Réalisation : Variateur 12 V



Page non-finie, mais lisez quand même !!

I - Introduction

Cette mini-réalisation est faite pour voir (ou revoir) une application du PWM dans un cas concret : Nous allons créer un **variateur** c'est à dire un circuit permettant d'alimenter un dipôle de puissance (lampe, moteur à courant continu) avec une tension variable par hacheur (PWM).

Mise en situation

- Le variateur sera commandé par deux boutons poussoirs, un permettant d'augmenter la tension (UP) et un permettant de la diminuer (DOWN)
- On va créer un variateur pouvant commander un dipôle consommant **0.5A**, mais il est possible de commander de plus grand (en modifiant les transistors et avec des dissipateurs)



Il faudra donc faire attention à la température des transistors qui pourraient griller !

Qui est qui ?

les entrées

- On dira que la pin digitale 1 est connectée à UP
- De même pour la pin 2 qui sera connectée à DOWN

les sorties

- Il n'y en a qu'une, il faudra que cette pin de sortie **soit équipée du PWM** , dans mon exemple je prendrais la pin digitale 3, mais il **faudra vérifier**... cela dépend du modèle de carte que vous utilisez.

II - Ressources Électronique

A/ Un problème de conversion de puissance

Bien sûr, **l'Arduino** ne peut pas délivrer plus de **30mA**, il faudra qu'elle demande assistance à un collègue costaud, j'ai nommé le **transistor de puissance** !
Pour cela, il faudra le connecter en sortie de l'arduino, sur la patte qui délivrera le pas de PWM, avec quelques précautions (résistance protectrice de 10K par exemple).



Bon à savoir : ** le PWM n'est pas continu… il est composé de créneaux d'une fréquence environ égale à 500Hz, pour ce montage, cela ne pose pas de problème car : - Le moteur à courant continu est un **dipôle inductif, un bon vieux circuit RL, ayant pour propriété de lisser la tension quand il est en série.

1. L'ampoule à incandescence quant à elle n'est **pas affectée** par ces phénomènes de fréquence relativement haute (il est impossible de les voir à l'œil nu : c'est l'intensité lumineuse moyenne qu'on perçoit)

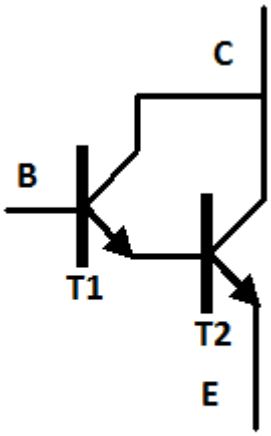


Si l'on veut lisser **le courant**, on pourra utiliser un circuit RC mais c'est inutile ici…

B/ Le montage Darlington

Afin de pouvoir alimenter votre moteur CC, il faut utiliser un montage amplificateur qui permet de travailler avec des courants assez élevés. Je vous propose donc d'utiliser …

** Le montage Darlington !!! ** 



Montage Darlington

Ce montage est composé de 2 transistors :

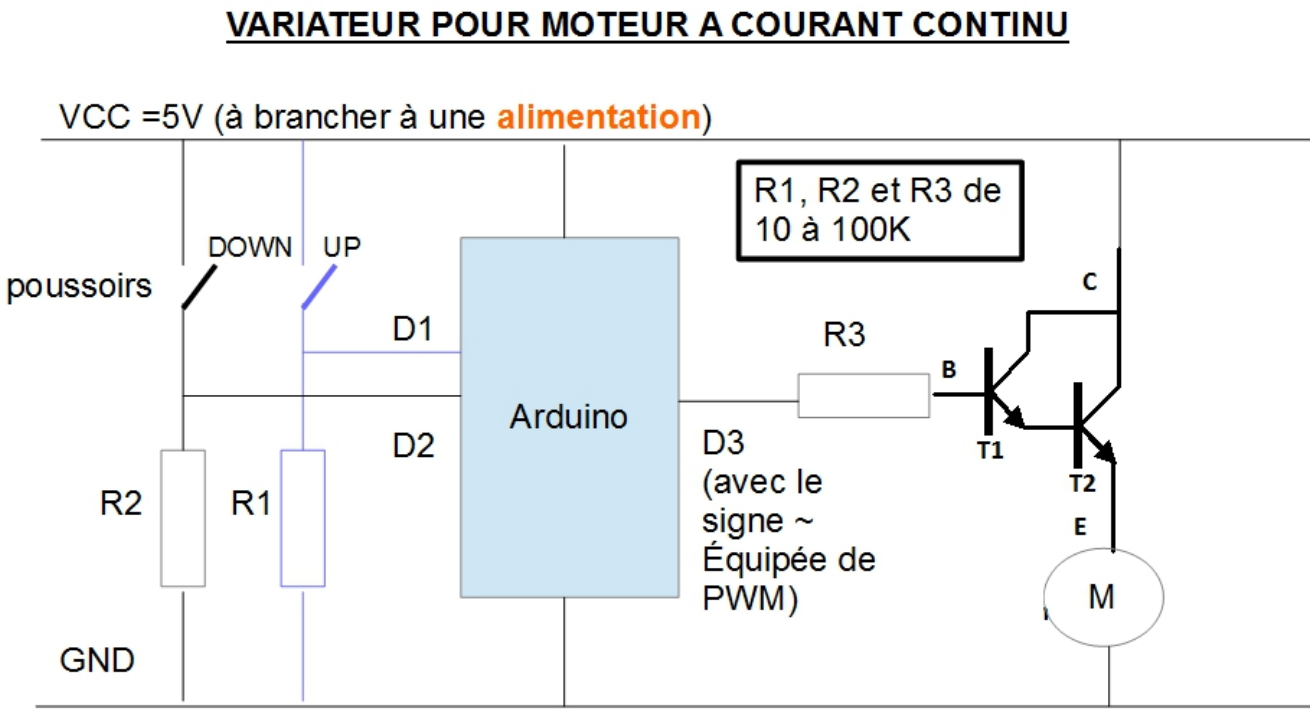
- T1 est un **petit transistor** qui est doté d'un grand gain mais il ne peut supporter une grande charge…
- T2 est un **GRAND transistor** qui est pourvu d'un petit gain mais il peut supporter des grandes

