

# Comment contrôler un moteur à courant continu

Par Kilian Sanfins

# Introduction

Le moteur à courant continu est un moteur très simple d'utilisation et le moins cher. Pour son fonctionnement il suffit de lui appliquer une tension pour sa mise en rotation. Plus la tension appliquée est élevée, plus le moteur tournera vite et plus le courant sera fort, plus il aura de couple (attention tout de même à ne pas dépasser les valeurs limites fournies par le constructeur!).

Cependant le courant requis peut être trop élevé pour qu'il soit fourni directement par le microcontrôleur. On va donc utiliser un dispositif appelé contrôleur (driver en anglais) tel que le L298 pour le faire tourner à pleine puissance.

## Avantage

- Peu coûteux
- Facile d'utilisation
- Pour les moteurs les plus petits il est possible de les contrôler directement avec le microcontrôleur.
- Contrôle en vitesse aisé.

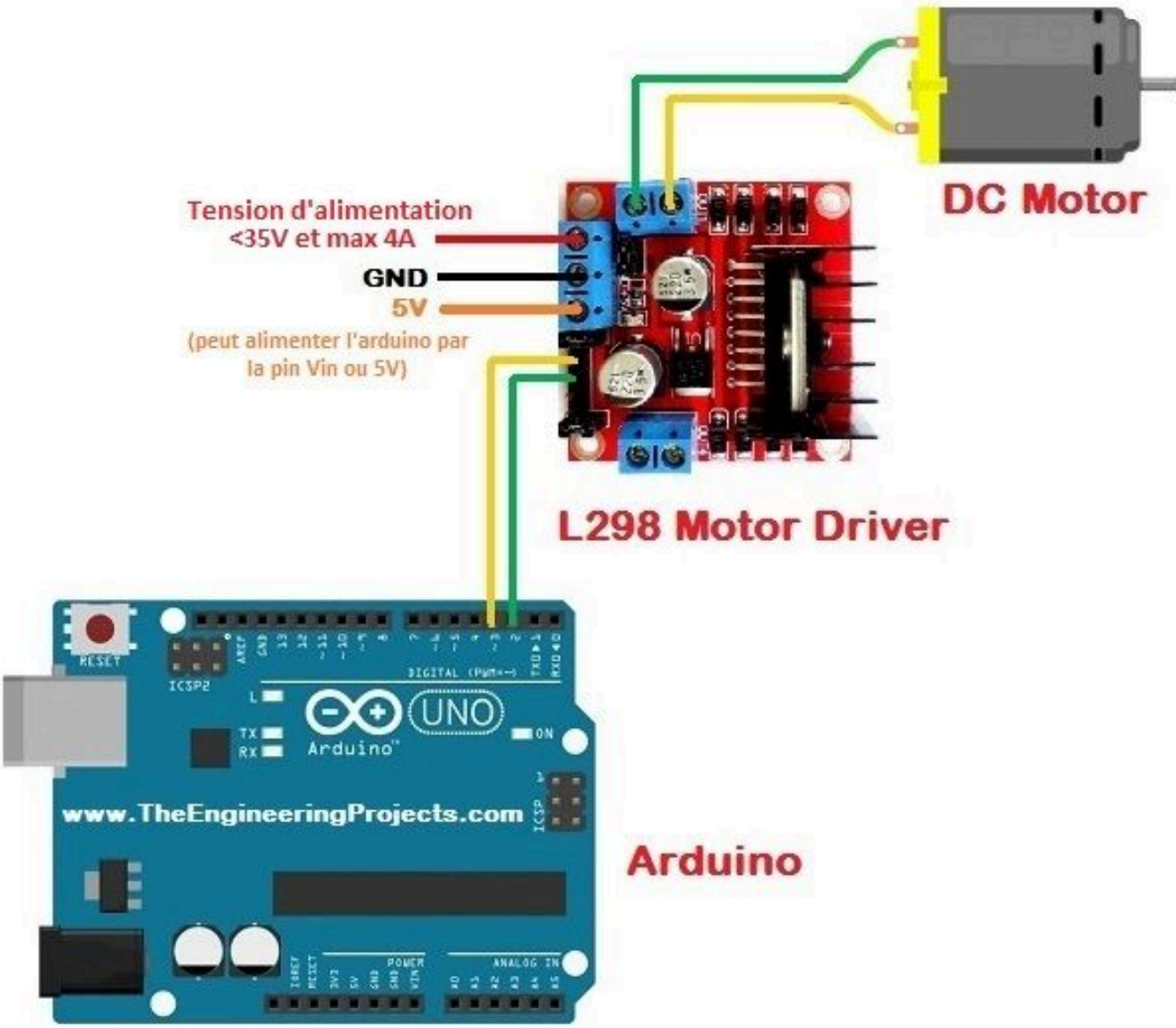
## Désavantage

- Le contrôle en position nécessite un encodeur et l'asservissement du moteur
- C'est une technologie peu compacte (ratio puissance/volume/poids faible)
- Un moteur à courant continu tourne vite, il nécessite une boîte de réduction pour obtenir un couple décent en sortie (ou alors il sera sur dimensionné pour produire un gros couple à faible vitesse)
- A cause des balais, ce n'est pas le moteur le plus silencieux

## Cas d'utilisation

- Mettre un système en mouvement à bas coût et avec simplicité
- Où l'on souhaite une certaine vitesse de rotation, même si avec un bon asservissement, il est possible de le contrôler en couple ou en position aisément.

# Schéma électrique



# Code

```
/*
 * Ce programme a pour but de contrôler un moteur à courant en continu avec le
 contrôleur L298.
 * La pin 2 est reliée à IN1 sur le carte du L298.
 * La pin 3 est reliée à IN2.
 * Le moteur est branché sur le bornier dédié à cet effet.
 * On alimente la carte avec une tension comprise entre 6V et 35V. (Très important,
