

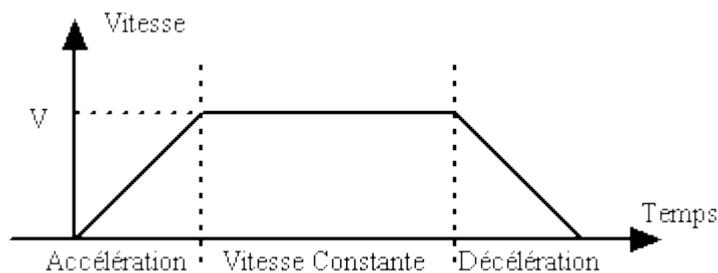
Rapport sur les cartes électroniques utilisées à la Coupe 2005

Le but est d'expliquer le fonctionnement des cartes que l'on a effectivement utilisées. Je me base sur le rapport de la PR de A04 et du rapport de Guillaume « Carte d'asservissement numérique finale avec odométrie ». Je les reprendrai au maximum pour afin de ne pas s'égarer dans la documentation, mais vous pouvez vous y reporter.

Je vais donc présenter les problèmes rencontrés, leurs solutions et les améliorations à envisager.

On étudiera tout d'abord la « carte mère » (avec le pic et les deux LM629, créée pour la PR), la carte de puissance (les deux LMD18200T), et enfin la carte de commande faite par Guillaume.

Le principe global utilisé est : une carte mère qui gère tout (les LM629, les différents capteurs (en fait juste le capteur infrarouge qui détecte le pont), et le contrôle des servomoteurs (par le biais d'un SD20 maison)). La carte mère donne les ordres de position, vitesse, accélération au LM629 (un par moteur). Ce dernier s'occupe du retour du codeur du moteur et génère une PWM afin d'avoir une trajectoire trapézoïdale :



Profil de vitesse trapézoïdal

Le moteur est directement commandé par un LMD18200T qui prend en entrée la PWM du LM629 et un bit de signe.

I/ La carte mère

Je reprends la description du rapport de la PR :

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.

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.

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

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.

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

\overline{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.

La carte est alimentée en 5v directement par la carte des TRACO. On y a rajouté pleins d'entrées/sorties (pour la liaison avec les autres cartes), comme on peut le voir sur la photo :

