

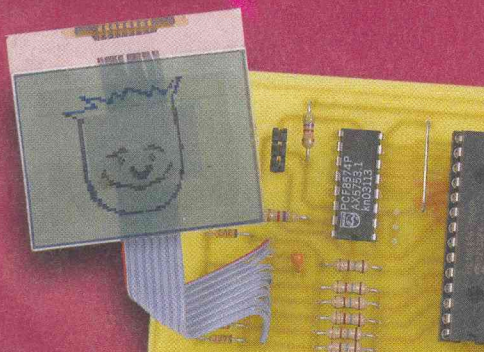
# AMÉLIOREZ LE COMPORTEMENT DE VOS ROBOTS



Mobile à déplacement aléatoire



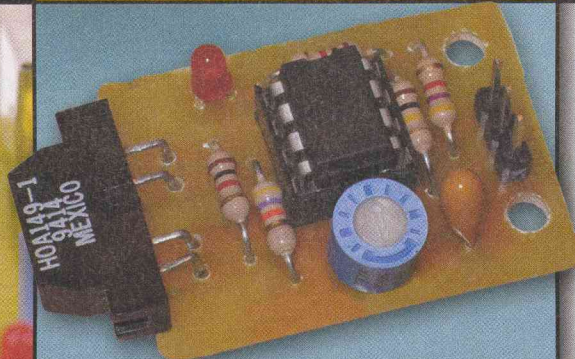
Robot infrarouge



Donnez un air à vos robots



Commande USB pour moteur



Détecteur de ligne

## RETROUVEZ AUSSI :

- ▷ Capteur différentiel
- ▷ Mise en œuvre d'un Perceptron
- ▷ etc

FRANCE : 4,50 € • DOM Avion : 5,70 €  
 BEL : 5,20 € • CH : 7,50 FS  
 CAN : 5,95 \$ CAN • ESP : 4,60 €  
 GR : 4,60 € • TUN : 4700 TNM  
 MAR : 50 DH • PORT CONT : 4,60 €  
 DOM SURF : 4,60 €

T 02437 - 296 - F : 4,50 € - RD



# SOMMAIRE

## ELECTRONIQUE PRATIQUE

Fondateur : Jean-Pierre VENTILLARD

N° 296 - JUILLET/AOÛT 2005

TRANSOCEANIC

S.A.S. au capital de 574 000 €

Président : Patrick VERCHER

2 à 12, rue Bellevue, 75019 PARIS

Tél. : 01.44.84.88.26 - Fax : 01.44.84.88.18

Internet : <http://www.electroniquepratique.com>

### RÉDACTION :

Directeur de la publication :

Patrick VERCHER

Conseiller technique :

Bernard DUVAL

Secrétaire de rédaction :

Annie LECOMTE

Mise en page :

Didier BELLANGER

Couverture :

Dominique DUMAS

Illustrations :

Alain BOUTEVEILLE SANDERS

Photos :

Isabelle GARRIGOU

Avec la participation de : G. Ehretsmann, F. Giamarchi, G. Isabel, L. Josse, V. Le Mieux, E. Lemery, P. Morin, P. Oguic, O. Viacava

La Rédaction d'Electronique Pratique décline toute responsabilité quant aux opinions formulées dans les articles, celles-ci n'engagent que leurs auteurs.

### DIFFUSION/VENTES :

Bertrand DESROCHE

Tél. : 01.44.84.84.54

N° vert réservé aux diffuseurs et dépositaires de presse :

0800.06.45.12

### PUBLICITÉ :

2 à 12, rue de Bellevue - 75019 Paris

Tél. : 01 44 84 88 26

Fax : 01 44 84 88 18

E Mail : [pub.ep@wanadoo.fr](mailto:pub.ep@wanadoo.fr)

### ABONNEMENTS/VPC :

18-24, quai de la Marne - 75019 Paris

Tél. : 01 44 84 85 16 - Fax : 01 42 00 56 92.

Préciser sur l'enveloppe «SERVICE ABONNEMENTS»

**Important :** Ne pas mentionner notre numéro de compte pour les paiements par chèque postal. Les règlements en espèces par courrier sont strictement interdits.