 sinon la moteur ne tournera pas ou alors on grille la carte)
 *
 * Code écrit par Kilian Sanfins
 */

#define IN_1 2 // On assigne le nom des pins en fonction
#define IN_2 3 // de la direction du moteur

void setup() {
  pinMode(IN_1, OUTPUT); // On initialise les pins comme des
  pinMode(IN_2, OUTPUT); // sorties

  ArreterMoteur(); // On s'assure que le moteur est arrêté au démarrage du programme
}

void loop() {
  // Fais tourner le moteur dans un sens à vitesse maximum
  TournerMoteur(225, 0);
  delay(1000); // On attend 1 sec

  // On arrête le moteur
  ArreterMoteur();
  delay(1000);

  // Puis dans l'autre sens à la moitié de la vitesse
  TournerMoteur(125, 1);
  delay(1000);
}
```

```
/*  
    Si sens = 1 alors le moteur tourne dans le sens horaire.  
    Si sens = 0 alors le moteur tourne dans le sens anti-horaire. (Si ce n'est pas le  
cas, inverser le branchement du moteur)  
    La variable vitesse définit la tension appliquée au moteur (et donc sa vitesse de  
rotation)  
    vitesse est comprise entre 0 et 255  
*/
```

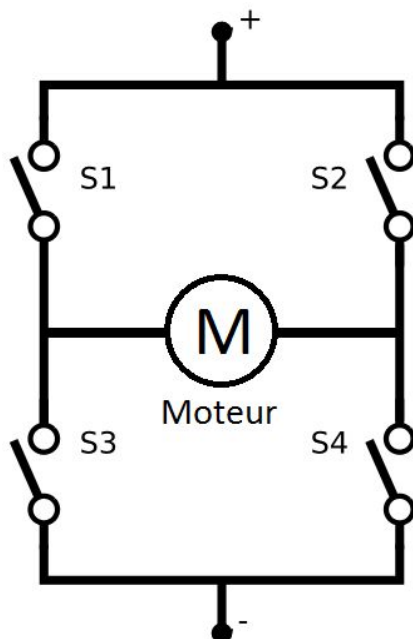
```
void TournerMoteur(int vitesse, bool sens) {  
    // On s'assure que la valeur ne dépasse pas les limites de commandes  
    if (vitesse > 255) {  
        vitesse = 255;  
    }  
    else if (vitesse < 0) {  
        vitesse = 0;  
    }  
  
    // En fonction du sens souhaité, on active un bras ou l'autre du pont en H,  
résultant dans un sens de rotation du moteur différent.  
    if (sens == 1) {  
        digitalWrite(IN_2, LOW);  
        analogWrite(IN_1, vitesse);  
    }  
    else {  
        digitalWrite(IN_1, LOW);  
        analogWrite(IN_2, vitesse);  
    }  
}
```

```
/*  
    Permet d'arrêter la rotation du moteur  
*/
```

```
void ArrêterMoteur() {  
    // On coupe tous les bras du pont en H  
    digitalWrite(IN_1, LOW);  
    digitalWrite(IN_2, LOW);  
}
```

