

Pierre Emmanuel CARLIER  
Damien DE SOUZA  
Damien DUMAY  
Germain FONDEUR  
Thomas MONTJALON  
Sylvain RODRIGUES  
Guillaume TRANNOY



## UV PR

# REALISATION DE LA BASE ROULANTE DU ROBOT DE LA COUPE E=M6

-----  
**ASSOCIATION COUPE E=M6**  
-----

Suiveurs Projet :

**FORGEZ Christophe**

**LANFRANCHI Vincent**

**Automne 2004**

# Sommaire

<b>PRÉSENTATION</b>	page 3
1. La coupe E=M6	page 3
2. Cahier des charges de l'UV PR	page 4
3. L'équipe chargée de l'UV PR	page 5
<b>PARTIE A : DESCRIPTION GÉNÉRALE DE LA BASE ROULANTE</b>	page 6
<b>PARTIE B : LA MÉCANIQUE</b>	page 7
1. Veille technologique	page 7
2. Le choix des moteurs	page 8
3. Les supports moteurs	page 13
4. Les roues	page 17
5. La base	page 18
6. Résultats	page 21
<b>PARTIE C : L'ÉLECTRONIQUE</b>	page 23
1. Les batteries	page 23
2. La commande des moteurs	page 26
3. Système PID	page 27
4. Les microcontrôleurs	page 35
5. Module AVR	page 37
6. Modules d'asservissement analogique « Escap »	page 38
7. Conclusion	page 42
8. Carte d'asservissement numérique « artisanale »	page 42
9. Carte compteur d'impulsions codeurs	page 43
10. Carte de puissance	page 44
11. Carte microcontrôleur	page 45
12. Algorithme	page 49
13. Conclusion	page 50
14. Remarques	page 50
15. Circuits d'asservissement numérique « LM629 »	page 51
16. Conclusion	page 53
17. Commande des circuits d'asservissement numériques « LM629 »	page 53
18. Conclusion	page 58
19. La carte mère	page 59
<b>PARTIE 4 : DIVERS</b>	page 61
1. Budget	page 61
2. Fournisseurs	page 62
3. Usinages	page 63

---

**CONCLUSION**

page 64

**REMERCIEMENTS**

page 64

**ANNEXES**

page 65

# Présentation

## 1. La coupe E=M6

La *Coupe de France de Robotique* est une compétition, au cours de laquelle viennent s'affronter les grandes écoles et universités françaises. Elle est organisée par la chaîne de télévision nationale M6 et l'Association Planète Sciences. Cette compétition se déroule chaque année au mois de mai sur le site de la Ferté Bernard à l'occasion du festival « Arts et Technologie », plus connu sous le nom de ARTEC.

Le défi proposé par la Coupe consiste à concevoir et fabriquer un robot mobile, totalement autonome à partir d'un sujet imposé par les organisateurs de la coupe et présenté au mois de septembre.

L'UTC participe à cette compétition depuis 1996, où un groupe d'étudiants motivés s'était constitué de lui-même pour participer à ce challenge. Aujourd'hui notre force est de profiter de l'enseignement des anciens participants et du savoir-faire acquis.

L'ensemble des personnes impliquées est vraiment déterminé à atteindre des objectifs précis : améliorer nos performances et éliminer nos faiblesses, déjà pour accéder à la phase finale et peut-être remporter la victoire. Il y a aussi la volonté de représenter notre école dans une compétition dont les retombées médiatiques sont de plus en plus importantes, et de la faire jouer dans la même cour que les plus célèbres écoles françaises. Par ailleurs, notre équipe s'est classée 12<sup>ème</sup> en 2002 (sur 190 équipes sélectionnées), l'expérience des participations précédentes ayant été systématiquement mise à profit.

Cette activité inclut également un aspect formateur à ne pas négliger. Pour mener à bien ce projet, il est nécessaire de faire les meilleurs choix dans les délais les plus brefs, de réagir vite et de travailler efficacement. Par ailleurs, ce projet possède un aspect pratique important pour des étudiants ayant des connaissances plutôt théoriques : « mettre la main à la pâte » permet d'apprendre ce qu'implique d'avoir à commander une pièce ou de la fabriquer, d'utiliser un tournevis, de tenir un fer à souder, parcourir une documentation ou encore de déceler l'origine d'une panne.

De plus, ce challenge technique et humain nous permet, d'une part, de mettre en pratique nos connaissances scientifiques à travers une réalisation concrète, et d'autre part, d'acquérir une certaine expérience dans le travail en équipe affilié à un projet complet, au travers de toutes les disciplines impliquées (management, comptabilité, communication, stratégies, mécanique, électronique, informatique...).

## 2. La Coupe de France de Robotique 2005

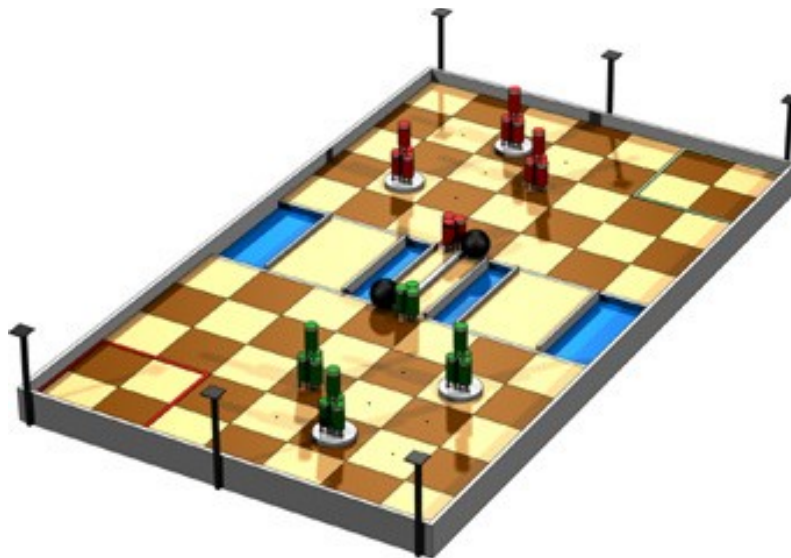
### **Cette année, les robots jouent au bowling !!**

Chaque équipe doit construire un ou deux robots entièrement autonomes. Les matchs se jouent entre deux équipes et durent 1 minute 30.

Une couleur de quilles est attribuée à chaque équipe. Sur le terrain, 15 quilles de chaque couleur sont disposées aléatoirement. Au départ, celles-ci se trouvent de l'autre côté d'un fossé, dans le camp adverse. Des ponts permettent de rejoindre l'autre côté mais ceux-ci sont placés aléatoirement, en début de partie. Il est également possible d'utiliser des balles de type « balle de squash » pour faire du tir balistique et ainsi espérer faire tomber des quilles à distance.

Pour gagner la partie, les robots doivent renverser le plus de quilles possible. Les robots peuvent aussi relever les quilles, de la couleur adverse, renversées par l'autre équipe.

L'équipe qui a le plus de quilles de sa couleur renversées en fin de match est déclarée vainqueur.



*Illustration de la zone de jeu 2005.*

### 3. Cahier des charges de l'UV PR

Chaque année, les stratégies de locomotions des robots se ressemblent, cependant nous partons à chaque fois de zéro.

C'est pourquoi il nous a semblé utile d'essayer de construire une fois pour toute une base roulante réutilisable et fiable.

Nous nous sommes fixé comme objectifs principaux :

- De réaliser une base roulante
  - Facilement réutilisable
  - Fiable
- Commandable :
  - En vitesse
  - En distance
  - En rotation
- Caractéristiques :
  - Pouvant supporter 15kg
  - Ne faisant pas plus de 260 mm de large

### 4. L'équipe chargée de l'UV PR

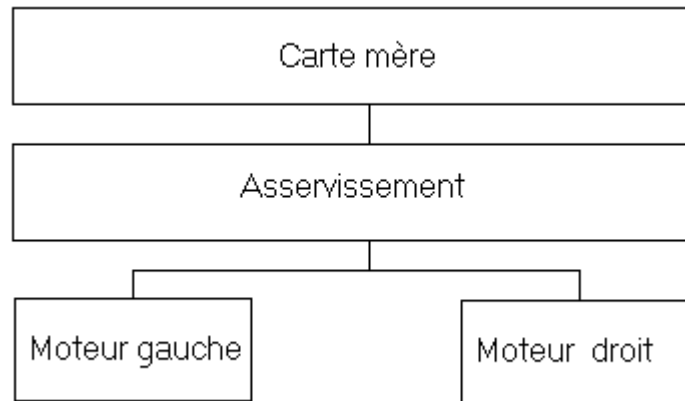
#### Etudiants :

CARLIER Pierre Emmanuel	GM02
DE SOUZA Damien	GSM05 - CMI
DUMAY Damien	GM02
FONDEUR Germain	GM05 - MARS
MONTJALON Thomas	GI05 -
RODRIGUES Sylvain	GM05 - MARS
TRANNOY Guillaume	GM02

#### Suiveurs Projet :

FORGEZ Christophe (laboratoire d'électromécanique de l'U.T.C.)  
 LANFRANCHI Vincent (laboratoire d'électromécanique de l'U.T.C.)

# Partie A : Description générale de la base roulante



## 1. Carte mère :

Il s'agit de la carte « intelligente » du robot, elle commande le déplacement mais aussi les différents actionneurs et capteurs du robot. Cette partie correspond à :

- Motorola HCS12 (MC9S12DP256B)
- Microchip PIC 18F458
- Rabbit (RCM3400)
- PC104
- ...

## 2. Asservissement :

Cette partie correspond au contrôle des actionneurs, elle est l'interface entre la carte mère et les effecteurs. Il peut s'agir de :

- Carte de puissance
- Pont en H
- Contrôle de vitesse
- ...



## Partie B : La mécanique

### 1. Veille technologique :

Afin de pouvoir éviter de proposer ou de faire des choses dépassées, inutiles ou non fiables, nous avons procédé à une petite veille technologique de tout ce qui se fait à la Coupe de France de Robotique. Pour cela, nous nous sommes basés sur le site de l'ANSTJ, qui est l'association qui organise la Coupe de France. Sur son site, l'ANSTJ a recensé tous les posters des équipes de la compétition de 2004. Ces posters sont des documents remis à l'organisation en mai 2004 lors de la dernière compétition. A partir de ces posters, nous avons regardé comment les équipes avaient fait leur(s) robot(s) :

- Quel système de déplacement (roues, chenilles, pattes) ?
- Quel type de moteurs ?
- Comment et où les moteurs étaient placés ?
- Les accouplements (directs ou indirects) ?
- La forme des robots ?
- La modularité du « bloc moteur » ?

Toutes ces informations sont recensées dans un tableau Excel. Ce document est en annexes.

Les principales informations recueillies de cette veille sont :

- La plupart des équipes utilisent une liaison directe,
- La motorisation se fait par des roues,
- Les roues sont placées à l'arrière du robot,
- Les robots sont de formes rectangulaires ou carrées.

Il faut préciser que ces informations sont relatives au règlement de la compétition de 2004. Ces choix sont donc fortement liés au règlement. On peut noter que dans le passé, plus de robot avec transmission à renvoi d'angle existaient.

Deux éléments sont donc à fortement retenir, les chenilles ont quasiment disparu et les robots cylindriques sont en forte voie de disparition.

Nous tâcherons donc d'éviter ces deux aspects pour la conception de notre base roulante.

## 2. Le choix des moteurs :

### a. Technologies :

Le règlement de la compétition ne laissant pas de réels choix quant à la source d'énergie utilisée, nous nous sommes directement orientés vers des moteurs électriques. Il existe plusieurs familles de moteurs électriques :

- Les moteurs Brushless,
- Les moteurs à Courant Continu,
- Les moteurs à Courant Alternatif,
- Les moteurs Pas à Pas.

Forts de nos expériences passées, les moteurs Pas à Pas ont été directement mis de côté, à cause de leur faible vitesse de rotation et couple. En ce qui concerne les moteurs à Courant Alternatif, leur mise en œuvre est assez compliquée pour des systèmes embarqués. Nous avons donc le choix entre les moteurs Brushless et les moteurs à Courant Continu. Nous avons opté pour les moteurs à Courant Continu car ceux-ci sont plus simples à contrôler, moins coûteux, et mieux connus que les Brushless.

### b. Dimensionnement :

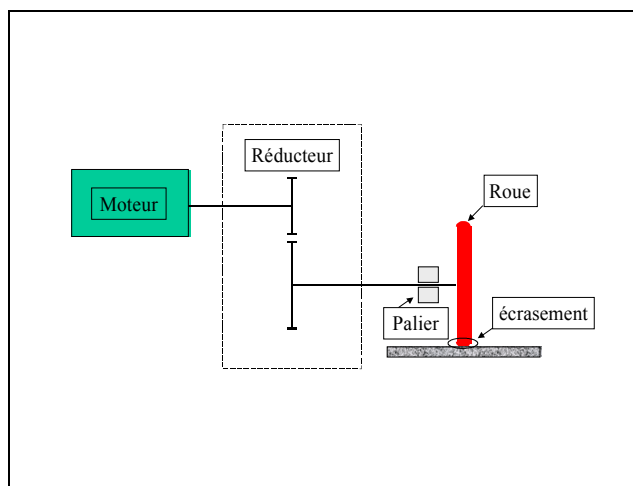
Le cahier des charges nous impose les caractéristiques suivantes :

- Diamètre des roues : 8cm,
- Poids du robot : 15 kg,
- Vitesse maximum d'entraînement :  $1\text{m}\cdot\text{s}^{-1}$ ,
- Accélération maximum :  $1\text{m}\cdot\text{s}^{-2}$ .

L'association avait à sa disposition 2 motoréducteurs de types AMAX 26G PLG 26 de chez MDPmotor, donc le premier objectif était de vérifier si l'on pouvait se servir de ces moteurs pour répondre aux spécifications que l'équipe s'était fixées.

Nous allons donc vous expliquer la démarche que nous avons suivie.

On peut représenter l'axe moteur comme ci dessous :



Tout d'abord il faut déterminer la force de résistance au roulement, celle-ci va dépendre essentiellement de deux paramètres :

- force de frottement essieu / palier donnée par la relation suivante :

$$F_f = \left[ \mu_f Mg \times \frac{d}{2} \right] \times \frac{2}{D_r}$$

$\mu_f$  : Coefficient de frottement essieu palier

$D_r$  : Diamètre de la roue

$\frac{d}{2}$  : Rayon de l'essieu

- force d'écrasement :

$$F_e = Mg \times f \times \frac{2}{D_r}$$

$M$  : Masse du robot

$f$  : Écrasement (0,0001 m)

Ce qui nous donne l'expression de la force de résistance au roulement, celle-ci devant être vaincue par le moteur pour mettre le robot en mouvement :

$$F_{res} = \frac{2}{D_r} Mg \left[ \mu_f \frac{d}{2} + f \right]$$

A partir de cette force on peut déterminer le couple résistant permanent à vaincre :

$$C_{rp} = \frac{D_r}{2} \times F_{res}$$

Pour pouvoir appliquer le principe fondamental de la mécanique, il nous faut déterminer l'inertie de la charge. Afin d'obtenir ce paramètre nous allons considérer qu'il y a conservation de l'énergie cinétique entre le mouvement de rotation (de la roue) et de la translation (du robot), ce qui nous donne :

$$\frac{1}{2} J_c \Omega_c^2 = \frac{1}{2} M V_c^2 \Rightarrow J_c = M \left( \frac{D_r}{2} \right)^2$$

Si l'on ramène cette inertie au moteur cela donne :

$$\frac{1}{2} J_c \Omega_c^2 = \frac{1}{2} J_{c,m} \Omega_m^2 \eta_r$$

$$J_{c,m} = J_c \times \frac{\Omega_c^2}{\Omega_m^2 \eta_r} \Leftrightarrow J_{c,m} = J_c \times \frac{1}{k \eta_r}$$

$k$  : Rapport de réduction

$\eta_r$  : Rendement du réducteur

$J_c$  : Inertie de la charge

$J_{c,m}$  : Inertie ramenée au moteur

Maintenant nous avons tous les éléments pour écrire le principe fondamental de la mécanique :

$$J_{tot,m}\gamma_m = C_m - C_{rp,m}$$

Ici compte tenu de notre application, pour déterminer le couple moteur nécessaire, il faut tenir compte du nombre de moteurs :

$$C_m = \frac{1}{nb_{moteur}} [(J_m + J_{c,m})\gamma_m + C_{rp,m}]$$

Dans l'application le nombre de moteurs s'élève à deux, donc si l'on regarde l'expression du couple moteur pour chaque phase cela donne :

- phase d'accélération :

$$C_{m,acc} = \frac{1}{2} \left[ \left( J_m + M \left( \frac{D_r}{2} \right)^2 \times \frac{1}{k^2 \eta_r} \right) \times \gamma_a + \frac{C_{rp}}{k \eta_r} \right]$$

$$P_{acc} = C_{m,acc} \times \Omega_{m,max} \times \frac{1}{\eta_m}$$

- phase de freinage :

$$C_{m,fr} = \frac{1}{2} \left[ \left( J_m + M \left( \frac{D_r}{2} \right)^2 \times \frac{1}{k^2 \eta_r} \right) \times \gamma_{fr} + \frac{C_{rp}}{k \eta_r} \right]$$

$$P_{fr} = C_{m,fr} \times \Omega_{m,max} \times \frac{1}{\eta_m}$$

- régime permanent :

$$C_{m,perm} = \frac{1}{2} \left[ \frac{C_{rp}}{k \eta_r} \right]$$

$$P_{perm} = C_{m,perm} \times \Omega_{m,max} \times \frac{1}{\eta_m}$$

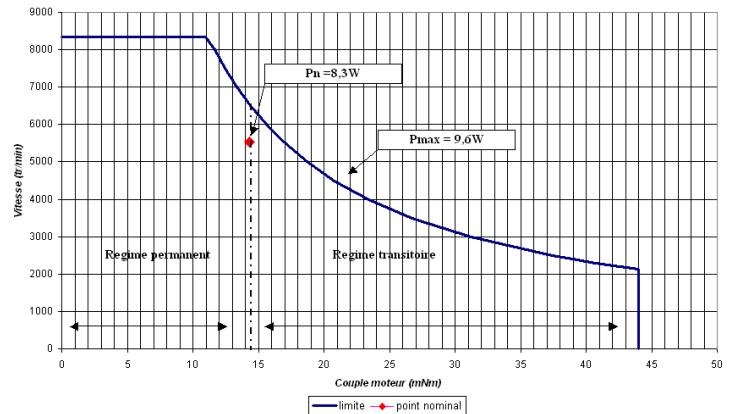
Maintenant que toutes les relations ont été mises en évidence on peut regarder les résultats que l'on obtient avec le moteur AMAX 26G, en gardant à l'esprit que pour l'application on considère que les phases d'accélération seront prépondérantes, c'est pourquoi on dimensionnera les moteurs pour les phases d'accélération :

coef de frot essieu-palier	0,1
ecrasement (m)	0,0001
masse (Kg)	15
diamètre roue (cm)	8
diametre essieu (cm)	0,4
accélération max (m/s <sup>2</sup> )	1
déccélération max (m/s <sup>2</sup> )	-1
vitesse max (m/s)	1
Nb de moteur	2
rapport de réduction	33
rendement moteur	0,75
rendement réducteur	0,96
inertie du moteur	0,00000126
accélération max moteur (rad/s <sup>2</sup> )	825
déccélération max moteur (rad/s <sup>2</sup> )	-825
inertie réducteur kgm <sup>2</sup>	0,0000016

Cr(Nm)	0,04
Fr (N)	1,10
aprox Jc,m	0,00002296
Cca (Nm)	0,600
Ccf (Nm)	-0,600
C <sub>m,acc</sub> (Nm)	0,011
C <sub>m,fr</sub> (Nm)	-0,009
C <sub>m,perm</sub> (Nm)	0,001
Vmot (tr/min)	7878

Pn (W)	0,77
Pacc (W)	11,76
Pfr (W)	-10,22

DOMAINE DE FONCTIONNEMENT DU MOTEUR AMAX26G



Apparemment le moteur aurait pu convenir si le réducteur avait eu un rapport de réduction plus faible. En effet le moteur peut délivrer le couple nominal nécessaire mais nous ne permet pas d'atteindre les vitesses souhaitées. Dans un premier temps nous avons revu nos spécifications à la baisse, envisagé la meilleure façon de commander le moteur de façon à ce que l'on démarre toujours en suivant la limite de puissance imposée par le collecteur mécanique. Ceci nous aurait permis d'avoir la meilleure dynamique du moteur. La conséquence de ce type de fonctionnement aurait été un échauffement excessif et un vieillissement prématuré du moteur. Donc les risques de casse auraient été multipliés.

Suite à un incident durant les phases de tests, un des moteurs c'est vu endommagé, c'est pourquoi il a fallu dimensionner d'autres moteurs.

Nous avons donc repris nos calculs avec un nouveau rapport de réduction, ce qui nous donne les résultats suivants :

coef de frot essieu-palier	0,1
ecrasement (m)	0,0001
masse (Kg)	15
diamètre roue (cm)	8
diametre essieu (cm)	0,4
accélération max (m/s <sup>2</sup> )	1
déccélération max (m/s <sup>2</sup> )	-1
vitesse max (m/s)	1
Nb de moteur	2
rapport de réduction	14
rendement moteur	0,75
rendement réducteur	0,7
inertie du moteur	0,0000005
accélération max moteur (rad/s <sup>2</sup> )	350
déccélération max moteur (rad/s <sup>2</sup> )	-350
inertie réducteur kgm <sup>2</sup>	0,0000016

Cr(Nm)	0,04
Fr (N)	1,10
aprox Jc,m	0,00017494
Cca (Nm)	0,600
Ccf (Nm)	-0,600
C <sub>m,acc</sub> (Nm)	0,034
C <sub>m,fr</sub> (Nm)	-0,029
C <sub>m,perm</sub> (Nm)	0,002
Vmot (tr/min)	3342

Pn (W)	1,05
Pacc (W)	15,75
Pfr (W)	-13,64

Nous avons donc recherché un moteur qui aurait pu avoir ses caractéristiques à savoir :

- Couple nominal : 34 mNm
- Vitesse nominale : 3342 tr/min
- Puissance nominale : 16 W
- Alimentation : 24V
- Réducteur : 1 : 14

Ce qui nous donnerait en sortie de réducteur :

- Couple nominal : 0,33 Nm
- Vitesse nominale : 239 tr/min

Nous avons eu beaucoup de mal à trouver un moteur qui réponde à ses exigences de manière précise et qui ait des caractéristiques géométriques compatibles avec les spécifications mécaniques. Ces spécifications mécaniques étant contraignantes compte tenu de la stratégie envisagée.

C'est pourquoi nous avons sélectionné un moteur un peu surdimensionné pour notre application, toutefois cela va dans le bon sens, comme nous sommes obligés de mettre un renvoi d'angle, ce qui va augmenter les pertes et l'inertie. Donc ce surdimensionnement devrait palier aux imprévus et combler nos négligences.

### *c. Choix possibles :*

Grâce aux informations qui précèdent, on a pu choisir 3 Références :

- Un motoréducteur de chez MDP-Motor, avec une vitesse en charge de 309 tr.min<sup>-1</sup> et un couple de 0.57 N.m, référence : AMAX32GHP/PLG42S, prix unitaire TTC : 310 € (avec -20% en tant que participant à la Coupe), disponibles sous 12 semaines, longueur sans arbre de 142 mm.
- Un motoréducteur de chez MDP-Motor, avec une vitesse en charge de 559 tr.min<sup>-1</sup> et un couple de 0.60 N.m, référence : RE035G/PLG52, prix unitaire TTC : 337 € (avec -20% en tant que participant à la Coupe), disponibles de suite, longueur sans arbre de 144 mm.
- Un motoréducteur de chez Faulhaber, avec une puissance de 45 W, références : moteur 2657, réducteur série 26, Rr =14), prix unitaire TTC : 507 €, disponibles de suite, longueur sans arbre : 125 mm.

Voir en annexes pour les caractéristiques de ces références.

Aux vues des longueurs obtenues, il nous est rapidement apparu qu'une transmission directe Motoréducteur / Roue n'était pas envisageable, du fait d'une longueur totale minimum de  $(125 + 30) * 2 = 310$  mm, avec 125 la longueur d'un motoréducteur, 30 la largeur estimée d'une roue. Nous avons donc opté pour une solution à renvoi d'angle.

*d. Choix définitif :*

Grâce aux caractéristiques précédentes, nous avons choisi les RE035G/PLG52, du fait de leurs disponibilité, caractéristiques, et prix. En ce qui concerne la transmission par renvoi d'angle, nous avons opté pour des systèmes déjà tout fait, qui permettent une plus grande précision, un meilleur rendement et un meilleur rapport qualité prix que les montages artisanaux. Voici la référence que nous avons retenue :

- Référence 748-437 chez Radiospares ou B332.31.2 chez HPCeurope, avec les caractéristiques suivantes :  $R_r = 1$ ,  $C_{\max} = 0.68 \text{ Nm}$ ,  $V_{\max} = 1500 \text{ tr.min}^{-1}$ ,  $\eta = 0.94$ , prix RS : 57.50 € pièce et prix HPC : 52 € pièce (avec -10% en tant que participant à la Coupe). (Voir les autres caractéristiques en annexes.)

