

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 microcontrôleur.

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>

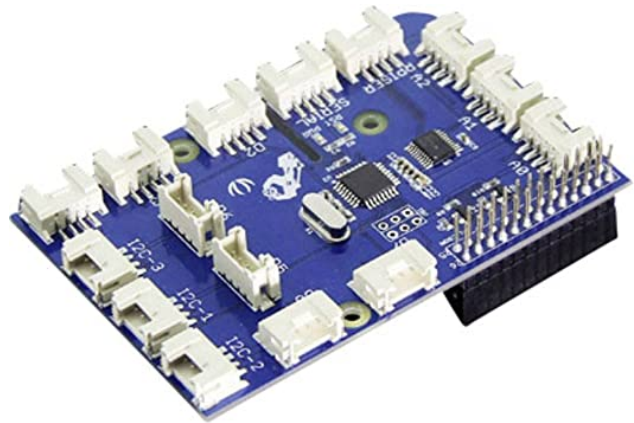
1 Architecture mixte

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.



Raspberry Pi 3 avec un SoC Broadcom BCM2837



GrovePi+ avec un microcontrôleur ATmega328

Nous allons tout d'abord installer un système d'exploitation sur la carte à base de microprocesseur avant l'installer le programme sur le 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 (Raspberry Pi)

2.1 Installation du système Raspbian

Nous avons vu lors des séances précédentes comment 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 bureau) comme c'est le cas des autres distributions Raspbian) à l'adresse suivante :

<https://www.raspberrypi.org/software/operating-systems/>



TD n° 11

Architectures mixtes

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 (balenaEtcher ou win32diskimager par exemple).

2.2 Configuration pour une connexion Wifi

Vous pouvez bien sûr connecter la Raspberry sur votre réseau filaire mais aussi en WiFi. Pour configurer la connexion au réseau Wifi, avant de démarrer l'image, il est nécessaire d'ajouter sur la partition de boot de la carte SD, un fichier `wpa_supplicant.conf` avec le contenu suivant:

```
country=fr
update_config=1
ctrl_interface=/var/run/wpa_supplicant

network={
    scan_ssid=1
    ssid=" "
    psk=" "
}
```

Dans le paramètre `ssid`, mettre le nom du Wifi auquel vous souhaitez vous connecter.

Dans le paramètre `psk`, mettre le mot de passe de connexion à ce WiFi.

Attention à ne pas connecter votre Raspberry Pi en filaire à votre PC qui devrait alors être configuré spécialement pour router le trafic de la Raspberry Pi à Internet, ce qui sera nécessaire pour le TD.

2.3 Autorisation de la connexion au serveur SSH

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

2.4 Configuration minimale du système de base

Maintenant que votre carte SD est prête à l'emploi et correctement configurée, il est temps de démarrer le système.

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`.

En tant que super-utilisateur, lancez l'application `raspi-config` pour modifier le nom de votre plateforme (`hostname`) pour avoir un nom unique sur le réseau local (dans le cas où il y a plusieurs Raspberry Pi installée de cette manière sur votre réseau local. Un redémarrage est nécessaire pour prendre en compte le changement de nom de la plateforme. Vous pouvez vous connecter 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 update
```



TD n° 11

Architectures mixtes

3 Carte fille à base de microcontrôleur (GrovePi+)

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 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
```

Vous devez voir le résultat suivant :

```
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- 04 -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- --
```

Si ce n'est pas le cas, c'est que votre plateforme GrovePi+ n'est pas détectée par votre Raspberry Pi.

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 (il doit y en avoir un par défaut, mais nous allons quoi qu'il arrive le mettre à jour avec la version disponible).

3.2 Flasher un programme pour le traitement des entrées/sorties

Pour tester que le programme reconnaît bien le port `gpio` pour flasher le microcontrôleur ATmega (pour information, c'est le même que pour l'Arduino que vous avez manipulé précédemment), lancez la commande suivante :

```
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`.

Regardez sur la carte GrovePi+ : vous devez voir la petite LED verte allumée (signe que la plateforme est bien alimentée) et la petite LED rouge juste à côté qui ne l'est pas (c'est la reset LED ; elle doit être éteinte). Si ce n'est pas le cas, vous pouvez utiliser la commande suivante pour resetter le microcontrôleur :

```
avrdude -c gpio -p m328p
```

Le programme `avrdude` qui s'exécute sur la Raspberry Pi est donc maintenant capable de contacter la carte GrovePi+ afin de flasher un nouveau firmware sur le microcontrôleur.

3.2.1 Mise à jour du firmware sur le microcontrôleur

Un script est disponible pour flasher le firmware sur la carte GrovePi+ :