**ATTENTION !** Si vous êtes déjà abonné, vous faciliterez notre tâche en joignant à votre règlement soit l'une de vos dernières bandes-adresses, soit le relevé des indications qui y figurent.

Aucun règlement en timbre poste.

Forfait photocopies par article : 5,00 €.

**Abonnements USA - Canada :** Pour vous abonner à

**Electronique Pratique** aux USA ou au Canada,

communiquez avec Express Mag :

USA : P.O.Box 2769, Plattsburgh, N.Y. 12901-0239

CANADA : 8155 rue Larrey, Anjou (QC) H1J2L5

**Téléphone :** (514) 355-3333

**Sans frais :** 1 800 363-1310

**Fax :** (514) 355-3332.

**Courriel :** [expsmag@expressmag.com](mailto:expsmag@expressmag.com)

**Internet :** [www.expressmag.com](http://www.expressmag.com)

Le tarif d'abonnement annuel pour les USA et le Canada est de 53,90 €.

I.S.S.N. 0243 4911

N° Commission paritaire : 0909 T 85322

Distribution : Transport Presse

Imprimé en France/Printed in France

Imprimerie : SIEP/Bois le Roi - 77590

DEPOT LEGAL JUILLET 2005

Copyright © 2005 - TRANSOCEANIC

« Ce numéro  
a été tiré à 36 500  
exemplaires »