# Explication

Le contrôleur L298 est un pont en H. Ce nom lui vient de la forme de son schéma électrique.



En fonction des interrupteurs activés (nommé S1, S2, S3, S4), il est possible de faire tourner le moteur dans un sens ou dans l'autre. Certaines combinaisons sont interdites.

Voici une liste des connections les plus importantes:

S1	S2	S3	S4	Résultat
✓	✗	✗	✓	le moteur tourne dans le sens 1
✗	✓	✓	✗	le moteur tourne dans le sens 2
✗	✗	✗	✗	le moteur tourne pas
✗	✓	✗	✓	court circuit <b>NE PAS FAIRE</b>
✓	✗	✓	✗	court circuit <b>NE PAS FAIRE</b>
✓	✓	✓	✓	court circuit <b>NE PAS FAIRE</b>

✓ interrupteur fermé (laisse passer le courant)

✗ interrupteur ouvert (ne laisse pas passer le courant)

Il est possible de fabriquer son propre pont en H avec des transistors mais pour simplifier la réalisation, nous avons choisi un composant qui gère tout et simplifie le contrôle du moteur. En particulier, le moteur se contrôle avec 2 pins, l'un pour faire tourner le moteur dans un sens et l'autre pin pour l'autre sens (le sens de rotation dépend du branchement du moteur).



Si les 2 pins sont activés en même temps, il y a court-circuit!

Le code sert précisément à cela, il désactive une pin pour activer l'autre et ainsi contrôler le moteur! Les fonctions fournies dans le code permettent un contrôle facile et compréhensible du moteur.

En activant et en coupant rapidement les bras de pont le L298 produit de la Modulation de Largeur d'Impulsion (abrégé MLI ou PWM en anglais pour Pulse Width Modulation).

Sans rentrer dans les détails, pour chaque période, le signal est à l'état haut pendant une certaine durée  $t$  et à l'état bas le reste du temps. On définit alors le ratio  $\frac{\text{Temps à l'état haut}}{\text{Durée d'une période}}$  comme le rapport cyclique (duty cycle en anglais).

### 25% Duty Cycle



### 50% Duty Cycle



### 75% Duty Cycle



← T →

Ainsi, un rapport cyclique de 10 % voudrait dire que 10% du temps, le signal est à l'état haut, le reste à l'état bas. Dans le cas du contrôle du moteur, l'état haut correspond à la tension maximum fournie par l'alimentation. L'état bas est la masse.

Donc si on applique un signal PWM au moteur avec un rapport cyclique de 10%, le moteur ne verra que 10% de la tension maximum. Ce principe là est utilisé pour faire varier la vitesse du moteur grâce au L298.

Cependant le moteur n'est pas protégé de la tension fournie par l'alimentation, si le moteur ne supporte que 12v et qu'une tension de 24v est appliquée au L298, il est possible de le faire tourner en envoyant une faible commande (rapport cyclique bas, la tension ressentie par le moteur sera alors inférieure à 12v) mais Si l'on envoie la commande maximum (rapport cyclique maximum donc 100%, ce qui correspond dans notre cas à 24v) alors le moteur sera détruit!

## Sources pour approfondir

- HowToMechatronics.com, site en anglais, avec une vidéo explicative.  
<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>
- Toutes les informations concernant le L298n. (français)  
[https://wiki.mchobby.be/index.php?title=Pont-H\\_L298N](https://wiki.mchobby.be/index.php?title=Pont-H_L298N)
- La datasheet du L298n. (anglais)  
[https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)