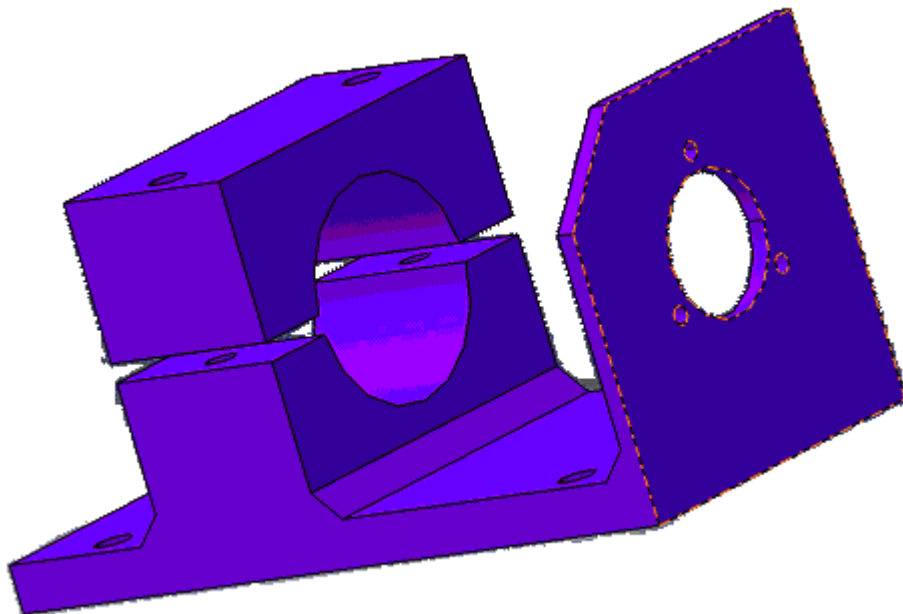
Pour des raisons de disponibilité, nous avons commandé ces 2 pièces chez Radiospares.

En ce qui concerne la liaison entre l'arbre moteur et l'arbre de la transmission, nous avons décidé d'usiner un accouplement rigide, dont les plans se trouvent en annexes. Du côté moteur, le blocage en rotation est assuré par une vis de pression qui se trouve dans la gorge prévue initialement pour la clavette. De l'autre côté, nous avons usiné un méplat sur l'arbre du réducteur, méplat sur lequel vient appuyer une autre vis de pression.

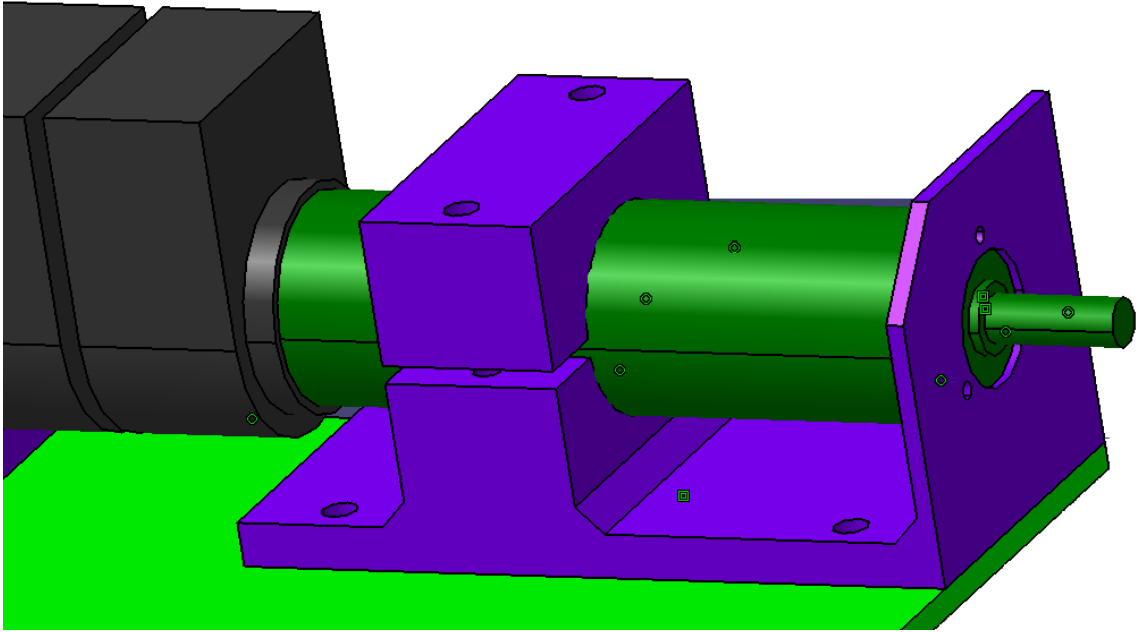
*3. Les supports des moteurs :*

A l'origine nous avons pensé mettre les moteurs et les roues coaxiaux, nous avons donc imaginé un système de bride qui permettrait de bloquer le moteur selon 2 axes de translation et 2 axes de rotation grâce à 2 vis de serrage, associée à une équerre qui permettrait le blocage selon le dernier axe de translation et de rotation. Le tout était fixé sur une plaque de fond, qui constituerait la base du robot.

Voici donc à quoi ressembler cette bride (pour un moteur) :



*Bride de maintien pour moteur de faible longueur.*

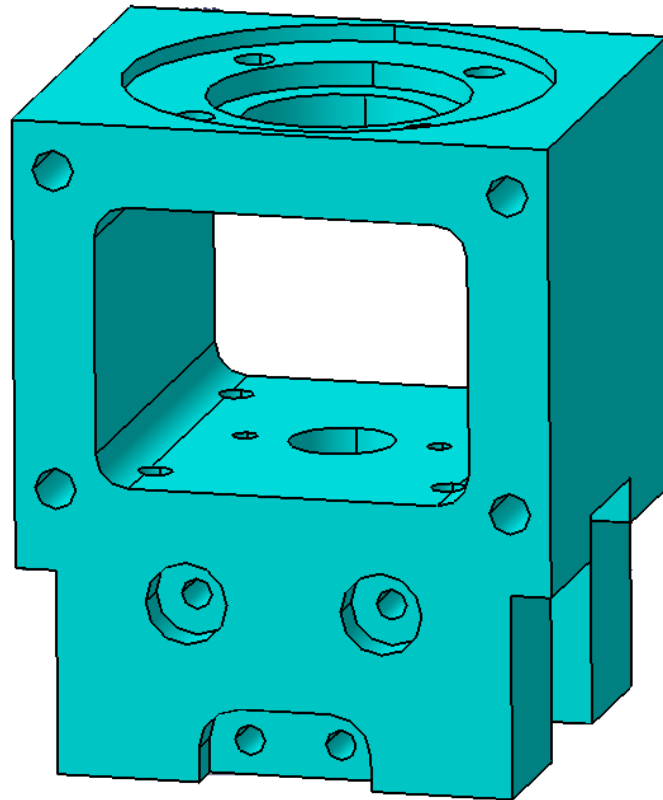


*Bride montée sur la plaque de fond avec un moteur.*

Ensuite, lorsque nous sommes passés à la solution avec renvoi d'angle, nous avons décidé d'usiner une pièce certes complexe, mais qui proposait une universalité, notamment au niveau des différentes positions des moteurs (verticale et horizontale) ; aspect qui satisfait pleinement au cahier des charges. Cette pièce, que par la suite nous appellerons Bloc Moteur, doit permettre la mise et le maintien en position du motoréducteur, la mise et le maintien en position du réducteur, la fixation de l'ensemble au sein du robot, notamment sur la plaque de fond.

Après une petite étude nous sommes arrivés à une pièce de ce type :



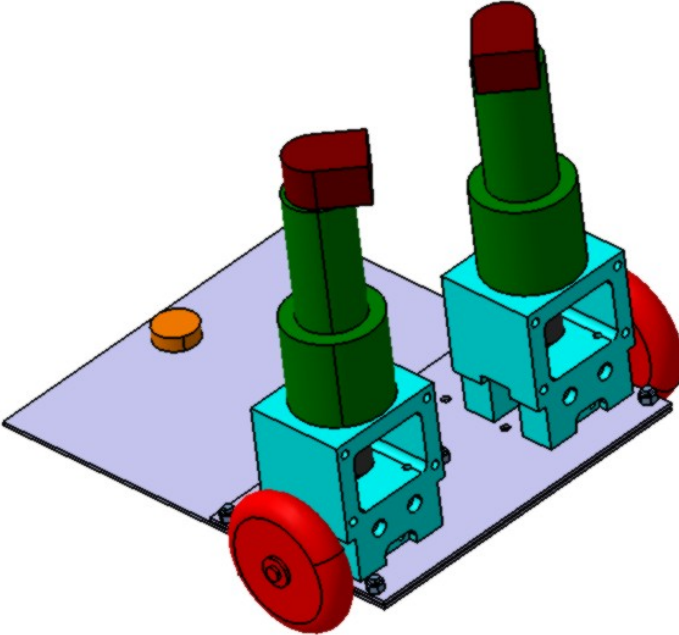


*Bloc moteur pour la solution avec renvoi d'angle.*

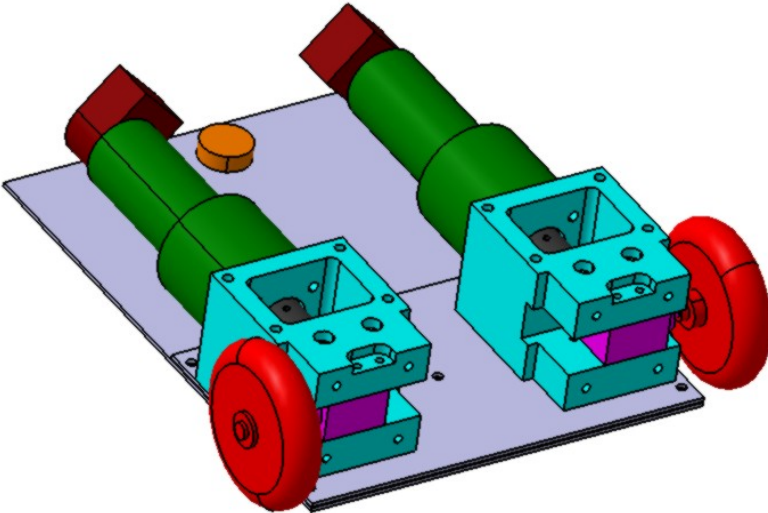
Cette nouvelle pièce a de nombreux avantages :

- Elle nous permet de pouvoir incorporer le renvoi d'angle
- De maintenir les pièces entre-elles et de limiter les efforts directs sur les différents éléments
- D'être installé à l'horizontale ou à la verticale

Effectivement, lors de la conception du bloc moteur, nous avons pensé aux années futures et nous avons donc prévu un montage des moteurs à la verticale ou à l'horizontale.



*En position verticale.*



*En position horizontale.*

#### 4. Les roues :

Comme vu lors de la veille technologique, les chenilles ont quasiment disparu de la compétition. Ceci est dû au fait que cette technologie est difficile à mettre en œuvre et que les règlements proposent des aires de jeux totalement planes.

Or, cette année, un fossé fait son apparition dans le règlement. Ce fossé de 36 mm de profondeur vient séparer deux aires de jeu. Une nécessité de traverser le fossé apparaît donc pour aller renverser les quilles de l'adversaire.

Deux solutions sont alors possibles :

- Traverser sur le pont aléatoire
- Traverser le fossé

Si nous choisissons de traverser par le pont, l'idée de garder notre système de roues est correcte, par contre, un système à chenille permettrait de passer par le fossé.

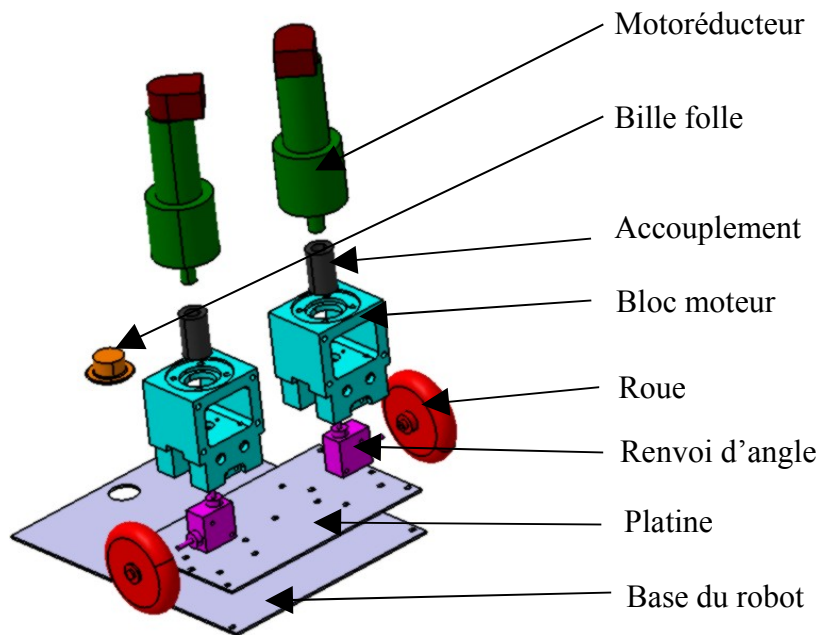
Après réflexion de l'équipe PR et des membres de la Coupe E=M6, la solution de traverser par le pont est retenue. Ce qui nous a conduit tout droit vers la solution à roues.

Les roues utilisées sont des roues de roller de diamètre 80mm et d'une largeur de 22mm. Ces roues sont reliées à l'arbre d'un renvoi d'angle par l'intermédiaire d'un accouplement en aluminium.

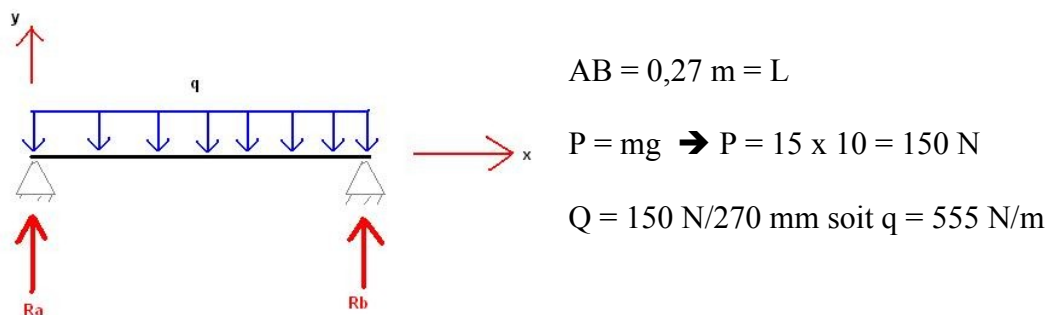


### 5. La base :

Pour la base du robot, il nous a parut indispensable de dimensionner la plaque de fond en épaisseur. De plus, afin de maintenir la solution de modularité de l'ensemble moteurs – roues, nous avons décidé de fixer les deux supports moteurs sur un platine qui sera elle-même fixée à la plaque de fond du robot.



Premièrement, nous avons effectué un dimensionnement succinct de la platine.



#### a. Réaction d'appui :

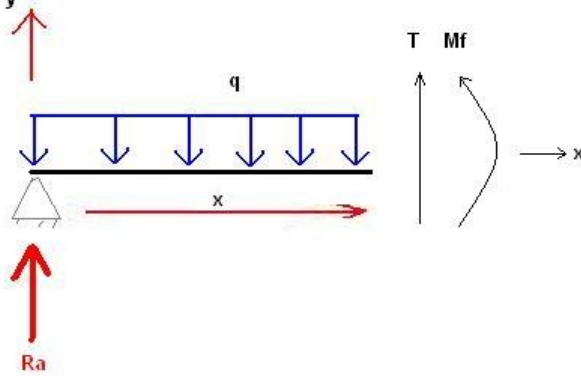
$$\sum F_x = 0 \rightarrow \text{OK !}$$

$$\sum F_y = 0 \rightarrow -qL + R_a + R_b = 0 \text{ or } R_a + R_b = \frac{1}{2}qL$$

$$R_a = \frac{1}{2}qL$$

b. Moment de flexion :

$$Mf = \sum M_{/c} = 0$$



$$Mf + \frac{qx^2}{2} - Rax = 0$$

$$Mf = \frac{1}{2}qLx - \frac{1}{2}qx^2$$

c. Flèche :

$$EIY'' = Mf(x)$$

$$EIY'' = \frac{1}{2}qLx - \frac{1}{2}qx^2$$

$$EIY' = \frac{1}{4}qLx^2 - \frac{1}{6}qx^3 + C1$$

$$EIY = \frac{1}{12}qLx^3 - \frac{1}{24}qx^4 + C1x + C2$$

d. Conditions aux limites :

$$Y(x=0) = 0 \rightarrow C2 = 0$$

$$Y(x=L) = 0 \rightarrow \frac{qL^4}{12} - \frac{qL^4}{24} + C1L = 0$$

$$LC1 = \frac{qL^4}{24}$$

$$C1 = \frac{qL^3}{24}$$

On obtient donc :

$$EIY = \frac{1}{12}qLx^3 - \frac{1}{24}qx^4 + \frac{1}{24}qL^3x$$

De même :

$$Y = \frac{q}{EI} \left( \frac{Lx^3}{12} - \frac{x^4}{24} + \frac{L^3x}{24} \right)$$

On sait que :

- $E = 210\,000 \cdot 10^6$
- $Y = 1 \cdot 10^{-3} \text{ m}$
- $Q = 555 \text{ N/m}$
- $L = 0,27 \text{ m}$
- $I = \frac{bh^3}{12} \rightarrow b = 0,06 \text{ m}$

$$I = \frac{q}{EY} \left( \frac{L \left( \frac{L^3}{8} \right)}{12} - \frac{\left( \frac{L^4}{16} \right)}{24} + \frac{L^3 \left( \frac{L}{2} \right)}{24} \right)$$

$$h = \sqrt[3]{\frac{12 \times 555 \times 1,52 \cdot 10^{-4}}{0,06 \times 210000 \cdot 10^6 \times 0,001}}$$

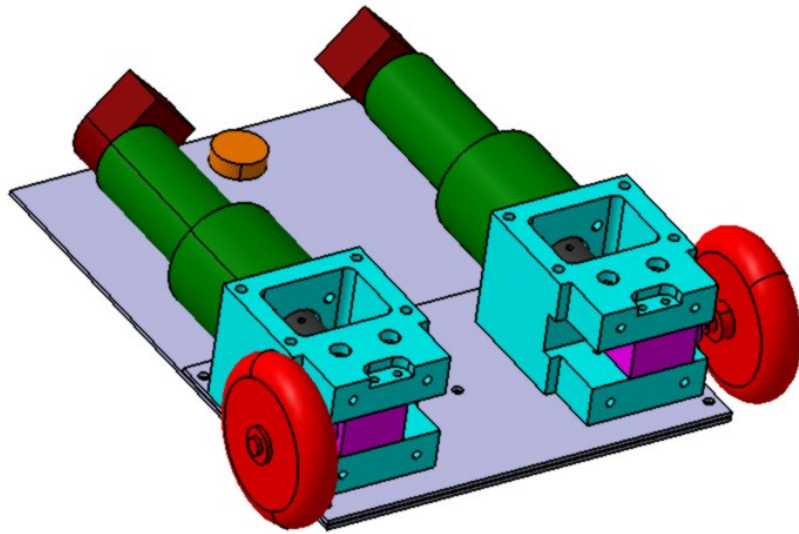
Et enfin on trouve h l'épaisseur de la platine :  $h = 4,3 \text{ mm}$

Après réflexion, nous nous sommes rendu compte que ce calcul n'était pas rigoureux car nous avons considéré la théorie des poutres, or dans notre cas, nous ne pouvons satisfaire cette théorie. (Longueur de la section = 10 x la largeur de la section)

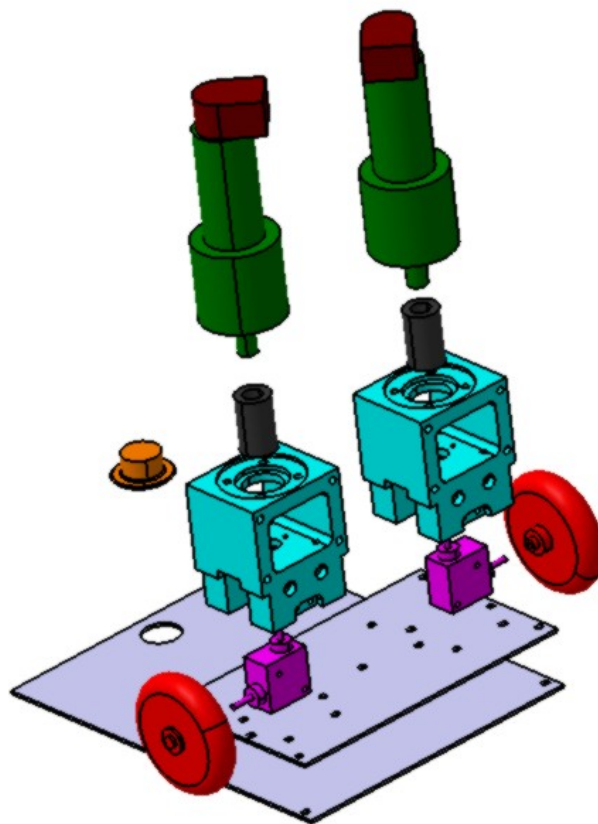
Il nous faut donc utiliser un modèle plus précis, c'est pourquoi nous avons décidé de faire un calcul par éléments finis sous *Idéas*. Ainsi, nous trouvons une épaisseur  $h = 3,2 \text{ mm}$ .

Il nous faut donc une platine de plus de 3 mm. Etant donné que la platine sera solidairement fixée à la base du robot, nous avons décidé de prendre une épaisseur de platine de 2 mm et une épaisseur de base de robot de 2 mm. Nous obtenons donc une épaisseur totale de 4 mm. Ainsi nous devrions avoir, sous une charge de 15 kg, moins d'un millimètre de flexion de la base du robot.

6. Résultats :

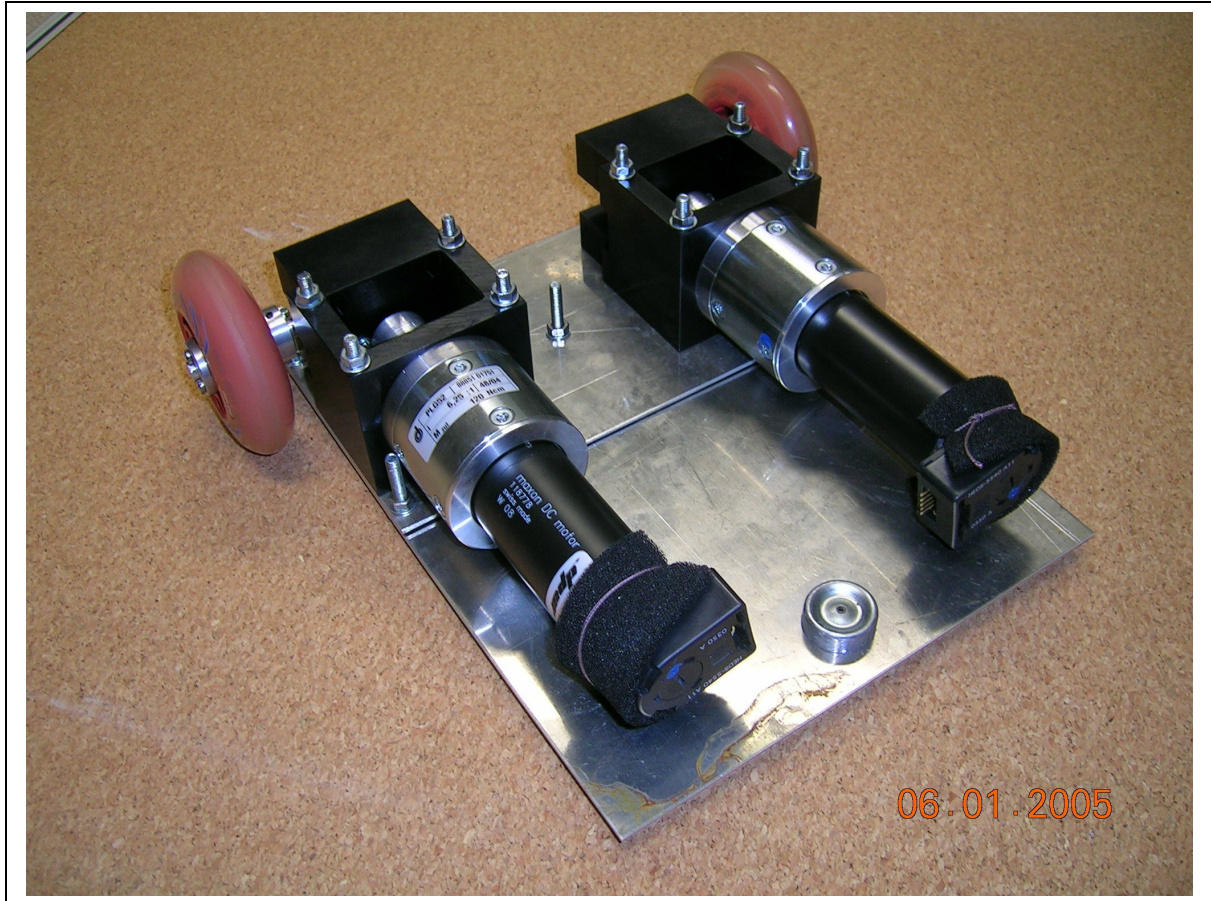


*Assemblage en position horizontale*



*Vue éclatée en position verticale*





*Photo de l'assemblage roues – blocs moteurs – moteurs – platine.*



# Partie C : L'électronique

## 20. *Les batteries*

Afin de pouvoir choisir les batteries appropriées à notre application, nous devons dans un premier temps faire une estimation de la consommation d'énergie totale avec les pointes de courant envisagées dans notre robot.

