



# Servomoteur- Arduino

## 1. Objectifs :

- Comprendre le fonctionnement d'un servomoteur.

## 2. Matériel nécessaire

- Arduino UNO
- Servomoteur positionnel
- Servomoteur continu (Défi)
- Quelques fils pour les branchements

## 3. Ce que fait le programme

Ce programme permet de faire tourner un servomoteur positionnel d'un angle de 0° à 180°.

## 4. Étapes de réalisation

- Préparer le montage
- Ouvrir l'application Arduino IDE, écrire le code, le compiler et le téléverser dans la carte.

## 5. Fonctionnement du servomoteur

Le servomoteur positionnel possède des engrenages permettant de contrôler précisément sa position. Les servomoteurs n'ont pas besoin de circuit de contrôle extérieur, il est déjà inclus à l'intérieur. Il est possible de les alimenter directement avec la carte Arduino, mais ils peuvent se montrer gourmands en courant. On doit alors se tourner vers une alimentation extérieure qui peut être branchée directement sur le moteur. Il existe deux types de servomoteurs : le positionnel et le continu.

Les servomoteurs positionnels peuvent se placer à des valeurs d'angle comprises souvent entre 0° et 180°. Ces valeurs peuvent changer d'un fabricant à l'autre. Les contrôler

revient à les placer à une valeur d'angle précis. En fait, c'est l'arbre de transmission qui prendra une position angulaire précise. Il est à noter que le servomoteur placé à un angle donné exercera une force pour se placer à cet angle et ensuite pour maintenir sa position si on exerce une force extérieure dessus. Plus la force exercée par le servomoteur est grande, plus celui-ci consommera de courant. On évite donc d'empêcher le servomoteur d'atteindre la position programmée. De la même façon, contrainte par une force extérieure (main ou autre) un servomoteur à changer de position demanderait au servomoteur un courant assez grand et pourrait aussi briser les engrenages qu'il contient.

Les servomoteurs continus ne sont pas bloqués entre 0° et 180°. On les contrôle alors en ajustant leur vitesse de rotation. On les programme de la même façon, mais la valeur d'angle utilisée donne alors la vitesse de rotation de l'arbre de transmission. Pour un servomoteur continu qui permet des « angles » de 0° à 180°,

Angle	Vitesse
0	Maximum dans un sens
90	0
180	Maximum dans l'autre sens

Avec Arduino, on utilise une bibliothèque pour programmer les servomoteurs. Qu'ils soient positionnels ou continus, ce sont les mêmes fonctions qui sont utilisées.

Attention : L'utilisation de la bibliothèque « Servo.h » avec une carte de type Arduino UNO enlèvera la fonctionnalité PWM sur les broches 9 et 10, qu'il y ait un servomoteur branché dessus ou non.

## 6. Plus en détails

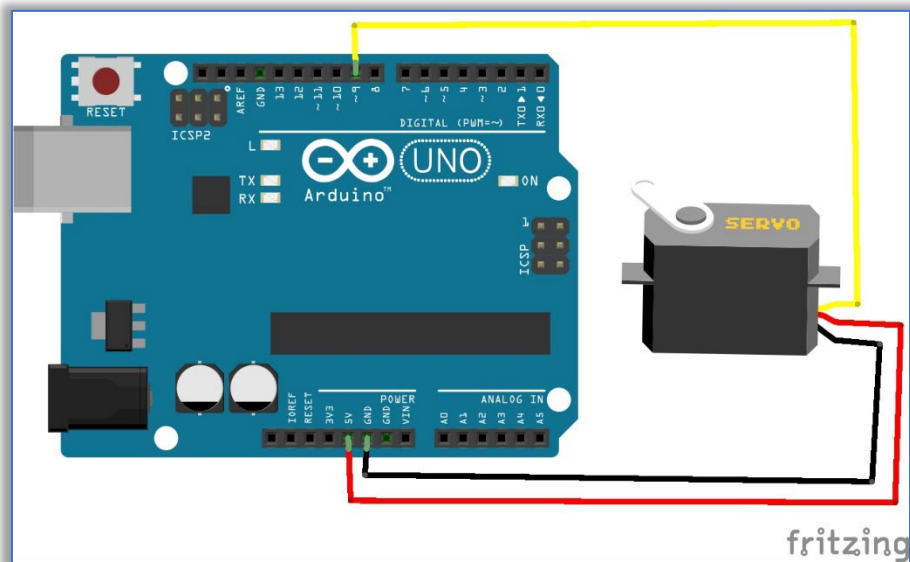
On contrôle un servomoteur à l'aide d'un signal semblable au PWM (pour les détails sur ce type de signal, voir le projet de contrôle d'intensité des DEL). Pour que cela fonctionne, on doit avec un signal avec une largeur d'impulsion bien précise et qui se répète avec une fréquence bien précise. Par exemple, un signal d'un angle de 90 degrés peut correspondre à une largeur d'impulsion de 1.5 ms pour un signal qui se répète aux 20 ms. Le tout est programmé dans la bibliothèque « Servo.h ».

