

CONTRÔLE MOTEUR DC VIA L293D (H BRIDGE DRIVER, CIRCUIT INTÉGRÉ PONT H)

Introduction

Si le montage est un peu fastidieux, il permet de comprendre comment fonctionne un pont H. Pour l'utilisation de tous les jours, il existe des Ponts H pré-assemblés dans des circuits intégrés (ça c'est le pied).

Dans la gamme de circuit intégré "H Bridge Driver" disponibles, les plus courants sont le L293D (un peu désuet) et le L293E (disponible chez Farnell en Belgique).

De l'information en Français

Le net regorge d'informations concernant l'utilisation d'un L293D avec Arduino ou un PIC.

Une simple requête sur Google avec les mots clés "L293D + Arduino + DC Motor" retourne une liste impressionnante.

Cependant la plupart de ces références sont en anglais.

Contrôler un moteur DC avec Arduino et un circuit intégré L293D

Source: Cet article est une traduction "at the best" de l'article "[Control a DC motor with Arduino and L293D chip](#)" de Luky Larry.

Voici un guide rapide avec quelques information complémentaire (configuration du circuit intégré etc) des choses que j'ai apprissent tout au long de mes essais pour utiliser un **L293D** avec un **Arduino**.

Ce guide présentant comment nous pouvons:

1. Utiliser **une alimentation supplémentaire pour fournir la puissance** nécessaire au moteur DC.
2. Utiliser le circuit intégré L293D pour piloter le moteur
3. Utiliser un bouton poussoir pour changer la direction du moteur.

Mise à jour: Si vous avez l'intention d'utiliser un L293D pour faire de la robotique alors vérifiez l'article suivant pour apprendre comment exploiter ce circuit intégré au maximum – j'ai découvert le circuit **SN754410** plus facile à utiliser que le L293D, il est exactement le même à part qu'il est capable de gérer un courant plus important Arduino obstacle avoidance robot

Note personnelle: Il existe également le LD293E disponible chez Farnell.

Il est également capable de conduire plus de courant et le montage reste similaire. Pour plus d'information sur les différences du L293E, vous pouvez consulter l'article "Contrôle Moteur DC via L293E".

Même si je n'ai utilisé qu'un seul moteur, il est possible de commander deux moteurs (dans les deux sens) avec un seul L293D chip. Dans ce cas, vous devez vous assurer de pouvoir fournir assez de courant pour alimenter convenablement les deux moteurs pour les charges maximales (peak load).

Souvenez-vous que si vous utilisez deux moteurs, l'alimentation aura toujours la même tension mais le courant sera doublé - un bon point de départ pour répondre à cette demande est de modifier la façon dont vous connectez vos batteries en parallèles ou en série.

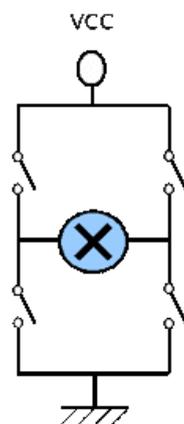
“Le L293D un circuit intégré monolithique, à haut voltage, grand courant, 4 canal pilote.”

Cela veut dire que ce circuit intégré peut être utilisé pour des moteurs DC et alimentation jusqu'à 36 Volts (ce sont déjà des jolis petits moteurs) et que le circuit peut fournir un maximum de 600mA par canal.

Le L293D est aussi connu pour être une sorte de Pont-H. Typiquement, un pont H est un circuit électrique qui permet d'appliquer une tension en sortie sur une charge dans une direction ou l'autre. Par exemple, un moteur.

Cela signifie principalement que vous pouvez inverser la direction du courant et donc renverser le sens de rotation du moteur. Le principe de fonctionnement est basé sur 4 éléments du circuit communément nommés contacts et identifiés comme suit: Haut Côté Gauche, Haut Côté Droit, Bas Côté Droit et Bas Côté Gauche.

En utilisant différentes combinaisons de fermetures, il est possible de démarrer, Stopper ou Inverser le courant.



Il est possible de faire ce montage à partir de relai mais il est plus simple d'utiliser un Circuit Intégré - Le L293D est double Pont-H convenable avec 1 Pont-H de chaque côté du circuit (soit un Pont-H par moteur).

La seule chose à laquelle il faut faire attention dans tout cela, ce sont les 2 broches d'entrée (Input Pins) qui commande la logique pour chaque moteur. Le plus important pour nos besoins, c'est que ces entrées sont commandables depuis un Board Arduino.

Il n'est pas vraiment nécessaire de se préoccuper de la régulation de la tension parce que le L293D accepte deux sources de tensions.