Il est à noter que nous ne connaissons pas encore en détail tous les éléments annexes qui seront implantés dans le robot comme par exemple les capteurs et actionneurs diverses qui seront choisis en fonction des objectifs technologiques atteints ou non, de l'association et de la stratégie finale.

Dans le but de pouvoir utiliser à nouveau dans l'avenir notre base roulante, nous avons choisi de surestimer les besoins en énergie de sorte à ne pas être pris au dépourvu.

La consommation pourrait se décomposer ainsi :

- 2 moteurs de 24V sous 2.1A, soit 100.8 Watts pendant 100 % du temps
- 3 actionneurs de 21 Watts chacun soit 63 Watts pendant 50 % du temps
- 3A pour l'électronique sous 5V soit 15 Watts pendant 100 % du temps  
(Asservissement, carte de commandes, capteurs)

Ce qui nous fait, au total, une consommation maximale ~180 Watts.

La consommation permanente devrait être ~147 Watts permanents, il faut donc que les batteries puissent débiter un courant de ~6 A sous 24 V durant le fonctionnement du robot.

Après une longue recherche, on a pu réaliser le tableau comparatif des batteries suivant :

type	Densité d'énergie (Wh/Kg)	Densité d'énergie (Wh/L)	auto-décharge (%:mois)	voltage d'une cellule (V)	temps de charge rapide
plomb	35	111	5 à 10	2	8h à 15h
NiCd (Nickel Cadmium)	50	94	10 à 20	1,2	1h30min
NiMH (Nickel Metal Hydride)	80	135	20 à 30	1,3	2h à 4h
Lithium Ion	100	196	5 à 10	3,6	2h à 4h
Lithium polymer	150	200	5 à 10	3,6	2h à 4h
Alkaline rechargeable	80			1,5	2h à 3h

type	cycle de charge-décharge avant que capacité passe de 100% à 80%	Courant max.	Courant max. pic	effet mémoire	estimation (€/Wh)	prix/cycles
plomb	200 à 300	0,2C	0,5C	M	0,895	0,003
NiCd (Nickel Cadmium)	1500	1C	20C	H	4,16	0,003
NiMH (Nickel Metal Hydride)	300 à 500	0,5C	5C	L	2	0,004
Lithium Ion	500 à 1000	1C	2C	L	8	0,008
Lithium polymer	300 à 500	1C	2C à 5C	L	3,2	0,006
Alkaline rechargeable	50	0,2C	0,5C	H	1,78	0,036

Effet mémoire : L (Low), M (Medium), H (High)

Courant max. : C signifie capacité, par exemple une batterie qui peut délivrer un courant max. de 1C de 4A/h pourra donc délivrer 4A au maximum sans risque de détérioration.

Nous nous sommes intéressés au courant maximum et au temps de charge rapide pour notre choix de batteries.

Le type de batterie qui semble le plus approprié pour nous est le type Nickel-Cadmium, parfaitement adapté au pic d'appel de courant que provoquent des moteurs, pouvant supporter des charges rapides en 1h30min seulement afin de devoir prévoir trop de jeux de batteries différents.

Les batteries LI-Ion ou LI-Polymer présentent une densité énergétique très intéressante, mais leur prix est beaucoup plus élevé à l'achat, et de plus dans notre application roulant sur un sol peu accidenté, la masse pouvant être transportée est suffisante pour le type NiCd.

Les batteries LI-Ion ou LI-Polymer sont plus adaptées à l'aéromodélisme ou au petit appareil électronique.

Quant aux batteries au plomb qui sont vraiment bon marché, le courant maximum supporté est trop faible est supposerait d'avoir une forte capacité.

Nous voulons avoir une autonomie de 30 minutes, il nous faut donc une énergie de 73.5 Wh.

Or sous 24 V, les batteries doivent avoir une capacité de **~3 Ah**.

Comme les batteries NiCd peuvent débiter un courant maximum de 1C (une fois leur capacité), nous devons donc prendre des batteries de **6Ah** sous 24 V ce qui nous donnera une autonomie de **1 heure**.

Il faudra donc, soit acheter un bloc accumulateur NiCd de 24 V 6 Ah s'il existe dans le commerce, soit le réaliser nous même en assemblant différents éléments.

Rappelons que l'association en série permet d'additionner les tensions des éléments associés et que l'association en parallèle permet d'additionner les courants débités.

Par exemple, des éléments de 1.2 V chacun pour 3 Ah s'associeraient ainsi :

1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah
1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah	1.2 V 3Ah

Soit deux séries de 20 éléments équivalentes à 24 V et faisant une capacité de 3 Ah chacune.

Il est possible de répartir les accumulateurs en différents paquets pour faciliter le rangement dans le robot ou répartir la masse, ou éventuellement trouver des packs dans le commerce.

Dans le cas de l'UV PR, le choix des batteries reste pour le moment sur les NiCd, mais pour les essais pratiques, on utilisera des batteries au plomb que l'on possède déjà, d'autant plus que les essais sont réalisés avec les moteurs de propulsion et l'asservissement seul, ce qui nécessite un débit de courant de  $\frac{105}{24} = 4.375$  A sur 2 batteries, soit  $\approx 2.187$  A.

Comme les batteries au plomb peuvent débiter 0,2.C, il nous en faudrait des de  $\approx 10.9$  A.h

Or les essais de la base roulante ne seront que de courtes durées, et l'on se contentera donc d'utiliser des batteries de 6 A.h pour des raisons de volume même si elles ne sont pas vraiment bien dimensionnées.

## 21. La commande des moteurs :

Dans un premier temps, nous verrons un bref récapitulatif des cartes et modules d'asservissement du commerce.

Pour la commande des moteurs, plusieurs solutions ont été envisagées :

- Un asservissement 'tout fait' analogique : modules « Escap »
- Une solution 'artisanale' : carte avec asservissement numérique
- Un asservissement 'tout fait' numérique : circuits « LM629 »

### a. Comparatifs des modules d'asservissement et calcul des correcteurs

Veille technologique sur les systèmes d'asservissement utilisés par les différentes écoles de la coupe E= M6.

Le but de cette partie est d'étudier les principales techniques d'asservissement utilisées par les écoles de la Coupe E=M6. Ce document est loin d'être exhaustif et servira de base à compléter et actualiser pour les semestres et équipes prochaines.

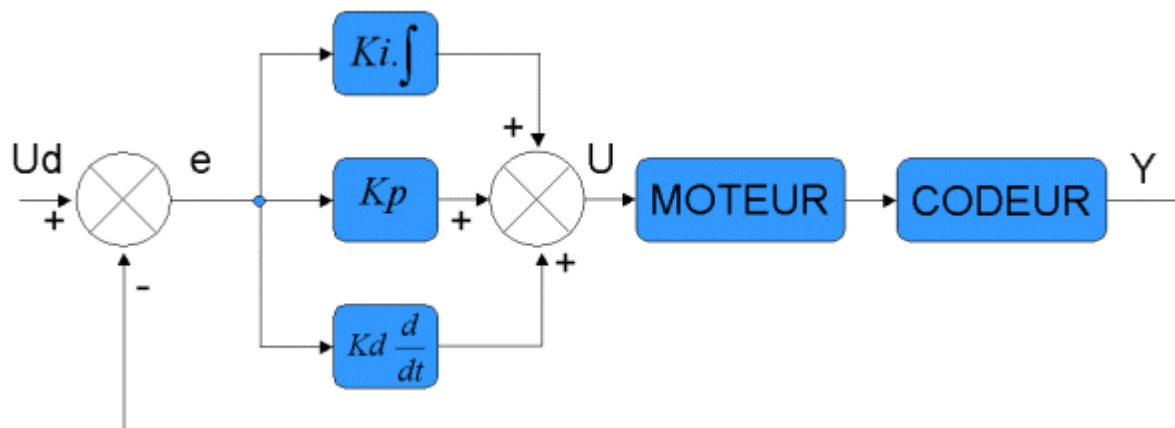
Après une étude sur plus d'une trentaine d'écoles, il ressort trois formes d'asservissement, la plus utilisée, à 85%, est du type PID en majorité réduit au PI, ensuite viennent les microcontrôleurs PIC associés au PID et les modules AVR, nous allons par la suite détailler ces trois solutions. Il existe également des systèmes de contrôle dits à retour d'état jamais utilisés jusqu'à présent pour la Coupe, c'est pourquoi nous ne développerons pas son approche.

### 3. Système PID :

<http://www.ancr.org/fichtech/action/pid/index.htm>

#### a. Principe

Pour pouvoir tester un asservissement, il faut d'abord réaliser une plate-forme de développement, le plus généralement du type suivant. . Mais pour commencer un brin de théorie illustré par des exemples. Le schéma de principe est représenté ci-dessous :

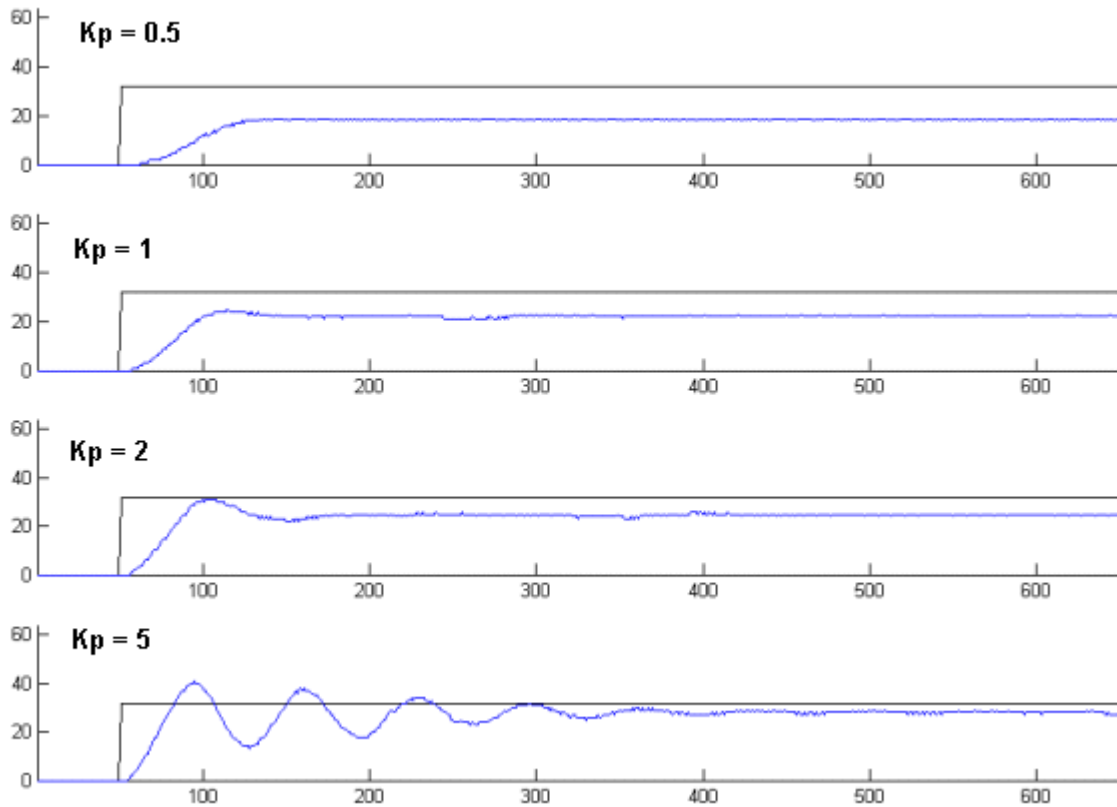


- $U_d$  : vitesse désirée
- $e$  : erreur entre la vitesse désirée et la vitesse réelle
- $U$  : Consigne appliquée au moteur
- $Y$  : Vitesse réelle

Nous avons bien un schéma en boucle fermée puisque la sortie du système est réinjectée dans l'entrée. Les trois blocs bleus correspondent chacun à une fonctionnalité :  $K_i$  est le coefficient intégral,  $K_p$  le coefficient proportionnel et  $K_d$  le coefficient dérivé. Chacun de ces coefficients a une fonctionnalité bien précise. Commençons par le plus simple : le proportionnel.

### b. Asservissement P

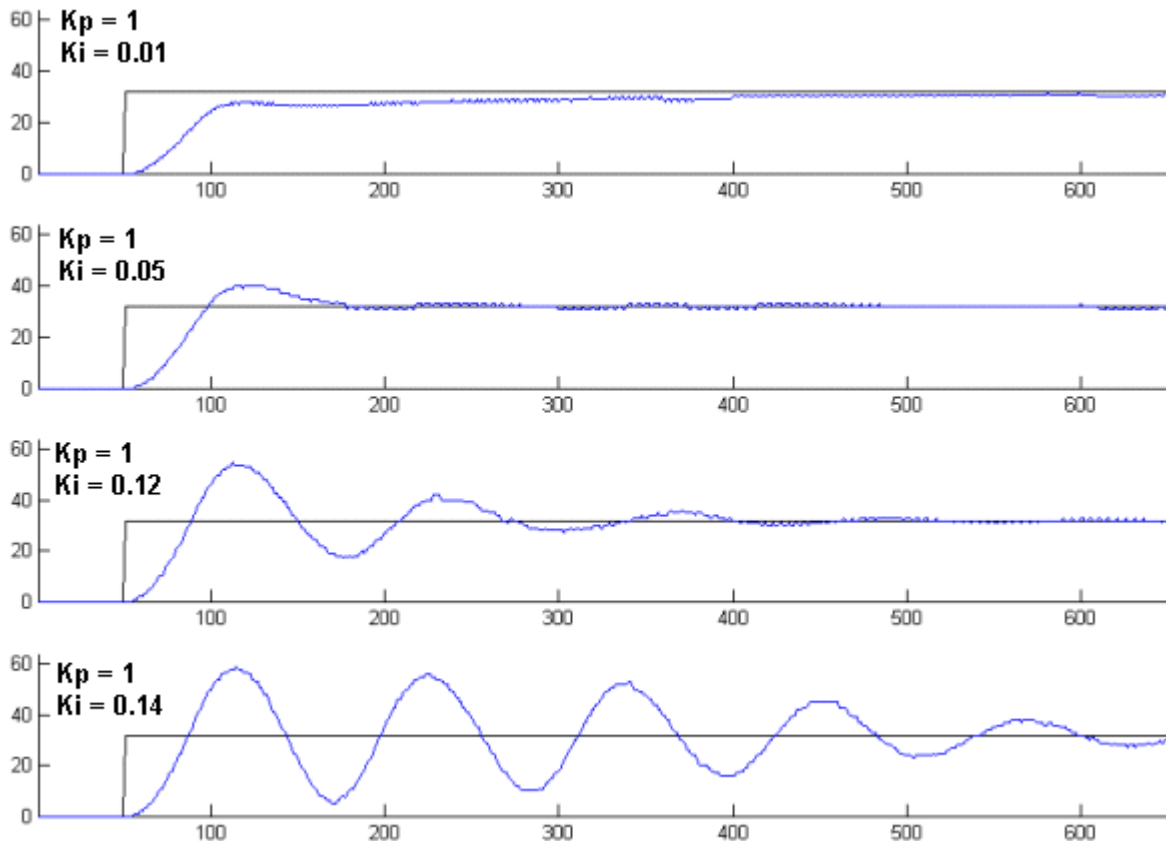
L'asservissement proportionnel est essentiel au fonctionnement du PID. Il permet essentiellement de donner de la puissance au moteur. Voici quelques résultats d'expérimentations avec  $K_i=0$  et  $K_d=0$ .



La courbe noire représente la consigne que l'on désire atteindre ( $U_d$ ) c'est un échelon de valeur 32. La courbe bleue représente la vitesse réelle du système ( $Y$ ). Comme vous pouvez le constater, plus  $K_p$  est grand, plus le système converge vite vers sa valeur finale. Mais en contrepartie, pour des valeurs de  $K_p$  trop grandes, le système oscille. Mais là n'est pas notre plus gros problème, en effet sur ces courbes on voit nettement que la vitesse du moteur n'atteint jamais la vitesse désirée. C'est ce que l'on appelle l'erreur statique, elle correspond à la différence entre la vitesse réelle et la vitesse désirée en régime établie (une fois que le système s'est stabilisé). Pour compenser cette erreur statique, on rajoute le terme intégral.

c. Asservissement PI

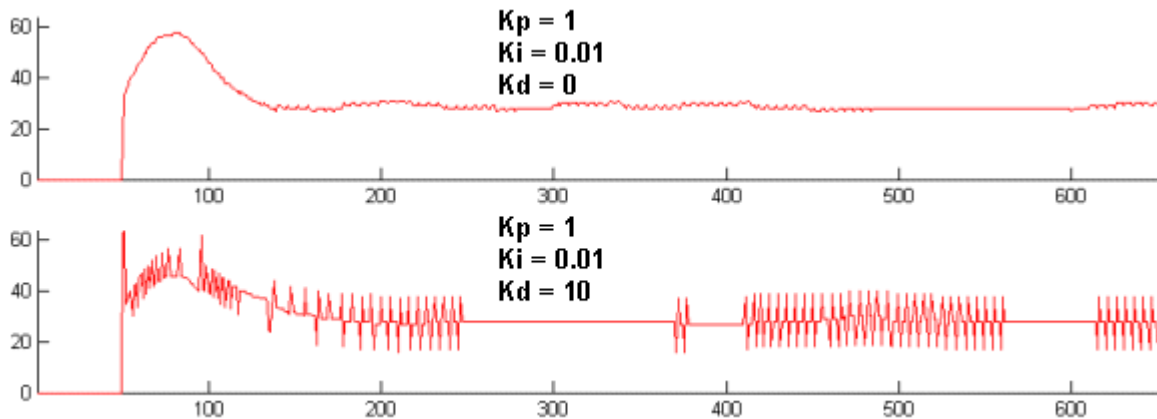
Le correcteur intégral sert principalement à supprimer l'erreur statique. L'idée principale est de "charger" ou intégrer l'erreur depuis le début et d'ajouter cette erreur à la consigne jusqu'à ce qu'elle devienne nulle. Lorsque cette erreur est nulle, le terme intégral se stabilise et il compense parfaitement l'erreur entre la consigne et la vitesse réelle. Démonstration en image :



On voit cette fois-ci que le terme intégral a bien fonctionné et que l'erreur statique est nulle. On constate aussi que plus le gain  $K_i$  est grand, plus le système converge vite. En revanche, plus  $K_i$  est grand, plus le système oscille et plus le dépassement est grand. Sur des asservissements en position le terme dérivé permet de diminuer, le dépassement et les oscillations. Nous allons voir que dans notre cas, nous avons quelques problèmes.

d. Asservissement PID ou à avance de phase

On a donc rajouté le terme dérivé, comme tout bon livre d'automatique le préconise, et voici les résultats :



Cette fois ci, j'ai choisi de représenter la tension appliquée au moteur ( $U$ ), le problème est bien plus flagrant ici. Vous voyez que lorsque l'on ajoute le terme dérivé, la commande est extrêmement bruitée. En réalité, on somme à la commande la dérivée de la vitesse réelle, que l'on multiplie par un gain  $K_d$ . La dérivée d'une vitesse est une accélération, cela signifie que l'on amplifie tous les bruits d'accélération et en plus, on les multiplie par un gain  $K_d$ . Donc rajoute du bruit amplifié dans la commande. Alors pourquoi mettent t-ils ce terme "dérivée" dans les livres ? Simplement parce que ce bruit est typique des asservissement en vitesse. Sur des asservissements en position, ce problème est moins fréquent, car la dérivée de la position est la vitesse.

Ici, le problème était visible, parce que nous avons réalisé toutes ces courbes sur un système réel. Si nous avions simulé ces résultats, l'accélération n'aurait pas été bruitée et les courbes auraient été parfaites. Pour implémenter ce PID sur notre robot, il a fallu programmer en numérique une dérivée et une intégrale.



### e. PID numérique et réglage

Notre premier problème est de programmer le terme intégral. Il existe une méthode qui consiste à estimer la surface entre deux échantillons, mais dans notre cas, nous avons des problèmes dus aux bruits et à la fréquence d'échantillonnage. Alors, il faut sommer toutes les erreurs depuis le début.

Notre second problème est de programmer le terme dérivé. La méthode ici est de calculer la pente entre deux échantillons de  $e$ . Théoriquement cela revient à faire le calcul suivant :

$$\frac{de}{dt} = \frac{e(t) - e(t-1)}{\Delta t}$$

Avec  $\Delta t$  la période d'échantillonnage. Pour faire un asservissement numérique, il faut impérativement avoir une période fixe. Pour gagner du temps de calcul, généralement on intègre directement  $\Delta t$  dans le gain  $K_d$ .

Voici un l'algorithme exemple :

```
Ti = 0; /* Initialise le terme intégral */
for ( i=1 ; i <= Nb_cycles ; i++ )
{
    Init_timers (1); /* Initialise le timer à notre période : 1 ms */
    Y = Codeur (); /* Stoque la valeur du codeur dans Y */
    Rst_codeur (); /* Remet le codeur à zéro */
    Ud = Consigne (); /* Mise à jour de la consigne */
    e_old=e; /* Stoque e(t-1) */
    e = Ud - Y; /* Calcul de l'erreur */
    Ti = Ti + e; /* Calcul de l'intégral */

    /* Calcul du PID */

    U = Kp * e /* Terme proportionnel */
    + Ki * Ti /* Terme intégral */
    + Kd * ( e - e_old ); /* Terme dérivé */

    Moteur (U); /* On applique la consigne */
    while ( ! end_timer() ); /* On attend la fin de la période */
}
}
```

Pour le réglage des gains  $K_p$ ,  $K_i$  et  $K_d$ , il existe différentes méthodes. Ici nous l'avons fait de manière empirique et le résultats est plutôt bon. L'inconvénient de ces méthodes est qu'elles demandent un modèle du système, ce qui est généralement dur à produire. Il existe

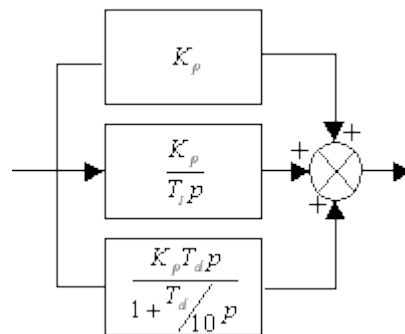
aussi des composants programmables qui permettent de faire directement l'asservissement numérique (HCTL 1100, LM629). Ils sont exactement basés sur la méthode présentée ici.

Pour ce qui est du réglage des correcteurs, l'optimal est d'utiliser la méthode de l'optimum dissymétrique une fois que l'on a réalisé le modèle du système. La plus part des écoles ne réalisent pas ce modèle et ce limite donc à une correction empirique en réalisant des tests ou en utilisant la méthode Ziegler-Nichols à partir de la réponse à un échelon de tension.

Par la suite nous rappellerons les méthodes de Ziegler-Nichols et de l'optimum dissymétrique :

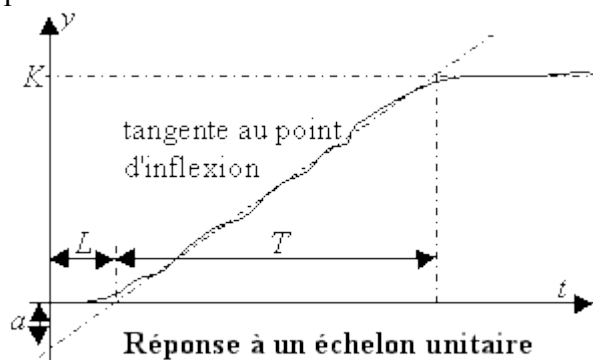
### f. Les méthodes de Ziegler-Nichols

Ces méthodes fournissent des réglages pour des correcteurs de type PID (pouvant ne comporter qu'une action Proportionnelle, une action Proportionnelle et une action Intégrale ou les 3 actions simultanément), dont la structure est la suivante :



#### • MÉTHODE DE ZIEGLER NICHOLS TEMPORELLE

Elle est adaptée aux systèmes stables en boucle ouverte, qui ont une réponse apériodique. La réponse à un échelon unitaire du système physique doit être enregistrée. Différentes caractéristiques de la réponse obtenue sont mesurées :



Le modèle approché utilisée par Ziegler Nichols comporte une constante de temps et un retard pur ou bien un retard pur et un intégrateur :

$$H(p) = \frac{Ke^{-Lp}}{1+Tp} \quad \text{ou} \quad H(p) = \frac{ae^{-Lp}}{Lp}$$

Si le système physique comporte un intégrateur, les mesures de  $a$  et de  $L$  demeurent disponibles sur le début de la réponse.