intensités.

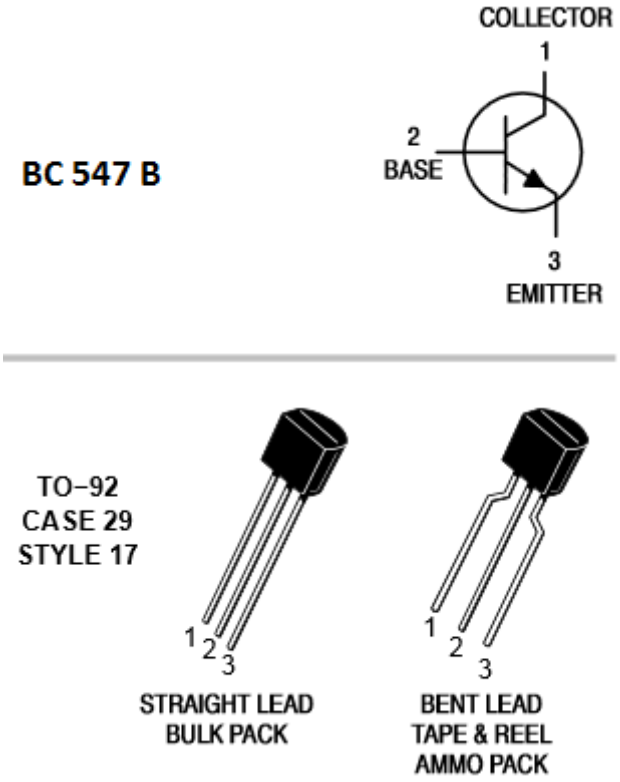
Bref, en combinant les deux, on obtient un transistor équivalent qui dispose d'un **GRAND gain** et qui peut supporter **une GRANDE intensité** en bref, qui peut alimenter un **moteur CC** !!

On ajoutera néanmoins une résistance en amont de la base pour protéger votre Arduino adorée

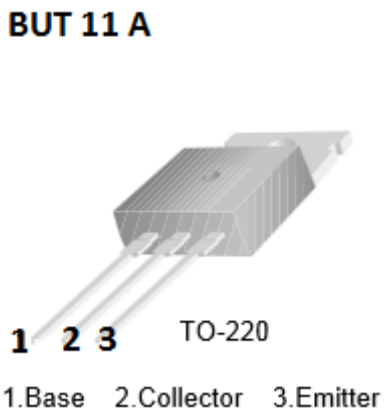
C/ Schéma électronique



On prendra T1 (le transistor faible) qui est un BC547.
** Brochage : **



On prendra T2 (le transistor fort) qui est un BUT11A.
** Brochage : **



III - Programmation

// Vu que je n'ai pas encore testé ce programme faute de boutons-poussoirs (honte à moi !) je mets ici un programme alternatif n'ayant pas de boutons poussoirs... // **NB**: j'ai prévu trois sorties pour LEDS pour me faire pardonner^^

```
int sortie = 3;
int rouge = 8;
int jaune = 9;
int vert = 10;
void setup () {
  pinMode(sortie, OUTPUT);
  pinMode(rouge, OUTPUT);
  pinMode(jaune, OUTPUT);
  pinMode(vert, OUTPUT);
```

```
}
void loop () {
  digitalWrite( rouge, LOW);
  int i = 10;
  while ( i<50) {
    analogWrite( sortie, i*5);
    i++;
   if( i < 20) {
    digitalWrite( vert, HIGH); }
  if( (i>20)&&(i<30)){
    digitalWrite( vert, LOW);
    digitalWrite( jaune, HIGH);
  }
  if(i>30) {
    digitalWrite( jaune,LOW);
    digitalWrite( rouge, HIGH);
    }
  delay(1000);
   }

}
```

From:
https://wiki.centrale-med.fr/fablab/ - **WiKi fablab**

Permanent link:
https://wiki.centrale-med.fr/fablab/start:projet:arduino:variateur

Last update: **2015/01/20 15:11**