## Infos/Nouveautés Opportunités

4

10

## Internet pratique

## Initiation

12

*Les asservissements linéaires*

14

*Contrôle d'un moteur à courant continu en MLI*

18

*Initiation au PLC*

24

*Mise en œuvre d'un perceptron*

32

*Personnalisation d'un montage USB*

## Réalisez vous-même

40

*Robot infrarouge*

46

*Détecteur de ligne*

48

*Donnez un air à vos robots*

52

*Mobile à déplacement linéaire*

58

*Commande de moteurs par la prise USB*

64

*Capteur directionnel*

70

*Commande moteur par GAL*

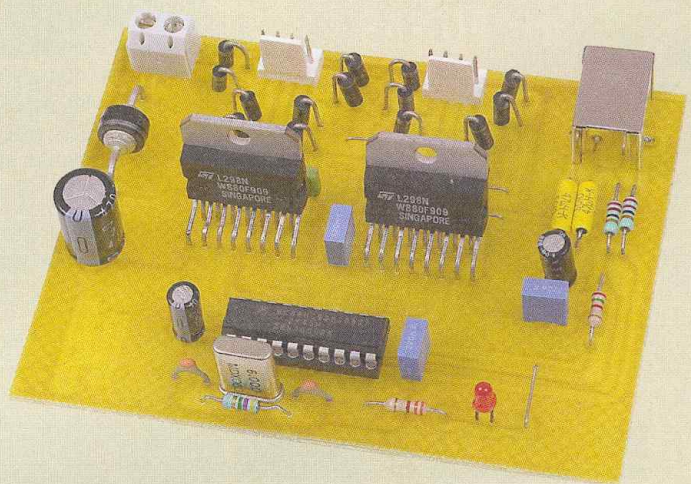
## Divers

30

*Robot HITEC*

38

*Tournoi de sumos de Nîmes*

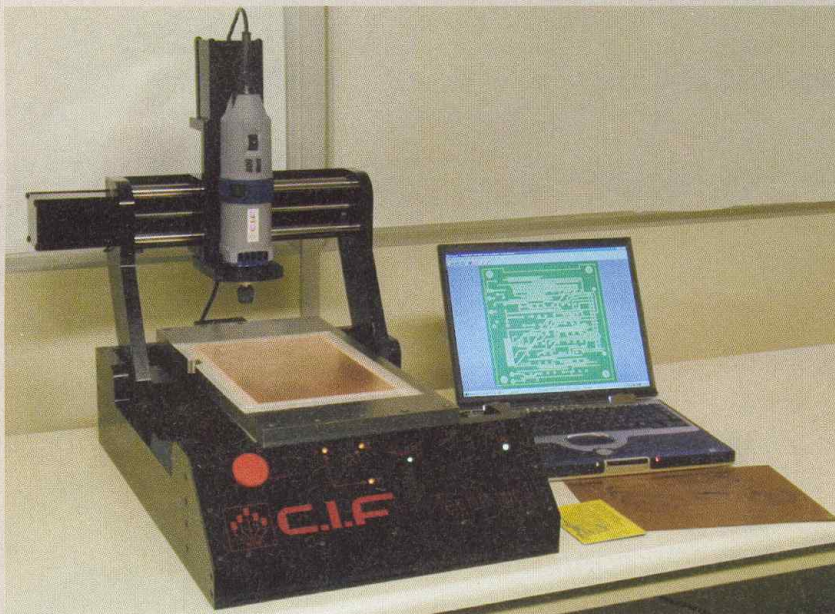


**LE PROCHAIN NUMÉRO D'ÉLECTRONIQUE PRATIQUE  
SERA EN KIOSQUE LE 5 SEPTEMBRE 2005**

# Nouveautés • Infos

## C.I.F.

**Perceuse fraiseuse 3 axes – 3 D de haute précision à commande numérique - Technodrill**



La perceuse fraiseuse 3 axes - 3D de Technodrill en action

Le système Technodrill est idéal pour réaliser tous les projets de découpe, gravure, perçage, détournage, usinage...

Il peut également être utilisé pour graver les faces avant en aluminium ou plastique.

Les courses X, Y, Z sont de 215, 350 et 80 mm pour une surface utile de 210 x 310 mm. La résolution varie de 0,003 à 0,01 mm et la reproductibilité est de 0,05 mm.

La Technodrill est équipée d'une broche de

900 W tournant à une vitesse de 8000 à 26000 tours/min avec mandrin 3.17 et 8 mm. La vitesse de déplacement est au maximum de 50 mm/s et le palpeur de positionnement outil est intégré. Par ailleurs, le système Technodrill est livré avec l'ensemble de logiciels Windows Galaad 3 (6000 licences vendues en France) destinés au pilotage intelligent de machines à commandes numériques de 1 à 5 axes.



Le guidage très précis des axes apporte une haute résolution et une grande reproductibilité

Les points forts de la Technodrill :

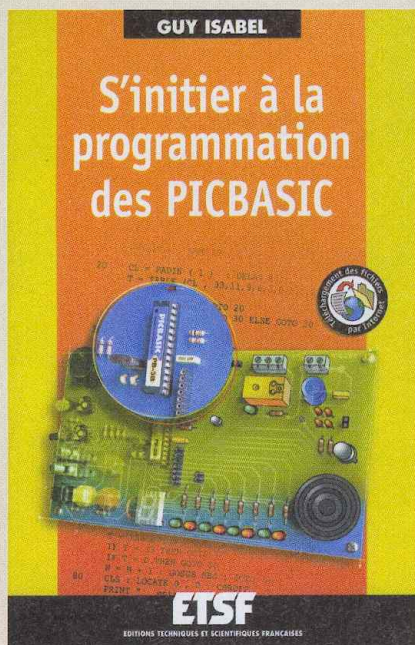
- Apprentissage immédiat
- Polyvalente : perçage, fraisage, détournage...
- Compatible avec tous les types de fichiers
- Guidage très précis des axes
- Logiciel GALAAD 3
- Palpeur d'outil
- Capot en option



11, Rue Charles Michels  
92227 Bagneux Cedex - France

Tél. : 33 (0) 1 4547 4800  
Fax : 33 (0) 1 4547 1614

E-mail : [cif@cif.fr](mailto:cif@cif.fr)  
Web : [www.cif.fr](http://www.cif.fr)



## S'initier à la programmation des PICBASIC

Les PICBASIC révolutionnent aujourd'hui le monde des microcontrôleurs : outre leur faible coût, une grande vitesse d'exécution et une excellente fiabilité, leur programmation en langage BASIC (grâce à un simple PC) permet d'établir **rapidement et sans difficulté leur configuration et leur mise en œuvre.**

À travers la construction d'une platine d'expérimentation et de nombreux exemples de programmes, cet ouvrage propose une **approche très progressive** qui permet au lecteur de comprendre, par la pratique, la "philosophie" et les avantages de ce composant hors du commun. Les applications proposées en fin d'ouvrage présentent quelques aspects de ses **spectaculaires possibilités**, comme la commande des moteurs ou encore les solutions d'affichage LCD ou à LED.