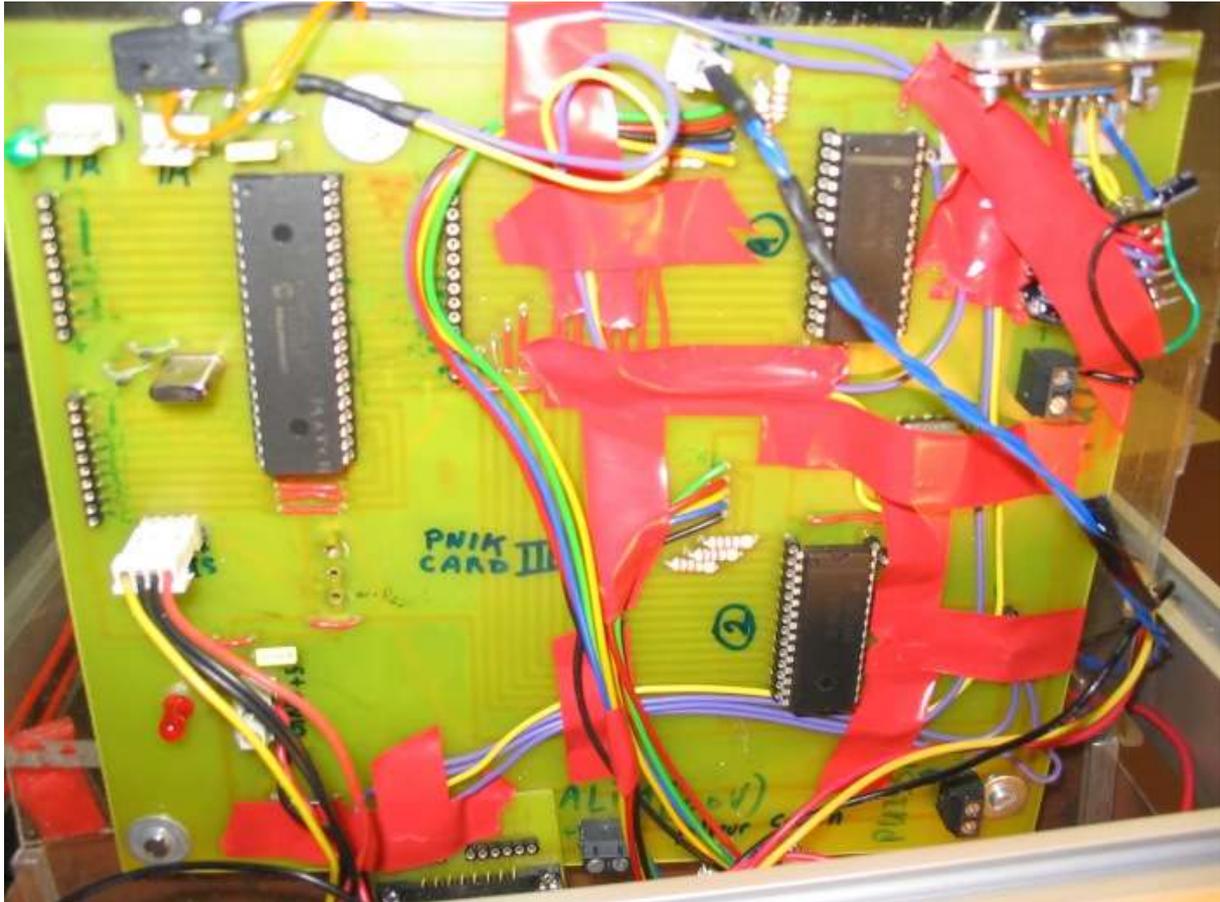


Photo 1

Description :

- _ en haut : le switch sert de reset du pic et des deux LM629
- _ en haut à gauche : une led branchée sur la patte A0 pour test
- _ juste en dessous : deux connecteurs pour les capteurs infrarouges (ils n'ont pas été utilisés à cause de forte interférences avec les moteurs)
- _ en dessous : le pic : 18F458 (par manque de place sur un 16F877, à cause du menu : trop de caractères) et son oscillateur
- _ encore en dessous : liaison et alimentation de la carte SD20
- _ juste en dessous : l'alimentation de la carte en provenance de la carte des TRACO et la led de contrôle de mise sous tension
- _ tout en bas : la prise DB-15 : liaison avec la carte puissance

_ en haut, au milieu de la carte : le fil bleu, c'est le contrôle du démarrage avec le jack
_ en haut à droite : la prise DB-9 pour liaison série avec le pc, avec le MAX232 sous le scotch, relié au pic par les ports RX et TX

_ en bas au milieu (juste visible sous les noms palpeur et canon à droite de l'ancienne alimentation de la carte) : liaison avec le capteur infrarouge et à sa droite, le contrôle du canon.

_ enfin on voit les deux LM629 et le retour des codeurs
_ sur la gauche de la carte, le branchement du LCD, qui a été utilisé pour la PR.

On n'a eu aucun problème avec la carte, malgré le fouillis de fils. On a eu de la chance, il faudrait faire à l'avenir des cartes un peu plus propre (on devait passer sur l'HCS12 et la carte de commande que je vais décrire ci-dessous, mais on n'a pas eu le temps).

II/ La carte de puissance

Je reprends la description du rapport de la PR :

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 μC 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°C ainsi que les bits de direction, brake et PWM.

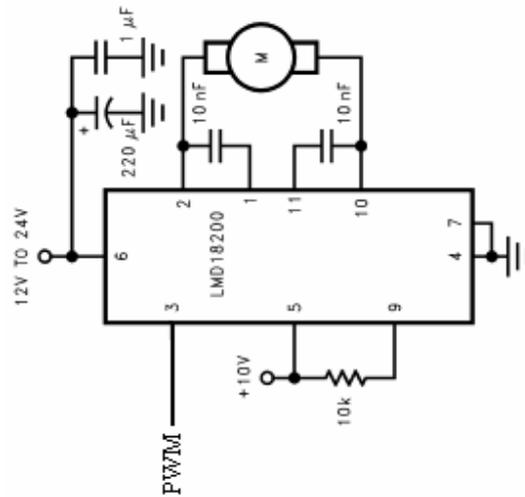
Il est également protégé contre une surchauffe avec une « *thermal shutdown* » à 170°C .

