



Les ServoMoteurs

Fiche Composant

C03_Servo-Moteur.pdf @

Y a quoi dedans ?

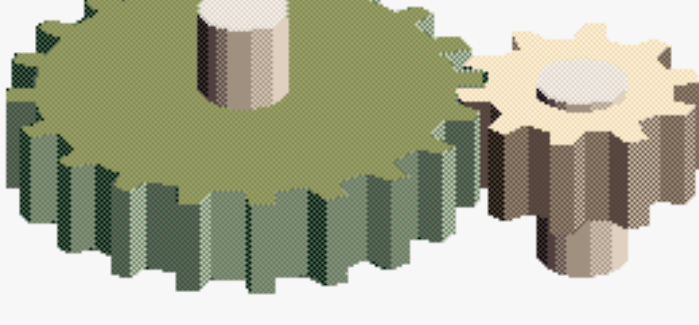
Des engrenages

Un engrenage est une roue dentée qui en tournant entraîne (engrène) une autre roue, dentée aussi.

Si la première roue dentée tourne dans un sens, la seconde tournera dans l'autre sens. On peut donc dire que le sens de rotation s'inverse d'une roue à l'autre dans un engrenage.

Si les deux roues dentées ne font pas la même taille, la vitesse de rotation de chaque roue sera différente.

En effet si la première roue a 6 dents, et la seconde 24 dents, la première effectuera 4 tours alors que la seconde n'en fera qu'un (6 dents* 4 tours= 24 dents).



Dans l'image ci-dessus, la grande roue effectue 1 tour pendant que la petite en effectue 2

Si vous voulez accélérer le mouvement, le moteur doit entraîner la grande roue qui entraînera la petite. Dans ce cas la puissance de rotation (le couple) de l'axe de la petite roue sera plus faible.

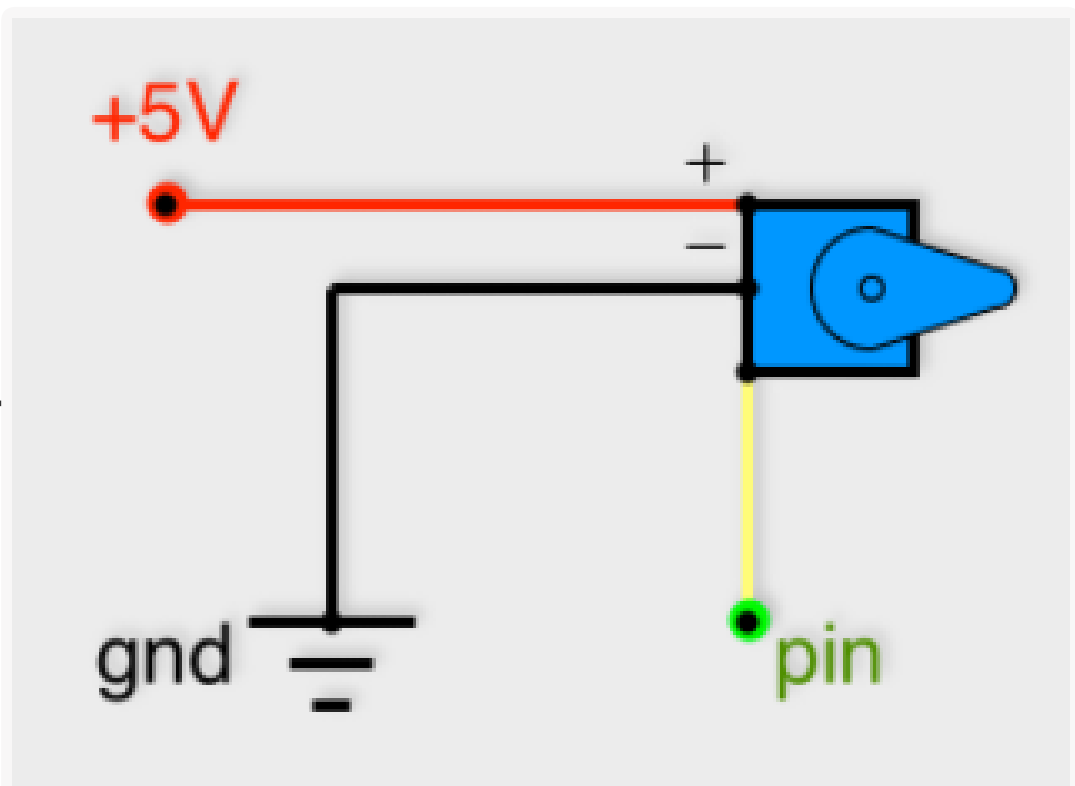
Si vous souhaitez gagner en couple, vous devrez entraîner la petite roue par le moteur, qui entraînera la plus grande. Dans ce cas, vous perdrez de la vitesse.

