmware
sudo ./firmware_update.sh
```

Une fois la mise à jour du programme effectuée, si la LED rouge RST sur la plate-forme GrovePi est allumée, il est nécessaire d'arrêter la Raspberry Pi, de la débrancher de son alimentation avant de la rebrancher.

4 Création d'une application utilisant les données des capteurs

4.1 Programme exploitant cette architecture mixte

Suivant le capteur qui vous a été fourni, trouvez un programme pour tester celui-ci dans le dossier :

```
cd ~/Dexter/GrovePi/Software/Python
```

4.2 Librairie exploitant l'accès aux informations fournies par la plateforme GrovePi

- Quelle librairie votre programme de test utilise-t-il ?
- Quelle est le type de bus de données que votre capteur utilise ?
- Quelles sont les fonctions qui permettent de récupérer les valeurs de ce capteur ?
- Et pour un capteur utilisant un autre bus de données ou d'un autre type (analogique/numérique) ?

5 Analyse du fonctionnement

Maintenant que nous avons mis en place et testé cette architecture, nous allons tenter de comprendre comment cela fonctionne.

TD n° 11
Architectures mixtes

5.1 Compréhension du mode de fonctionnement cette architecture mixte

Le code source du programme firmware que nous avons mis sur le microcontrôleur de la carte GrovePi se trouve dans le dossier suivant :

```
cd ~/Dexter/GrovePi/Firmware/Source/grovepi
```

A partir de ce code source, de la librairie `grovepi.py` se trouvant dans ce dossier et des informations disponibles sur les pages suivantes, déduisez-en le mode de fonctionnement de la communication des données dans cette architecture pour la mise à jour du firmware et pour la communication entre Raspberry Pi et GrovePi pour lire les données des capteurs.

http://www.dexterindustries.com/GrovePi/engineering/port-description/
http://www.dexterindustries.com/GrovePi/engineering/software-architecture/

5.2 Justification de cette architecture mixte

Nous allons prendre l'exemple d'un capteur spécifique : Ultrasonic Ranger.

http://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/

Ce capteur permet de mesurer une distance comprise en 3 et 400cm. Son mode de fonctionnement est le suivant : le capteur émet un son à 42KHz et se met à l'écoute de la réception de ce signal après réverbération sur une surface. La mesure de la distance est donc calculée à partir du temps que mets le signal pour revenir vers le capteur en fonction de la vitesse du son. Le temps mesuré étant le temps écoulé entre l'émission du signal et sa réception, le son a parcouru deux fois la distance (aller et temps retour après réverbération sur la surface).

Dans le code source de la librairie python, trouvez le code correspondant au capteur Ultrasonic ranger.

- Pourquoi ce capteur a-t-il une fonction dédiée particulière dans la librairie ?

Trouvez dans le firmware que nous avons mis sur l'ATmega si un code particulier est bien utilisé pour le capteur Ultrasonic Ranger.

- Quel est le code correspondant à ce capteur ?
- Pouvez-vous expliquer le fonctionnement de ce code ?
- Pourquoi n'est-t-il pas possible de faire fonctionner un code similaire directement en Python ou en C sur le micro-processeur de votre architecture ?
- Qu'est ce qui peut justifier de mettre en place une architecture mixte microprocesseur/microcontrôleur ?