```
cd ~/Dexter/GrovePi/Firmware
sudo ./firmware_update.sh
```

TD n° 11

Architectures mixtes

Ceci aura pour effet de flasher le programme `grove_pi_firmware.hex`.

Une fois la mise à jour du programme effectuée, si la LED rouge RST sur la carte GrovePi+ est allumée (c'est sûrement qu'il y a eu un problème), utiliser la commande suivante pour resetter le microcontrôleur :

```
avrdude -c gpio -p m328p
```

4 Application sur la Raspberry Pi utilisant les données des capteurs connectés à la GrovePi+

4.1 Programme exploitant cette architecture mixte

Vous avez plusieurs exemples fournis avec la distribution GrovePi, dans différents langages. Les exemples les plus développés sont en Python (premier à avoir été développé pour la GrovePi).

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

Je vous invite à aller tester avec une LED avec le programme `grove_led_blink.py`.

Modifiez le programme `grove_button.py` pour que l'appui sur le bouton déclenche la LED (et si on relâche celui-ci, la LED s'éteint).

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

Pour mettre en œuvre votre petit programme de test, vous n'avez pas tout écrit, vous avez utilisé des fonctions préexistantes.

- Quelle bibliothèque votre programme de test utilise-t-il ?
- Quelle est le type d'entrée/sortie utilisé par la LED et le bouton ?
- Quelles sont les fonctions qui permettent de récupérer ou envoyer les valeurs à ces capteurs/actionneurs ?
- Et pour le capteur potentiomètre (Rotary angle sensor), quel type d'entrée est utilisé ? Quelle fonction est utilisée pour lire la valeur du capteur ?

5 Analyse du fonctionnement de cette architecture

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

5.1 Compréhension du mode de fonctionnement de 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 bibliothèque `grovepi.py` et des informations disponibles sur les pages suivantes, déduisez-en le mode de fonctionnement de la communication pour lire/écrire les données dans cette architecture pour la communication entre Raspberry Pi et GrovePi+.

<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/



TD n° 11

Architectures mixtes

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).

Trouvez le programme de test Python qui permet de lire les valeurs de ce capteur.

- Quelle méthode est utilisée pour récupérer la valeur de ce capteur Ultrasonic Ranger.

Dans le code source de la bibliothèque Python (`grovepi.py`), trouvez le code correspondant à cette fonction appelez du programme de test.

- Quelle(s) méthode(s) (fonction, bus de communication) est(sont) utilisée(s) pour récupérer la valeur de ce capteur ?

Trouvez dans le firmware mis sur le microcontrôleur si un code particulier est utilisé pour le capteur Ultrasonic Ranger (consultez le fichier `grovepi.ino` dans les sources du firmware).

- Quel est le code correspondant à la mesure de distance pour 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 ?
- Serait-il possible d'utiliser le capteur potentiomètre en le connectant directement à la Raspberry Pi ? Justifiez.

6 Changer le firmware sur le microcontrôleur

Aujourd'hui, il existe un certain nombre de capteurs et d'actionneurs spécifiques et nécessitant un traitement comme c'est le cas pour le capteur Ultrasonic Ranger que nous venons d'étudier. Mais si demain un nouveau capteur arrive sur le marché, il faudra mettre à jour le firmware sur le microcontrôleur. C'est ce que nous allons voir dans cette section.

6.1 Installation de l'environnement de développement pour microcontrôleur

Comme nous l'avons vu précédemment, il n'est pas raisonnable de travailler sur la cible embarquée pour compiler ou produire le programme que l'on souhaite (temps de compilation long, environnement pas adapté, ...)

Nous allons donc dans un premier temps mettre à jour notre machine virtuelle travail pour compiler le nouveau firmware à mettre sur le microcontrôleur.

L'IDE permettant facilement de compiler le firmware GrovePi est PlatformIO qui est un plugin de Visual Studio Code. Voici quelques informations pour l'installer sur votre machine de travail.