Les subtilités du langage machine ne seront bientôt plus qu'un lointain souvenir. Les portes de la **création** vous seront dès lors grandes ouvertes, et qui sait jusqu'où votre imagination vous entraînera ?

**Guy ISABEL** est professeur en lycée professionnel dans le secteur électrotechnique (BEP et Bac pro). Il collabore régulièrement, depuis plus de 25 ans, à la revue Électronique Pratique.

DUNOD/ETSF - 26 €

# Nouveautés • Infos

## SOLUTION DE FABRICATION DE CIRCUITS IMPRIMÉS SUR FRAISEUSE A COMMANDE NUMERIQUE CHARLYROBOT

Concepteur et fabricant de fraiseuses à commande numérique depuis plus de 20 ans, Charlyrobot dévoile sa solution de fabrication de circuits imprimés sans chimie (ou gravure anglaise).

Cette solution permet le perçage, l'isolation des pistes et la découpe du circuit imprimé, sur fraiseuse à commande numérique.

Ce logiciel permet la réalisation de circuits double face, monocouche, avec un nombre de pastilles illimité, avec optimisation des liaisons (chevelus) en temps réel, auto placement des composants, auto routeur en contour (basé sur les formes, sans grille).

Le calcul des outils est réalisé directement par le logiciel de CAO, grâce à une liaison directe au pilote machine Charlyrobot avec gestion du repositionnement du circuit pour le double face.

L'ensemble des parcours est inclus dans un seul fichier d'usinage, le cycle de changement des outils (manuel ou automatique selon le modèle de fraiseuse Charlyrobot) est géré par le pilote machine.

La réalisation de circuits imprimés



conçus avec d'autres CAO électroniques par import des PCB (principalement au format XGERBER, compatibilité à tester suivant les logiciels et les versions).

Avec ce logiciel, Charlyrobot met son expertise de l'usinage au service de la fabrication mécanique de circuits imprimés.

### Charlyrobot

BP 22 - F-74350 CERNEX

Tél : 04 50 32 00 80

Fax : 04 50 44 00 41

[info@charlyrobot.com](mailto:info@charlyrobot.com)

[www.charlyrobot.com](http://www.charlyrobot.com)

## S'INITIER À LA PROGRAMMATION DES PIC BASIC ET ASSEMBLEUR 2<sup>ème</sup> édition



*Vous permettre de maîtriser les bases de l'électronique programmable, telle est l'ambition de ce livre.*

Nous avons choisi de nous appuyer sur les fameux et universels microcontrôleurs PIC avec lesquels on "fait" de l'électronique sans s'embarasser de schémas complexes.

Les modèles retenus ici sont les 16F84 et 16F628 au rapport performances/prix sans égal sur le marché.

Cet apprentissage, nous l'avons voulu progressif au travers de montages simples et ludiques, basés sur des applications actuelles comme les cartes à puce, la télévision ou les robots.

Ainsi, vous explorerez les immenses possibilités de cette électronique moderne en BASIC, assistés des logiciels BASIC F84 et F84+ développés pour vous par l'auteur, puis en assembleur. Bien plus qu'un manuel technique et qu'un simple ouvrage de programmation, ce guide est une véritable porte ouverte sur le monde de l'électronique numérique.

