



Cégep **André-Laurendeau**

## Travail pratique 3 – Partie 3

« **Moteurs et afficheur à segments** »

420-555-AL

Programmation appliquée aux objets connectés

**Note:                    / 100**

**Correction en classe le mardi 11 novembre**

(Remise des scripts Python et du circuit Fritzing le même jour avant minuit)







## Objectifs de l'évaluation

On manipulera ici un servo, un moteur et un afficheur graphique à l'aide d'une interface graphique en Python. Le servo pourra aussi être contrôlé par deux boutons.

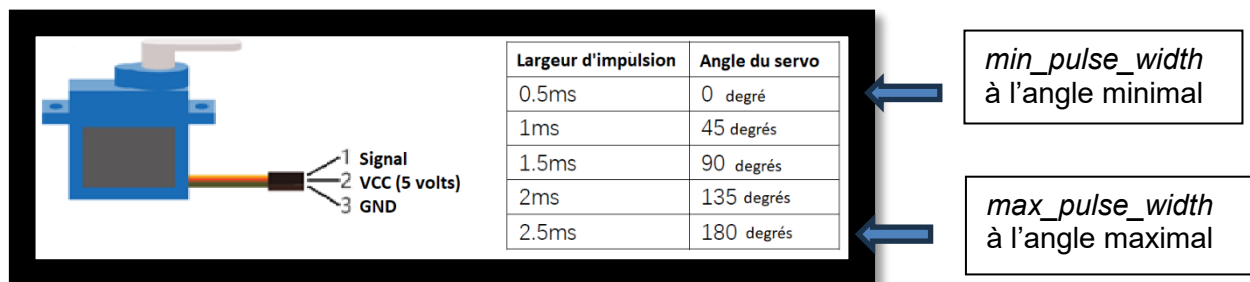
## Directives importantes

- Ce travail pratique **peut** être réalisé **en équipe de deux**.
- Divisez clairement vos platines en régions distinctes correspondant à chacun des composants
- Modalités de remise :
  - ✓ Correction en classe le mardi 11 novembre
  - ✓ Remise des scripts Python et du circuit Fritzing le même jour avant minuit

## Matériel requis et fonctionnement attendu

<p><b>Un Servo</b> <i>sur lequel on va exercer sur son bras un contrôle en le faisant tourner</i></p> 	<p><b>Un afficheur à sept segments</b> <i>pour afficher l'état du servo et faire défiler les lettres entrées dans une boîte de saisie</i></p> 
<p><b>Deux boutons</b> <i>pour contrôler le servo</i></p> 	<p><b>Un moteur</b> <i>à faire tourner dans les deux directions</i></p> 
<p><b>L293D : Un circuit intégré</b> pour piloter notre moteur à courant continu (DC)</p> 	<p>Source d'alimentation avec prise et battery 9V</p>  <p>Breadboard Power Module      9V Battery Cable</p>

**Servo (Tower Pro sg90):** un servomoteur (souvent abrégé en « servo », provenant du latin *servus* qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure<sup>1</sup>. Le bras du servo peut effectuer des rotations de 180 degrés. Une seule broche (fil orange) suffit pour exercer un contrôle à travers un signal. Voir ce [clip](#).



C'est votre programme Python qui aura la charge d'effectuer ces rotations. Les rotations seront par tranche de  $\pm 45$  degrés. Voici les informations nécessaires pour bien faire fonctionner votre servo:

1. Fixez le bras sur le dessus du servo:

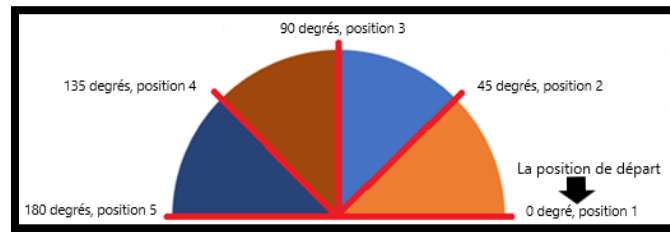


*Le bras montré est à +180 degrés: `servo.angle = 180`*

2. Branchez le servo à votre RasPi.
3. Pour éviter les tremblements (*jitter*), utilisez la pigpio factory :
  - a. Au besoin, installez-la sur votre RasPi : `$ sudo apt-get install pigpio python3-pigpio`
  - b. Démarrez-la : `$ sudo pigpiod` (`$ sudo killall pigpiod` pour la terminer au besoin)
  - c. Dans votre code :
    - i. Importez le module : `from gpiozero.pins.pigpio import PiGPIOFactory`
    - ii. Ajouter l'argument suivant à votre constructeur : `pin_factory = PiGPIOFactory()`
4. Utilisez la classe **AngularServo**<sup>2</sup> dans la bibliothèque **gpiozero** pour contrôler votre servo. Vous allez préciser au moins six arguments dans votre constructeur :
  - d. le numéro de la broche de contrôle
  - e. votre nouvelle pin factory
  - f. les angles minimum (0 degré) et maximum (180 degrés)
  - g. les largeurs d'impulsion minimale et maximale du servo, en secondes
 Fixez votre position de départ à 0 degrés (position 1).
5. Faites les ajustements pour que les angles du bras soient comme dans l'illustration suivante :

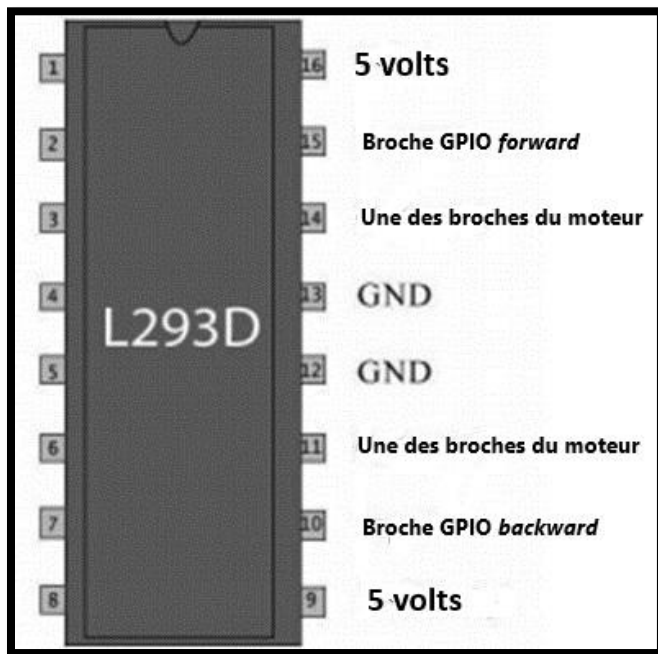
<sup>1</sup> <https://fr.wikipedia.org/wiki/Servomoteur>

<sup>2</sup> [https://gpiozero.readthedocs.io/en/latest/api\\_output.html#angularservo](https://gpiozero.readthedocs.io/en/latest/api_output.html#angularservo)



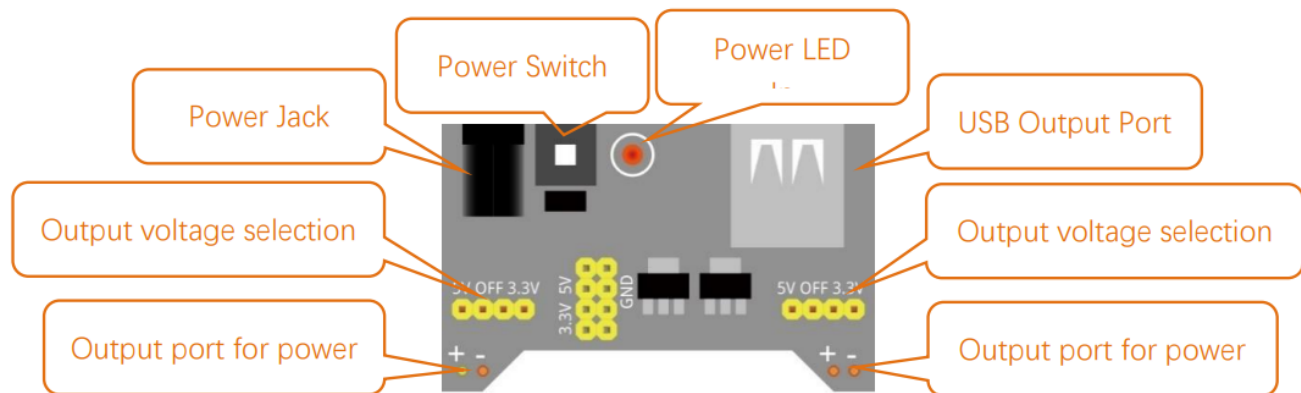
*Le numéro de la position est celui **affiché** sur les segments*

**Le moteur et le circuit intégré L293D:** Ces deux composants fonctionnent ensemble. Ils sont branchés de la manière suivante :



Ici la source de 5V doit être fourni pas la composant suivante :