Un moteur à courant continu

Un potentiomètre

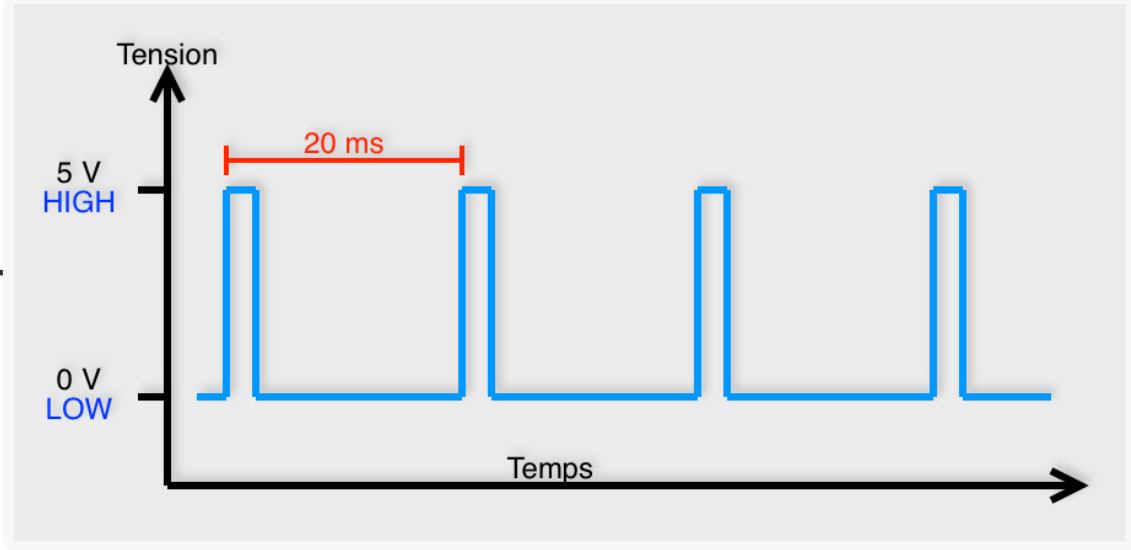
Une carte de contrôle

Le Câblage



Envoyez des ordres à un servo-moteur

Signal périodique

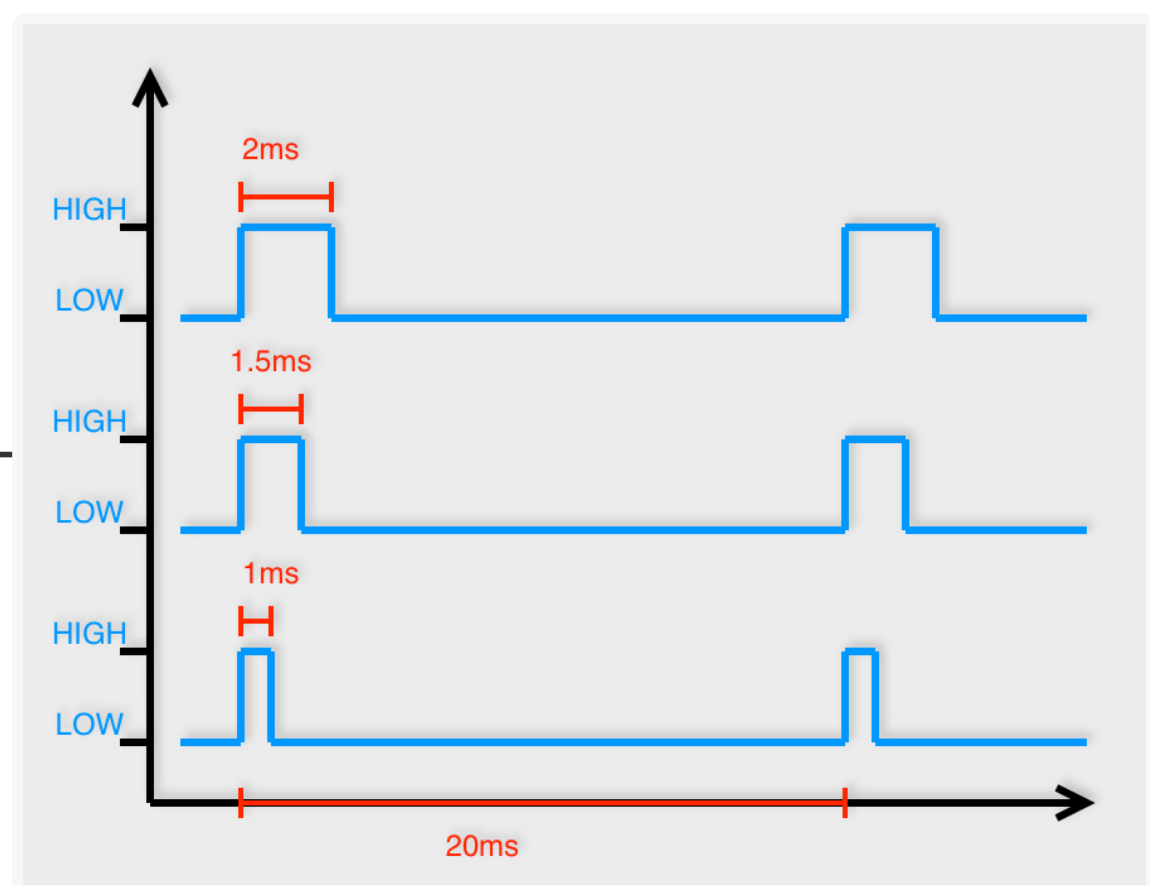


La valeur de la tension est soit +5V (HIGH de l'Arduino) soit 0V (LOW de l'Arduino)

Les impulsions sont envoyées régulièrement, ici toutes les 20 millisecondes. C'est la durée généralement utilisée pour piloter un servo-moteur.

La durée de l'impulsion peut varier. C'est d'ailleurs grâce à elle que nous allons piloter notre servo-moteur.

Modification de la durée d'impulsion



Lors d'une impulsion à 1ms : le servo-moteur se met à la position 0°

Lors d'une impulsion à 1.5 ms : le servo-moteur se met à la position 90°

Lors d'une impulsion à 2ms : le servo-moteur se met à la position 180°

On peut donc facilement faire un mappage des valeurs pour obtenir pour obtenir la durée de l'impulsion (µs) en fonction de l'angle

```
uint16_t dureeImpulsion = map( angle, 0, 179, 1000, 2000);
```

ATTENTION, l'intervalle d'impulsion peut être différent de 1000-2000, par exemple 500-2500

Programmation du signal

Nouvelle fonction :

```
delayMicroseconds(dureeImpulsion);
```

Exercices

Exercice 1

Générer un signal pour un angle de 90° suivant le schéma ci dessous

Exercice 2

Créer une fonction qui prendra en paramètre l'angle souhaité en utilisant le code de l'exercice 1

Exercice 3

Grâce à votre précédente fonction, créer un mouvement de votre servo moteur de gauche à droite comme un essuie-glace

La librairie servo.h

Une librairie c'est quoi ?

Elles sont réalisées par les ingénieurs de chez Aduino, ou par toute personne qui se sent capable d'en créer une qui répond à un besoin particulier. Nous verrons comment créer une bibliothèque bien plus loin dans un autre cours. Pour le moment, il suffit de savoir que votre programme peut s'enrichir du travail des autres, gratuitement, et souvent avec performance.