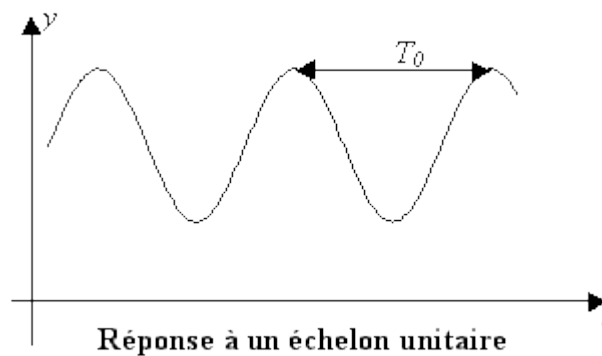
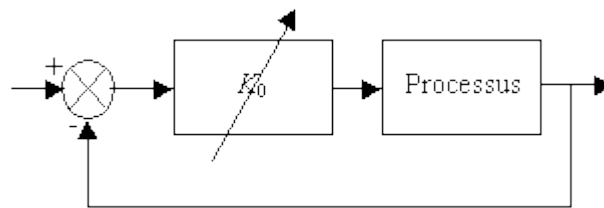
Les réglages suggérés par la méthode temporelle de Ziegler Nichols sont récapitulés dans le tableau suivant :

	$K_p$	$T_i$	$T_d$
correcteur P	$1/a$	-	-
correcteur PI	$0,9/a$	$3L$	-
correcteur PID	$1,2/a$	$2L$	$L/2$

Concernant le raffinement des réglages, augmenter  $K_p$  ou diminuer  $T_i$  augmentent les oscillations de la sortie.

- **MÉTHODE DE ZIEGLER NICHOLS FRÉQUENTIELLE**

Cette méthode convient particulièrement bien aux systèmes instables en boucle ouverte pour lesquels la méthode temporelle n'est pas envisageable. L'expérimentation se fait sur le système bouclé avec un simple gain proportionnel qui doit être stable. Le principe est de modifier la valeur du gain par essai-erreur jusqu'à atteindre la limite de stabilité (phénomène de pompage). Il reste à relever le gain limite  $K_0$  obtenu et à mesurer la période  $T_0$  des oscillations de la réponse obtenue.



Les réglages suggérés par la méthode fréquentielle de Ziegler Nichols sont récapitulés dans le tableau suivant :

	$K_p$	$T_i$	$T_d$
correcteur P	$0,5 K_0$	-	-
correcteur PI	$0,45 K_0$	$0,8 T_0$	-
correcteur PID	$0,6 K_0$	$0,5 T_0$	$0,125 T_0$

A la page Internet suivante vous trouverez un simulateur de cette méthode :

<http://www-hadoc.ensieg.inpg.fr/hadoc/ateliers/reglageEmpCorr/ReglEmpCorrDroit.html>

g. Méthode de l'optimum dyssimétrique (voir poly MC06 Chap 5 et 8)

Cette méthode s'applique une fois le modèle réalisé et mis sous la forme d'un second ordre :

$$\frac{K_Q}{s \left( 1 + \frac{s}{\omega_Q} \right)}$$

et on obtient le correcteur PI :  $K_P \cdot \left( 1 + \frac{\omega_N}{s} \right)$

pour une surtension désirée  $Q_{dB}$  :

$$Q = 10^{(Q_{dB}/20)}$$

$$\lambda = \sqrt{(Q^2/(Q^2-1))}$$

$$\Delta\varphi = \arccos(1/\lambda)$$

$$\alpha = ((1+\sin(\Delta\varphi))/\cos(\Delta\varphi))^2$$

$$\omega_N = \omega_Q/\alpha$$

$$\omega_0 = \omega_Q/\sqrt{\alpha}$$

$$K_P = \omega_0 * \lambda / K_Q$$

#### 4. Les microcontrôleurs :

##### a. Avant propos :

Cette fiche se veut être sous la forme d'un projet. Un projet non terminé, où chacun peut apporter sa contribution (améliorations, adaptations, corrections, etc...). De ce fait, ces pages sont amenées à évoluer, et vous-même pourrez participer à leur évolution.

Pourquoi avoir choisit un tel mode de travail ? Tout simplement parce que le sujet est vaste, et que des variations sont possibles suivant l'utilisation qu'en feront les différentes personnes.

Le but de ce projet est de réaliser un double asservissement de vitesse (type PID) de deux moteurs sur un seul PIC (type 16F877). L'utilité majeure étant de faire rouler un robot bien droit et à la bonne vitesse : A quoi ça sert plus précisément ?

Avec un seul PIC ce projet permet aux roboticiens amateurs et chevronnés de réaliser facilement leur asservissement de vitesse de moteurs à courant continu (en général les moteurs de propulsions de leur robot, mais ça peut être bien sûr d'autres moteurs). La commande de l'asservissement pourra se faire, à terme, en liaison série (déjà fait) , I2C ou par bus ISA.

##### b. Avantages

Il existe bien sûr d'autres systèmes qui gèrent un asservissement de vitesses. Des systèmes professionnels existent comme les [LM629](#). Ils ont l'inconvénient d'être cher (et ce pour un seul moteur) et relativement complexes.

Pourquoi chercher un fusil pour chasser des moustiques ? Le double asservissement sur PIC vous coûte ... le prix d'un PIC... soit environ 15Euro pour un 16F877 actuellement, et ce pour 2 moteurs. Ce double asservissement se veut relativement simple .De plus, vous pouvez en modifier le programme et l'adapter à vos besoins.

##### c. Inconvénients

Les personnes qui auront les moyens préféreront opter pour un système professionnel, certainement pour des raisons de fiabilité, mais c'est à leurs risques et périls ;-)  
La puissance du PIC étant limitée, il ne peut pas réaliser des PID trop rapides ou trop complexes. Il y a en particulier une limitation sur le nombre d'impulsions par seconde qu'il est capable de gérer.

Le langage utilisé étant le C, il faut une licence du logiciel pour pouvoir modifier et recompiler le code source c. Mais le projet se veut générique : Une version donnée pourra être utilisée sans recompilation en respectant quelques contraintes.

d. Asservissement :

En fait on pourrait imaginer qu'il suffirait d'envoyer au moteur une puissance constante pour qu'il tourne tout le temps à la même vitesse. Ce n'est pas vrai du tout. Tout d'abord 2 moteurs d'une même série n'auront pas forcément le même comportement. Ensuite parce que les niveaux de tension ou d'autres différences électriques peuvent jouer. Enfin parce que des influences extérieures (obstacles, frein, mécanique du robot) peuvent jouer.

Du coup il est très difficile de faire aller un moteur à la vitesse voulue. Parfois ce n'est pas gênant, mais ça peut l'être. Un exemple classique est le robot avec 2 roues motrices, type char, une configuration qui est très courante. Mais pour que ce robot aille droit, il faut que les 2 moteurs aillent exactement à la même vitesse !

La solution est de faire un asservissement. C'est à dire moduler dynamiquement la puissance envoyée aux moteurs suivant leur vitesse mesurée, afin d'adapter celle-ci à ce qui est désirée.

Cela implique un élément très important : pour asservir il faut un capteur de vitesse du moteur. En général il s'agit d'un compte-tour, qui enverra N impulsions (ou "pas") par tour de moteur. Ensuite un programme va calculer, en fonction de la vitesse mesurée du moteur, la puissance à utiliser.

Les asservissements sont constamment en train de chercher l'équilibre. Il faut bien comprendre qu'ils mettent un certain temps à l'atteindre, et ce dernier peut aussi être instable si l'asservissement est mal réglé...

L'asservissement qu'on va utiliser ici est un classique PID ou plus précisément un PI, car pour des raisons de vitesses de traitement la composante dérivée est laissée de côté (mais rien de difficile à rajouter).

Le PID ou Proportionnel Intégrale Dérivé est une méthode de calcul d'asservissement.

Soit  $K_p$ ,  $K_i$  et  $K_d$  les coefficients proportionnels, intégrale et dérivé.  
Soit  $e$  l'erreur : Vitesse mesurée moins vitesse demandée.  
Soit  $T_i$  le terme intégral, d'abord initialisé à 0.

Consigne à appliquer (PWM) =  $K_p * e + (1/K_i) * (T_i + e) + K_d * (e - e(t-1))$

On pourrait dire des pages et des pages sur le PID, et ce n'est pas mon but ici. Merci de vous reporter au lien ci-dessous :

Pour plus de détails sur le PID, la page de l'ANCR :

<http://www.ancr.org/fichtech/action/pid/index.htm>

Chez les Fribottes, la Simulation de Jean-Claude :

<http://www.wiki.free.fr/WiKiFri/index.php?Simulateur%20de%20PID>

e. Le pic :

Le PIC est un petit microcontrôleur, assez bon marché, et simple à programmer comme à utiliser.

La aussi je vais pas rentrer dans les détails, vous pouvez vous reporter à la bdtech du site Fribotte, et en particulier aux cours de Bigonoff.

Le PIC est ici programmé en C, voir la page Fribotte [ici](#). L'utilisation du langage C permet d'alléger considérablement le développement d'une application compliquée.

2. Module AVR :

<http://www.enseirb.fr/~matz/fr/avr.php>

Très peu de renseignement sur ce module sinon ces informations d'un étudiant de l'Enseirb :

Depuis plusieurs années je participe à la coupe de robotique E=M6 dont l'objectif pour chaque équipe est de fabriquer un robot autonome devant se conformer à un règlement. Pour plus d'infos, vous pouvez vous rendre sur le site de mon équipe, [Microb Technology](#), ou bien sur le site de [Planète Sciences](#), l'organisme bénévole qui organise cette compétition

J'ai découvert en 2001 les microcontrôleurs [Atmel](#) AVR, auxquels je me suis beaucoup attaché, d'une part parce qu'ils sont assez performants pour des microcontrôleurs 8 bits, mais aussi et surtout car le fameux compilateur libre GCC est capable de compiler du code C pour ces petites bêtes. Il existe de surcroît plusieurs outils libres permettant de les programmer, de les simuler, etc...

Le but de ce projet est donc de développer des modules servant d'interface générique avec le hardware du microcontrôleur, quel que soit la version (par exemple les UART inclus dans les AVR AtMega sont différents de ceux présents dans les petits AVR), et de développer des modules utiles pour certaines utilisations usuelles, et pas seulement pour la coupe E=M6 (codage de hamming, asservissement, chiffrement, etc...).

Vous pouvez contacter l'équipe de développement à [avr-dev@droids-corp.org](mailto:avr-dev@droids-corp.org). Pour le moment, aucune release n'a été faite, mais le travail est déjà bien avancé. Chaque soir un screenshot du CVS est réalisé. Il peut ne pas compiler mais normalement ça devrait fonctionner. Voici le [lien](#).

### 3. Modules d'asservissement analogique « Escap »

Nous avons essayé de récupérer les cartes de commande Escap pour moteurs Escap qui avaient été utilisées en 96 et 98 avec succès.

Nous les avons montés sur une base roulante qui pesait en tout 7,4 kg. Voici grosso-modo leurs avantages et leurs inconvénients (pour les notices techniques voir le rapport 96 ou 98) :

#### a. Avantages :

- Cette carte inclut directement un asservissement avec correcteur PI, la commande de puissance, les ponts en H.
- Elle est très facile à utiliser.
- Elle est spécialement conçue pour les moteurs Escap que nous avons, donc parfaitement adaptée.

#### b. Inconvénients :

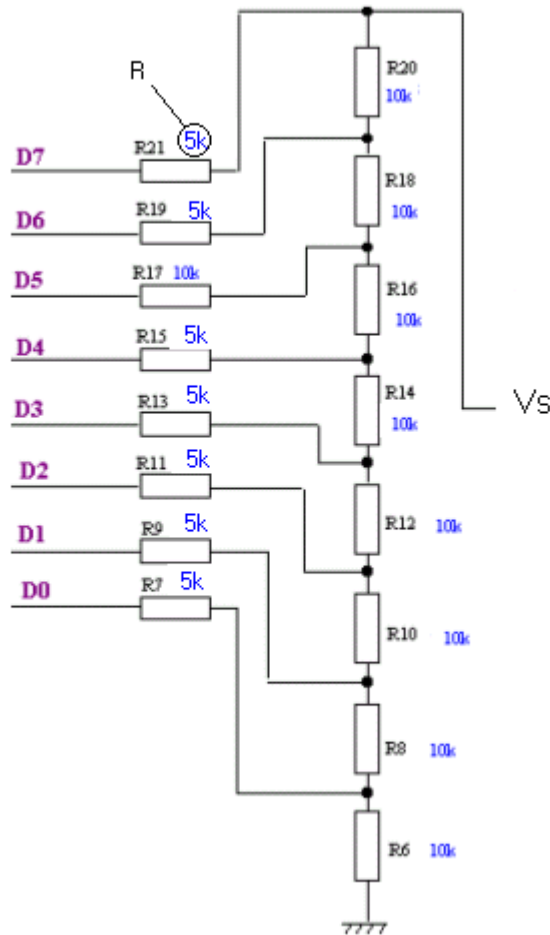
- Les moteurs ne fonctionnent correctement que sous 24v.
- Les cartes et les moteurs semblent assez gourment.
- Les cartes, nantis de gros radiateurs, sont relativement encombrantes.
- Les tachymètres montés sur les moteurs ne donnent pas une indication de position, mais de vitesse (qui plus est en analogique).

Nous avons préféré refaire la partie commande, car celle de 1996 souffrait de quelques défauts (cf. rapport 1996).



### La conversion numérique analogique :

Pour réaliser la conversion numérique analogique, nous avons choisit un pont à résistance R-2R.



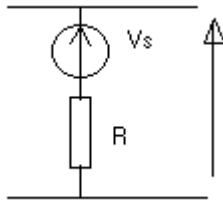
Nous pouvons considérer que chaque bit est relié à un générateur idéal de tension (qui délivre 0v ou 5v). Par conséquent la résistance équivalente, quelque soit l'état des bits et quelques soit le nombre de bits, vue entre la sortie et la masse sera égal à R.

La tension  $V_s$ , en fonction de l'état des bits sera :

$$V_s = \frac{1}{256} (D_0 * 2^0 + D_1 * 2^1 + D_2 * 2^2 + D_3 * 2^3 + D_4 * 2^4 + D_5 * 2^5 + D_6 * 2^6 + D_7 * 2^7)$$

Ce type de CAN étant très classique, nous n'indiquerons pas les démonstrations, toutefois vous pourrez les trouver très facilement sur Internet ou sur le Cd-rom d'annexes joint au dossier.

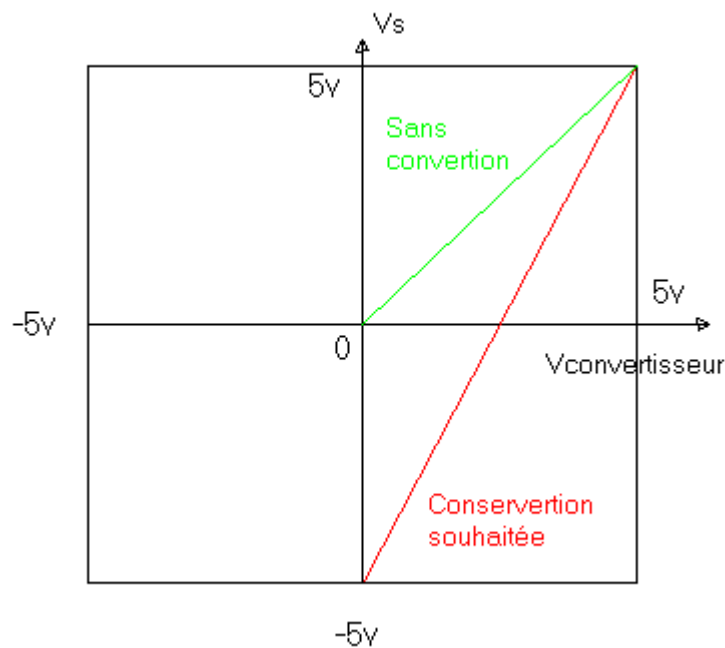
Le montage de Thevenin équivalent est donc :



Nous avons jugé qu'une discrétisation sur 8 bits était un peu superflu, c'est pourquoi nous nous contenterons de 4 bits (ce qui représente quand même 16 niveaux de vitesse).

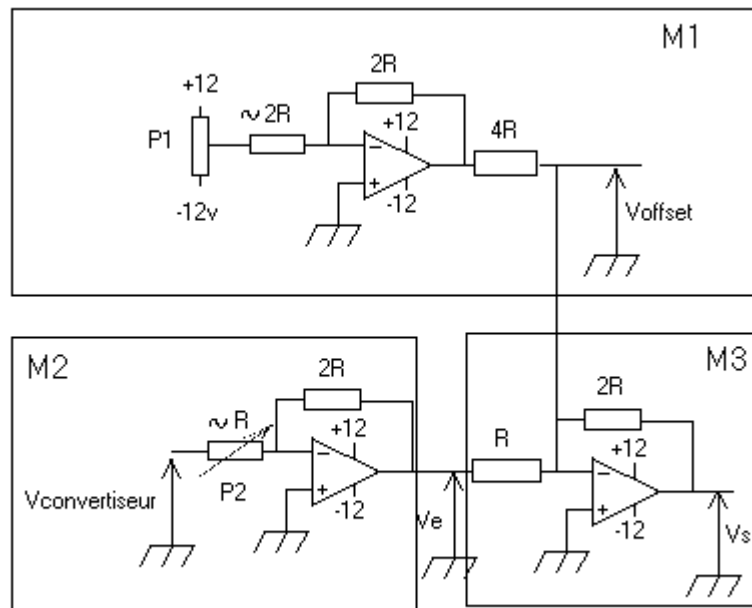
- L'adaptation du niveau de tension :

Nous avons à la sortie du CAN une tension qui s'étend de 0 à 5v, or la commande des cartes Escap ce fait entre -5v et +5v.



On peut voir cette adaptation comme l'ajout d'un offset de -2,5v et une amplification de gain 2.

Voici le montage réalisant cette conversion :



M1 : est un montage suiveur qui ‘recopie’ la tension aux bornes de P1 mais sous une impédance constante et égale à  $4R$ . En 1996 l’équipe avait branché P1 directement sur M3, mais cela a pour inconvénient de faire varier à la fois l’offset et la pente. Le potentiomètre P1 permet d’ajuster l’offset (permet par conséquent de régler l’arrêt complet de moteur). (La valeur théorique de l’offset est  $5V$  (cf. formule en M3)).

M2 : est aussi un montage suiveur. Le potentiomètre P2 permet d’ajuster la pente (qui doit avoir un coefficient directeur de 2). Comme M2 est connecté au CAN dont la résistance interne est de  $R$ , P2 doit être a peut près égal à  $R$ , de la sorte la branche aura une résistance équivalente de  $2R$ .

M3 : Sommateur ( $V_s = - 2 * (V_e + V_{offset})$ ).

Qu’elle valeur choisir pour  $R$  ?

Théoriquement, on peut choisir n’importe quelle valeur pour  $R$ , mais en pratique, si  $R$  est trop petite, une intensité importante serait consommée sans raison et si  $R$  est trop grande, la forte impédance du montage le rendra très sensible aux parasites (secteur, moteurs...). Ce montage a été testé et fonctionne correctement pour  $R=5k$ , les amplificateurs opérationnels utilisés sont des LM324 mais d’autres seraient tout à fait envisageables (peu de contraintes de vitesse et de tensions de déchets).

#### 4. Conclusion :

Nous avons préféré ne pas utiliser les cartes Escap car les moteurs ne renvoient pas une position sous forme d'impulsion mais une vitesse. Etant donné que nous avons envisagé un repérage par odométrie, cela compliquait considérablement la tâche.

Toutefois, il faut savoir que cette solution a été testée et fonctionne très bien (bonne précision, bonne dynamique), ainsi dans le cas où le repérage ne se fait pas par odométrie, mais des balises extérieures, sachez que cette solution est très pratique à utiliser. En effet, pour imposer une vitesse, il suffit au microcontrôleur d'appliquer sur un port un mot binaire correspondant à la vitesse voulue, ... et c'est tout ! La carte Escap se charge du reste.

#### 5. Carte d'asservissement numérique « artisanale »

La carte réalisée basée sur un microcontrôleur PIC 16F877A s'est avérée fonctionnelle, mais les tests d'algorithmes d'asservissement n'ont pas été effectués. En effet, l'utilisation de cette solution nécessite un microcontrôleur dédié à l'asservissement capable de communiquer, via un protocole à définir, avec la « carte mère » (centralisant les retours des autres capteurs du robot et agissant sur les différents actionneurs en fonction de la stratégie).

Or afin de limiter le nombre de microcontrôleurs sur le robot et du fait que l'on dispose d'un budget suffisant, nous avons décidé d'essayer d'utiliser des composants spéciaux réalisant l'asservissement.

Le système se décompose en :

- Un microcontrôleur pour appliquer la PWM, asservir et commander
- Une carte de puissance pour chaque moteur basée sur un LMD18200
- Une carte compteur d'impulsions pour les codeurs

Afin de favoriser l'interactivité et les réglages, j'ai ajouté une carte afficheur LCD avec 6 boutons sur la carte microcontrôleur.

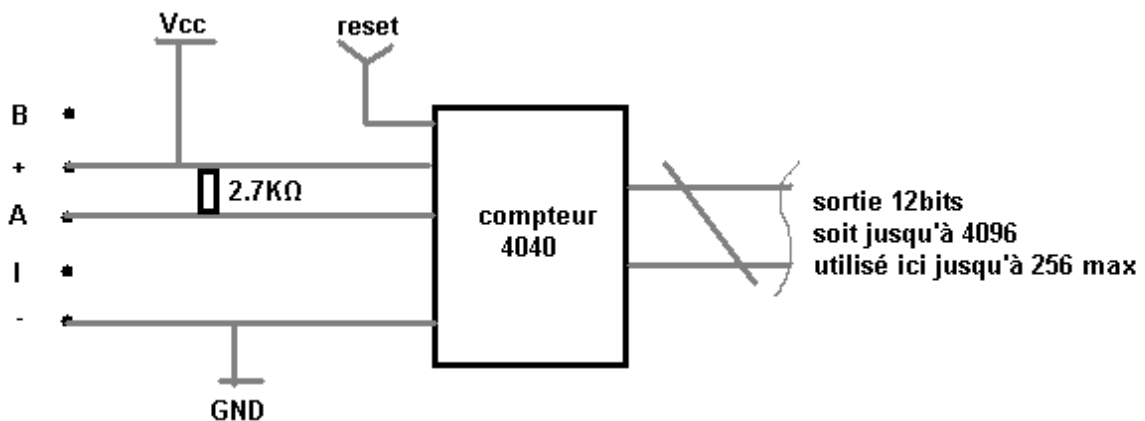
## 6. Carte compteur d'impulsions codeurs :

Son rôle est de décharger le  $\mu\text{C}$  d'une tâche simple mais qui demande toute la disponibilité du  $\mu\text{C}$  au détriment des autres fonctions de celui-ci, ou avec le risque de rater des impulsions et d'avoir une fausse indication de vitesse.

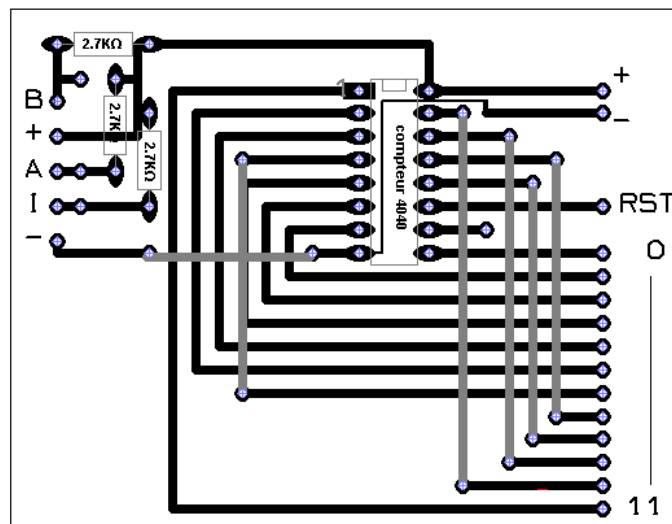
Conformément au datasheet des codeurs HEDS5540 utilisés, il y a 3 résistances de pull-up de  $2.7\text{K}\Omega$ , ainsi que le compteur qui est un classique CMOS 4040.

Tout le montage ainsi que les codeurs HEDS5540 fonctionnent en TTL, ce qui permet de les brancher directement avec le  $\mu\text{C}$ .

Schéma :



Typon :



## 7. Carte de puissance :

Elle s'articule autour du LMD18200 qui est un « pont en H » intégrant une logique TTL (0V ou 5V) qui permet de le connecter directement à un  $\mu\text{C}$  ainsi que de le commander en PWM avec des périodes allant jusqu'à 100KHz.

Le LMD18200 peut commander des moteurs courant continu pour des tensions de 12 à 55V et des intensités jusqu'à 3A nominal (et des pics de 6A sur 200ms).

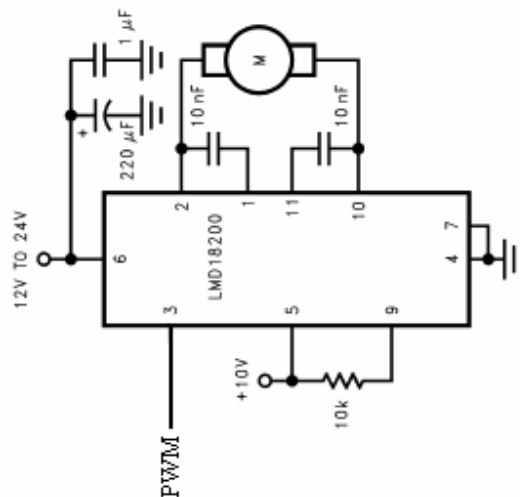
Le LMD18200 inclut également une sortie à *miroir de courant* (fournissant  $377\mu\text{A/A}$  et jusqu'à 12V de tension, connecté avec une résistance de rappel à la masse), un « *temperature flag* » en logique inverse déclenchant à  $145^\circ\text{C}$  ainsi que les bits de direction, brake et PWM.

