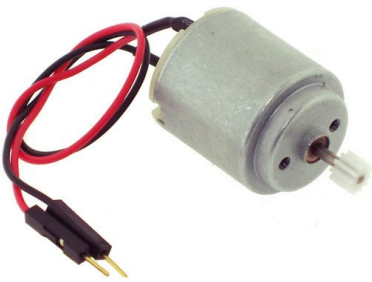


# Qu'est ce qu'un servo-moteur ?

## I Introduction

Lorsque vous créez un projet qui intègre des éléments robotique vous arrivez forcément à devoir faire un choix de moteur pour automatiser votre création. Plusieurs solutions existent, comme par exemple des moteurs à courant continu:



Ceux-ci ont l'avantage d'être très simple d'utilisation. Il suffit de connecter leurs deux fil d'alimentation à une batterie et ils se mettent à tourner.

Une autre solution serait d'utiliser des moteurs pas à pas :



Ces moteurs sont très pratique pour déplacer quelque-chose à un endroit précis, du fait que vous pouvez précisément le positionner à un angle donné. Par contre leur utilisation n'est pas aussi triviale que pour un moteur à courant continu.

Ce que je vous propose dans ce tutoriel c'est de découvrir les Servomoteurs. il en existe deux type :

- A contrôle d'angle
- A contrôle de vitesse de rotation

Les premiers ne tournent pas en continu, mais en général entre 0° et 180° et vous pouvez contrôler leur angle de rotation. Très pratique quand vous voulez faire un bras robotisé, piloter l'orientation des roues avant d'une voiture, ou bien contrôler un petit mécanisme. Les Servomoteurs suivants sont très pratiques lorsque vous souhaitez faire tourner un objet en contrôlant sa vitesse, par exemple des roues! :

[tiptopboards.com/169-thickbox\\_mini-servo-moteur-sg90-9g-mod\\_c3\\_a9lisme-arduino.jpg](https://www.tiptopboards.com/169-thickbox_mini-servo-moteur-sg90-9g-mod_c3_a9lisme-arduino.jpg)

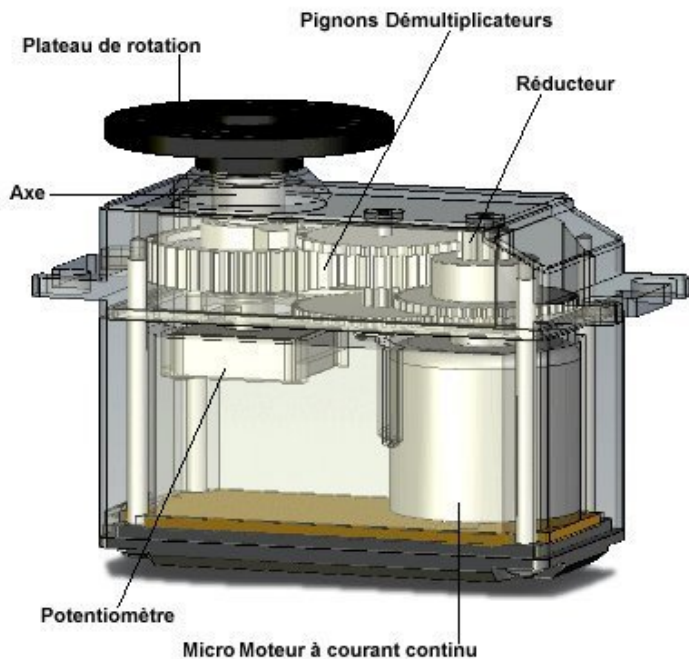
Ce sont des Servomoteurs à contrôle d'angle sur 180° et le tutoriel va donc porter sur ce type de moteur.

## II Fonctionnement

<https://fr.wikipedia.org/wiki/Servomoteur> --> fonctionnement

Les Servomoteurs intègrent au sein d'un même boîtier un moteur à courant continu, un potentiomètre, un réducteur et un circuit de contrôle. L'idée est que la valeur d'angle est mesurée grâce au potentiomètre et le circuit de contrôle fait tourner le moteur et corrige l'orientation. Voilà une image qui donne une idée du fonctionnement interne:

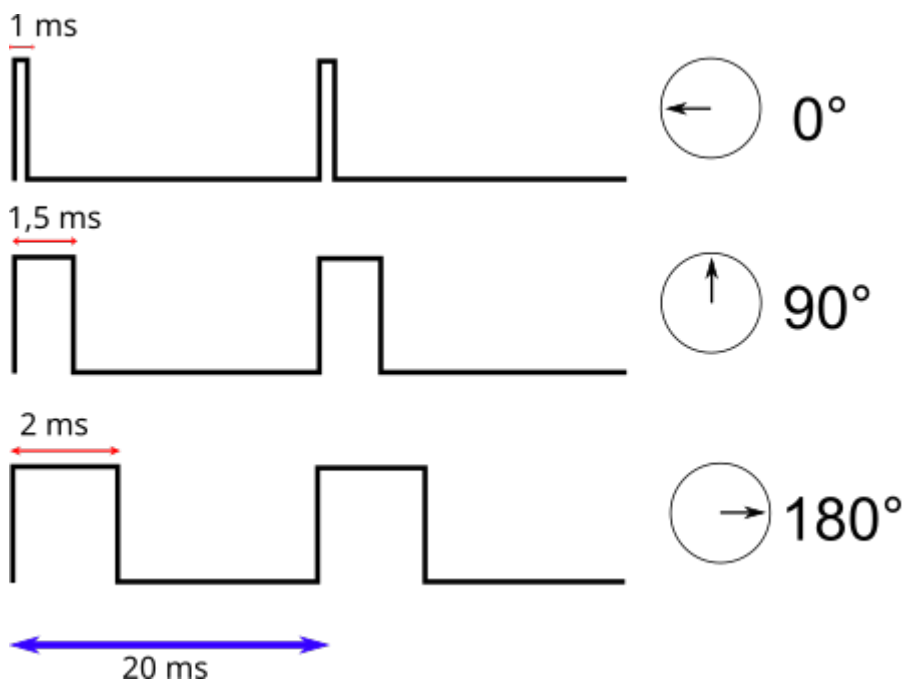
[upload.wikimedia.org/wikipedia/commons/d/d3/micro\\_servo.jpg](https://upload.wikimedia.org/wikipedia/commons/d/d3/micro_servo.jpg)



*Eric G*

Vous l'avez sûrement remarqué, le Servomoteur a trois fils. Le fil marron correspond à la masse, le fil rouge au 5 Volts et le fil orange à l'envoi de données. C'est par le fil orange que nous allons envoyer le signal pour la commande de l'angle voulu au Servomoteur.

L'instruction par le fil orange s'envoie sous la forme d'un signal PWM (Pulse Width modulation=Modulation en largeur d'impulsion). Le principe est que l'envoi d'instruction se fait par un signal électrique qui passe de façon régulière et rapide (30-50Hz ou 300Hz) de 0 à 5 Volts. La valeur de l'angle voulu est définie par le rapport entre le temps où le signal est à 5 Volts et le temps où celui-ci est à 0 Volt. Par exemple pour un angle de 0°, on envoie 5V pendant 1ms puis 0V pendant 19ms : Le signal est à 5V pendant 5% du temps (1ms/20ms). Une image vous donnera une meilleure idée:

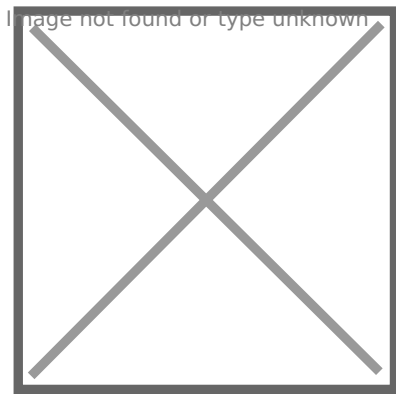


Source : <https://upload.wikimedia.org/wikipedia/commons/thumb/f/f6/TiemposServo.svg/220px-TiemposServo.svg.png>

## III Mise en pratique

Pour consolider votre compréhension nous allons mettre en pratique avec un Arduino. L'Arduino permet de générer des signaux PWM avec la fonction `AnalogWrite()` mais nous ne pouvons pas l'utiliser avec un Servomoteur, car sa fréquence est trop élevée (environ 500Hz). Pour ce faire nous allons plutôt utiliser la librairie `servo.h` qui est disponible de base avec le programme Arduino.

Connectez d'abord un Servomoteur à l'Arduino comme sur l'image:



et on utilisera pour l'exemple le code suivant

```
#include <Servo.h>

Servo myservo;  // On crée un objet MyServo par lequel on envoie les instructions au servo

void setup()
{
  pinMode(6, OUTPUT);
  myservo.attach(6);  // On associe notre objet au Pin connecté au fil de données du servo
}

void loop()
{
  myservo.write(0);    //donne l'ordre de mettre le Servo à son angle minimum
  delay(1000);
  myservo.write(255);  //donne l'ordre de mettre le Servo à son angle maximum
  delay(1000);
}
```

Si tout se passe bien, vous devriez voir le Servomoteur tourner de 180° toutes les secondes. Après ça, je vous conseille de vous amuser en modifiant les valeurs et/ou le code afin de vous familiariser avec son utilisation.

Vous savez maintenant le minimum nécessaire pour correctement utiliser un Servomoteur.

---

Revision #9

Created 28 August 2023 07:00:03 by FabManager

Updated 29 February 2024 13:25:24 by FabManager