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 :



Pour la commande, on a utilisé la première méthode avec le bit de signe.



photo 2

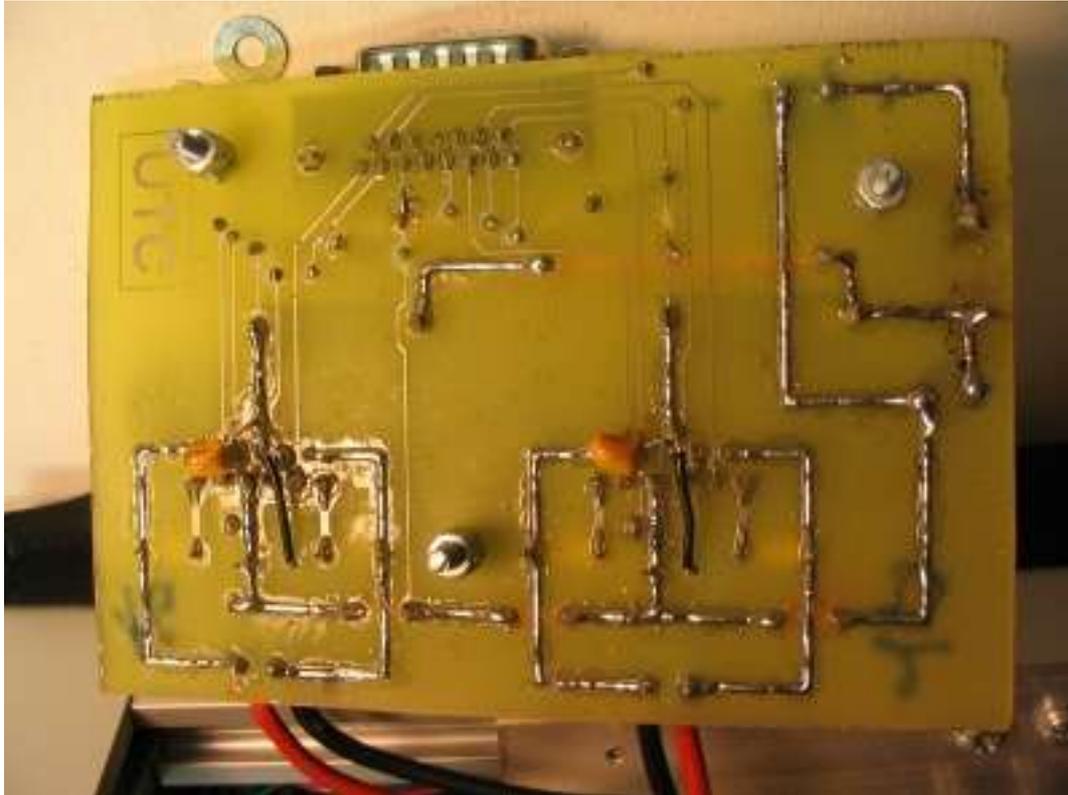


photo 3

Pendant deux mois, on a eu de sérieux problèmes avec les LMD18200T : ceux-ci claquaient sans raison. Ce problème était récurrent sur les anciennes cartes puissance (une pour chaque moteur). En effet, la sortie TH_FLAG était reliée par une résistance au +. D'après nous, cette résistance ne devait pas être assez grosse. La solution a été de ne pas brancher du tout cette sortie et on a plus eu de problème. Il faudra faire attention si l'utilisation de la nouvelle carte de commande est envisagée, car le TH_FLAG est relié au pic d'odométrie, ce qui pourrait poser problème.