Pour les servomoteurs continus, la vitesse de rotation devrait être nulle lorsqu'on programme un « angle » de 90 degrés. Cependant, la vitesse peut être nulle pour quelques degrés au-dessous ou en-dessous de 90 degrés. Il peut aussi arriver que la vitesse ne soit pas nulle à 90 degrés. Il y a, à l'arrière de certains modèles, un petit potentiomètre que l'on peut tourner avec un tournevis pour ajuster la vitesse nulle plus précisément. Il est possible que la vitesse reste nulle pour quelques degrés dans tous les cas.

Théoriquement, la vitesse augmente en fonction de l'angle. Cette augmentation n'est souvent pas complètement linéaire et elle atteint un plateau à partir d'un angle. Ces valeurs peuvent varier d'un moteur à l'autre, même s'ils proviennent du même fabricant.

Pour tester un servomoteur, on peut toujours l'alimenter directement avec la carte Arduino, mais ils sont gourmands en courant. Il devient rapidement utile et recommandé d'utiliser une source d'alimentation extérieure pour tous les moteurs, en fait. Pour un servomoteur demandant entre 4.8V et 6V, on peut utiliser un bloc de 4 piles AA rechargeables ou non, par exemple. Il faudra cependant brancher la borne négative de cette alimentation ( « - » ) avec une broche GND de la carte Arduino.

## 7. Circuit



## 8. Code

```
#include <Servo.h>

//crée un objet de type Servo :
Servo monServo;

//variable définissant la position du servo
int pos = 0;

void setup() {
//On doit attacher l'objet Servo à une broche, ici la 9.
  monServo.attach(9);
}
```

```
void loop() {  
    for(pos = 0; pos <= 180; pos += 1) {  
        //Mettre le servomoteur à l'angle « pos »  
        monServo.write(pos);  
        delay(15);  
    }  
}
```

## 9. Fonctions utilisées

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

La bibliothèque « Servo.h » est une bibliothèque de base déjà installée avec le compilateur Arduino. Elle permet le contrôle des servomoteurs.

```
Servo monServo;
```

Crée un objet de type *Servo* qui se nomme *monServo*. Il est possible de lui donner un autre nom.

```
monServo.attach(9);
```

Le servomoteur *monServo* est branché sur la broche 9.

```
monServo.write(pos);
```

Cela donne un angle d'une valeur « pos » au servomoteur *monServo*. Utiliser seulement des valeurs comprises entre 0 et 180 ici.

### Défis :

- Faire revenir le servomoteur à un angle de 0° avant de recommencer.
- Modifier la vitesse de balayage du servomoteur.
- Contrôler le servomoteur à l'aide d'un potentiomètre.
- Remplacer le servomoteur positionnel par un servomoteur continu :
  - Trouver les valeurs d'angle qui donnent une vitesse nulle
  - Tourner le potentiomètre derrière le moteur pour que la vitesse nulle soit centrée à 90 degrés.
  - Mesurer plusieurs vitesses de rotation pour des « angles » différents