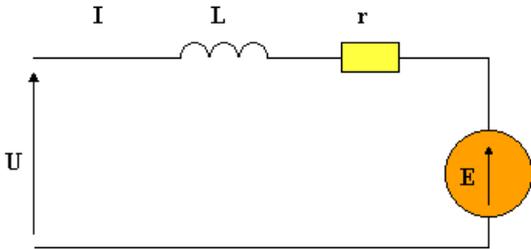


Commande PWM des moteurs

Comment faire varier la vitesse d'un moteur ?

Rappels:

Un moteur à courant continu alimenté par une tension U peut être modélisé par une résistance r en série avec une inductance L , E étant la force électromotrice du moteur:



$$U = E + ri + L \frac{di}{dt}$$

De plus, on sait par les lois de l'électromagnétisme que :

$$E = k \cdot \omega$$

La force électromotrice est proportionnelle à la vitesse

$$C_p = k \cdot i$$

Le couple est proportionnel à l'intensité

En **régime établi**, l'intensité i est constante (donc $di/dt = 0$) et $E = U - r \cdot I$

Conséquences sur la vitesse ω

$$\omega = \frac{E}{K} = \frac{U - ri}{K}$$

La vitesse de rotation ω en rd/s dépend de la tension U et de i

↳ Pour modifier la vitesse de rotation d'un moteur, il faut faire varier la fem E en agissant sur:

- la tension U **Commande en tension**
- le courant $I \Rightarrow$ **Commande du couple**

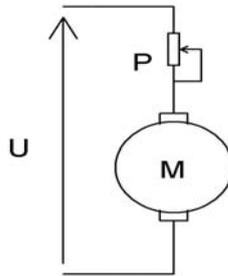
Solution 1: Variation de vitesse par potentiomètre

C'est la solution qui paraît la plus simple.

On limite le courant en augmentant la résistance.

On agit alors essentiellement sur le couple.

La fem $E = U - (r+P) \cdot I$ diminue également ce qui se traduit par une variation de vitesse.



exemple: $r = 1 \Omega$ $P = 10 \Omega$ et $U = 12V$

Si l'on veut diviser la vitesse par 2 soit $E \approx 5V$

On a: $5 = 12 - 11 \cdot I \Rightarrow I = 0.5A$

Le couple est également divisé par 2.

La puissance totale est $P_t = U \cdot I = 6W$

La puissance dissipée par le potentiomètre est dans ce cas est $P_p = r_i^2 = 2.5W!$

\Rightarrow rendement 60%!

Conclusion : Solution pas terrible (baisse du couple, de la puissance, du rendement)

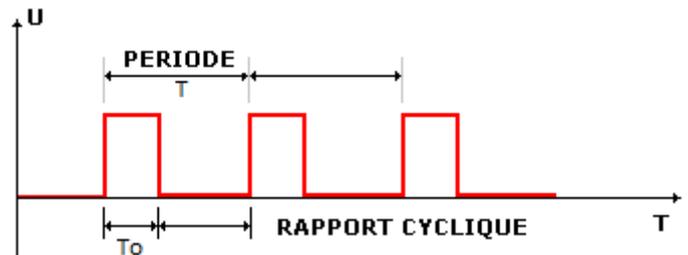
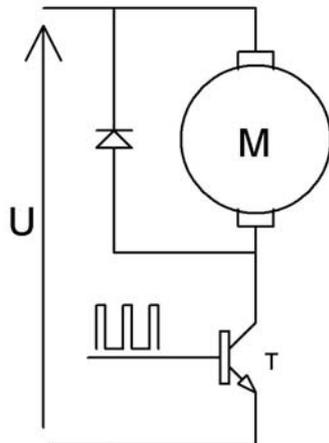
Solution 2: La commande PWM ou MLI

(Pulse width modulation)(Modulation en Largeur d'Impulsions)

Cela consiste à alimenter le moteur avec une tension en créneaux.

La tension moyenne dépend alors du rapport cyclique $\frac{T_o}{T}$.

La vitesse varie en fonction de cette tension moyenne.



$$U_{moy} = \frac{1}{T} \int_0^T u(t) dt = \frac{1}{T} [U_{max} \cdot t]_0^{T_o} = U_{max} \cdot \frac{T_o}{T}$$

Le rapport cyclique est exprimé en %

Pourquoi la commande PWM est-elle intéressante ?

