

## Chapitre 6

# Moteurs et robots

### 1. Moteurs à courant continu

Ce type de moteur est très courant, vous pouvez facilement en récupérer sur un appareil hors d'usage. Il suffit simplement de savoir sous quelle tension il fonctionne. Souvent, c'est écrit dessus, sinon regardez sa source d'alimentation (piles ou batterie).



Exemple de moteur à courant continu

Le branchement est très simple : il n'a que deux fils, il suffit donc de brancher le (+) à une borne et le (-) à l'autre, pour qu'il tourne dans un sens. Si vous inversez les deux bornes, il tourne dans l'autre sens. Et si vous baissez la tension de l'alimentation, il tourne moins vite.

### Remarque

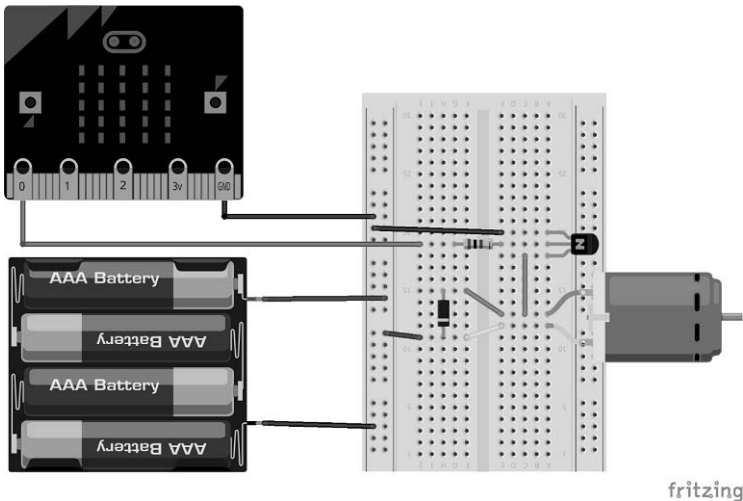
Attention, vous ne devez jamais connecter directement un moteur à un port du micro:bit, car la puissance de sortie est limitée et les moteurs génèrent des courants parasites qui risqueraient d'endommager la carte.

## 1.1 Utilisation d'un transistor

Pour protéger le micro:bit, utilisez un transistor de type NPN (comme le 2N222A, que nous avons déjà utilisé avec le buzzer) et une diode Schottky (de roue libre), pour protéger le transistor d'un retour de courant. Le port du micro:bit doit aussi être protégé, avec une résistance de  $1\text{ k}\Omega$ .

Le micro:bit et le moteur possèdent des alimentations séparées, le transistor sert seulement d'interrupteur. Ce branchement permet au micro:bit de contrôler sans risque un moteur d'une tension supérieure à 3,3 V. Cependant, le moteur ne peut tourner que dans un seul sens.

### 1.1.1 Branchement



Branchement d'un moteur avec un transistor

## 1.1.2 Programmation

En mode numérique, vous ne pourrez pas contrôler sa vitesse, car vous n'aurez que deux possibilités (0 ou 1), soit le moteur est à l'arrêt, soit il tourne à la vitesse maximum. Par contre, en mode analogique, vous avez le choix entre 1 024 vitesses différentes (de 0 à 1 023). La régulation de la vitesse du moteur fonctionne comme avec une LED (cf. *chapitre Entrées et sorties - Actionneurs - LED (sortie analogique)*).

Dans cet exemple, vous pouvez augmenter la vitesse avec le bouton B et la réduire avec le bouton A.

### MakeCode

```

au démarrage
  définir vitesse à 0
  écrire sur la broche P0 la valeur vitesse

lorsque le bouton A est pressé
  changer vitesse par -100
  si vitesse < 0 alors
    définir vitesse à 0
  +
  écrire sur la broche P0 la valeur vitesse

lorsque le bouton B est pressé
  changer vitesse par 100
  si vitesse > 1023 alors
    définir vitesse à 1023
  +
  écrire sur la broche P0 la valeur vitesse
  
```

Contrôle de la vitesse du moteur

## MicroPython

```
# MoteurTransistor
# Appel de la bibliothèque microbit
from microbit import *
# variable
vitesse = 0
# Applique la vitesse au moteur
pin0.write_analog(vitesse)

# Boucle infinie
while True:
    # Si bouton A appuyé
    if button_a.is_pressed():
        # Réduis la vitesse de 100
        vitesse = vitesse - 100
        # Si vitesse inférieure à 0
        if vitesse < 0:
            # Affecte la valeur 0 à vitesse
            vitesse = 0
        # Applique la vitesse au moteur
        pin0.write_analog(vitesse)
        # Pause pour relâcher le bouton
        sleep(350)
    # Sinon et si bouton B appuyé
    elif button_b.is_pressed():
        # Augmente la vitesse de 100
        vitesse = vitesse + 100
        # Si vitesse supérieur à 1023
        if vitesse > 1023:
            # Affecte la valeur 1023 à vitesse
            vitesse = 1023
        # Applique la vitesse au moteur
        pin0.write_analog(vitesse)
        # Pause pour relâcher le bouton
        sleep(350)
    # Petite pause (0,01 seconde)
    sleep(10)
```

## 1.2 Pont en H

Il s'agit d'un astucieux branchement constitué de quatre transistors (avec leurs diodes de protection). Il permet d'inverser le sens de rotation du moteur, en utilisant deux ports du micro:bit (un pour chaque sens).