Il est également protégé contre une surchauffe avec une « *thermal shutdown* » à  $170^\circ\text{C}$ .

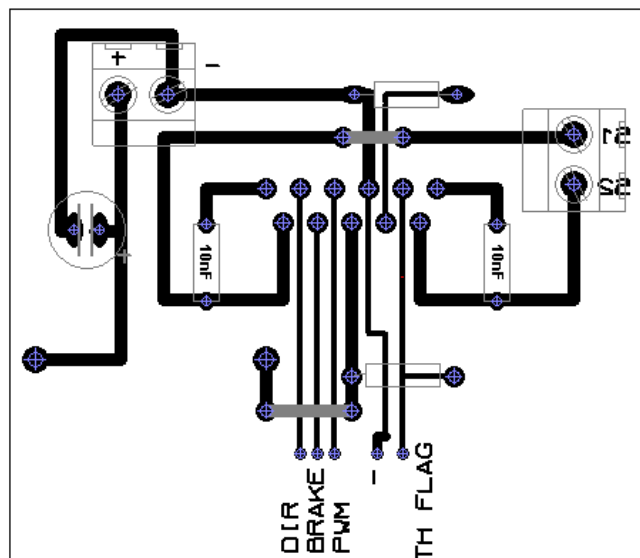
La commande peut être faite de 2 manières :

- \_ Avec un bit pour la direction et la commande PWM variant d'un rapport de 0 à 100%
- \_ Avec le bit PWM au 1 logique et la commande PWM sur le bit direction, de sorte qu'un rapport cyclique à 0% correspond au maximum dans un sens, que 50% correspond au point mort et que 100% correspond au maximum dans l'autre sens.

Schéma :



Typon :



## 8. Carte microcontrôleur :

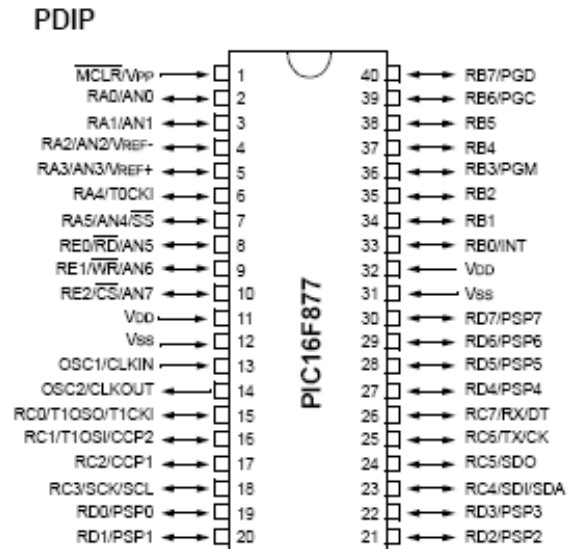
Il s'agit d'une banale carte à 16F877 en communication directe avec les autres cartes par liaison parallèle.

La carte LCD comprend les commandes classiques des LCD à interface 8bits, commandé ici en mode 4bits en raison du nombre de broches disponible sur le 16F877.

### PIC 16F877(A) caractéristiques :

Mémoire : Flash de 8Koctets  
RAM de 368 octets  
Eeprom de 256 octets

5 MIPS (million d'instructions par secondes)  
avec un quartz de 20 MHz



Le LCD possède 16 broches :  
 \_ +5V et masse  
 \_ contraste  
 \_ RS (pour choisir entre commande et donnée)  
 \_ WR (pour lire ou écrire sur le LCD)  
 \_ D0 à D7 pour envoyer ou recevoir les informations  
 \_ E (enable pour valider l'envoi ou la réception)  
 \_ et deux broches pour le rétro-éclairage quand il est présent

Les bits RS et E du LCD ne peuvent pas se connecter sur RA4 et RA5 du fait que RA4 est une sortie à drain ouvert, ce qui implique qu'elle ne fournit pas de 5V TTL.

Ces 2 bits ont donc été branchés sur RE0 et RE1, de sorte que RA4, RA5 et RE2 puisse facilement se réunir en une information de 3 bits pour les boutons qui renvoient en binaire le numéro du bouton pressé.

En effet, si on prend les quartets MSB de RA et LSB de RE :

RA	: 0011
+ RE	: 0100
⇒	0111 (3 bits pour les touches)

RC1 et RC2 sont les sorties PWM incluent dans le PIC donc fixes.

J'ai préféré choisir des ports entiers donc plus facile à lire pour relever les compteurs : port B et port D.

Les bits RC6 et RC7 appelé *démarrage carte puissance* serve à pouvoir arrêter ou n'appliquer aucune consigne sur les moteurs temps que les PWM ne sont pas établies lors de l'initialisation du PIC.

Et enfin les bits RC4 et RC5 servent à remettre à zéro les compteurs après lecture.

Avant d'aller plus loin voici le tableau de brochage du 16F877 :

broche	port 16F877A	fonction
1		
2	RA0	LCD bit 4
3	RA1	LCD bit 5
4	RA2	LCD bit 6
5	RA3	LCD bit 7
6	RA4	Bouton 0
7	RA5	Bouton 1
8	RE0	LCD bit RS
9	RE1	LCD bit E
10	RE2	Bouton 2
11		
12		
13		
14		
15	RC0	
16	RC1	sortie PWM2
17	RC2	sortie PWM1
18	RC3	
19	RD0	Compteur 1 bit 0
20	RD1	Compteur 1 bit 1
21	RD2	Compteur 1 bit 2
22	RD3	Compteur 1 bit 3
23	RC4	reset compteur 1
24	RC5	reset compteur 2
25	RC6	démarrage carte puissance 1
26	RC7	démarrage carte puissance 2
27	RD4	Compteur 1 bit 4
28	RD5	Compteur 1 bit 5
29	RD6	Compteur 1 bit 6
30	RD7	Compteur 1 bit 7
31		
32		
33	RB0	Compteur 2 bit 0
34	RB1	Compteur 2 bit 1
35	RB2	Compteur 2 bit 2
36	RB3	Compteur 2 bit 3
37	RB4	Compteur 2 bit 4
38	RB5	Compteur 2 bit 5
39	RB6	Compteur 2 bit 6
40	RB7	Compteur 2 bit 7



Typon de la carte « LCD-boutons » :

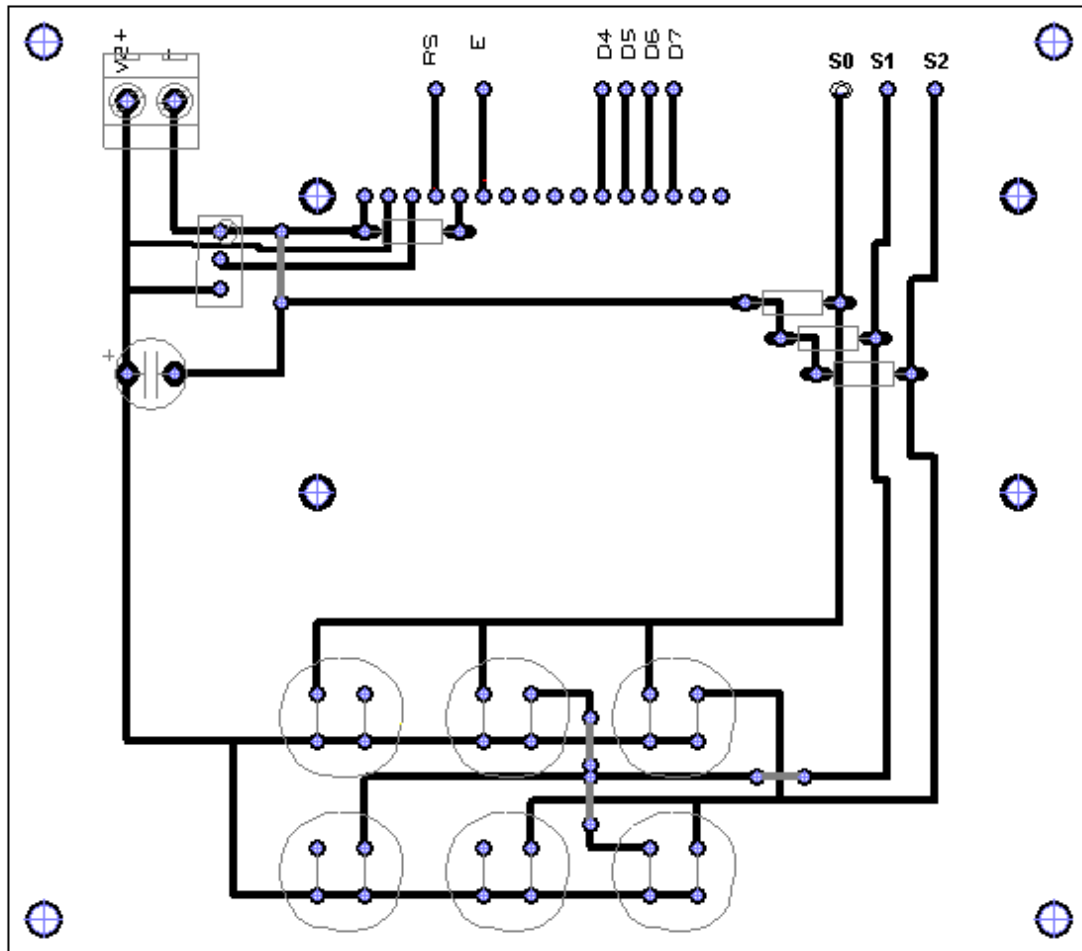
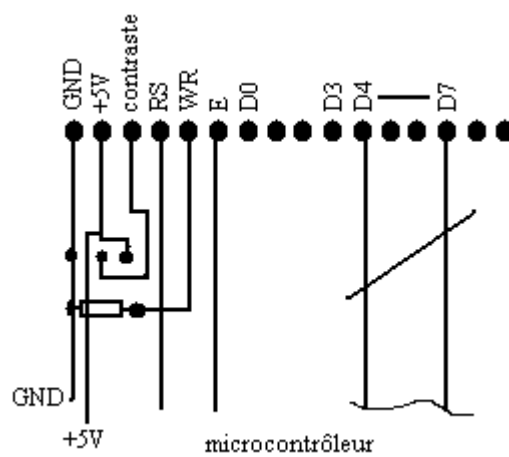


Schéma de la connectique du LCD :



### Commande possible du LCD :

RS	R/W	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	Description	
0	0	0	0	0	0	0	0	0	1	Efface l'afficheur et met le curseur au 1er caractère	
0	0	0	0	0	0	0	0	0	1	* Remet le curseur en haut à gauche	
0	0	0	0	0	0	0	0	1	ID	S Déplacement du curseur après apparition d'un caractère	
0	0	0	0	0	0	1	D	C	B	Options d'affichage du curseur	
0	0	0	0	0	1	SC	RL	*	*	Décalage du curseur ou de tout l'affichage	
0	0	0	0	1	DL	N	F	*	*	Choix 4/8 bits, choix des lignes utilisées et de la taille caractères	
0	0	0	1	-	-	-	-	-	-	- Adresse du générateur de caractères	
0	0	1	-	-	-	-	-	-	-	- Adresse de la mémoire de données	
0	1	BF	-	-	-	-	-	-	-	- Lecture du busy flag et de l'adresse de la position du curseur	
1	0	-	-	-	-	-	-	-	-	- Ecrire dans la mémoire de donnée ou le générateur de caractères	
0	1	-	-	-	-	-	-	-	-	- Lire données dans la mémoire ou le générateur de caractères	
ID	0	déplacement du curseur à gauche								1	déplacement du curseur à droite
S	0	L'affichage ne se décale pas								1	L'affichage se décale
D	0	L'affichage est désactivé								1	L'affichage est activé
C	0	Le curseur ne se voit pas								1	Le curseur se voit
B	0	Le caractère actif ne clignote pas								1	Le caractère actif clignote
SC	0	Déplacement du curseur								1	Déplacement de l'affichage
RL	0	Décalage vers la gauche								1	Décalage vers la droite
DL	0	Mode 4 bits								1	Mode 8 bits
N	0	Une ligne utilisée								1	Deux lignes utilisées
F	0	Taille des caractères = 5*7								1	Taille des caractères = 5*10
BF	0	Le LCD est prêt								1	Le LCD est occupé

### 9. Algorithme :

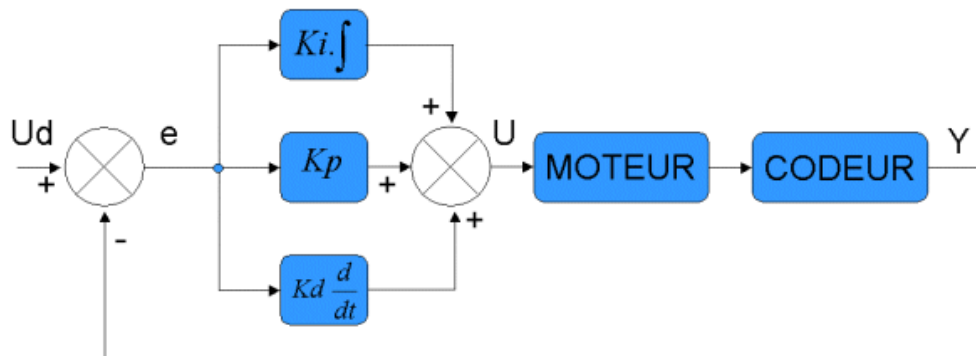
Nous avons d'une part une lecture des codeurs toutes les  $\sim 820 \mu\text{s}$  qui nous informe de la vitesse de rotation des moteurs.

Or les codeurs HEDS5540 renvoient 500 impulsions par tour, or la sortie du moteur entraîne un réducteur 1/33.2 avant d'entraîner la roue.

Pour un tour de roue on a donc  $500 * 33.2 = 16600$  impulsions par tour réel de roue.

Or après calcul, on peut compter au maximum 312000 impulsions par secondes, soit pour des roues de 4cm de rayon, une vitesse maximum de 4.7 m/s => ce qui est largement supérieur à la vitesse maximale réelle des moteurs.

L'asservissement doit se faire en boucle fermée suivant le schéma :



- $U_d$  : vitesse désirée
- $e$  : erreur entre la vitesse désirée et la vitesse réelle
- $U$  : Consigne appliquée au moteur
- $Y$  : Vitesse réelle

En ce qui concerne la programmation de l'asservissement de manière software, il faudra procéder suivant une démarche similaire à celle décrites précédemment dans la partie « Comparatifs des modules d'asservissement et calcul des correcteurs : système PID ».

### 10. Conclusion :

Cette solution est envisageable, et déjà utilisée par différentes associations de robotique.

Le point le plus délicat reste le calcul du correcteur par le microcontrôleur qui réclame une grande puissance, et cela constamment et le plus rapidement possible pour que le système reste réactif.

Si cette solution est retenue, il est nécessaire d'utiliser un microcontrôleur dédié pour l'asservissement, et d'établir ensuite un protocole de communication avec la carte principale qui appliquera les consignes en fonction du parcours que doit faire le robot.

### 11. Remarques :

Lors des essais, il y eut un problème avec le port B sur la carte microcontrôleur => il était causé par l'option LVP (Low Voltage Programming) dans le registre du PIC 16F877A qu'il faut veiller à désactiver, car dans le mode LVP, la broche 3 du port B fait commuter ce dernier en mode programmation et stoppe le fonctionnement en cours.

Suite à la perte d'un codeur après une mauvaise manipulation, nous n'avons pas pu réaliser de tests de déplacement avec asservissement => en revanche la commande des moteurs avec variation de vitesse était satisfaisante.

Le LCD présente quelques problèmes d'affichage propre à un défaut de fabrication de celui recyclé sur la carte LCD de tests => il faudrait un LCD neuf.

## 12. Circuits d'asservissement numérique « LM629 »

Nous avons réalisé une carte mettant en œuvre les circuits LM629.

Cette carte permet de rassembler les circuits intégrés d'asservissement (LM629) et un microcontrôleur (PIC 16F877A) pour envoyer les commandes et interpréter les résultats.

La carte contient un régulateur de tension (5V jusqu'à 1,5 A) qui permet de fournir une tension TTL de 5V afin d'alimenter les différents circuits intégrés : PIC et LM629.

Un interrupteur nous permet de couper l'alimentation du PIC, de sorte que l'on puisse faire fonctionner le circuit d'asservissement avec un autre circuit de commande (autre microcontrôleur externe).

Les LM629 sont cadencés à leur vitesse maximale de 6MHz, sachant qu'il existe aussi des modèles à 8MHz, mais qu'ils n'étaient pas disponibles au moment de la commande.

On trouve également un oscillateur 20MHz avec deux petits condensateurs pour fixer la fréquence du microcontrôleur PIC ainsi que des résistances de pull-up pour les codeurs HEDS5540 comme préconisé dans le datasheet.

Enfin, notons que la communication avec les deux LM629 se fait sur un bus commun, et le choix d'adressage du LM629 souhaité s'effectue grâce aux lignes  $\overline{CS}$  (chip select) respectives.

L'alimentation du circuit se fait directement sous 24 V et est répartie aux interfaces de puissance.

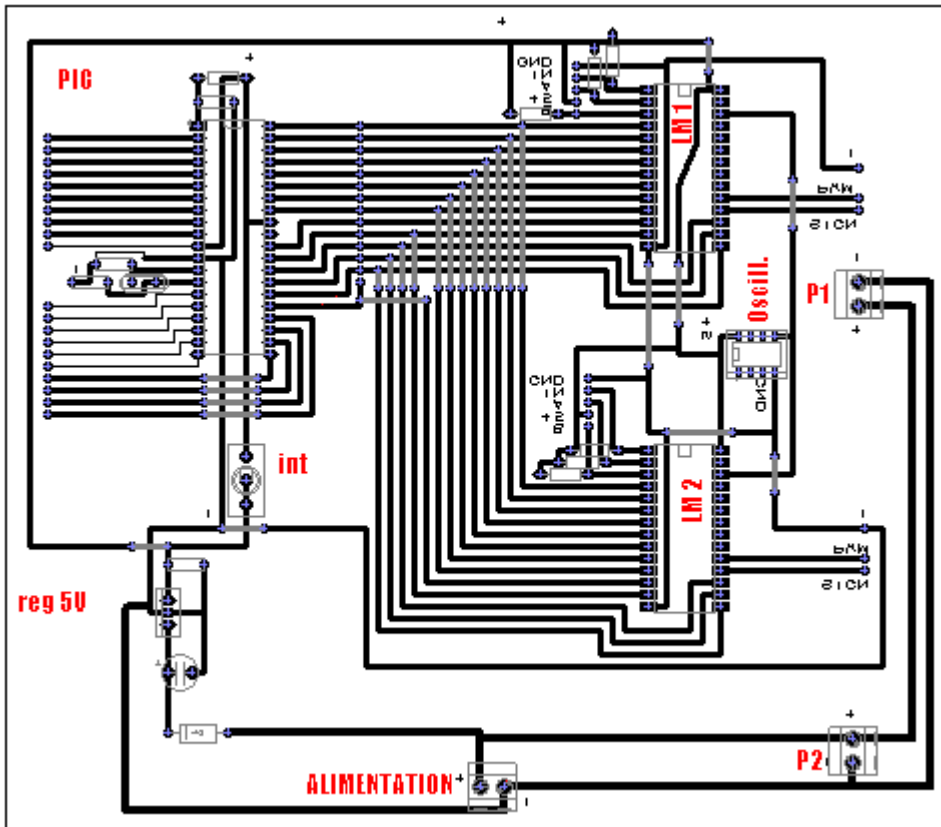
Les interfaces de puissance pour alimenter les moteurs sont des « pont en H » à LMD18200 comme décrit précédemment dans la section « carte d'asservissement numérique artisanal ».

Les principaux avantages du circuit sont d'une part sa simplicité technique du fait que tous les composants sont en TTL et peuvent donc se connecter directement entre eux, et d'autre part sa puissance de calcul qui offre différents types d'asservissement avec une très grande précision.

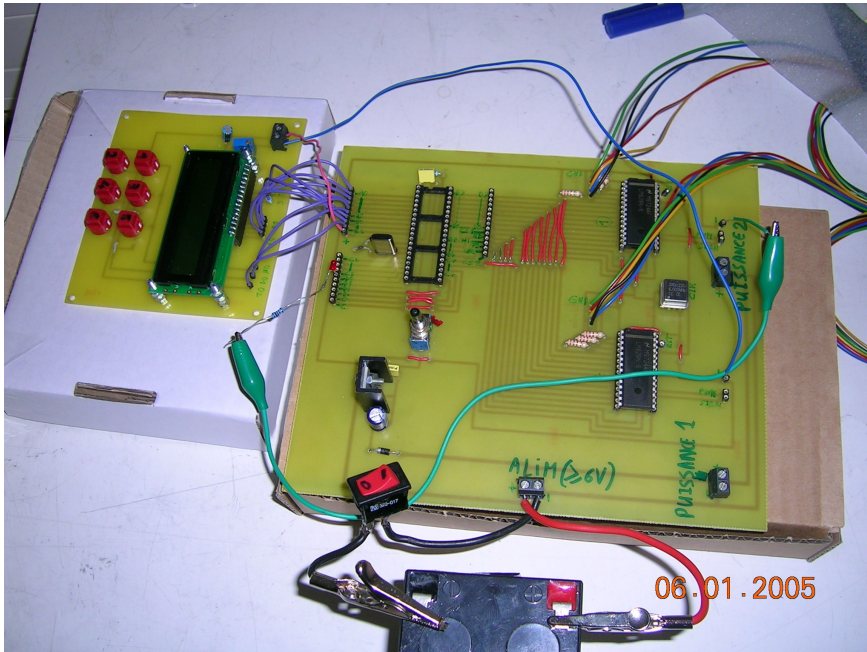
### Correspondance des bornes du microcontrôleur PIC utilisé :

Broches du PIC	Broches du LM
RB 0	D0
RB 1	D1
RB 2	D2
RB 3	D3
RB 4	D4
RB 5	D5
RB 6	D6
RB 7	D7
RD 4	PS
RD 5	HI
RD 6	RD
RD 7	CS 1
RC 6	CS 2
RC 7	WR

*Typon :*



*Photos :*



Les câbles de liaison aux codeurs de type HEDS5540 sont soudés et prêts à être connecté (câble vert = GND), et les sorties P1 et P2 ainsi que les lignes PWM et signe des LM, permettent de brancher du même côté les cartes de puissances.

Remarques :

- \_ prévoir plus de sorties +5V et GND pour brancher les cartes annexes d'essais (carte LCD)
- \_ envisagé de placer les résistances de pull-up des codeurs HEDS5540 plus près de ces derniers pour limiter au maximum les interférences.
- \_ temps de lecture/écriture minimum :
  - \_ écriture commande : 160ns
  - \_ écriture donnée : 250ns
  - \_ lecture donnée : 330ns

13. Conclusion :

Cette carte permet de rassembler de façon pratique les composants nécessaires pour la mise en œuvre des circuits d'asservissement LM629.

Elle n'a été réalisée que dans le cadre d'essai de la base roulante pour l'UV PR, et une nouvelle carte sera probablement imprimée pour la compétition de robotique de l'association E=M6, avec une forme plus appropriée à la place disponible dans le robot, et avec le choix définitif du microcontrôleur de commande.

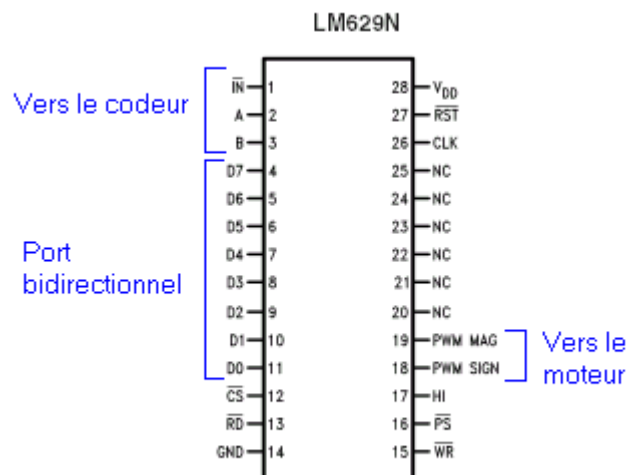
On peut également envisager d'intégrer sur la carte les interfaces de puissance à base de LMD 18200 afin de limiter la quantité de cartes sur le robot et les différentes connectiques qu'implique la communication entre ces cartes.

14. Commande des circuits d'asservissement numériques « LM629 »

Cet asservissement repose sur le LM629 qui va faire une grosse partie du travail pour nous. Ces circuits d'asservissement disposent d'un correcteur PID dont chaque paramètre (P, I et D) sont ajustables numériquement.

On peut lui fixer une consigne de vitesse ou de position, ainsi qu'une accélération.

Présentation des principales broches :



### Port bidirectionnel (D0 à D7) :

Permet de lire les registres du LM (registres de vitesse, de statuts...) et d'envoyer les commandes (assignation d'une vitesse, démarrage...).