CD-ROM inclus

**A. REBOUX - ETSF/DUNOD**

224 pages - 38 €

## Avalanche de nouveautés en ROBOTIQUE chez... Lextronic

Lextronic est désormais le distributeur exclusif pour la France des bases robotiques « Robotics Connection™ ». Proposés sous forme de kits à assembler, les modèles « Botster™ » se composent d'un châssis mobile en polycarbonate dont la motricité est assurée par 2 servomoteurs. Chacune des roues est équipée d'une "bande" anti-dérapante ainsi que d'un disque associé à un encodeur optique qui vous restituera des impulsions proportionnelles à la vitesse de rotation. La base dispose de plusieurs emplacements permettant de recevoir en option des batteries, votre platine électronique de commande, 3 détecteurs d'obstacles infrarouges ainsi qu'une tourelle pour la caméra de détection de couleurs « CMUcam2 ». La photo montre la base équipée de divers accessoires optionnels. Le modèle « Botster™ » libre de toute option est proposé à 135 € TTC.

Lextronic distribue la plate forme de développement « RDS-X01 ». Livrée en kit, elle se compose de plusieurs plaques supports, associées à 2 blocs moteurs ainsi qu'à une série de capteurs (antennes « palpeuses », détecteurs d'obstacle et de lumière...) et à une platine de gestion microcontrôlée destinée à coordonner l'ensemble des fonctions du robot (toutes les platines électroniques sont montées et testées). L'ensemble est fourni avec un logiciel de programmation graphique sur PC (type organigramme) permettant via une liaison série de modifier les réactions du robot (action sur ses sorties logiques, sur ses moteurs avec plusieurs vitesses) en fonction de l'état de ses capteurs ou de calculs pouvant être effectués sur des variables. La plate-forme complète est proposée à 159 € TTC.

Développée par la société "Designer Systems™" et distribuée par Lextronic, la base « DS-X4 » s'apparente à une plate-forme mobile « tout terrain » avec 4 roues motrices (actionnées par servomoteurs). Ces servomoteurs sont pilotés par une petite platine (livrée montée) qui pourra recevoir des ordres séries en provenance de votre électronique de commande afin que vous puissiez faire avancer ou reculer la base ( avec 9 vitesses au choix), la faire tourner à droite ou à gauche, la stopper, activer ses détec-



Le Botster avec son châssis mobile en polycarbonate, sa motricité est assurée par 2 servo-moteurs



Le DS-X4, plate forme mobile tout terrain

teurs d'obstacles infrarouges intégrés... La base est disponible en version standard (avec support pour piles) à 159 €

TTC ou dans une version plus complète avec un accumulateur et un chargeur à 215 € TTC.

Parmi les autres nouveautés dédiées à la robotique, Lextronic propose également plusieurs petits modules hybrides :

- Petite carte accéléromètre/gyroscope subminiature.
- Module de commande pour moteur « CC » 3 A avec entrée d'asservissement (par tension via un potentiomètre de recopie ou via le signal d'un encodeur).
- Platine microcontrôlée intégrant un µP Atmel™ + un étage de puissance pour pilotage de moteurs « CC » + afficheur LCD.
- Module de commande miniature pour moteur « CC » de 30 A max !
- Module de communication infrarouge bi-directionnel sécurisé.
- Module de commande pour moteur « pas à pas » très évolué avec fonction mémoire.

La description de l'ensemble de ces nouveaux produits est disponible sur le site [www.lextronic.fr](http://www.lextronic.fr)

### LEXTRONIC

36/40 rue du Général de Gaulle  
94510 La Queue en Brie  
Tél. : 01 45 76 83 88  
Fax : 01 45 76 81 41