- Une source d'alimentation directe (jusqu'à 36 Volts) pour alimenter les moteurs.
- Une autre source de tension (5 Volts) pour alimenter la logique du circuit intégré. *Cette alimentation peut provenir directement de la carte Arduino.*

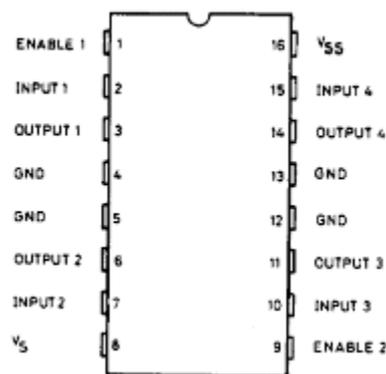
Puisque l'alimentation de mon moteur est de 6 Volts, Je vais directement l'utiliser (si la tension d'alimentation du moteur était plus élevée, je devrais considérer l'usage d'un transistor ou d'un régulateur de tension).

La seule chose dont il faut se souvenir, c'est qu'il faut raccorder ensemble la masse (GND) d'Arduino et celle de l'alimentation externe.

La masse doit être commune entre les alimentations (d'Arduino et du Pont-H).

Note Personnelle: Il est aussi possible (voire recommander) d'alimenter la logique du L293D avec Arduino et de garder l'alimentation de "puissance" uniquement pour le circuit de puissance.

Vous pouvez voir ci-dessous la configuration des broches du L293D et la table de la logique de commande.



Source: [Datasheet du L293D](#)

- VSS - Alimentation de la logique de commande (5V).
A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino).

- VS - Alimentation de puissance des moteurs.
Par exemple, s'il s'agit d'un ancien véhicule téléguidé que vous avez canibalisé, il s'agira de la borne positive (+9.2v) de votre accumulateur.

- GND - Doit être raccorder à la masse (GND) de la source d'alimentation de puissance VS (donc le négatif de l'accumulateur) **et** à la masse de la source d'alimentation de VSS (donc GND Arduino).

Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.

- OUTPUT1, OUTPUT2 - Broches à raccorder à la charge (le moteur). C'est via ces broches de
- INPUT1, INPUT2 - Broche de commande du Pont-H. Se raccorde a Arduino.
- ENABLE1 - permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2.

ENABLE1 commande l'activation du premier Pont-H.

Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas.

Si ENABLE1 = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 & INPUT2.

Importance de la broche ENABLE1:

Si l'on ne désire pas faire de commande en vitesse et que le Pont-H sert uniquement en commande tout ou rien, alors dans ce cas, il suffit de raccorder ENABLE1 à VSS.

Par contre, ENABLE1 offre l'avantage de pouvoir moduler la vitesse du moteur en y appliquant un train d'onde PWM.

Il est ainsi possible de commander assez finement une moteur DC avec 3 broches Arduino (dont une PWM).

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

D'une façon générale, une moteur DC nécessite bien plus de courant et tension que le board Arduino est capable de lui fournir.

Dans ce cas pratique, le moteur que j'utilise necessite que 5 à 6 Volts.

je pourrais aussi alimenter mon Arduino avec une source de tension de 12 Volts. Mais cela la viderais assez vite si elle devait tout alimenter, plus spécialement si j'ajoutais un autre moteur et un couple de Servo.

Donc, a la place, mon Arduino fonctionne avec une source d'alimentation séparée de 9 Volts que j'ai faite moi même.

Capacités de découplages

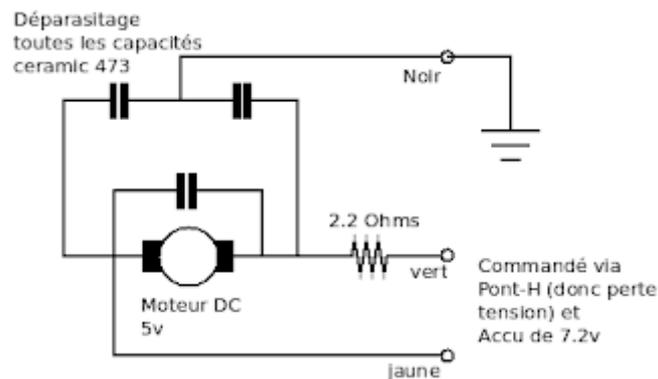
Vous avez aussi besoin de quelques capacités dans votre circuit pour réguler la puissance de charge aux moteurs autant que possible et ainsi éviter les pointes de courant et stabiliser le

courant.

J'utilise une capacité de 10 microFarad sur la source d'alimentation - je suggère d'utiliser cela comme valeur minimale.

Note personnelle: J'ai eu l'occasion d'utiliser une capacité polarisée de 33 μ F.

Vous pouvez aussi ajouter des capacités sur chaque moteur que vous utilisez - quelque-chose comme des capacités ceramique 220nF multi-couche devrait être bon pour de petits moteurs.