Il y a deux avantages à piloter la vitesse d'un moteur en PWM:

a) Le moteur peut être piloté par la sortie numérique d'un microcontrôleur

b) Amélioration considérable du rendement énergétique:

En effet, le transistor T fonctionne en commutation (saturé/bloqué).

Lorsqu'il est **saturé** la tension **Vce est très faible** (presque nulle) donc la puissance dissipée par ce transistor est $P_t = V_{ce} \cdot I = 0$

Lorsqu'il est **bloqué**, c'est le courant **I qui est nul** donc $P_t = V_{ce} \cdot I = 0$

On dit qu'il ne consomme que **pendant les transitions**, c'est à dire pendant un temps très court. Les dissipateurs thermiques sont donc de ce fait plus petits voir inutiles. Ceci est encore plus vrai lorsque T est un MOSFET de puissance.

Toute l'énergie est donc consacrée à la rotation du moteur. Ce qui est intéressant pour des systèmes embarqués fonctionnant sur batterie.

Quelle fréquence de commande ?

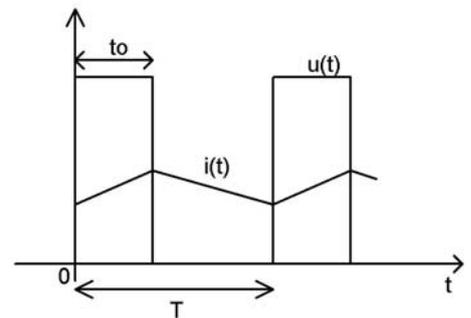
Fréquence élevée:

L'allure du courant est donnée ci contre (système du 1er ordre).

Si la fréquence $1/T$ est suffisamment grande, la variation $di(t)$ sur une période est petite et **$i(t)$ peut être considéré constant donc le couple constant.**

On considère qu'il faut : $\omega \gg \omega_0$

Une diminution du rapport cyclique se traduit par une diminution du courant moyen dans le moteur donc une diminution du couple moyen.



Les petites variations de $i(t)$ se traduisent par des variations de couple qui entraînent des vibrations de l'axe moteur. La plupart du temps ces vibrations sont intégrées par la mécanique accouplée à l'axe moteur et sont sans conséquences sur la vitesse.

Mais pas trop:

⚡ Il ne faut pas choisir une fréquence trop importante. Le signal MLI serait, dans ce cas, trapézoïdal et non plus rectangulaire. La durée des transitions ne serait plus négligeable devant la période du signal MLI et le transistor dissiperait davantage d'énergie.

⚡ Il faut vérifier la **fréquence maximale du hacheur** de la carte de commande : Nous utilisons la carte Pololu MD03A avec des transistors Mofsets Vnh2sp30 dont la fréquence max est de **20 KHz**.

⚡ Si F est choisie dans la **bande audible**, les vibrations mécaniques du moteur seront elles mêmes audibles.