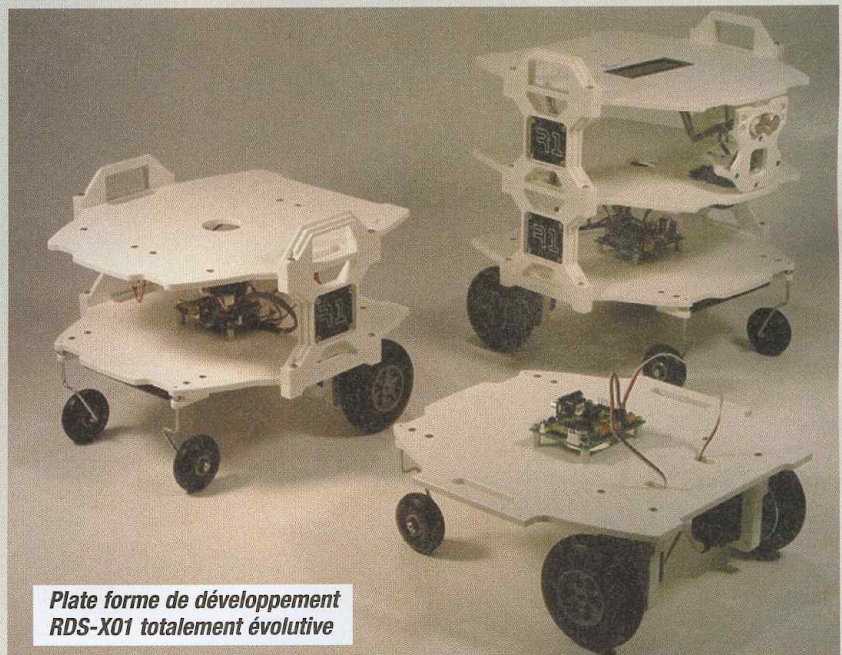


Plate forme de développement RDS-X01 totalement évolutive

# internet PR@TIQUE

**Les DSP sont des processeurs optimisés pour réaliser des traitements complexes sur des signaux audio, vidéo et autres. La spécialisation de ce type de processeurs est telle que dans l'esprit de beaucoup de personnes le terme DSP désigne la fonction égaliseur que l'on rencontre dans les amplis audio modernes ou dans des logiciels de lecture de fichiers audio tels que Winamp. Pourtant l'utilisation des DSP va bien au-delà du filtrage élaboré d'un signal audio. Pour découvrir l'étendue des applications offertes aux DSP nous vous invitons à visiter quelques sites Internet avec nous.**

Pour entrer directement dans le vif du sujet nous vous invitons à charger la page suivante dans votre navigateur préféré : « <http://perso.wanadoo.fr/lapiste/DSP/dsp.htm> ». Cette page brosse rapidement le tableau de quelques-unes des particularités d'un DSP sans pour autant noyer le lecteur dans un flot de détails trop techniques. Après avoir lu les quelques explications sur le format en virgule flottante qu'un DSP est capable de manipuler très rapidement, on ressent le besoin de trouver un site qui donne un peu plus de détails.

C'est donc tout naturellement que nous vous invitons à char-

**1** <http://perso.wanadoo.fr/lapiste/DSP/dsp.htm>

ger ensuite la page « [http://genelaix.free.fr/iufm-genelec-forum/DSP/DSP\\_Prog.htm](http://genelaix.free.fr/iufm-genelec-forum/DSP/DSP_Prog.htm) » dans votre navigateur.

Ce site comporte plusieurs documents très intéressants au format PDF. En particulier, l'introduction disponible à l'adresse « [http://genelaix.free.fr/iufm-genelec-forum/DSP/Tome1\\_DSP.pdf](http://genelaix.free.fr/iufm-genelec-forum/DSP/Tome1_DSP.pdf) » est particulièrement utile pour bien

**2** [http://genelaix.free.fr/iufm-genelec-forum/DSP/Tome1\\_DSP.pdf](http://genelaix.free.fr/iufm-genelec-forum/DSP/Tome1_DSP.pdf)

**3. Architecture des processeurs :**

L'architecture d'un microprocesseur, et donc d'un DSP, est un élément important qui conditionne directement les performances d'un processeur. Il existe deux types fondamentaux de structures, dites « Von Neuman » et « Harvard », telles que présentées par la figure 4.