Modifications :

- _ rajout d'un condo de $100\mu\text{F}$ entre le + et le - de chaque LMD18200
- _ rajout d'un condo de découplage (en marron sur la photo)
- _ élargissement des pistes (par sécurité)
- _ une des petites pistes de liaison était coupée, il faudra donc faire attention à la conception soit de bien vérifier le circuit, soit d'élargir un peu les pistes (une petite coupure est vite arrivée)

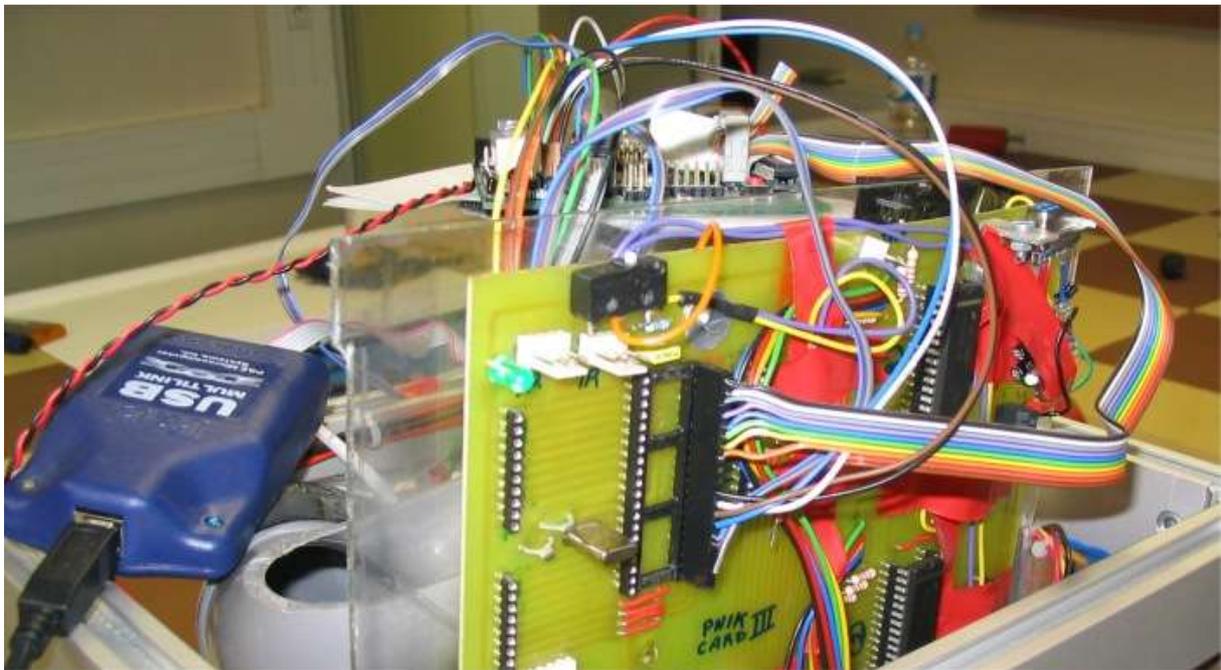
Conseils :

- _ faire bien attention au sens de branchement des codeurs sur les moteurs : on peut se demander longtemps pourquoi ça ne fonctionne pas, alors que la cause est très simple...
- _ bien faire attention au branchement des moteurs en sortie de la carte puissance (voir photo2, les fils du moteur partent vers le bas)
- _ ne pas coller de la mousse directement en contact des cartes, la mousse peut cramer et mettre le feu (expérience vécue à la pré-coupe d'Evry)
- _ laisser de la place sur les cartes pour faire des trous pour pouvoir les fixer dans le robot (c'est très important).

