

TP Entrées-sorties (2)

Bus I2C & Raspberry

Module F.A.S. - IG3 - Polytech Montpellier

Nous allons maintenant connecter, à la Raspberry Pi, du matériel (des modules) autre que l'écran LCD. Choisissez-en un dans la liste suivante :

- Bouton
- Led de différentes couleurs
- Capteur de distance à ultra-sons
- Capteur de température/humidité
- Capteur de luminosité
- Capteur d'intensité acoustique (sonore)
- Capteur de mouvement (PIR Motion)
- Potentiomètre
- Buzzer (ne pas abuser de l'utilisation de ce dernier, car il fait un son horrible).

Sur le site Web suivant, vous avez plein d'exemples d'utilisation de ces modules matériels, à partir de programmes Python (faites attention à ne pas copier les exemples de programmes écrits pour l'Arduino, donnés souvent en premier) :

<http://wiki.seeedstudio.com> (Menu : Grove > Sensor/Actuator)

Ex : http://wiki.seeedstudio.com/Grove-Sound_Sensor/

Pour tester ces programmes nous aurons besoin du module Python (ne pas confondre « module matériel » et « module logiciel Python ») grovepi, qui est le driver du système Grove (le pont/*bridge* fixé sur la raspberry + les capteurs/actionneurs que l'on connecte dessus).

Voilà l'exemple d'un programme Python qui manipule un capteur d'intensité sonore :

```
import time
# Importer le driver pour manipuler les modules Grove (capteurs/actionneurs)
import grovepi

# Connecter le capteur au port analogique A1 du pont Grove
sound_sensor = 1

# On indique le sens de transfert des données (INPUT)
grovepi.pinMode(sound_sensor,"INPUT")

while True: # Boucle d'attente active
    try:
        # Lire le niveau d'intensité sonore
        sensor_value = grovepi.analogRead(sound_sensor)
        # Pour un capteur connecté à un port digital : grovepi.digitalRead(...)
        # Pour des actionneurs : analogWrite(...) et digitalWrite(...)

        print("sensor_value = %d" %sensor_value)
        time.sleep(.5) # Faire une pause d'1/2 seconde avant de continuer

    except IOError:
        print ("Error")
```

Dans ce programme, notez la boucle d'attente active (while True) qui, à l'exécution, va attendre, de façon continue, que quelque chose se passe dans l'environnement des modules (l'utilisateur appuie sur un bouton, ...).

Pour exécuter le programme (après l'avoir copié dans un fichier .py bien sûr :-), téléchargez le driver grovepi.py sur la page Web Moodle du cours et placez-le à côté de votre programme (votre_programme.py).

Exécutez le programme : python votre_programme.py

Pour interrompre l'exécution du programme, Ctrl-C

Testez maintenant l'utilisation de deux capteurs/actionneurs dans un même programme. Concertez vous (entre binômes) pour l'utilisation du matériel.

Exemples :

1. lorsqu'on appuie sur un bouton, une led s'allume ou un buzzer sonne une fois
2. lorsque l'intensité sonore ou lumineuse dépasse un certain seuil, une led s'allume
3. lorsque la distance qui sépare un objet du capteur à ultra-sons atteint un certain seuil, un buzzer sonne

...

Si le temps le permet, écrivez un programme Python qui simule un radar de recul (vous utiliserez un capteur de distance à ultra-sons, un buzzer et une led). Lorsqu'un objet se rapproche du capteur à ultra-sons, on doit faire clignoter la led et faire sonner le buzzer. Au fur et à mesure que l'objet s'approche du capteur, on fait clignoter la led et sonner le buzzer plus souvent.