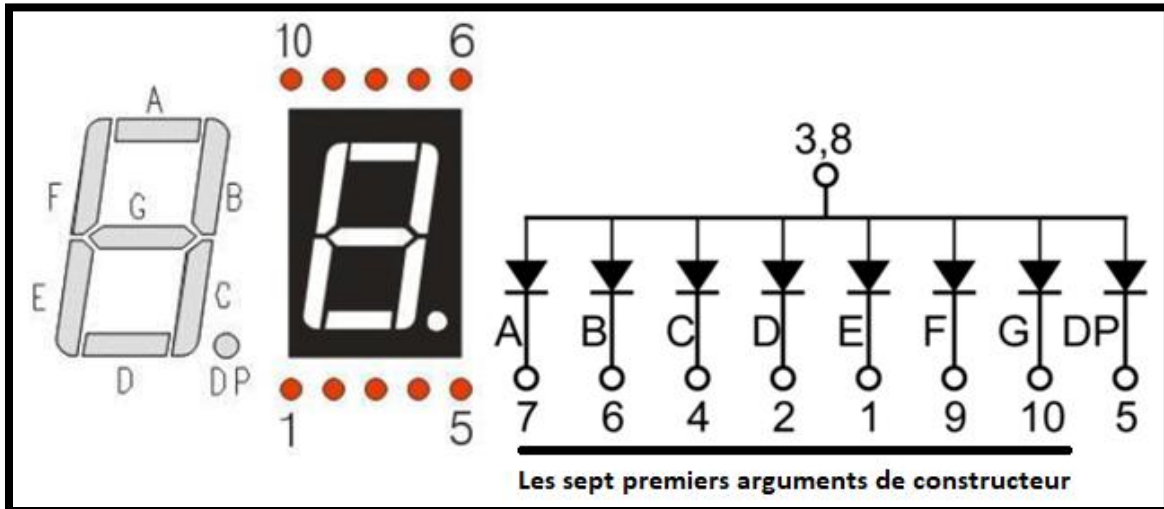
**VOUS NE DEVEZ PAS UTILISER LE rPI POUR ALIMENTER LE MOTEUR**



Pour ce type de branchement vous devez vous assurer que les GND de votre PI et de votre source alternative sont brancher ensemble. Uniquement les GND doivent être reliés pas les autres sources.

Référez-vous à la démonstration [gpiozero<sup>3</sup>](#) pour le code nécessaire. Le bouton installé sur l'axe de rotation du moteur vous permettra de bien visualiser sa vitesse de rotation. Ce [clip](#) montre les deux sens de rotation.

**L'afficheur à sept segments:** Un affichage à sept segments est une technique d'affichage basé sur sept segments qui peuvent être activés ou désactivés en fonction du motif graphique à produire<sup>4</sup>. L'afficheur que vous utiliserez possède les branchements suivants:



Par exemple, le segment A est activé par la broche 7. La broche 5 sert à afficher un point décimal (DP) au besoin. Cet afficheur est à **anode commune** (les broches 3 et 8 sont donc connectées à **3.3 volts**); autrement dit, si on veut qu'un segment s'allume, il faut lui envoyer un signal BAS (il y a un paramètre dans le constructeur pour spécifier ce comportement). Vous pourriez contrôler vous-même les segments à allumer, mais vous pouvez utiliser une classe de **gpiozero** qui s'occupera de le faire pour vous: **LEDCharDisplay<sup>5</sup>**. Notez bien, ce sont des LEDS, elles ont chacune besoin d'une résistance!

Pour afficher le point sur l'afficheur, vous pouvez utiliser la méthode `LEDCharDisplay.dp.on()`.

En testant cette classe, vous allez remarquer que les lettres [K, M, V, W, X, Z] provoquent une exception puisqu'elles n'existent pas dans l'alphabet par défaut. Vous devez donc créer un nouvel alphabet en utilisant la classe **LEDCharFont<sup>6</sup>** et l'intégrer à votre afficheur pour obtenir tous les caractères suivants, en plus des chiffres de 0 à 9:



Voici un [clip](#) montrant l'affichage de la chaîne suivante : « ACEFGHIJ0123456789 ».