$\overline{WRRD}$  : Ecriture ou lecture des registres. A l'état de 'repos', ces bits doivent être à 1.

$\overline{PS}$  : Port select, quand PS=0, on envoie une commande, quand PS=1 on envoie une donnée.

$\overline{IN}$  : Index, le codeur du moteur y envoie une impulsion par tour. On veillera à mettre une résistance de pull up.

$\underline{A}, \underline{B}$  : Impulsions en provenance du codeur.

$\underline{HI}$  : Signale une interruption (c'est-à-dire un événement particulier telle une opération non conforme, un parcours fini, une erreur trop grande...).

$\overline{RST}$  : Reset (redémarrage) du LM.

$\overline{CS}$  : (Chip Select) active le LM. Ce bit nous a permis d'utiliser le même port bidirectionnel pour commander les deux LM à la fois.

### Principe de fonctionnement:

#### Le démarrage :

Au démarrage, ou après un reset, tous les registres (en 8, 16 ou 32 bits) du LM sont nuls. Ainsi la première étape consiste à les configurer.

1. Les registres du filtre PID :

$$V_{Robot} = P.e + I.\int_0^T e.dt + D.\frac{\Delta e}{\Delta T}$$

e : l'erreur

T : la période d'échantillonnage

$$T = \frac{2048}{F_{Clock}} \quad \text{Dans notre cas, } F_{Clock} = 6\text{MHz.}$$



Les coefficients P, I et D sont codés en 16 bits, on peut donc leur assigner une valeur comprise entre 0 et 65535.

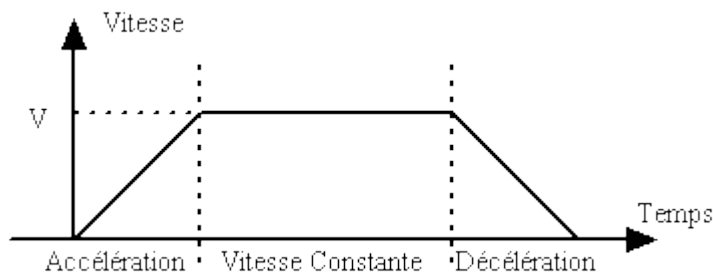
## 2. Le chargement d'une trajectoire :

La trajectoire est définie par son profil trapézoïdal, c'est à dire :

- une accélération
- une vitesse maximale
- une position à atteindre (uniquement en mode position).

Chacun de ces éléments peut être défini en absolu ou en relatif par rapport aux éléments actuels, c'est à dire par rapport à :

- l'accélération actuelle.
- la vitesse maximale de consigne actuelle.
- la position à atteindre actuelle.



**Profil de vitesse trapézoïdal**

### Remarque :

Nous appelons *COUNT* le nombre d'impulsions du codeur\*4. La multiplication correspond à la méthode utilisée par le LM pour décoder les impulsions générées par le codeur. Pour une roue codeuse (codeur) ayant 500 points par tour, le LM va décoder 2000 impulsions (*COUNT*) par tour. Cette unité dépend donc du codeur utilisé. En fonction du nombre de points du codeur par tour, du montage mécanique, un point du codeur peut représenter un déplacement de  $x$  mm, ou bien une rotation de  $x$  degrés, etc. Le compteur de position, par exemple utilise le *COUNT* pour mesurer les déplacements.

Nous appellerons *SAMPLE* la période d'échantillonnage du LM.

Dans le registre du LM, les paramètres de trajectoire ont un codage spécifique :

- La position à atteindre est exprimée par un nombre signé de 32 bits, en C-OUNT
- La vitesse maximale est exprimée par un nombre non signé en virgule fixe de 16.16 bits, en COUNT/SAMPLE
- L'accélération maximale est exprimée par un nombre non signé en virgule fixe de 16.16 bits, en COUNT/SAMPLE/SAMPLE

Pour chaque élément, il n'est pas possible d'utiliser toutes les valeurs, seule une plage de valeurs est possible.

<b>Donnée</b>	<b>Plage autorisée</b>
<b>Position</b>	De $-2^{30}$ à $2^{30}-1$ , soit <b>0xC0000000</b> à <b>0c3FFFFFFF</b> ou encore de <b>1073741824</b> à <b>1073741823</b>
<b>Vitesse</b>	<b>Toute la plage. (0 à 65536,99998 soit 0x00000000 à 0xFFFFFFFF)</b>
<b>Accélération</b>	De <b>0</b> à $2^{30}-1$ , soit de <b>0</b> à <b>0x3FFFFFFF</b> ou encore de <b>0</b> à <b>16383,99998</b>

Voici les formules qui permettent de trouver la correspondance entre ces unités et les grandeurs physiques (sachant que le codeur dont nous disposons délivre 500 impulsions par tour et que nb\_impulsion représente le nombre réel d'impulsion et que le moto réducteur a un rapport 6,25 :1) :

Pour le déplacement :

$$d = \frac{nb_{impulsion}}{4} * \frac{2 * \pi * R_{Roue}}{500 * 6,25}$$

Le nombre d'impulsions étant codé dans un registre 32 bits en code complément à deux (pour pouvoir coder aussi des déplacements négatif) par conséquent :

➤ Nous pouvons coder un déplacement maximum de plus ou moins : 21,6 km !

En effet :

$$d_{Max} = \frac{2^{30} - 1}{4} * \frac{2 * \pi * 0,04}{500 * 6,25} = 21588m$$

➤ Avec un quantum théorique de : 0,02 mm

$$d_{Quantum} = \frac{2^0}{4} * \frac{2 * \pi * 0,04}{500 * 6,25} = 2^{-5}m$$

Pour la vitesse :

$$v = \frac{nb_{impulsion}}{4} * \frac{2 * \pi * R_{Roue}}{500 * 6,25} * \frac{F_{Clock}}{2048}$$

- Vitesse maximum : 1072 Km/h (bien sûr la vitesse maximum réelle dépendra du moteur utilisé.

$$v_{Max} = \frac{65536,99998}{4} * \frac{2 * \pi * 0,04}{500 * 6,25} * \frac{6 * 10^6}{2048} = 3860 m / s$$

- Avec un quantum de :  $1,17 * 10^{-17}$  m/s

$$v_{Quantum} = \frac{2^{-16}}{4} * \frac{2 * \pi * 0,04}{500 * 6,25} * \frac{6 * 10^6}{2048} = 1,17 * 10^{-17} m / s$$

Pour l'accélération :

- Accélération maximale codable : 2827433 m/s<sup>2</sup>

$$a = \frac{16383,99998}{4} * \frac{2 * \pi * R_{Roue}}{500 * 6,25} * \left( \frac{F_{Clock}}{2048} \right)^2$$

- Avec un quantum de : 2,63 mm/s<sup>2</sup>

$$a = \frac{2^{-16}}{4} * \frac{2 * \pi * R_{Roue}}{500 * 6,25} * \left( \frac{F_{Clock}}{2048} \right)^2 = 2,63 * 10^{-3} m / s^2$$

### 15. Conclusion:

Ce composant peut donc servir dans un très grand nombre d'application et avec une très grande précision théorique. En pratique, la précision dépendra beaucoup de la partie mécanique du système.

## 16. La carte mère

### a. Choix de la carte microcontrôleur

La carte centrale du robot doit être suffisamment puissante pour supporter des utilisations futures gourmandes en ressources. Ainsi, on pourra réutiliser la même d'année en année et économiser du temps de développement et de l'argent.

Elle doit également avoir assez d'entrées-sorties et de bus pour pouvoir brancher tous les périphériques.

Puisque c'est l'organe central du robot, mieux vaut un microcontrôleur accompagné de sa carte dont on sait qu'elle fonctionne très bien.

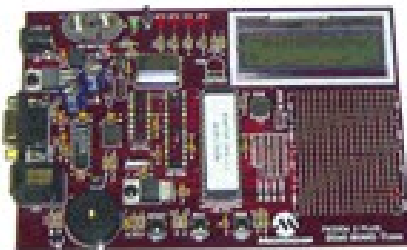
Enfin, le prix de l'ensemble doit être raisonnable.

Pour choisir cette carte, nous avons étudié les modèles de cartes pour les microcontrôleurs suivants, dont les descriptions détaillées sont en annexes :

- Motorola HCS12 (MC9S12DP256B)
- Microchip PIC 18F458
- Rabbit (RCM3400)
- Protee
- PC104
- Hitachi SH-1

Il a finalement été décidé d'acheter une T-board comme carte centrale et un PICdem2plus pour développer des modules à base de PICs.

### b. Programmation du microcontrôleur HCS12 sur la carte T-Board



Une carte PICdem2plus et une carte T-Board

c. Quelques détails sur la carte " HCS12 T-Board "

La carte " HCS12 T-Board " peut être utilisée telle quelle pour un développement direct (de part l'existence de son moniteur intégré). Toutefois, le module optionnel "ComPod" permet de télécharger plus rapidement le programme au sein de la carte tout en permettant de récupérer l'espace mémoire attribué au moniteur.

Si on ne veut pas utiliser l'assembleur livré comme langage de programmation, le compilateur optionnel "ICC12" permet de développer en "C" de façon très rapide et professionnelle. Enfin, l'option "NoIce" (associée au "ComPod" et à "ICC12") permet de disposer d'un environnement de développement puissant et convivial avec mode "DEBUG" (points d'arrêts, mode pas-à-pas, etc...).

Evidemment, aucune de ces options n'a été achetée. Il faut donc se débrouiller avec le minimum.

Le CD-rom livré avec la carte contient la version d'évaluation du compilateur C (\DevTools\ICC12).

Une fois que le programme est compilé grâce à ce logiciel, il faut le transférer dans le microcontrôleur. Cette étape se fait grâce au moniteur situé dans le microcontrôleur. Le fichier devant être transféré est le programme sous forme hexadécimal (extension .s19).

Il est possible de le faire avec l'HyperTerminal de Windows. Une fois connecté via la liaison série, la commande LOAD permet de transférer un programme par copier coller dans le terminal.

Un programme de terminal (OC-Console) est livré sur le CD-rom (DevTools\Terminal).

En tapant "help" dans le terminal, on a la liste de toutes les commandes du moniteur.

Enfin, on peut exécuter le programme en tapant G suivi de l'adresse de la première instruction du programme.

Voici le "quick-start" extrait du manuel :

- Connect the Evaluation Board via RS232 to a PC. The connection between HCS12 T-Board (interface SCIO, connector X2) and PC is simply made using the flat ribbon cable which is in the box.
- On the PC, start a Terminal Program. An easy to use Terminal Program is OC-Console, which is available at no charge from our Website!
- Select a baudrate of 9600 Bd. Dissable all hardware or software protocols.
- **Caution:** make sure that jumper JP1 is in position 1-2! (this is the factory default setting)
- Connect a power supply to X1, delivering approx. 9V (8..12V, center pin positive)
- Please note: wall plug power supplies are usually not stabilized and they provide a voltage that is higher than the nominal (full load) voltage. Therefore, in order to get "real" 9V, using 6V or 7.5V is normally sufficient. The higher the input voltage, the more heat will be produced by IC3.
- Once powered up, the Monitor program will start, display a message and await your commands. For a list of monitor commands, please refer to the D-Bug12 Reference Guide (see separate document on the accompanying CD-ROM)

# Partie 4 : Divers

Dans cette partie, nous allons nous intéresser au budget de l'UV PR, aux fournisseurs et usinages dans les différents domaines qui nous intéresse.

## 1. Budget :

<b>Budget PR A04</b>		
	Désignations	en Euros
Mécanique		
	2 moteurs	674,54
	2 transmissions	122,37
	2 roues de roller	15,00
	2 axes de roues	0,00
	2 accouplements rigides	0,00
	2 supports	30,00
	Platine support	0,00
	Plaque de fond	0,00
	Usinage	0,00
	Visserie	1,67
Electronique / Informatique		
	Carte de développement Pic	119,32
	Carte mère	183,05
	2 LMD 18200	63,08
	Composants divers	62,65
	Tirage de carte	0,00
Divers		
	Impressions rapports	60
	Démarchage téléphonique et fax	2,00
Total		1 333,68
Objectif fixé		2 000,00
Conclusion	Bénéficiaire	666,32

## 2. Fournisseurs :

En ce qui concerne les fournisseurs de moteurs, nous avons contacté :

- MDPmotor : [www.mpdmotor.fr](http://www.mpdmotor.fr)  
Contact : Mme Anne Champétinaud.  
Téléphone : 04 72 01 83 00  
Commande passée au nom de : **Coupe E=M6 BDE de l'U.T.C.**  
M. DUMAY D. Accueil B. Franklin  
Rue Roger Couttolenc  
60203 COMPIEGNE

Commentaires :

Livraison en 48H gratuite si en stock.  
Si pas en stock délais très long.  
Réduction de 20% si participation à la Coupe.

- MUVMO : [www.faulhaber.com](http://www.faulhaber.com)  
Contact : M. Benoît Turbe  
Téléphone : 01 30 80 45 00

Commentaires :

Moteurs très performants mais aussi très chers.

Pour ce qui est des renvois d'angle, la référence que nous avons retenue était disponible chez 2 fournisseurs :

- Radiospares : [www.radiospares.fr](http://www.radiospares.fr)  
Adresse : Radiospares  
Rue Norman King B.P. 453  
60031 BEAUVAIS  
Commentaires : Les prix sont un peu plus élevés que la concurrence, mais les frais de port sont gratuits et la livraison en moins de 24 heures.  
On a aussi -10% en tant que participant à la Coupe.
- HPC Europe : [www.hpceurope.com](http://www.hpceurope.com)  
Commentaires : Idéal pour tout ce qui est mécanique (réduction, moteur, entraînement,...), vérifier que les pièces sont bien en stock.

Pour tout ce qui est visserie :

- Daniel :  
Commentaires : Vente de vis à l'unité, d'outils... prix très compétitifs.

En ce qui concerne les fournisseurs de matériel électronique, nous avons passé nos commandes chez :

- Electronic Diffusion : [www.electronic-diffusion.fr](http://www.electronic-diffusion.fr)
- Conrad : [www.conrad.fr](http://www.conrad.fr)
- Radiospares : [www.radiospares.fr](http://www.radiospares.fr)
- Selectronic : [www.selectronic.fr](http://www.selectronic.fr)
- Lextronic : [www.lextronic.fr](http://www.lextronic.fr)



- Microship : [www.microship.com](http://www.microship.com)

## 22. Usinages :

La halle GSM dispose d'une fraise et d'un tour à commande numérique, il faut contacter M. Pouille, les délais d'usinage sont très importants (2 ou 3 mois), il faut maîtriser TN24 et donc CATIA, et éventuellement acheter les outils et la matière.

La Halle GC dispose d'un tour manuel, idéal pour l'usinage de petites pièces ne nécessitant pas une précision supérieure à 0.1mm. Là il faut contacter M. Poirel, impérativement fournir la matière, et des plans manuel suffisent. Les délais sont de l'ordre de un à deux jours. Nous y avons fait usiner les axes de roues et les accouplements.

Pour tout ce qui est perçage, taraudage, découpage de tôle... qui ne nécessite pas une très grande précision, l'atelier TN04 est tout à fait suffisant, il faut demander aux responsables avant de se servir des machines.

Pour ce qui est de la Halle GM, M. Boulnois n'a jamais le temps, donc c'est une idée à mettre de côté.

En ce qui concerne les 2 supports, ils ont été usinés par M. Jean Pierre MONTE, Chef d'atelier chez A.S.M.P. (Atelier Service Mécanique Précision), à Ris-Orangis. Il s'agit d'un parent d'un de nous.

# Conclusion

L'objectif de ce projet était de réaliser la base roulante du futur robot de la Coupe E=M6. Nous pensons avoir atteint notre objectif, même si d'autres tests et améliorations doivent être envisagés et effectués. Le dossier technique ici présent constitue une base documentaire, qui pourra dans l'avenir être consulté, et donc faire gagner du temps aux futures équipes. Cette UV PR va permettre à l'association de partir sur de bonnes bases pour cette édition, mais aussi pour les prochaines.

# Remerciements

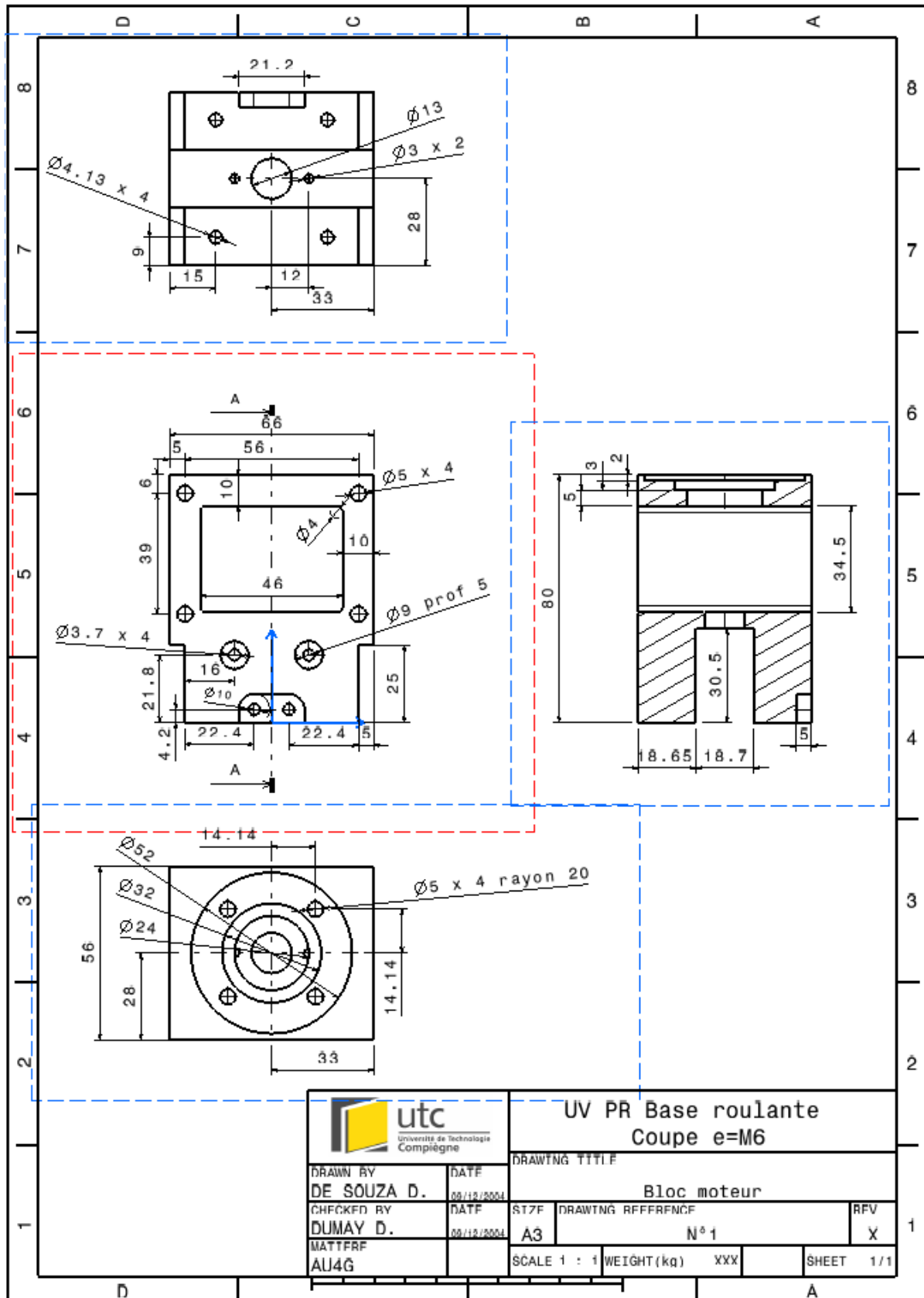
Nous tenons à remercier :

- M. Loriaux et M. Baillot, responsables de l'atelier TN04.
- M. Lanfranchi et M. Forgez, enseignants chercheurs au laboratoire d'électromécanique de l'U.T.C., et suiveurs de notre PR.
- M. Poirel, technicien de la halle G.C..
- M. Pouille, responsable de la halle G.S.M..
- M. Monte, Chef d'atelier chez A.S.M.P..
- M. Arnoult, Enseignant chercheur à l'U.T.C. (pour les calculs de R.D.M.).

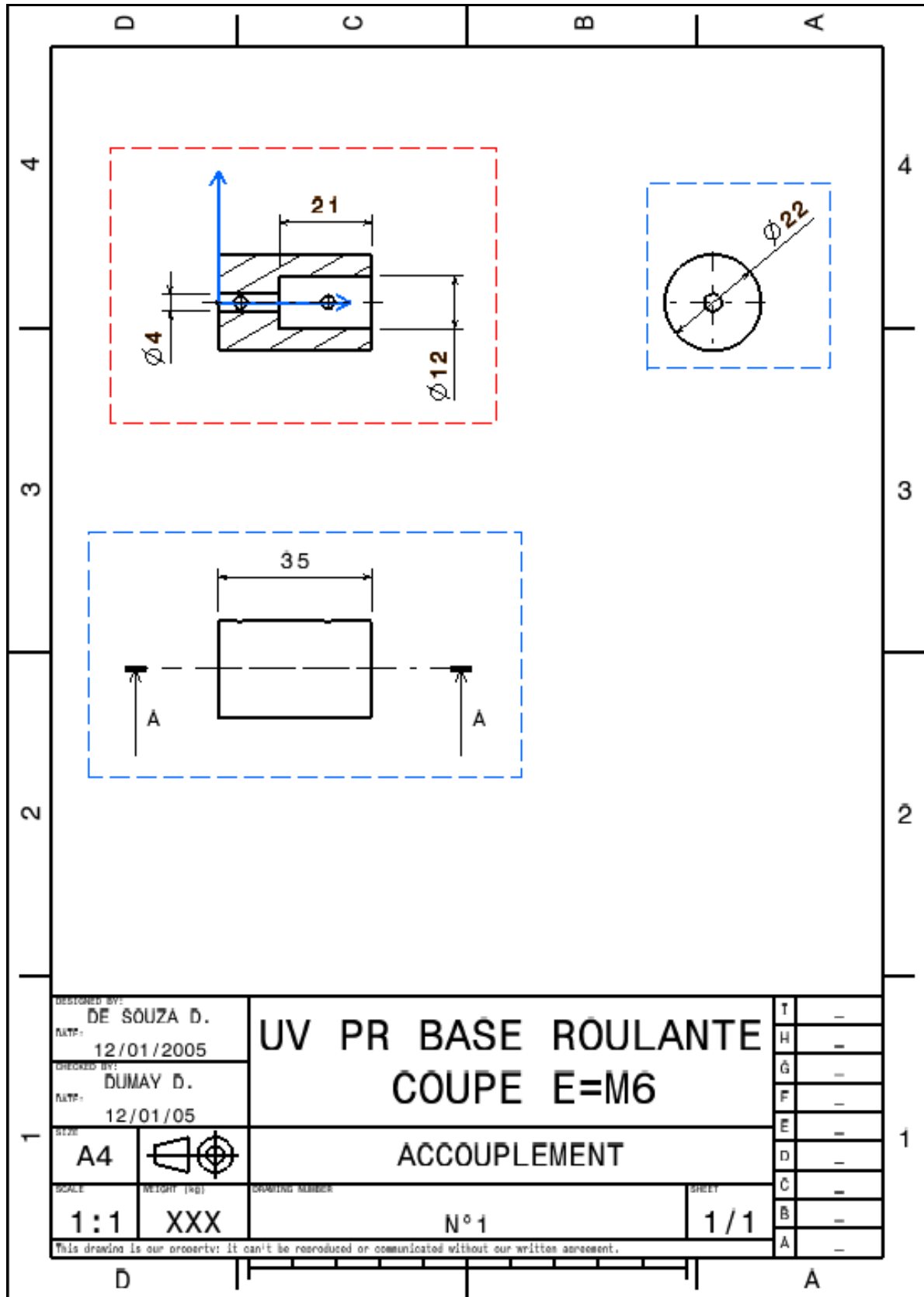
# **ANNEXES**

N°	Nom des équipes	Ecole - Ville	Pts			Poster	Base	Accouplement	Motorisation
N°3	Robotics Venturies	UFR Menton Ville 2	Pts 45	2003	2002	Poster	Base	Accouplement	Motorisation
N°4	Nom des équipes	ENSIC Lille	Pts 6	3		Poster	Base	Accouplement	Motorisation
N°8a	Nom des équipes	ENSIC Lille	Pts 7	2	4	Poster	Base	Accouplement	Motorisation
18a	Les 23 Robots du mois	INSEP Paris	Pts 7	2	4	Poster	Base	Accouplement	Motorisation
18a	ITIP'S	Caen	NH 7			Oui	Rectangle	90°	Arrière
18b	Les Dragons d'air	ENSV Valenciennes	NH 7			Oui	Rectangle	Directe	Arrière
100	Adwin Monta	ILM Montluçon 1	NH 7	5	2	Oui	Rectangle	Directe	Arrière
100	Woua Team 2004	IFI Avin Paris	NH 7	8		Oui	Carré	Courbe	Milieu
107	Les Mouches de France	TMC Evry	NH 7	11		oui	?	?	?
108	Club Robotique de Nantes	Supélec Nantes	NH 7			Oui	Rectangle	Directe	Arrière
108	AM Robotique Paris	ENSA Paris	NH 7	1	3	Oui	Rectangle	Directe	Arrière
140	Osprey Robot	Isia - ENSI Esinsa	NH 7			Oui	Carré	Directe	Milieu
169	Robotique de Roanne	St-Etienne de Roovray	NH 7			Oui	Rectangle	Directe	Arrière
169	Robotique 6	Rueil-Malmaison	NH 7			Oui	Carré	Courbe	Arrière
182	A2 Robotique de Lyon	ISEN Lyon	NH 7			Oui	Rectangle	Directe	Arrière
195	AGLUB Robotique	Lycée de Brest	NH 7			Oui	Rectangle	Directe	Arrière
199	EPA Robotique	ENSA Nancy	NH 7			Oui	Rectangle	Directe	Arrière
106	TCORP Robotique	ENSIMX Caen SERG	NH 7			Oui	Rectangle	Directe	Arrière
185	Robotique de Paris	Centra Paris	NH 7	6	1	Oui	Rectangle	Directe?	Arrière
188	Robotique de Paris	Epita - Ecole Supsa	NH 7			Oui	Rectangle	Directe	Arrière
189	Association des Roboticiens ENSPS	ENSIC Nancy	NH 7			Oui	Rectangle	Directe	Arrière
187	Club de Robotique de Lille	Univ. de Lille	NH 7			Oui	Quelconque	Directe	Milieu
189	Robotique de Strasbourg	INSA de Strasbourg	NH 7			Oui	Rond	Directe	Milieu
656	Montpellier Robotique	CFR Montpellier	NH 7	4		Oui	?	?	?
108	CAIPONS ROU	ENSA de Rouen	NH 7			Oui	Rectangle	Directe	Arrière
122	EPIC Robotique	ENSA de Lille	NH 7			Oui	Rectangle	Embranché	Arrière
107	SUDONDIK	ESM de Lille	NH 7			Oui	Rectangle	Embranché	Arrière
108	Robotique de Lille	ENSA de Lille	NH 7	9		Oui	Rectangle	Chaîne	Arrière
122	Robotique de Lille	Lycée de Lille	NH 7			Oui	Rectangle	Directe	Milieu
108	Robotique de Lille	INSA de Lille	NH 7			Oui	Rectangle	Directe	Arrière
189	Asso. Robotique de Lille	ENSIC de Lille	NH 7			Oui	Rectangle	Directe	Arrière
189	Newton Robot Club	Lycée de Lille	NH 7			Oui	Rectangle	Directe	Arrière
189	La Botanique	URW de Lille	NH 7			Oui	ccq	??	??
116	Club Robotique de Nantes	Polytech de Nantes	NH 7			Oui	Carré	Directe	Arrière
102	Club de Robotique de Nantes	INSA de Nantes	NH 7	64	12	Oui	Carré	Directe	Arrière
127	MOSAIK	Mines de Nantes	NH 7			Oui	Rond	Directe	Milieu
109	Robotique de Nantes	ENSA de Nantes	NH 7			Oui	Carré	Directe	Arrière
188	Centre de Robotique de Nantes	ENSA de Nantes	NH 7			Oui	Carré	Directe	Arrière
108	Alliance CIST-IFMA	CUSIFMA Paris	NH 7	7	9	Oui	Hexa	Directe	Milieu
60	Agence de Nantes	AER de Nantes	NH 7			Oui	Octogone	Directe	milieu
152	ESIC Paris	Supélec de Paris	NH 7			Oui	Rectangle	Directe	Arrière
152	Les Robots	Univ. de Paris	NH 7			Oui	Rectangle	Directe	Arrière
93	ORL Robotique Libre	ENSI de Paris	NH 7			Oui	Rectangle	Directe	arrière