### Remarque

Pour simplifier la lecture et expliquer le principe de fonctionnement du pont en H, sur le schéma, les transistors sont remplacés par des interrupteurs.

Les ports du micro:bit contrôlent deux transistors en même temps. Si T1 et T4 laissent passer le courant, le moteur tourne dans le sens des aiguilles d'une montre. Mais si c'est T2 et T3, le moteur tourne en sens inverse. Bien sûr, rien n'empêche de modifier la vitesse avec le mode analogique.

Il est possible d'ajouter un 5<sup>e</sup> transistor (T5) qui sert d'interrupteur général (quel que soit le sens du moteur). Dans ce cas, il sert aussi à changer la vitesse.

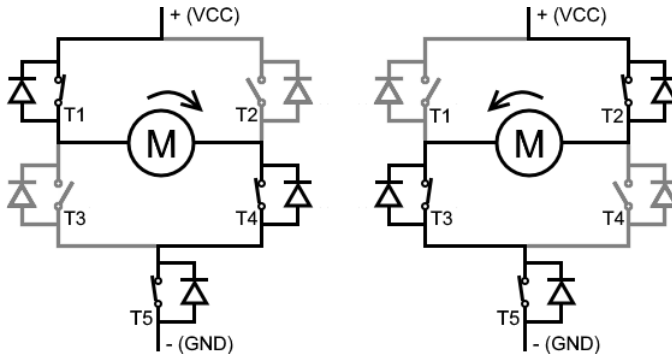
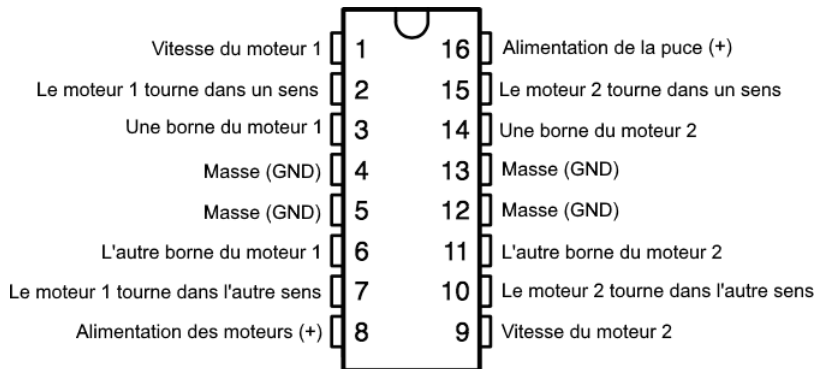


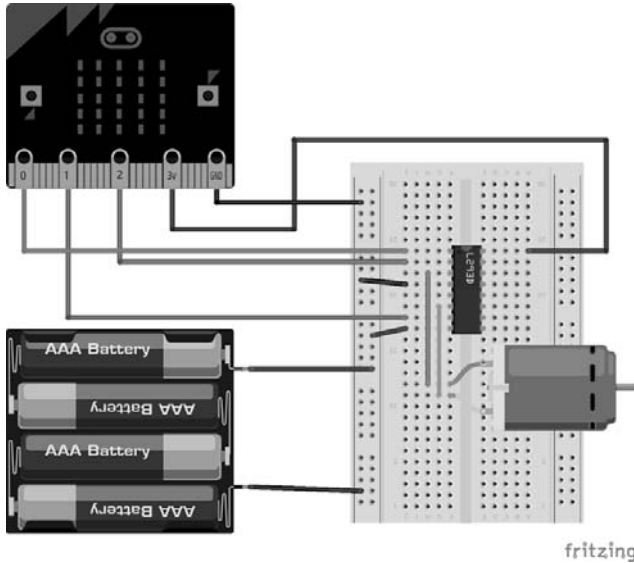
Schéma du pont en H

Vous pouvez bien sûr réaliser vous-même le pont en H. Cependant, la puce L293D fait non seulement la même chose, mais en plus, elle contient deux ponts en H. Elle permet donc de piloter deux moteurs dans les deux sens (ou quatre moteurs dans un seul sens). Elle simplifie aussi énormément les branchements.



Les broches du circuit intégré L293D (double pont en H)

### 1.2.1 Branchement



Branchement d'un moteur avec la puce L293D (double pont en H)

### 1.2.2 Programmation

Le sens de rotation est géré par les ports P1 et P2, en mode numérique (0 ou 1). Si P1 est à 0 et P2 à 1, le moteur tourne dans un sens. Et si P1 est à 1 et P2 à 0, le moteur tourne dans l'autre sens. Le port P0 gère la vitesse de rotation ; en mode analogique, à 0 le moteur est arrêté. Le moteur tourne plus ou moins vite, en fonction de la variable **vitesse** (de 0 à 1023). Comme dans le programme précédent, les boutons A et B diminuent ou augmentent la valeur de la variable **vitesse**. Mais maintenant, elle peut être négative (dans ce cas, le moteur tourne en sens inverse).

### MakeCode



#### Remarque

Attention, il existe un risque de confusion de ces deux blocs. Le premier gère le port P0 comme une sortie numérique et le deuxième comme une sortie analogique.