**Deux boutons (rouge et bleu):** ces boutons servent à augmenter ou diminuer l'angle du bras du servo :

<sup>3</sup> <https://gpiozero.readthedocs.io/en/stable/recipes.html#motors>

<sup>4</sup> [https://fr.wikipedia.org/wiki/Affichage\\_à\\_sept\\_segments](https://fr.wikipedia.org/wiki/Affichage_à_sept_segments)

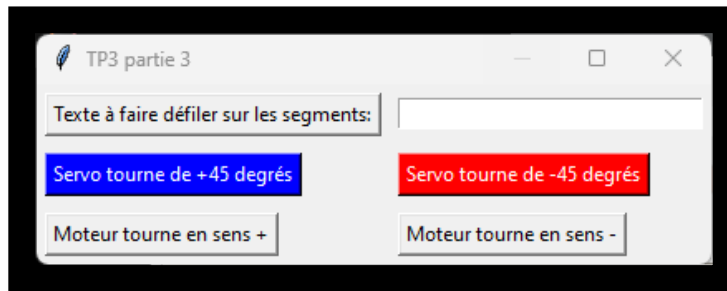
<sup>5</sup> <https://gpiozero.readthedocs.io/en/stable/recipes.html#ledchardisplay>

<sup>6</sup> Si cette classe n'est pas incluse avec le SE que vous avez sur le Raspberry Pi, il faudra refaire l'installation :

```
sudo apt remove python3-gpiozero
sudo pip3 install gpiozero
```

1. **Bouton bleu:** tourne le bras du servo de +45 degrés. Si le bras est déjà à sa position maximale de 180 degrés, on le ramène à sa position minimale. Il faut aussi indiquer la position finale sur l'afficheur (entre 1 et 5). *Cette fonctionnalité sera aussi disponible via l'interface graphique.*
2. **Bouton rouge:** tourne le bras du servo de -45 degrés. Si le bras est déjà à sa position minimale de 0 degré, on le ramène à sa position maximale. Il faut aussi indiquer la position finale sur l'afficheur (entre 1 et 5). *Cette fonctionnalité sera aussi disponible via l'interface graphique.*

**Interface graphique:** votre programme s'exécutera à travers une interface graphique ayant l'allure suivante:



Votre interface fournira les fonctionnalités suivantes à l'aide de cinq boutons et une boîte de texte:

1. **Texte à faire défiler sur les segments:** affichera le mot écrit dans la boîte, une lettre par seconde. Votre programme ne permettra pas d'afficher un mot avec un caractère qui n'est pas dans votre nouvel alphabet. Une fois la dernière lettre du mot écrite, le segment affichera un point.
2. **Servo tourne de +45 degrés:** même fonctionnalité que le bouton bleu.
3. **Servo tourne de -45 degrés:** même fonctionnalité que le bouton rouge.
4. **Moteur tourne en sens + :** le moteur tournera en sens horaire durant une seconde à une vitesse (*speed*) de 0.4
5. **Moteur tourne en sens - :** le moteur tournera en sens anti-horaire durant une seconde à une vitesse (*speed*) de 0.4

## Barème de correction

<b>Composant : Fonctionnalité</b>	<b>Circuit</b>	<b>Python</b>	<b>Fritzing</b>	<b>Total</b>
<u>Moteur</u> : rotation +	3	1	1	5
<u>Moteur</u> : rotation -	3	1	1	5
<u>Servo</u> : position de départ à 0°	3	1	1	5
<u>Servo</u> : position maximale à 180°	3	1	1	5
<u>Servo</u> : mouvement de +45° normal	3	1	1	5
<u>Servo</u> : mouvement de +45° extrême, donc retour à 0°	3	1	1	5
<u>Servo</u> : mouvement de -45° normal	3	1	1	5
<u>Servo</u> : mouvement de -45° extrême, donc retour à 180°	3	1	1	5

<u>Segments</u> : affichage de la position du servo (1 à 5)	3	1	1	5
<u>Segments</u> : affichage d'un mot avec ajout du point à la fin	3	1	1	5
<u>Segments</u> : nouvel alphabet pour majuscules et chiffres	3	1	1	5
<u>Bouton bleu</u> : mouvement du servo de +45°	3	1	1	5
<u>Bouton rouge</u> : mouvement du servo de -45°	3	1	1	5
<u>Interface graphique</u> : aspect visuel	3	1	1	5
<u>Interface graphique</u> : bouton pour le défilement	3	1	1	5
<u>Interface graphique</u> : boîte de texte	3	1	1	5
<u>Interface graphique</u> : bouton pour tourner le servo à +45°	3	1	1	5
<u>Interface graphique</u> : bouton pour tourner le servo à -45°	3	1	1	5
<u>Interface graphique</u> : bouton pour faire tourner le moteur +	3	1	1	5
<u>Interface graphique</u> : bouton pour faire tourner le moteur -	3	1	1	5
<b>Total</b>	<b>60</b>	<b>20</b>	<b>20</b>	<b>100</b>