**Architecture de type Von Neuman**

```
graph LR; MP[Mémoire Programme et Données] --- A[Adresse]; MP --- D[Donnée]; A --- US[Unité Arithmétique et Logique]; D --- US;
```

**Architecture de type Harvard**

```
graph LR; MP[Mémoire Programme] --- A1[Adresse]; MP --- D1[Donnée]; A1 --- US[Unité Arithmétique et Logique]; D1 --- US; A2[Adresse] --- MD[Mémoire Données]; D2[Donnée] --- MD; US --- MD;
```

figure 4 représentation schématique des structures de Harvard et de Von Neuman

**3.1. structure de Von Neuman**

Un microprocesseur basé sur une structure Von Neuman stocke les programmes et les données dans la même zone mémoire. Une instruction contient le code opératoire et l'adresse de l'opérande. Ce type de microprocesseur incorpore principalement deux unités logiques de base :

**3** <http://perso.ens-lyon.fr/arnaud.tisserand/docs/semlip-at-dsp.pdf>

**Application type : MODEM ADSL**

- Modems et modems
- Télécommunications
- Modems
- Téléphonie cellulaire
- Disques durs
- Appareils audio
- Lecteurs CD, DVD, MP3
- Instruments médicaux
- Instrumentation de la parole
- Vidéo
- Automobile
- Robotique

**Algorithme type : les Filtrés FIR**

Pour un filtre à réponse impulsionnelle finie (FIR) linéaire à coefficients réels de taille N (nombre de coefficients), on doit effectuer un calcul du type :

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k]$$

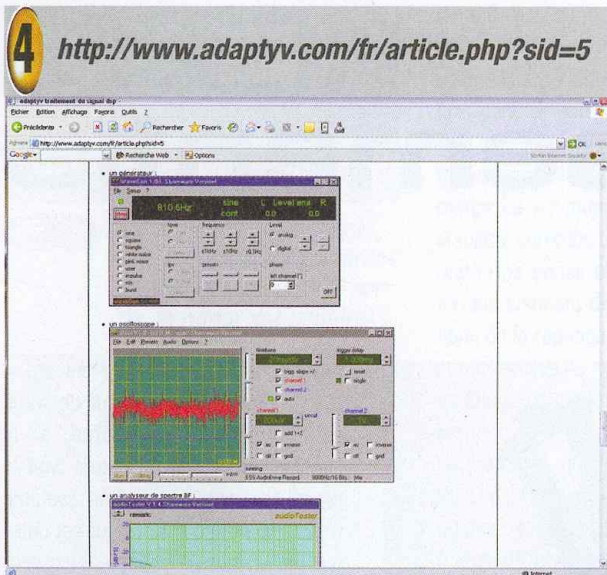
Besoin en calcul de MM à 150 MIPS

QAM = Quadrature Amplitude Modulation

TEQ = Time domain Equalizer

FEC = Frequency domain Equalizer

Contraintes : n/dé coefficients (16 – 300), précision (18 – 24 bits, un peu plus en interne) ; vitesse de fonctionnement.

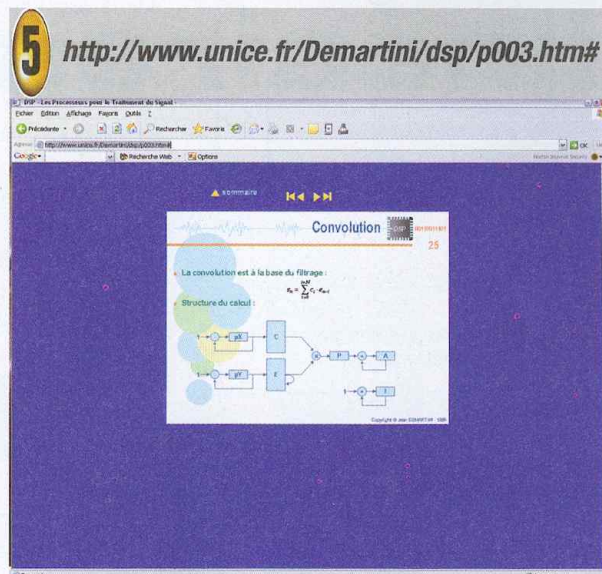


comprendre ce qui fait la force d'un DSP face à un microprocesseur classique dans le domaine du traitement de signal.