Commande de direction d'un véhicule Nikko.
La direction est commandée par Pont-H à transistor.

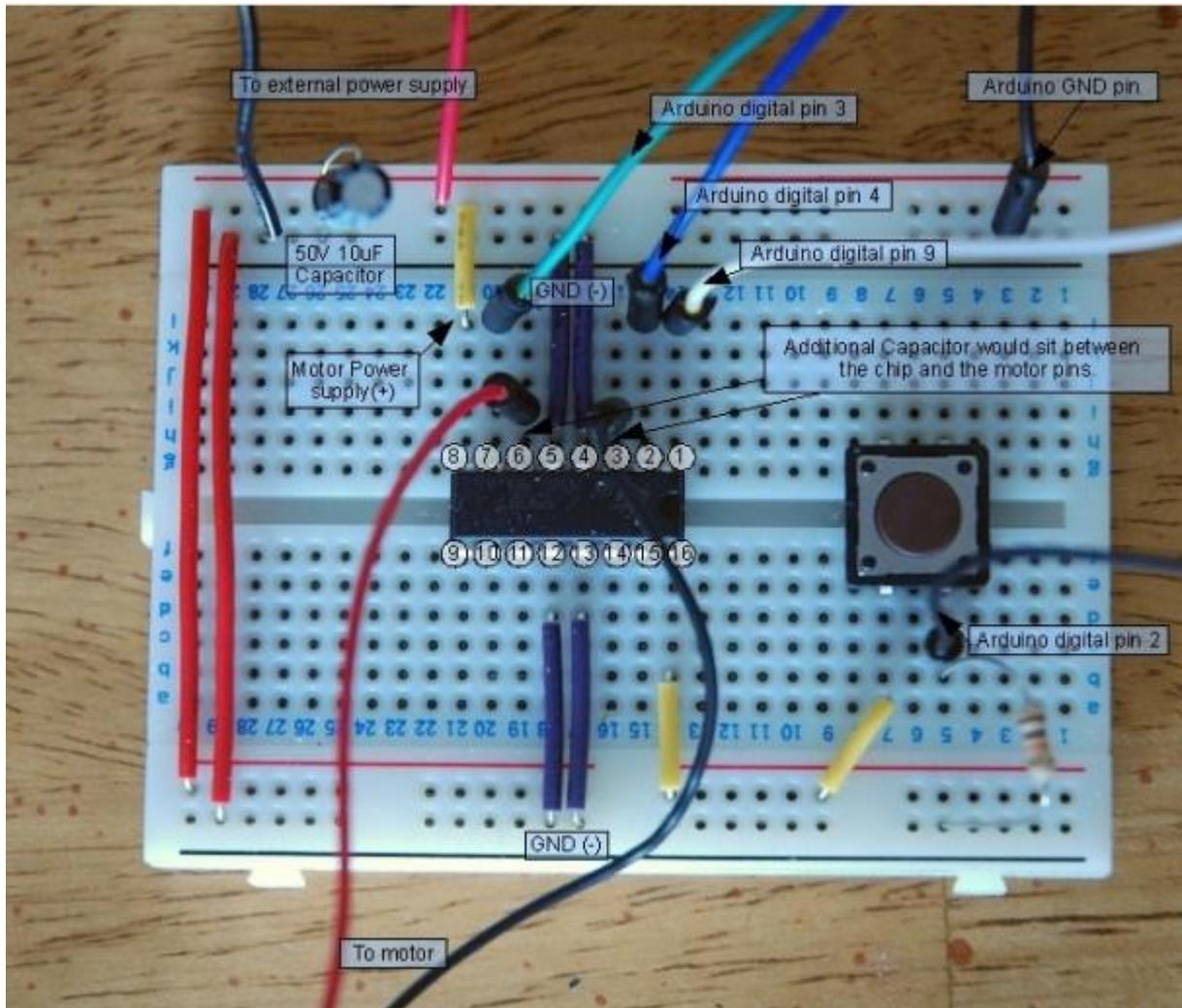
Montage du circuit L293D pour Arduino

Matériel

- Résistance de 10 K Ohm (brun, noir, orange, or)
- Capacité 50V 10 μ F
- Moteur DC 6V
- H Bridge L293D (circuit intégré)
- Un switch (bouton poussoir, etc..)
- Un Arduino (ex: Deumilanove w/ ATMEGA328)
- Un breadboard ou board de prototypage
- Fils de connection
- Container pour 4 piles AA
- 4 piles AA
- Capacité céramique multicouche 220nF (Y5V) - optionel
- Alimentation externe [9V DC](#) ou utiliser le câble USB pour alimenter le board Arduino

Montage

Commençons par la pins 16 sur le L293D et tout ce qui doit y être raccordé. Comme vous pouvez le constater, il y a deux côté sur le L293D, 1 côté pour chaque moteur.