Vous avez déjà utilisé la bibliothèque Serial (Serial.begin(9600) , Serial.print() , Serial.println()). Ces commandes sont des fonctions incluses dans la bibliothèque Serial.

On le repère d'ailleurs par la présence du mot-clé Serial devant chaque nom de fonction.

Inclure une librairie

```
#include <Servo.h>
```

Ceci est une directive de pré-compilation. C'est-à-dire que l'IDE, avant même de chercher à transformer votre programme en langage machine, va aller chercher la bibliothèque sur votre ordinateur, ajouter ses lignes de codes aux vôtres et créer un programme complet.

Ensuite seulement, il va compiler.

Observez bien la structure : #include puis <nomDeLaBibliotheque> et pas de point-virgule à la fin.

A Placer en 1ere ligne de votre programme avant la déclaration des constantes et variables globales

Déclaration de l'objet Servo

Souvent pour utiliser une bibliothèque, il faut créer un objet lié à cette bibliothèque. C'est le cas pour la bibliothèque Servo. Nous allons créer un objet de type Servo et lui donner un nom :

```
Servo monServo;
```

Maintenant, à chaque fois que nous ferons référence à "monServo" il s'agira de cet objet que nous venons de créer. Pour qu'il soit accessible partout (pensez au scope des variables) on le crée généralement avant le setup()

Utilisation de l'objet

```
monServo.attach(8);
```

Permet de relier l'objet "monServo" à la broche 8

```
monServo.write(angle);
```

Placer le servo à "angle" degré

Exercices

Exercice 4

Reprendre l'exercice 3 de l'essuie-glace en utilisant la librairie servo.h

Exercice 5

Rajouter un potentiomètre comme appris lors de partie precedente

Faire correspondre la position du servo avec la valeur du pota

Exercice 6

Supprimer le pota et rajouter un bouton poussoir

Le servo est à 0° au départ

Le servo se déplace vers sa position maximale tant qu'on appuie, et revient lorsqu'on relâche.