```
apt install libnotify4 libsecret-common libsecret-1-0 python3-pip git
```

Après avoir téléchargé le dernier package de Visual Studio Code, installez-le avec une commande du type :

```
dpkg -i code_1.44.2-1587059832_amd64.deb
```

Vous exécuterez ensuite en tant que super utilisateur :

```
chown user:user /run/user/*  
xhost +
```

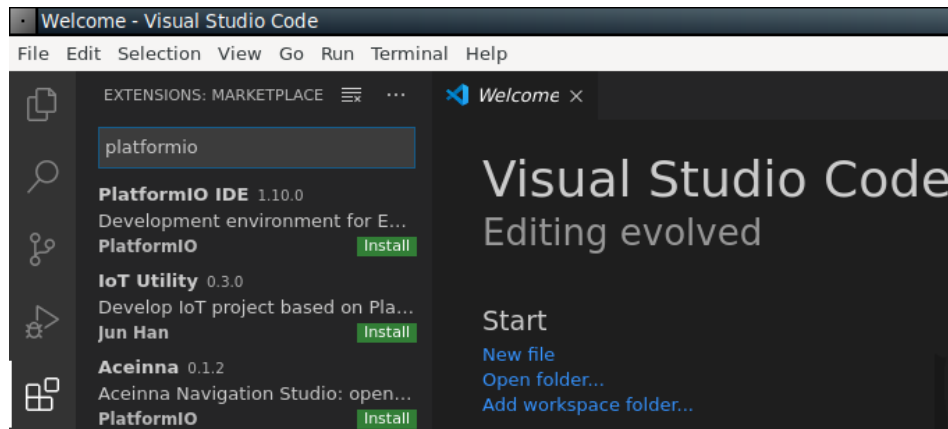
Puis dans un terminal en tant que user (Visual Studio Code n'aime pas être exécuté en tant que super utilisateur) :

```
code
```

Dans l'interface, ajoutez le module PlatformIO comme indiqué ci-dessous :

TD n° 11

Architectures mixtes

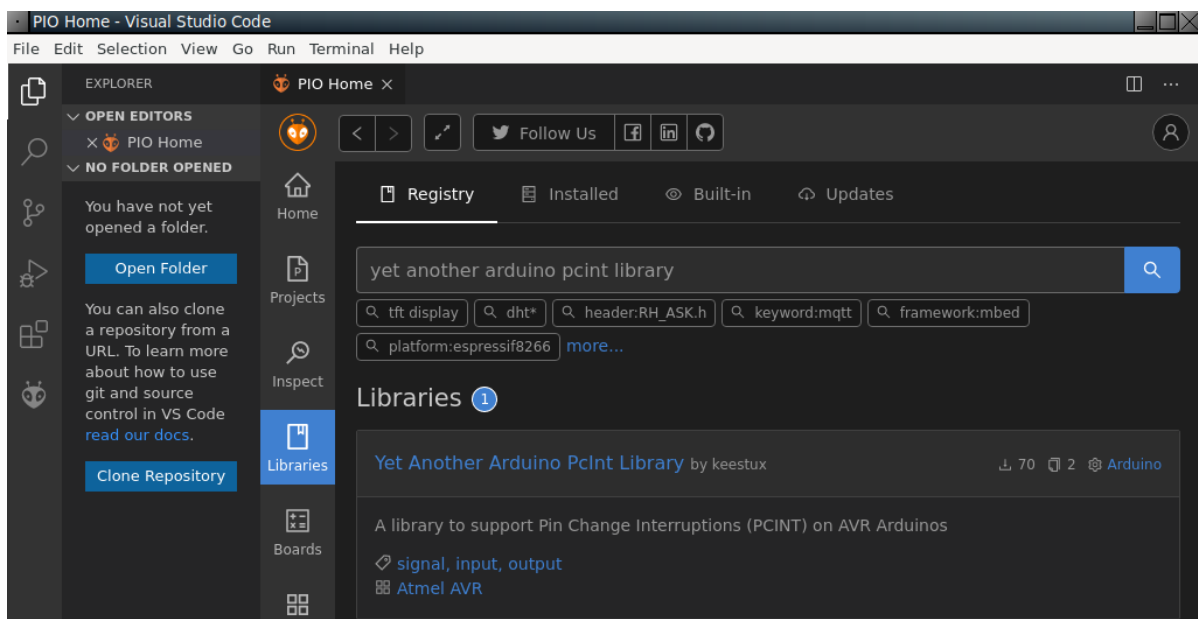


Il faut bien entendu récupérer les sources du firmware. Depuis votre terminal en tant que user :