Source: [Article de Luky Larry](#)

1. Active/désactive le moteur peut importe qu'il soit en marche ou non (high ou low). Raccordé à la pin 9 d'Arduino (digital PWM)
2. Contrôle logique du moteur pour le moteur (INPUT qui est soit high ou low raccordé sur la pin 4 d'Arduino (digital)
3. Est connecté à une borne du moteur (soit +, soit -)
4. GND
5. GND
6. Est connecté sur l'autre borne du moteur.
7. Contrôle logique pour le moteur (INPUT qui est soit high ou low) raccordée sur la pin 3 d'Arduino (digital)
8. Alimentation de puissance du moteur. Devrait assumer la charge en courant (et la tension) pour le moteur. Dans cet exemple, c'est une alimentation de 6V.
9. Active/désactive le second moteur.
10. Contrôle logique du second moteur (INPUT qui est soit high soit low)
11. Est connecté à une borne du second moteur.
12. GND
13. GND
14. Est connecté à l'autre borne du second moteur.

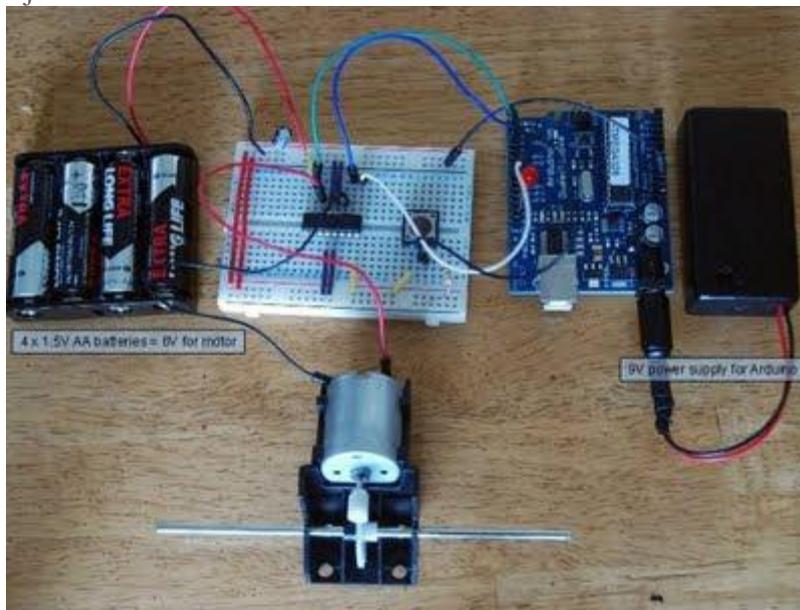
15. Contrôle logique du second moteur (INPUT qui est soit high soit low)
16. Alimentation du circuit intégré +5V.
Dans ce cas, la même alimentation que celle utilisée pour le moteur.

Vous pouvez voir sur la photo ci-dessous comment le L293 est raccordé au différents composants externes.

J'ai raccordé mon bouton poussoir sur la pin 2 d'Arduino (digital) and raccordé la masse (GND) d'Arduino sur le rail de masse de mon breadboard.

J'ai aussi ajouté une capacité entre les pôles de l'alimentation - faite attention de bien respecter la polarité.

Finalement, j'ai ajouter des fils pour alimenter les deux côtés de mon breadboard (les 2 fils rouges sur la gauche) et ajouté le moteur et son alimentation.



Source: [l'article de Luky Larry](#)

Le code d'exemple

```
int switchPin = 2;    // switch input

int motor1Pin1 = 3;   // pin 2 on L293D

int motor1Pin2 = 4;   // pin 7 on L293D

int enablePin = 9;    // pin 1 on L293D

void setup() {

    // set the switch as an input:

    pinMode(switchPin, INPUT);
```

```
// set all the other pins you're using as outputs:

pinMode(motor1Pin1, OUTPUT);

pinMode(motor1Pin2, OUTPUT);

pinMode(enablePin, OUTPUT);

// set enablePin high so that motor can turn on:

digitalWrite(enablePin, HIGH);

}

void loop() {

// if the switch is high, motor will turn on one direction:

if (digitalRead(switchPin) == HIGH) {

    digitalWrite(motor1Pin1, LOW); // set pin 2 on L293D low

    digitalWrite(motor1Pin2, HIGH); // set pin 7 on L293D high

}

// if the switch is low, motor will turn in the opposite direction:

else {

    digitalWrite(motor1Pin1, HIGH); // set pin 2 on L293D high

    digitalWrite(motor1Pin2, LOW); // set pin 7 on L293D low

}

}
```