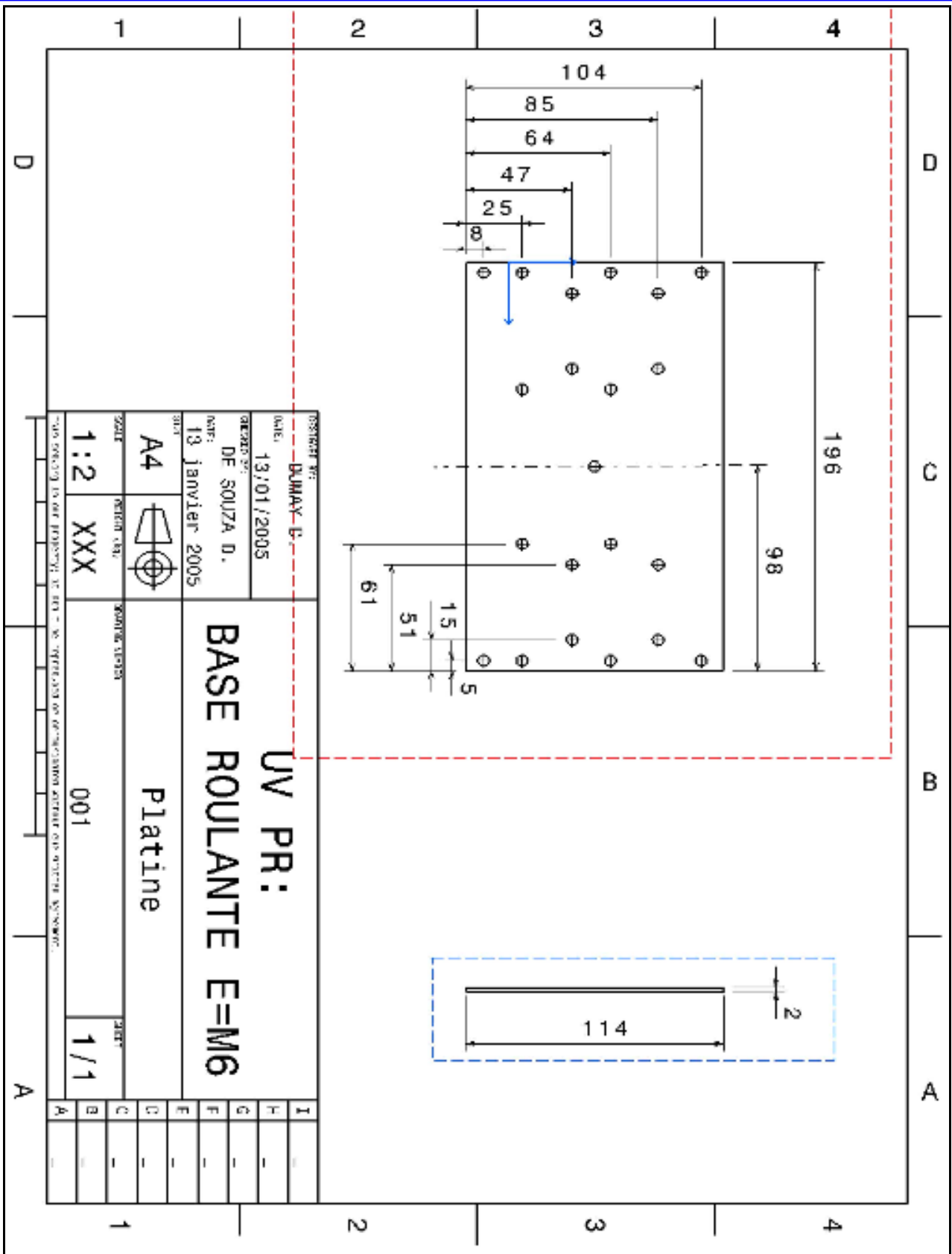
*Veille technologique sur l'aspect mécanique.*



*Dessin de définition des blocs moteurs*

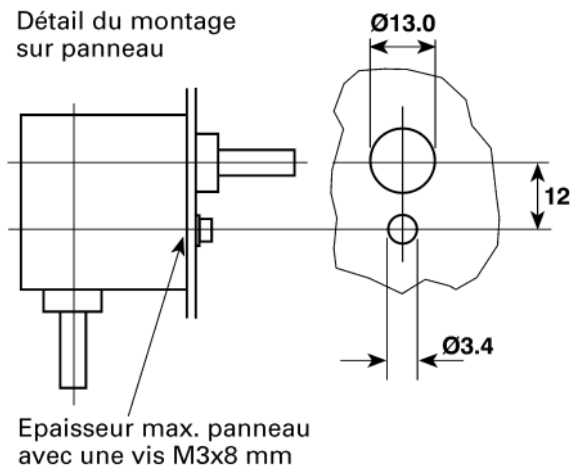
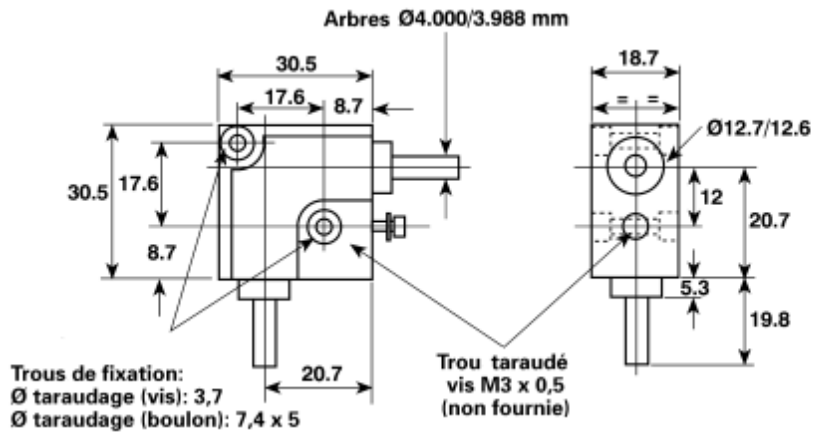


*Dessin de définition de l'accouplement.*



*Dessin de définition de la platine.*





- Les réducteurs miniatures à engrenage conique sont proposés en deux versions pour répondre aux besoins des composants standard avec différents niveaux d'application et d'économie.
- Le type 2 possède des engrenages en acier trempé collés sur des axes en acier meulé et trempé.
- Le type 3 possède des engrenages en acétal homopolymère moulés sur des axes en acier meulé et trempé.
- Tous les modèles possèdent un boîtier moulé par injection avec deux trous de passage lamés pour le montage sur châssis conventionnel et une douille taraudée au-dessous de chaque axe pour le montage vertical.
- Dans ce plan, le boîtier de la portée évite les pivotements sous l'effet de la charge.
- Tous les modèles ont un rapport de 1:1.
- Sur le type 2, les axes sont isolés électriquement par rapport au boîtier.
- Le type 3 possède une isolation électrique entre les axes d'entrée et de sortie et entre les axes et le boîtier.
- La boîte d'engrenage et les portées sont lubrifiées définitivement sur les deux versions. Toutes deux conviennent pour les transmissions manuelles et à court terme.

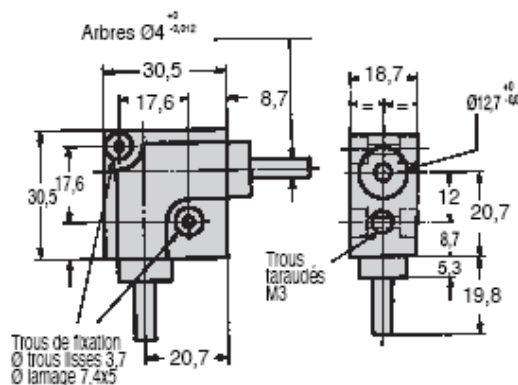
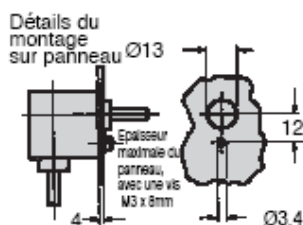
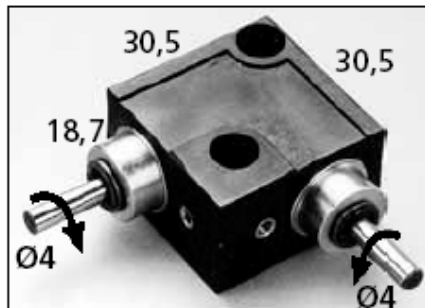
## Les réducteurs de vitesse



### Renvoi d'angle miniature

**B332**

- Très léger
- Jeu en sortie 1°
- Vitesse maxi. en entrée: 3000 t/min
- Matières:  
boîtier: plastique  
engrenages: Type 1 = acier  
Type 2 = plastique
- Masse: 37g à 41g



**COUPLE DE SORTIE (Nm)**

Rapports	1:1	1:1
Vitesse d'entrée	Type 1 Acier	Type 2 Plastique
1500 t/min	0,68	0,11

**REMISES**

Qté	1+	5+	10+	25+	50+
Rem. Prix	-6%	-12%	-18%	Sur demande	

REFERENCE	Type	Rapport	Rendement à 1500 t/min	Vitesse d'entrée maxi. (t/min)	Stock	Prix Uni. 1 à 4
B332.31.2	1	1:1	94%	3000	✓	48,38 €
B332.31.3	2	1:1	94%	1500	✓	28,12 €

\*Dans la limite du disponible

Fax: **0 325 88 6000** www.hpceurope.com



catalogue 2004 **203**

*Documentations du réducteur à renvoi d'angle.*

**Motoréducteur AMAX32GHP/PLG42S**

0,15 Nm

14 Nm

MDP

**CARACTÉRISTIQUES**

**24V/0016**

1. Rapport de réduction		16
2. Nombre d'étages		2
3. Rendement		0.75
4. Vitesse à vide	tr/mn	403
5. Vitesse en charge	tr/mn	309
6. Couple nominal	Nm	0.57
7. Courant nominal	A	1.4

Modification de la tension d'alimentation



- + MOTEUR
- + COURBE
- + GÉNÉRALITÉS
- + PLANS

+ ACCESSOIRES & OPTIONS

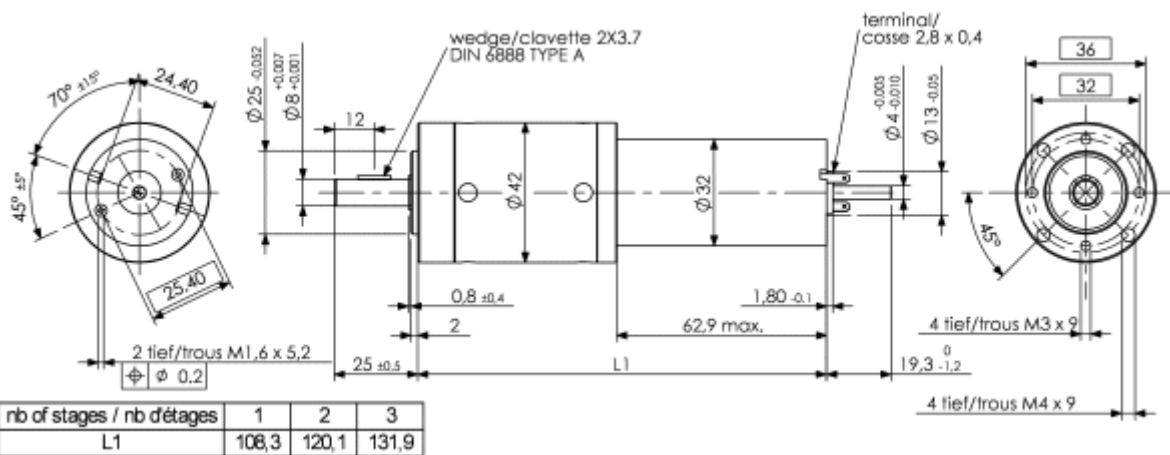
**CARACTÉRISTIQUES**

**Motoréducteur AMAX32GHP/PLG42S**

0,15 Nm

14 Nm

MDP





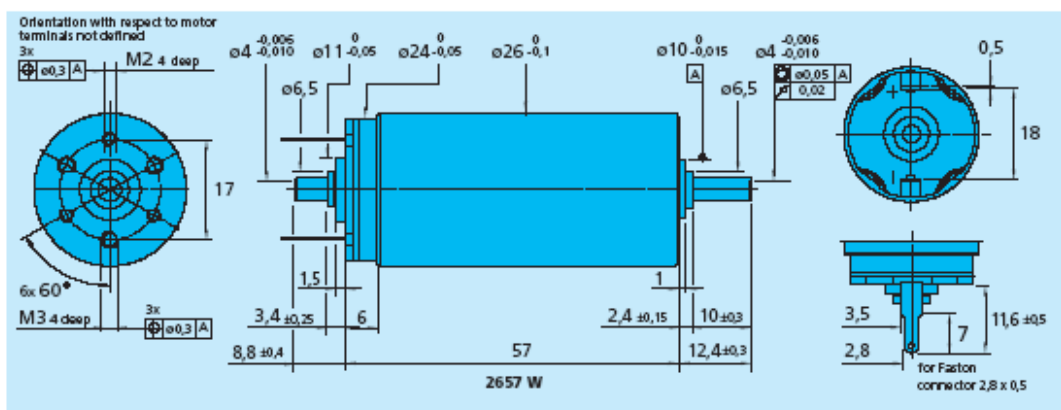
**DC-Micromotors**  
Graphite Commutation

**44 mNm**

For combination with  
Gearheads:  
26/1, 30/1  
Encoders:  
IE2, 5500, 5540

**Series 2657 ... CR**

	2657 W	012 CR	024 CR	048 CR	
1 Nominal voltage	$U_N$	12	24	48	Volt
2 Terminal resistance	R	0,71	2,84	12,50	$\Omega$
3 Output power	$P_2 \text{ max.}$	45,9	47,9	44,5	W
4 Efficiency	$\eta \text{ max.}$	84	85	84	%
5 No-load speed	$n_0$	6 300	6 400	6 400	rpm
6 No-load current (with shaft $\varnothing 4,0$ mm)	$I_0$	0,115	0,058	0,028	A
7 Stall torque	$M_H$	278	286	265	mNm
8 Friction torque	$M_f$	2	2	2	mNm
9 Speed constant	$k_n$	552	274	136	rpm/V
10 Back-EMF constant	$k_E$	1,81	3,65	7,37	mV/rpm
11 Torque constant	$k_M$	17,3	34,8	70,4	mNm/A
12 Current constant	$k_I$	0,058	0,029	0,014	A/mNm
13 Slope of n-M curve	$\Delta n/\Delta M$	22,7	22,4	24,2	rpm/mNm
14 Rotor Inductance	L	95	380	1 550	$\mu\text{H}$
15 Mechanical time constant	$\tau_m$	3,9	3,9	3,9	ms
16 Rotor Inertia	J	16	17	15	$\text{gcm}^2$
17 Angular acceleration	$\alpha \text{ max.}$	170	170	170	$\cdot 10^3 \text{rad/s}^2$
18 Thermal resistance	$R_{th 1} / R_{th 2}$	1,9 / 9			K/W
19 Thermal time constant	$\tau_{w1} / \tau_{w2}$	10 / 580			s
20 Operating temperature range:					
- motor		-30 ... +125			$^{\circ}\text{C}$
- rotor, max. permissible		+155			$^{\circ}\text{C}$
21 Shaft bearings		ball bearings, preloaded			
22 Shaft load max.:					
- with shaft diameter		4,0			mm
- radial at 3000 rpm (3 mm from bearing)		20			N
- axial at 3000 rpm		2			N
- axial at standstill		20			N
23 Shaft play:					
- radial	$s_r$	0,015			mm
- axial	$s_a$	0			mm
24 Housing material		steel, black coated			
25 Weight		156			g
26 Direction of rotation		clockwise, viewed from the front face			
<b>Recommended values</b>					
27 Speed up to	$n_{\text{re max.}}$	6 000	6 000	6 000	rpm
28 Torque up to	$M_{\text{re max.}}$	44	44	44	mNm
29 Current up to (thermal limits)	$I_{\text{re max.}}$	3,10	1,54	0,73	A





## Planetary Gearheads

3,5 Nm

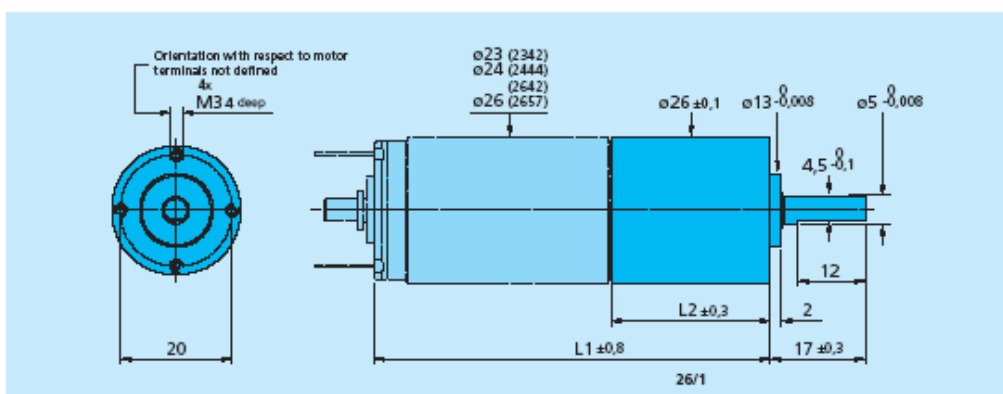
For combination with  
DC-Micromotors:  
2342, 2642, 2657  
Brushless DC-Servomotors:  
2444

### Series 26/1

	26/1
Housing material	Inox steel
Geartrain material	steel <sup>1)</sup>
Recommended max. Input speed for:	
- for continuous operation	4000 rpm
Backlash, typical, at no-load	≤ 1°
Bearings on output shaft	preloaded ball bearings
Shaft load, max.	
- radial (10 mm from mounting face)	≤ 150 N
- axial	≤ 100 N
Shaft press fit force, max.	≤ 150 N
Shaft play (on bearing output):	
- radial	≤ 0,015 mm
- axial	≤ 0,10 mm
Operating temperature range	-30 ... + 100° C

reduction ratio (nominal)	weight without motor g	length without motor L2 mm	length with motor			output torque		direction of rotation (reversible)	efficiency %
			2444 S	2642 W	2657 W	continuous operation	intermittent operation		
			L1 mm	L1 mm	L1 mm	M max. Nm	M max. Nm		
3,71 :1	93	28,4	72,4	70,4	85,4	1,1	2,3	-	88
14 :1	116	36,4	80,4	78,4	93,4	0,3 (3,5)	0,4 (4,5)	-	80
43 :1	139	44,4	88,4	86,4	101,4	1,0 (3,5)	1,2 (4,5)	-	70
66 :1	139	44,4	88,4	86,4	101,4	1,5 (3,5)	1,8 (4,5)	-	70
134 :1	162	52,5	96,4	94,5	109,5	2,5 (3,5)	3,5 (4,5)	-	60
159 :1	162	52,5	96,4	94,5	109,5	3,5 (3,5)	4,5 (4,5)	-	60
246 :1	162	52,5	96,4	94,5	109,5	3,5 (3,5)	4,5 (4,5)	-	60
415 :1	185	60,5	104,4	102,5	117,5	3,5 (3,5)	4,5 (4,5)	-	55
592 :1	185	60,5	104,4	102,5	117,5	3,5 (3,5)	4,5 (4,5)	-	55
989 :1	185	60,5	104,4	102,5	117,5	3,5 (3,5)	4,5 (4,5)	-	55
1 526 :1	185	60,5	104,4	102,5	117,5	3,5 (3,5)	4,5 (4,5)	-	55

<sup>1)</sup> Gearheads with ratio ≥ 14:1 have plastic gears in the input stage. The values for the torque rating indicated in parenthesis, are for gearheads, type 26/1 S with all steel gears.  
For extended life performance, the gearheads are available with all steel gears and heavy duty lubricant as type 26/1 S.



