

# Commander deux moteurs avec Arduino

Cette activité s'insère dans la partie Architectures matérielles et systèmes d'exploitation du programme de première NSI.

**Contenus :** Périphériques d'entrée et de sortie Interface Homme-Machine (IHM)

**Capacités attendues :** Identifier le rôle des capteurs et actionneurs. Réaliser par programmation une IHM répondant à un cahier des charges donné.

**Objectif:** Commander deux moteurs pour pouvoir réaliser des déplacements avec un robot à deux roues.

**Pré-requis :** Avoir déjà réalisé l'activité de découverte du micro-contrôleur Arduino pour être ainsi familiarisé avec le langage utilisé et le l'utilisation des entrées et sorties.

**Introduction :** Un robot munie de deux roues non orientables peut se déplacer grâce à deux moteurs entraînant indépendamment chacune des roues. Le but de cette activité est de programmer différents déplacements grâce à des fonctions.

**Matériel utilisé :** Un robot préfabriqué comportant deux moteurs à courant continu, un module L298N, un module émetteur récepteur d'ultrasons orientable grâce à un servomoteur et un Arduino UNO. Le module à ultrasons n'est pas utilisé ici et servira dans une autre activité.

## **Spécifications du moteur :**

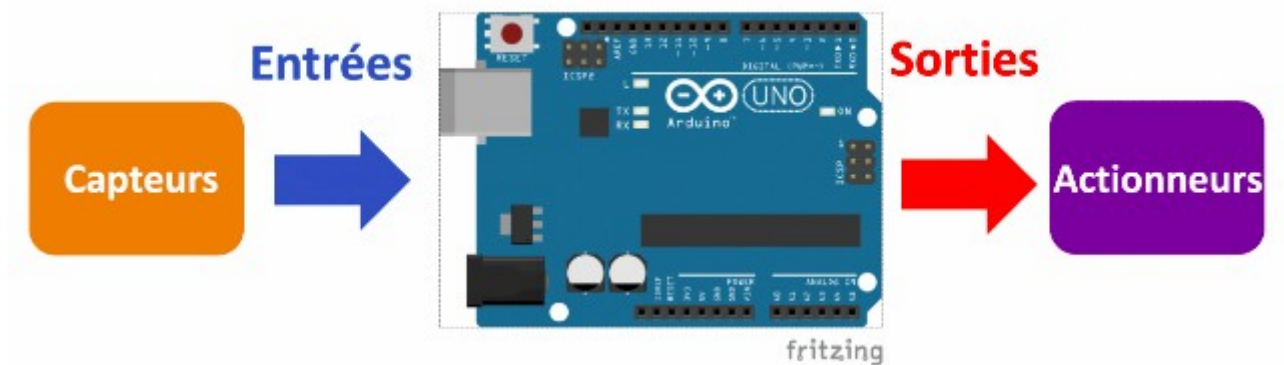
- tension nominale: 6V
- sans charge:
  - vitesse: 14500rpm
  - courant: 0.21A
- couple maximum: 340gcm

## **Spécifications du module L298 N de contrôle du moteur :**

- Puce: L298N (ST NOUVEAU)
- Tension logique: 5V;
- Tension d'entraînement: 5V-35V
- Courant logique: 0mA-36mA
- Courant d'entraînement: 2A
- Puissance max .: 25W
- Poids: 30g
- Taille: 43 x 43 x 27 mm

## Manipulation 1: Faire tourner un moteur

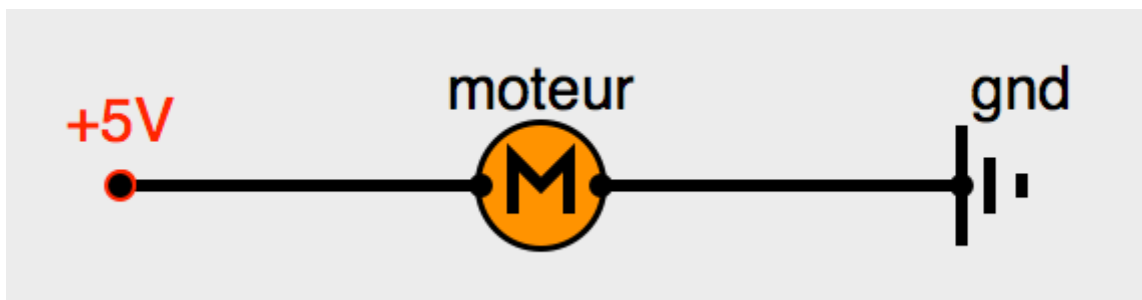
Un microcontrôleur est un circuit intégré de faible dimension qui rassemble les éléments essentiels nécessaires au fonctionnement d'un ordinateur : processeur, mémoires (ROM et RAM), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs sont fréquemment utilisés dans des systèmes embarqués, en robotique et en domotique. Relié à différents types de capteurs et d'actionneurs, ils peuvent permettre de réaliser une voiture télécommandée ou autonome



Dans cette activité, nous allons commander un moteur avec Arduino.

1- Le moteur est-il un actionneur ou un capteur ? Pourquoi ?

Vous pouvez dans un premier temps connecter directement votre moteur sur le +5V et le ground de l'Arduino comme ceci :



2- Que se passe-t-il si vous inversez les fils du moteur ?

Réponse : Le moteur tourne dans l'autre sens.

3- L'inconvénient principal de ce montage est de ne pas pouvoir commander la rotation du moteur. Quelle solution pouvez vous envisager pour pouvoir piloter le moteur?