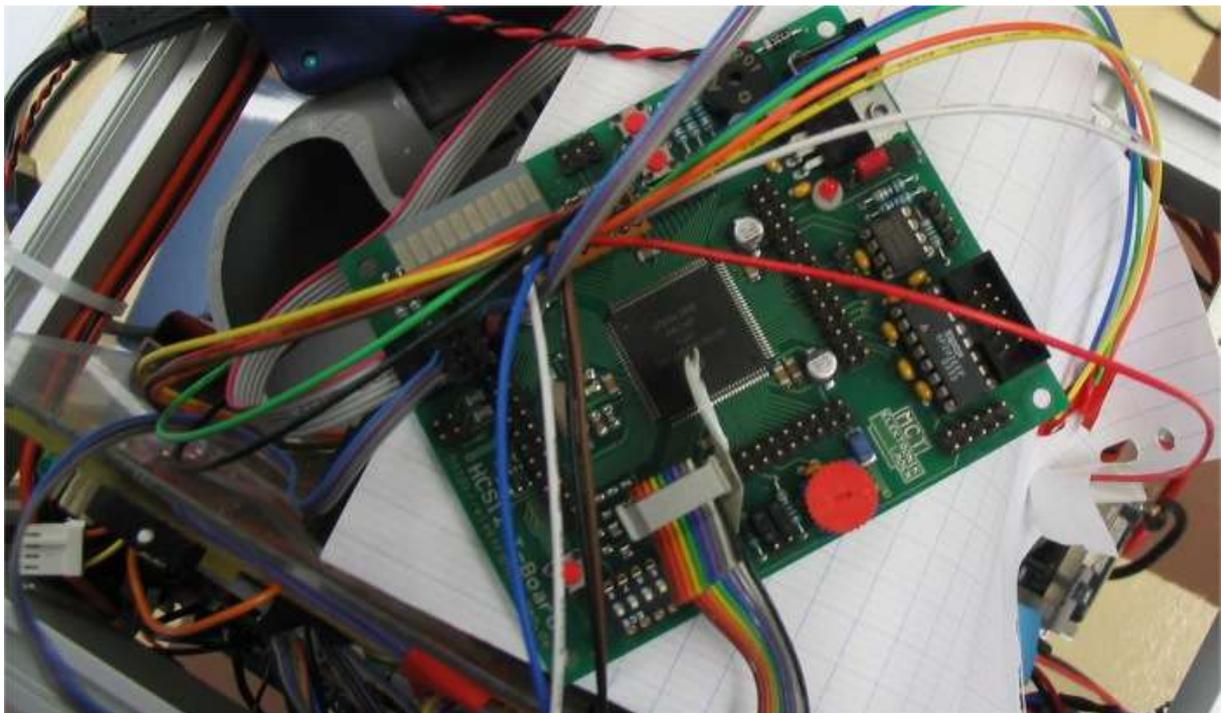
III/ HCS12

L'utilisation de l'HCS12 a été proposée lors de la PR. Je ne vais décrire ici que la partie matérielle et les essais que l'on a effectués.

On est parti de la carte mère avec le pic et on a utilisé un adaptateur pour la liaison avec les LM629 (le but était juste de tester l'HCS12), comme on peut le voir sur la phot ci-dessous :



Cela donne sur l'HCS12 :



Je détaille ici les branchements :

Pic	LM629	HCS12
RB0	D0	PA0
RB1	D1	PA1
RB2	D2	PA2
RB3	D3	PA3
RB4	D4	PA4
RB5	D5	PA5
RB6	D6	PA6
RB7	D7	PA7
RD4	PS	PB1
RD5	HI	PB2
RD6	RD	PB3
RD7	CS1	PB4
RC6	CS2	PB7
RC7	WR	PB5

D'autres fils sont branchés, mais ils ne sont pas utilisés.

Pour les ports de l'HCS12, se reporter à la documentation (deux feuilles A4 dans la boîte de l'HCS12).

On n'a pas réussi à faire fonctionner les LM629 avec l'HCS12. Les hypothèses sont soit un mauvais branchement, soit un faux-contact, soit un problème dans la programmation (au point où on en était arrivé, le problème se situait dans la fonction write, voir rapport sur l'HCS12)

Conclusion : L'utilisation des LM629 est très intéressante. Il nous facilite beaucoup l'asservissement, on n'a eu aucun problème avec et beaucoup d'autres équipes l'utilisent. Le seul problème est le sifflement des moteurs (qui résonnent à la fréquence de la PWM de sortie des LM629, environ 16kHz ; ce n'est pas dangereux, c'est juste fatigant à entendre, mais on sait que le robot est allumé. La solution est de passer à des LM629 qui fonctionnent à 8MHz, soit en prenant ceux existant (d'après certains, ça ne pose pas de problème) ou en rachetant des nouveaux).

L'HCS12 est très prometteur, un OS a été codé par Cyril Nahon, il serait bien de l'utiliser à la prochaine coupe. Il dispose d'un nombre d'E/S important, l'environnement de programmation est bien adapté.

Si vous avez des questions, n'hésitez pas à me contacter : teddy.trachsler@etu.utc.fr