*Documentations des différents motoréducteurs non retenus.*

**Motoréducteur RE035G/PLG52**

0,3 Nm

24 Nm

MDP

**CARACTÉRISTIQUES**

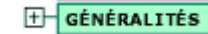
**24V/0006**

1. Rapport de réduction		6.3
2. Nombre d'étages		1
3. Rendement		0.85
4. Vitesse à vide	tr/mn	688
5. Vitesse en charge	tr/mn	559
6. Couple nominal	Nm	0.60
7. Courant nominal	A	2.1

Modification de la tension d'alimentation



COURBE

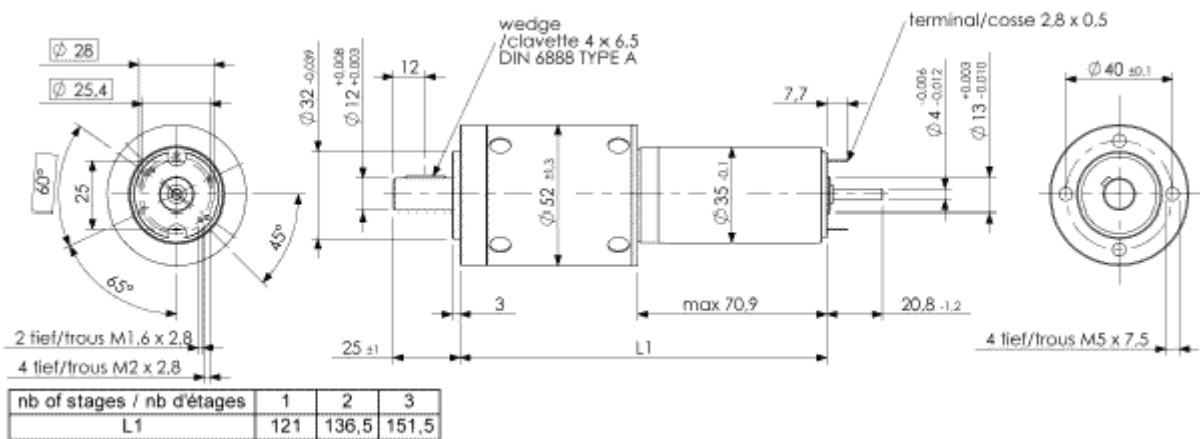


**Motoréducteur RE035G/PLG52**

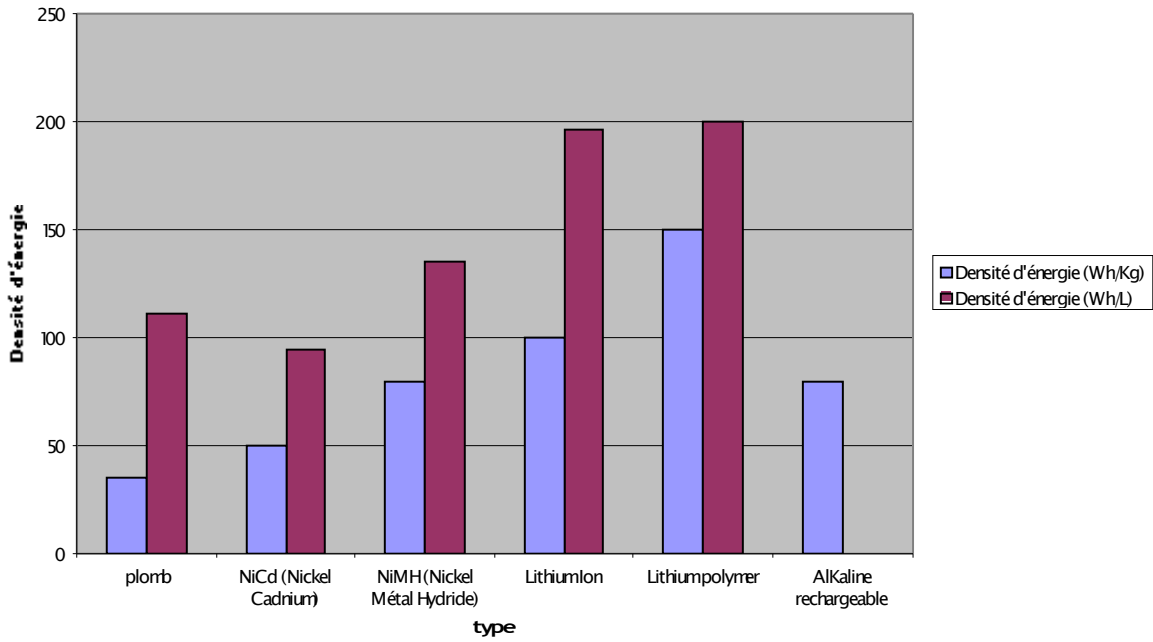
0,3 Nm

24 Nm

MDP







*Densité énergétique de différents types de batteries :*



### HCTL 1100

**Features**

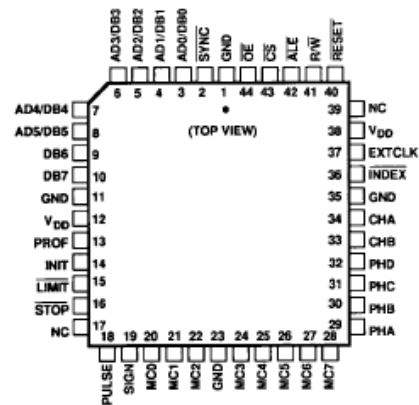
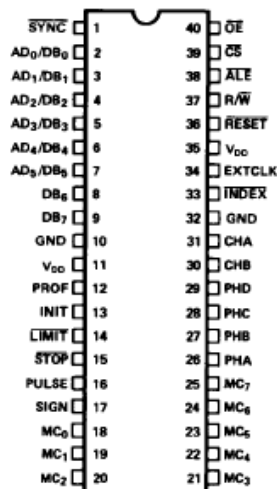
- Low Power CMOS
- PDIP and PLCC Versions Available
- Enhanced Version of the HCTL-1000
- DC, DC Brushless, and Step Motor Control
- Position and Velocity Control
- Programmable Digital Filter and Commutator
- 8-Bit Parallel, and PWM Motor Command Ports
- TTL Compatible
- SYNC Pin for Coordinating Multiple HCTL-1100 ICs
- 100 kHz to 2 MHz Operation
- Encoder Input Port

**Description**

The HCTL-1100 series is a high performance, general purpose motion control IC, fabricated in HP CMOS technology. It frees the host processor for other tasks by performing all the time-intensive functions of digital motion control. The programmability of all control parameters provides maximum flexibility and quick

design of control systems with a minimum number of components. In addition to the HCTL-1100, the complete control system consists of a host processor to specify commands, an amplifier, and a motor with an incremental encoder (such as the HP HEDS-5XXX, -6XXX, -9XXX series). No analog compensation or velocity feedback is necessary.

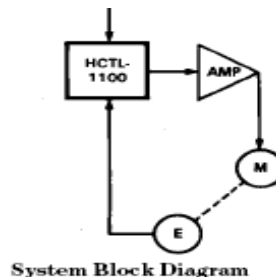
**Pinouts**



medical instruments, material handling machines, and industrial automation.

**HCTL-1100 vs. HCTL-1000**

The HCTL-1100 is designed to replace the HCTL-1000. Some differences exist, and some enhancements have been added.



**Comparison of HCTL-1100 and HCTL-1000**

Description	HCTL-1100	HCTL-1000
Max. Supply Current	30 mA	180 mA
Max. Power Dissipation	165 mW	950 mW
Max. Tri-State Output Leakage Current	150 nA	10 µA
Operating Frequency	100 kHz-2 MHz	1 MHz-2 MHz
Operating Temperature Range	-20°C to +85°C	0°C to 70°C
Storage Temperature Range	-55°C to +125°C	-40°C to +125°C
Synchronize 2 or More ICs	Yes	-
Preset Actual Position Registers	Yes	-
Read Flag Register	Yes	-

## LM628/LM629

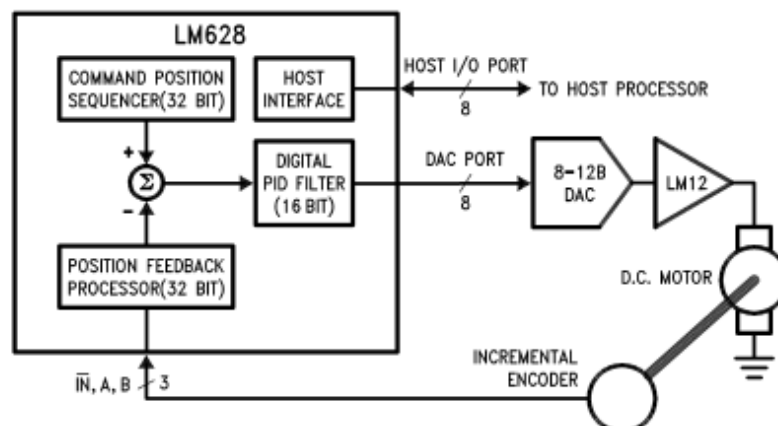
### Precision Motion Controller

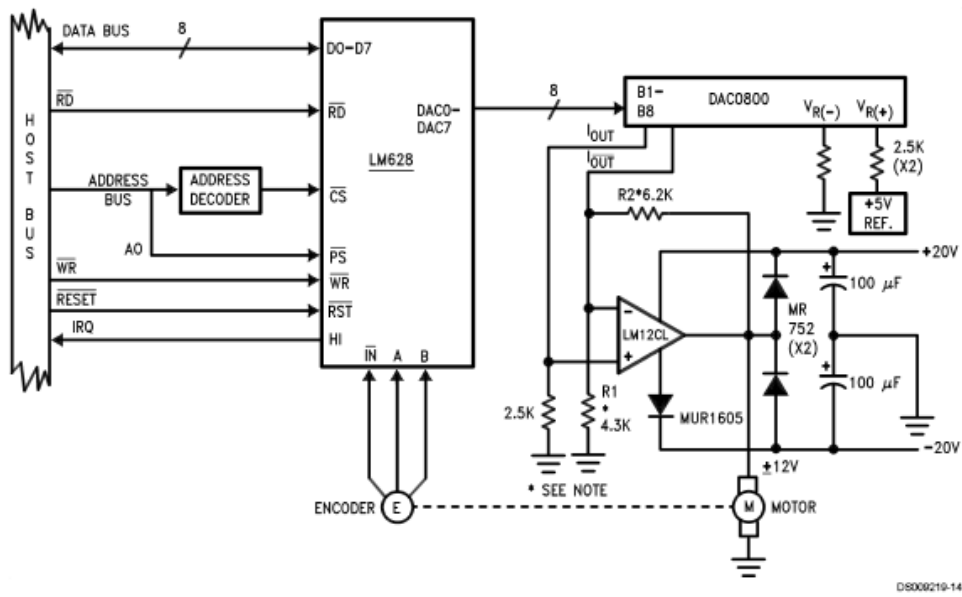
#### General Description

The LM628/LM629 are dedicated motion-control processors designed for use with a variety of DC and brushless DC servo motors, and other servomechanisms which provide a quadrature incremental position feedback signal. The parts perform the intensive, real-time computational tasks required for high performance digital motion control. The host control software interface is facilitated by a high-level command set. The LM628 has an 8-bit output which can drive either an 8-bit or a 12-bit DAC. The components required to build a servo system are reduced to the DC motor/actuator, an incremental encoder, a DAC, a power amplifier, and the LM628. An LM629-based system is similar, except that it provides an 8-bit PWM output for directly driving H-switches. The parts are fabricated in NMOS and packaged in a 28-pin dual in-line package or a 24-pin surface mount package (LM629 only). Both 6 MHz and 8 MHz maximum frequency versions are available with the suffixes -6 and -8, respectively, used to designate the versions. They incorporate an SDA core processor and cells designed by SDA.

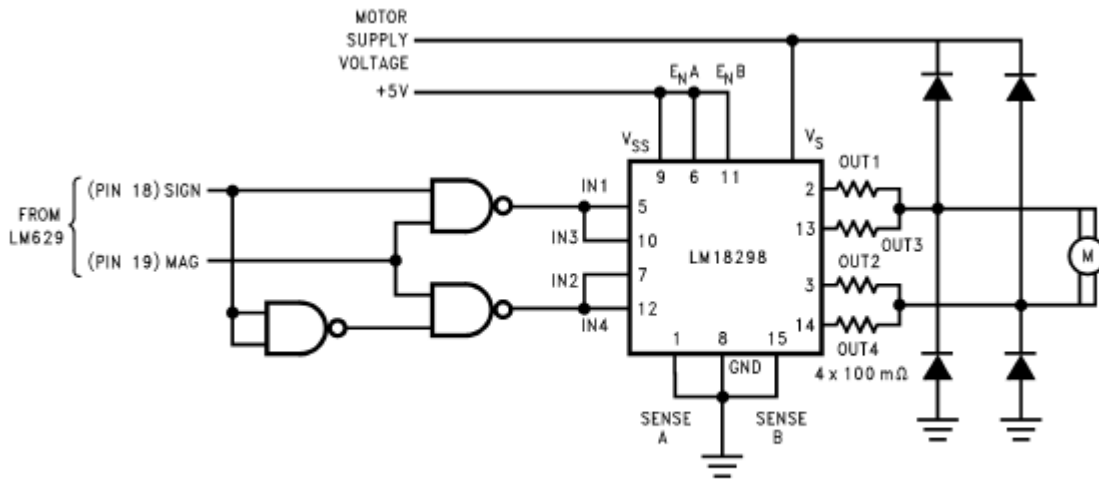
#### Features

- 32-bit position, velocity, and acceleration registers
- Programmable digital PID filter with 16-bit coefficients
- Programmable derivative sampling interval
- 8- or 12-bit DAC output data (LM628)
- 8-bit sign-magnitude PWM output data (LM629)
- Internal trapezoidal velocity profile generator
- Velocity, target position, and filter parameters may be changed during motion
- Position and velocity modes of operation
- Real-time programmable host interrupts
- 8-bit parallel asynchronous host interface
- Quadrature incremental encoder interface with index pulse input
- Available in a 28-pin dual in-line package or a 24-pin surface mount package (LM629 only)





DS000219-14



DS000219-18

FIGURE 16. PWM Drive for Brush/Commutator Motors



**Code en C du programme de la carte d'asservissement numérique artisanale (logiciel utilisé C2C) :**

```
#pragma CLOCK_FREQ 2000000

#include "lcd.h"

// Déclaration des variables globales
int codeur1,codeur2,mcodeur1,mcodeur2,pwm1,pwm2;

void affiche()
{
  num_int(pwm1);
  phrase("_____");
  num_int(pwm2);
}

/*****
*      Fonction principale          *
* - Initialise le PIC              *
* - attente touches pour réglage pwm *
*****/
void main()
{
  //intialistaion du PIC

  pc = 0;

  bank1(); //banque 1
  trisa = 0x30; //afficheur/clavier
  trisb=0xFF; //Port B = lecture compteur
  trisc=00000000b;
  trisd=0xFF; //Port D = lecture compteur
  trise = 0x04; //RE0/1 en sortie et RE2 en entrée

  ADCON1=00000110b; //tous les ports en numérique (conversion A/D désactivée)
  OPTION_REG=00000011b; //prescaler TMR0 au 1/16ème
  INTCON = 0x00;

  PR2=0x64; //valeur de la période TMR2

  bank0(); //banque 0

  T2CON=00000000b; //Prescalaire du timer 2 à 1 soit avec PR2=0x64 => fréquence de
  la PWM ~49.5KHz donc zone inaudible
  CCP1CON=00101100b;
  CCP2CON=00101100b; //Mode PWM

```

```
CCPR1L=0x32;      //PWM à 50% résolution 10bits donc 512
CCPR2L=0x32;

set_bit(T2CON,2); //démarrer timer2

pb=0;
pd=0;

LCDinit();
delay_ms(100);
phrase(" initialisation");
delay_s(1);

pc = 0xC0;

//initialisation des variables

char vrep=0x32,k;

codeur1 = 0;
codeur2 = 0;
mcodeur1 = 0;
mcodeur2 = 0;

pwm1 = vrep;
pwm2 = vrep;
CCPR1L = vrep;
CCPR2L = vrep;

// début
set_bit(INTCON,5); //activation interruption timer0
enable_interrupt(GIE);

while(1)
{

k = touche();
if (k != 0)
{
putcmd(0x01);
delay_ms(100);
switch (k)
{
case 1 : pwm1++;
CCPR1L = pwm1; affiche();
break;
case 2 : pwm1--;
CCPR1L = pwm1; affiche();
```

```

        break;
    case 3 : pwm2++;
        CCPR2L = pwm2; affiche();
        break;
    case 4 : pwm2--;
        CCPR2L = pwm2; affiche();
        break;
    //case 5 :
    case 6 : CCPR1L = vrep; CCPR2L = vrep;

    }

    }
    putcmd(0xC0);
    num_int(mcodeur1);
    phrase("_____");
    num_int(mcodeur2);
}
}

/*****
*      Fonction d'interruption      *
* - Vérifie l'interruption qui l'a appelée *
* - Exécute le code correspondant   :   *
*   => Ici lit les compteurs.       *
*****/
void interrupt()
{
    if(INTCON & 4) //interruption de TMR0
    {
        codeur1 = pb;
        codeur2 = pd;

        pc = pc | 0x30; //reset des compteurs
        delay_us(16);
        pc = pc & 0xCF;

        mcodeur1 = (mcodeur1 + codeur1) / 2; //moyenne des codeurs
        mcodeur2 = (mcodeur2 + codeur2) / 2;

        clear_bit(INTCON,2); //efface le flag d'interruption
    }
}

```

**bibliothèque lcd.h :**

```

/*-----*/
/* Mode : HD44780 type LCD */
/* */
/* Interface : SEL = Port A bit - RE01 */
/* RS = Port A bit - RE00 */
/* Data_7 = Port A bit - RA03 */
/* Data_6 = Port A bit - RA02 */
/* Data_5 = Port A bit - RA01 */
/* Data_4 = Port A bit - RA00 */
/* */
/* */
/* This version is for 4 bit mode */
/* */
/* */
/* */
/* */
/*****/

```

```

char pa@0x05;
char pb@0x06;
char pc@0x07;
char pd@0x08;
char pe@0x09;

```

```

char trisa@0x85;
char trisb@0x86;
char trisc@0x87;
char trisd@0x88;
char trise@0x89;

```

```

char ADCON1@0x9F;
char ADCON0@0x1F;
char PR2@0x92;
char CCPR1L@0x15;
char CCPR2L@0x1B;
char CCP1CON@0x17;
char CCP2CON@0x1D;
char T2CON@0x12;

```

```

void bank0(void);
void bank1(void);
void bank2(void);
void bank3(void);
char touche(void);
void putdata(char a);
void putcmd(char a);

```



```
void putchar(char a);
void LCDinit(void);
void phrase( const char *lcdptr );
void num_int(int num);
```

```
void bank0()
{
    clear_bit(STATUS,RP0);
    clear_bit(STATUS,RP1);
}
```

```
void bank1()
{
    set_bit(STATUS,RP0);
    clear_bit(STATUS,RP1);
}
```

```
void bank2()
{
    clear_bit(STATUS,RP0);
    set_bit(STATUS,RP1);
}
```

```
void bank3()
{
    set_bit(STATUS,RP0);
    set_bit(STATUS,RP1);
}
```

```
char touche()
{
    char a;

    a = pa & 0x30;
    a >>= 4;
    a = a + (pe & 0x04);

    return a;
}
```

```
void putdata(char a)
{
    char x,b;
    delay_ms(4);
    x = a;
    asm swapf _x_putdata,1
    output_port_a(pa & 0xF0);
    output_port_a(pa | (x & 0x0F));
}
```

```
b = pe | 0x02;
pe = b;
asm nop
b = pe & 0xFD;
pe = b;
```

```
output_port_a(pa & 0xF0);
output_port_a(pa | (a & 0x0F));
```

```
b = pe | 0x02;
pe = b;
asm nop
b = pe & 0xFD;
pe = b;
}
```

```
void putcmd(char a)
{
    char b;
    b = pe & 0xFE;
    pe = b;

    putdata(a);
}
```

```
void putchar(char a)
{
    char b;
    b = pe | 0x01;
    pe = b;

    putdata(a);
}
```

```
void LCDinit(void)
{
    delay_ms(20);
    putcmd(0x28);
    delay_ms(15);
    putcmd(0x28);
    putcmd(0x28);
    putcmd(0x28);
    putcmd(0x28);
    putcmd(0x28);
    putcmd(0x28);
    putcmd(0x0C);
    putcmd(0x06);
    putcmd(0x01);
}
```

```
}  
  
void phrase( const char *lcdptr )  
{  
    char pi;  
  
    pi = 0;  
    while( lcdptr[pi] != 0 )  
    {  
        putchar( lcdptr[pi++] );  
    }  
  
}  
  
void num_int(int num)  
{  
    putchar( '0' + (num / 10000));  
    putchar( '0' + (num / 1000) % 10 );  
    putchar( '0' + (num / 100) % 10);  
    putchar( '0' + (num / 10) % 10 );  
    putchar( '0' + num % 10 );  
}
```

## **MOTOROLA HCS12 (MC9S12DP256B)**

- 16-bit
- 256 Kb Flash EEPROM
- 12 Kb RAM
- 4 Kb EEPROM
- 2 asynchronous serial communications interfaces (SCI)
- 3 serial peripheral interfaces (SPI)
- 8-channel IC/OC enhanced capture timer
- 2 8-channel
- 10-bit analog-to-digital converters (ADC)
- 8-channel pulse-width modulator (PWM)
- 89 discrete digital I/O channels (Port A, Port B, Port K and Port E)
- 20 discrete digital I/O lines with interrupt and wakeup capability
- 5 CAN 2.0
- 1 I<sup>2</sup>C Bus

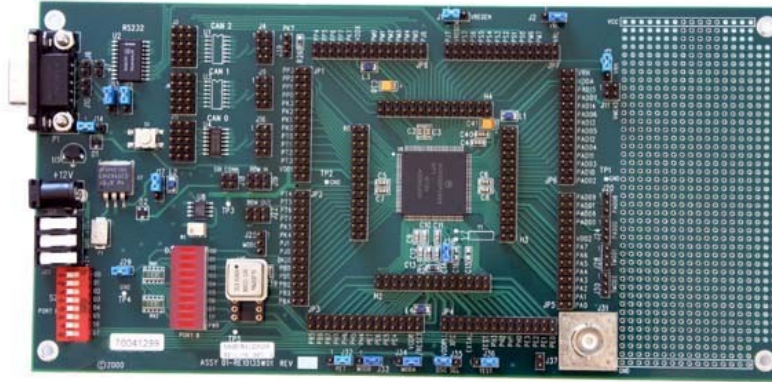
2 cartes sont envisageables pour ce microcontrôleur.

La T-Board de Elektronik Laden coûte moins de 200 € mais l'interface de programmation (ComPod) est en option. Cette dernière permet de charger un programme rapidement et de le débogger facilement.

Cette carte présente l'intérêt d'avoir beaucoup d'entrées-sorties tout en étant très petite.



La deuxième carte possible est le kit de Motorola (M68KIT912DP256). Elle coûte environ 500 € mais contient tout le nécessaire pour programmer et débogger facilement. Elle contient également une zone de prototypage. Outre ses fonctionnalités, sa plus grande particularité est sa taille.



### **MICROCHIP PIC 18F458**

- 10 MIPS
- 8-bit
- 40 MHz Max. Speed
- Program Memory Size (bytes) : 32768
- RAM Size (bytes) : 1536
- Data EEPROM Size (bytes) : 256
- I/O pins : 34
- 5 PWM 10-Bit
- PSP
- 1 CAN 2.0
- 1 I<sup>2</sup>C
- ICD
- Self-Programming

Ce PIC de nouvelle génération peut être programmé en C avec le compilateur PIC18. Il comporte des bus I<sup>2</sup>C et CAN mais est limité en nombre de pattes donc de fonctions utilisables en même temps.

Il n'existe pas vraiment de carte commerciale de petite taille et ambivalente pour les PICs.

Les PICs sont peu onéreux voir gratuits s'ils sont commandé en "sample". C'est donc un microcontrôleur de choix pour les petites tâches. C'est pourquoi, il peut être intéressant de disposer d'une carte de développement. La PICdem2plus comporte des périphériques comme un afficheur LCD ou des boutons, elle dispose d'une zone de prototypage et elle permet d'utiliser des 16Fxxx comme des 18Fxxx.



### **RABBIT (RCM3400)**

- Small size: 1.16" × 1.37" × 0.31" (29 mm × 34 mm × 8 mm)
- Microprocessor: Rabbit 3000 running at 29.4 MHz
- 47 parallel 5 V tolerant I/O lines: 41 configurable for I/O, 3 fixed inputs, 3 fixed outputs
- Two additional digital inputs, one additional digital output
- Eight single-ended or four differential analog inputs
- One additional analog input
- External reset input
- Alternate I/O bus can be configured for 8 data lines and 6 address lines (shared with parallel I/O lines), I/O read/write
- Ten 8-bit timers (six cascadable) and one 10-bit timer with two match registers
- 512K flash memory, 512K SRAM, (options for 256K flash memory and 256K SRAM)
- 10-bit free-running PWM counter and four width registers
- Two-channel Input Capture can be used to time input signals from various port pins
- Two-channel Quadrature Decoder accepts inputs from external incremental encoder modules
- Five CMOS-compatible serial ports: maximum asynchronous baud rate of 5.5 Mbps. Four ports are configurable as a clocked serial port (SPI), and two ports are configurable as SDLC/HDLC serial ports.
- Supports 1.15 Mbps IrDA transceiver

Le rabbit est très simple d'utilisation mais difficile à intégrer. En effet, il est livré avec une carte de développement assez encombrante. Pour l'utiliser en situation, il faudrait l'intégrer à une carte développée en interne.

### **PROTEE**

- 20 MIPS
- beaucoup d'I/O
- CANs
- PWMs
- langage : Forth

Un membre de l'association (Joseph Pinkasfeld) a proposé d'utiliser une carte de petite série développée au sein de la société Protee. L'inconvénient principal de cette carte est l'apprentissage du langage qui lui est associé.

### **PC104**

Cette norme décrit une architecture sous forme de modules à empiler. De tels modules sont disponibles au sein de l'association. Malheureusement, la documentation est introuvable. De plus, cet ensemble est assez encombrant.

### **HITACHI SH-1**

Des cartes SH-1 sont disponibles dans l'association. Elles ont déjà été utilisées dans les années précédentes puis abandonnées au profit des PICs.

Leur puissance de calcul paraît satisfaisante.

Il semble qu'elles ne contiennent pas de ROM (dans leur état actuel). Il fallait donc recharger le programme à chaque redémarrage.

**LIENS**

Microcontrôleur HCS12 :

[http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=MC9S12DP256B&nod](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC9S12DP256B&nod)

Carte T-Board : [http://www.elektronikladen.de/en\\_hcs12tb.html](http://www.elektronikladen.de/en_hcs12tb.html)

Carte Motorola kit :

[http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=M68KIT912DP256&parentCode=MC9S12DP256B&nodeId=0162468636bJwn](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=M68KIT912DP256&parentCode=MC9S12DP256B&nodeId=0162468636bJwn)