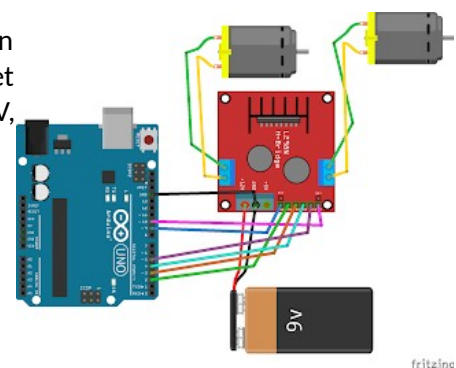
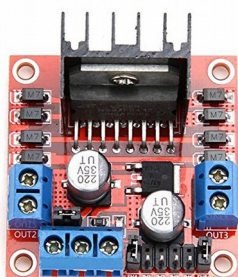
Réponse possible: Brancher le moteur sur un pin digital. Mais il faut éviter de le faire car l'intensité que peut fournir la sortie est insuffisante pour un moteur.

## Manipulation 2: Commander les deux moteurs du robot

Une solution possible pour pouvoir commander un ou deux moteurs est l'utilisation d'un module LH298 N.

L' Arduino sera branché aux entrées ENA, IN1, IN2, IN3, IN4 et ENB.

Si IN1 (arduino pin 8) est à 5 V (HIGH) pendant que IN2 ( arduino pin 9) est à 0 V (LOW), le moteur gauche branché aux sorties OUT1 et OUT2 tourne dans un sens. Si IN1 est à 0 V pendant que IN2 est à 5 V,



le moteur tourne dans l'autre sens. ENA (arduino pin ~ 10) est l'entrée "enable": le moteur tourne à la condition que cette entrée se trouve à un niveau logique haut.

Pour contrôler la vitesse de rotation du moteur, vous devez brancher le pin ENA à une sortie PWM de l'Arduino et écrire l'instruction `analogWrite(pinENA, 100);` Le second paramètre de la fonction `analogWrite` est compris entre 0 (arrêt) et 255 (vitesse de rotation rapide). C'est lui qui dirige la vitesse de rotation.

C'est le même principe pour les pins ENB (arduino pin ~11), IN3 (arduino pin 12) et IN4 (arduino pin ~13) , sauf que ces entrées permettent de contrôler le moteur droit relié aux sorties OUT3 et OUT4.

4- Compléter le tableau suivant indiquant l'état des 6 entrées du module LH298 N pour assurer 4 types de déplacement et l'arrêt du robot.

fonction	IN1	IN2	IN3	IN4	ENA	ENB
Arrêt	....	....	....	....	LOW	LOW
Avance	HIGH	LOW	HIGH	LOW	HIGH	HIGH
Reculé						
Tourne à droite	....	....	HIGH	LOW	LOW	HIGH
Tourne à gauche			....	....		

.... état indifférent

5- Le code arduino suivant possède les fonctions avance et recule. Écrire les fonctions `tourne_droite()` `tourne_gauche()` et `arret()`

```
/*
  Contrôle de deux moteurs cc avec un L298N
  sketch modifié d'après http://electroniqueamateur.blogspot.com/2016/06/controle-dun-ou-deux-moteurs-cc-avec.html
*/
```

```
// définition des pins de l'Arduino qui contrôlent le 1er moteur
#define pinIN1 8
#define pinIN2 9
#define pinENA 5 // doit être une pin PWM
```

```
// définition des pins de l'Arduino qui contrôlent le 2e moteur
#define pinIN3 12
#define pinIN4 13
#define pinENB 11// doit être une pin PWM
```

```
void setup() {
```

```
  // toutes ces pins sont des sorties
```

```
  pinMode(pinIN1, OUTPUT);
  pinMode(pinIN2, OUTPUT);
```

```

pinMode(pinENA, OUTPUT);

pinMode(pinIN3, OUTPUT);
pinMode(pinIN4, OUTPUT);
pinMode(pinENB, OUTPUT);

}

void loop() {

}

void avance(){
    // le premier moteur tourne en marche avant

    analogWrite(pinENA, 200);
    // si le moteur ne tourne pas du tout, augmentez la valeur
    digitalWrite(pinIN1, true);
    digitalWrite(pinIN2, false);

    // le deuxième moteur tourne en marche avant

    analogWrite(pinENB, 200);
    digitalWrite(pinIN3, true);
    digitalWrite(pinIN4, false);

    // on les laisse tourner comme ça pendant 1 seconde

    delay(1000);
}

void recule(){
    // le premier moteur tourne en marche arrière

    analogWrite(pinENA, 200);
    digitalWrite(pinIN1, false); // le contraire de ce qu'on avait en marche avant
    digitalWrite(pinIN2, true);
    // le deuxième moteur tourne en marche arrière
    analogWrite(pinENB, 200); // augmentez le nombre si votre moteur ne tourne pas du tout
    digitalWrite(pinIN3, false); // le contraire de ce qu'on avait en marche arrière
    digitalWrite(pinIN4, true);

    // on les laisse tourner comme ça pendant 1 seconde

    delay(1000);
}

void tourne_droite(){

}

void tourne_gauche(){

}

```

```
void arret(){  
}
```

6- Que fait le code précédent lorsqu'il est téléversé sur un Arduino ?

7- Modifiez la fonction loop() pour appeler vos fonctions et téléverser le sketch sur le robot pour observer ses déplacements.

Sources: <https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino/3285333-le-moteur-a-courant-continu-partie-1-transistors-et-sorties-pwm>  
<http://electroniqueamateur.blogspot.com/2016/06/controle-dun-ou-deux-moteurs-cc-avec.html>