Caractéristiques d'un moteur Maxxon

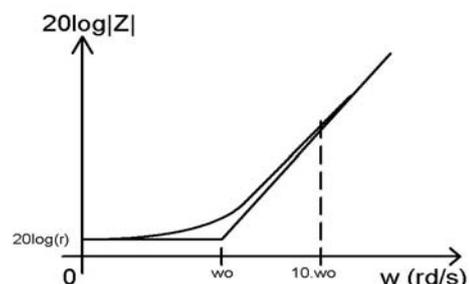


RE025 G -12V

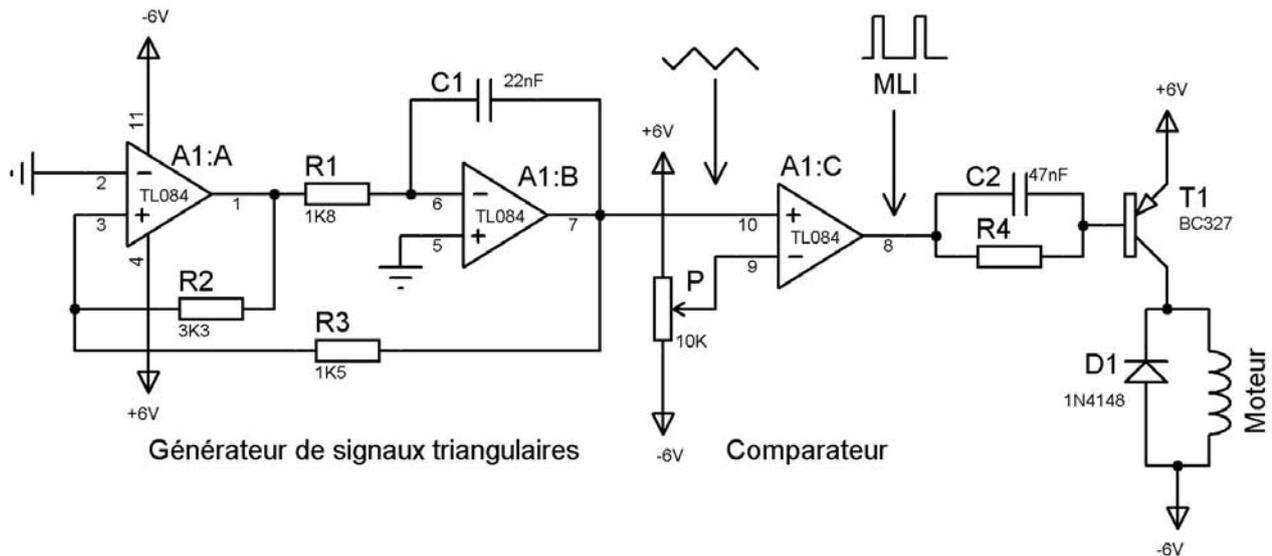
Tension	12 V
Masse	130 g
Puissance utile maxi	27.6 W
Couple de démarrage	155 mN.m
Courant de démarrage	9,53 A
Résistance	1.26 ohm
Self	0.115 mH
Constante de couple	0.0163 Nm/A
Vitesse de rotation	6800 tr/mn
Rendement	84%

Pour le moteur Maxxon utilisé,
 $r = 1.26 \text{ ohm}$ et $L = 0.115 \text{ mH}$
 La constante de temps du moteur est $\tau = L/r$:
 la pulsation $\omega_0 = 1/\tau = r/L$ et $f_0 = \omega_0/2\pi = 1744\text{Hz}$

Pour avoir un courant constant dans le moteur
 il faut choisir $\omega \gg \omega_0$ ($\omega > 10\omega_0$)
 On prendra **F = 18 KHz**
 (gamme audible par l'homme entre 20Hz et 20kHz)
 Dans la pratique on prendra: $1\text{KHz} < F < 20\text{KHz}$



Exemple de schéma de commande de moteur en MLI



Merci au travail réalisé par Dominique Nardi

Comment programmer le PWM ?

- Les microcontrôleurs ont des sorties PWM. Celles-ci utilisent les timers (ou compteurs) du microcontrôleur avec leurs options : choix de la résolution, valeur de débordement (ou comparateurs) et prédiviseur du temps (prescale).

- Fréquence d'horloge** : elle dépend du microcontrôleur utilisé :

exemple : 8 MHz

- Fréquence avec prédiviseur (Prescale)**: le prédiviseur peut être 1,2,4,8,16,32,64,128,256...1024

exemple avec un prédiviseur de 8 : 1MHz

- Timer (Compteur)** : La résolution du timer dépend du timer choisi (timer 0, 1 ou 2)

exemple traité : le timer 1 de l'atmega 32 est sur 16bits mais la fonction PWM ne peut utiliser que 8,9, ou 10 bits : choix de résolution sur 10 bits (soit 1024)

- Valeur de débordement** : fixé en fonction du rapport cyclique souhaité (To/T)

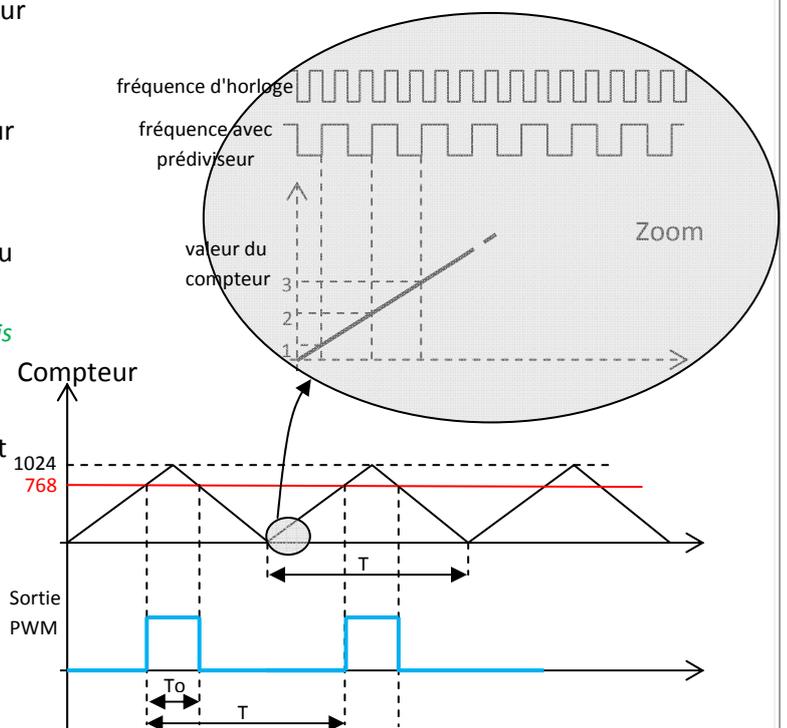
exemple : valeur = 768 donc rapport cyclique = (768/1024) = 075=75%