```
cd /work/td11
git clone https://github.com/DexterInd/GrovePi.git
```

Chargez le projet à l'aide de PlatformIO, de la manière suivante « Add Existing » et sélectionnez le dossier /work/td11/GrovePi/Firmware/Source/grovepi.

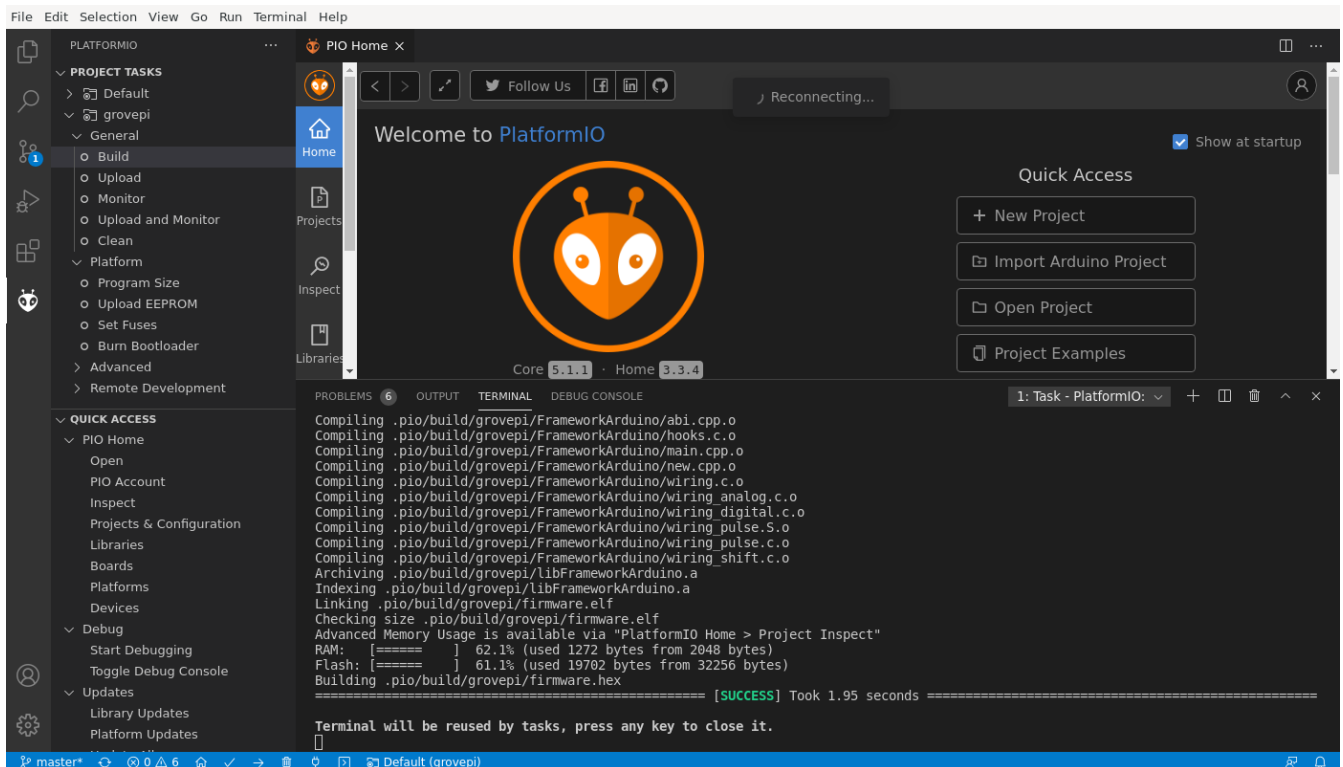
Vous devez installer une bibliothèque qui sera nécessaire pour la compilation du firmware : « Yet Another Arduino Pcnt Library » et l'ajouter à votre projet :



Il ne vous reste plus qu'à compiler le tout en allant sur l'icône de platformIO et en sélectionnant « grovepi / General / Build ».

TD n° 11

Architectures mixtes



Cela générera le firmware dans le dossier `.pio/build/grovepi/firmware.hex` du dossier `grovepi`.

6.2 Flasher le nouveau firmware sur le microcontrôleur

Vous copierez ce fichier sur la Raspberry Pi :

```
scp firmware.hex pi@raspberrypi.local:
```

Connectez-vous à la Raspberry Pi. Remplacez le fichier `Dexter/GrovePi/Firmware/grove_pi_firmware.hex` par le nouveau firmware que vous venez de produire :

```
cd ~/Dexter/GrovePi/Firmware
mv grove_pi_firmware.hex grove_pi_firmware.hex.sav
cp ~/firmware.hex grove_pi_firmware.hex
```

Il ne vous reste alors plus qu'à flasher celui-ci grâce au script comme vous avez pu le faire dans la section 3.2.1.

Vous venez de mettre à jour le programme sur le microcontrôleur avec éventuellement la nouvelle fonctionnalité que vous avez ajoutée. Il vous faudra alors aussi modifier la bibliothèque `grovepi.py` pour accéder à cette nouvelle fonctionnalité implémentée.