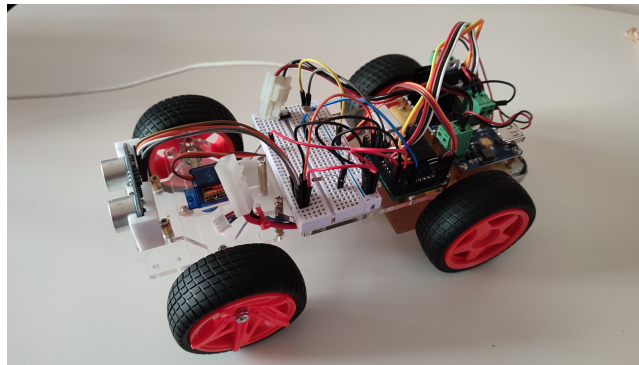


## I. Présentation

Vous disposez d'une voiture composé d'une pyboard, de divers capteurs, de leds, d'un servo pour la direction et deux moteurs avec réducteur.



Votre voiture devra se déplacer de façon autonome en évitant les obstacles et en s'adaptant au milieu dans lequel elle évolue.

Vous devrez aussi intégrer les contraintes suivantes :

- Si un obstacle est détecté la voiture doit ralentir de plus en plus jusqu'à l'arrêt complet;
- Si la voiture détecte une montée, elle doit accélérer, si elle détecte une descente elle doit ralentir (avec allumage de deux leds différentes sur la carte pyboard). On utilisera pour cela l'accéléromètre présent sur la carte;
- Si le robot est dans le noir, vous allumerez une led;
- Si on appuie sur le bouton USR de la carte le programme s'arrête (et la voiture aussi!).

Une fois que vous aurez réalisé ces différentes tâches, vous devrez faire un programme pour que la voiture se déplace seule sans toucher un obstacle. A vous de réfléchir à ce qui doit être fait s'il y a un obstacle, le véhicule ne devant pas être immobilisé plus de 2 secondes.

## II. Le matériel

Vous avez à votre disposition :

- Une carte pyboard.
- Deux moteurs et son contrôleur.
- Un capteur ultrason avec la fonction qui permet de l'utiliser dans le paragraphe suivant.



- Une cellule photoélectrique avec la fonction qui permet de l'utiliser dans le paragraphe suivant.



- Les LEDs de la carte pyboard.

**Remarques :** Vous devrez repérer sur quels port sont connectés chaque éléments.

Il est conseillé de faire des tests de tous les éléments séparément avant de vouloir faire un programme principal.

## III. La programmation

**Il est conseillé de faire des tests de tous les éléments séparément avant de vouloir faire un programme principal.**

**Vous pouvez faire une classe Voiture et créer des méthodes correspondantes à ce qui est demandé.**

Vous devrez modifier le fichier **main.py** sur le pyboard. Un fichier de départ vous est fourni **volta.py** avec les fonctions et classes évoquées pas la suite.

## 1) Détection d'obstacle

Pour détecter les obstacles, vous utiliserez la fonction suivante, qui retourne la distance en centimètres :

```
def dist_obstacle():
    trigger = pyb.Pin(pyb.Pin.board.Y5, mode=pyb.Pin.OUT, pull=None)
    trigger.low()
    echo = pyb.Pin(pyb.Pin.board.Y6, mode=pyb.Pin.IN, pull=None)
    trigger.high()
    time.sleep(0.00001)
    trigger.low()
    while echo.value() == 0:
        pass
    start = time.ticks_us()
    while echo.value() == 1:
        pass
    stop = time.ticks_us()
    return round((stop-start)*34/2000,1)
```

**Remarque :** la fonction `time.ticks_us()` renvoie la valeur d'un compteur en micro-secondes déclenché au début du programme.

## 2) Pilotage des moteurs

Pour le pilotage des moteurs on utilisera la classe `HBridge`.

L'instanciation est donnée dans le fichier, les méthodes sont assez explicites, à vous d'analyser le code pour comprendre le fonctionnement.

Vous devez réaliser des fonctions qui permettent de faire avancer la voiture (avec les deux moteurs), de tourner (attention la roue extérieure tourne plus vite que l'intérieure) et ne pas oublier de pouvoir l'arrêter!

## 3) Contrôle du servo moteur

Pour le contrôle du servo moteur de direction, une classe dédiée permet de l'utiliser simplement, exemple d'utilisation :

```
servo = pyb.Servo(1)
# fait tourner le servo de 30 degres
servo.angle(30)
```

## 4) Photorésistance

La fonction suivante renvoie une tension qui va être fonction de l'intensité lumineuse reçue. Cette tension va varier de 0 (pas de lumière) à 3,3 (très forte lumière). A vous de faire des tests pour définir des seuils utilisables.

```
def lectension():
    lecture = ldr.read()
    tension = (lecture*3.3)/4095
    pyb.delay(100)
    return tension
```

## 5) Ressources

Pour plus d'information, ce site vous permettra de savoir comment utiliser les autres composants :

<https://wiki.mchobby.be/index.php?title=MicroPython-Accueil>

Ou celui ci :

<https://docs.micropython.org/en/latest/pyboard/tutorial/index.html>