- Fréquence du PWM**:

Elle est donnée par la relation:

$$F_{PWM} = \frac{F_{clock}}{2 \times prescale \times resolution}$$

exemple : $F_{PWM} = 8000000 / (2 \times 8 \times 1024) = 488 \text{ Hz}$



- Remarques:**

Le microcontrôleur possède en réalité plusieurs modes de PWM (un rapide qui ne compte que à la montée puis repart de 0, un à fréquence corrigé...). Celui qui est décrit ici est le **mode PWM correct**, moins rapide mais plus adapté

à la commande de moteurs.

👉 Le mode décrit ici s'appelle **Clear up** : il est censé mettre à zéro (clear) au dessus (up) de la valeur de débordement. Pour une raison inconnue, il fait le contraire!

Il existe le mode inverse : clear down qui fait le contraire de clear up

👉 Un timer possède **2 sorties** pour le PWM. On peut les utiliser avec des valeurs de débordement différentes (rapports cycliques donc vitesses différentes de 2moteurs) ou des modes différents (clear up et clear down). Ces sorties ont une **adresse imposée** que l'on trouve dans la doc. du microcontrôleur.

👉 Dans le programme, on peut imposer la fréquence d'horloge. Celle ci écrase celle qui est définie dans les paramètres de compilation liés au microcontrôleur utilisé.

\$crystal = 11052000 impose la fréquence, même si le microcontrôleur est à 16Mhz (?)

Exemples de programmation avec Bascom:

extrait 1:

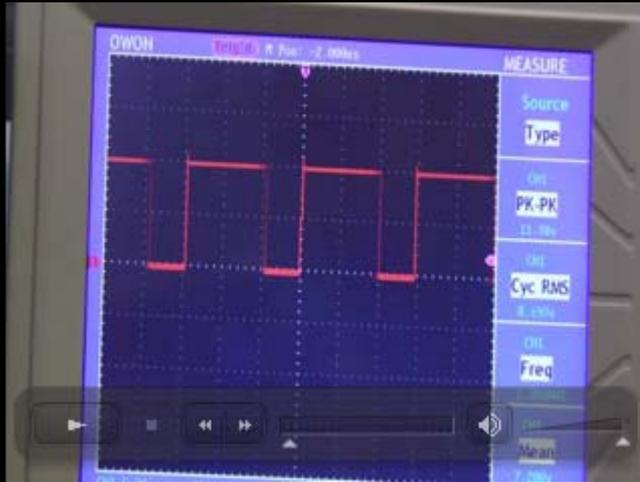
```
...
Config PinD.5 = Output           ports de sortie du pwm
Config PinD.4 = Output
'configuration en 10 bits, prescale de 8 mode clear up, utilisation des 2 sorties (facultative)
Config Timer1 = Pwm , Pwm = 10 , Prescale = 8 , Compare A Pwm = Clear Up , Compare B Pwm = Clear Down
Pwm1a = 768                      valeur de débordement
Pwm1b = 450
Start Timer1
End                               le timer ne s'arrête pas car il n'a pas rencontré de stop timer
```

extrait 2:

```
...
Config PinD.5 = Output
Dim J as word
'configuration en 10 bits, prescale de 8 mode clear up,
Config Timer1 = Pwm , Pwm = 10 , Prescale = 8 , Compare A Pwm = Clear Up
Pwm1a = 768
J = 1
Do
  Pwm1a = J
  J = J + 1                      rapport cyclique variable de 0 à 100%
  If J = 1024 Then J = 1
Loop
Start Timer1
End
```

Vidéos

👉 Une super explication en anglais sur le PWM :



👉 A venir : la vidéo de notre expérience