

# TP Bibliothèques logicielles

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

### 1. Écrire une bibliothèque pour l'écran LCD

- Reprendre le fichier `driverI2C.py` (TP entrées-sorties) ;
- Le compléter (si vous n'aviez pas eu le temps de le faire) ;
- Rajouter une fonction `setColor` prenant les noms de quelques couleurs (rouge, vert, blanc, etc.) en paramètre ;
- Faire en sorte que la fonction `setText` gère l'affichage de texte trop long (> 32 caractères) de deux façons différentes : i) mettre en pause pendant 1 à 2 secondes, effacer entièrement l'écran et écrire les 32 caractères suivants, et ii) effacer la première ligne (de 16 caractères), la remplacer par la deuxième et écrire les caractères suivants sur la deuxième ligne (scrolling) ;
- Rajouter une fonction `echoing()` qui affiche chaque caractère saisi sur le clavier directement sur l'écran (utiliser les deux instructions : `tty.setraw(sys.stdin.fileno())` `c=sys.stdin.read(1)` au lieu de `c=input(...)` ou `c=raw_input(...)`) ;
- Transformer le fichier en une bibliothèque appelée `lcd` ;
- Faire un fichier `testLcd.py` qui teste cette bibliothèque.

### 2. Écrire une bibliothèque pour la LED et le buzzer

- Créer un fichier `ledBuz.py` ;
- Dans ce fichier, on utilisera la bibliothèque `grovepi` ;
- Écrire une fonction `init(pin)` qui sélectionne le pin sur lequel la led/buzzer sont connectés (le pin doit être en mode "OUTPUT") ;
- Écrire une fonction `turnOn` qui allume la led/buzzer ;
- Écrire une fonction `turnOff` qui éteint la led/buzzer ;
- Écrire une fonction `blink` qui allume et éteint la led/buzzer, et prend en paramètre le temps entre les états allumé et éteint ;
- Transformer ce fichier en une bibliothèque appelée `ledBuz`.

### 3. Écrire une bibliothèque pour le capteur de distance ultrason

- Créer un fichier `ultrasonic.py` ;
- Dans ce fichier, on utilisera la bibliothèque `grovepi` ;
- Écrire une fonction `init(pin)` qui sélectionne le pin sur lequel le capteur est connecté ;
- Écrire une fonction `surveille()` qui renvoie `True` si un objet est trop près (distance seuil indiquée en paramètre)

### 4. Faire la même chose pour les capteurs de son et de lumière

### 5. Écrire une bibliothèque pour le capteur de température/humidité

### 6. Grouper les différentes fonctions sur les capteurs dans une même bibliothèque : `capteurs.py`

### 7. Faire de ces bibliothèques un package Python utils (répertoire avec `__init__.py`) et le tester