Si vous recherchez des explications un peu plus illustrées, nous vous conseillons de télécharger également le document PDF disponible à l'adresse « <http://perso.ens-lyon.fr/arnaud.tisserand/docs/semliip-at-dsp.pdf> ». Ce document illustre l'intérêt des instructions MAC propres aux DSP dans la mise en œuvre d'un filtre à réponse impulsionnelle finie (FIR). Ce document explique également pourquoi les modes d'adressage particuliers d'un DSP sont un avantage pour le traitement de signal.

Si l'idée de mettre vous-même en œuvre un DSP vous semble trop ardue vous constaterez à l'aide du site situé à l'adresse « <http://www.adaptyv.com/fr/article.php?sid=5> » que cela reste pourtant à la portée des amateurs passionnés. Notez au passage que, comme le démontre le site en question, la carte son de votre fidèle PC contient une partie des équipements nécessaires pour constituer à moindre frais un générateur BF et un analyseur de spectre. Le site indique les liens qui permettent d'accéder aux logiciels nécessaires pour cela, et ils sont gratuits. Alors ne vous en privez pas !

Pour terminer notre petite incursion dans le monde des DSP grâce à Internet, nous vous proposons finalement de charger la page « <http://www.unice.fr/Demartini/dsp/index.htm> » dans votre navigateur. Ce site propose un cours en ligne sur les DSP. Il décrit notamment les opérations de base du traitement



de signal tels que le produit scalaire, la convolution et la transformée de Fourier rapide avec des illustrations tout à fait intéressantes.

Bien entendu, la visite de quelques pages sur Internet sont insuffisantes pour prétendre maîtriser un sujet aussi vaste que celui du traitement de signal à l'aide d'un DSP. Mais désormais, lorsque vous entendrez parler de DSP dans une conversation profane, vous aurez le sentiment d'en savoir un peu plus sur le sujet. Si votre soif d'apprendre sur ce sujet n'est toujours pas éteinte, nous vous proposons de visiter également les sites mentionnés à la fin de cet article (les trois derniers sont en anglais) et nous vous donnons rendez-vous à la rentrée pour de nouvelles découvertes grâce à Internet.

P. MORIN

- <http://perso.wanadoo.fr/lapiste/DSP/dsp.htm>
- [http://genelaix.free.fr/iufm-genelec-forum/DSP/Tome1\\_DSP.pdf](http://genelaix.free.fr/iufm-genelec-forum/DSP/Tome1_DSP.pdf)
- <http://genelaix.free.fr/iufm-genelec-forum/DSP/indexdsp.htm>
- <http://perso.ens-lyon.fr/arnaud.tisserand/docs/semliip-at-dsp.pdf>
- <http://www.adaptyv.com/fr/article.php?sid=4>
- <http://www.adaptyv.com/fr/article.php?sid=5>
- <http://www.unice.fr/Demartini/dsp/index.htm>
- <http://www.creatis.insa-lyon.fr/~yougz/tsi/dsp.ppt>
- <http://www.f5mi.fr/fm/>
- <http://www.bores.com/courses/intro/chips/index.htm>
- <http://dspvillage.ti.com/>
- <http://www.dspguru.com/info/tutor/index.htm>

**T1** Liste des liens

# Les asservissements linéaires



*Les asservissements sont employés très fréquemment pour contrôler une grandeur physique à l'aide d'une consigne. Cela va de la vitesse d'un moteur à la position du bras articulé d'un robot, sans parler des nombreuses autres applications courantes (régulation de la température, climatisation, positionnement de la tête d'un lecteur de CD, etc.).*

L'étude des asservissements fait appel à la théorie des systèmes bouclés. Cette théorie ne s'applique pas seulement à l'électronique puisqu'elle permet également d'étudier de nombreux systèmes mécaniques (ABS des véhicules modernes, direction assistée, etc.). Sans vouloir prétendre résumer cette théorie en seulement quelques lignes, nous vous proposons de découvrir ici quelques-uns des grands principes mis en jeu dans un asservissement.

Tous les systèmes bouclés peuvent se ramener à l'étude du schéma bloc représenté en **figure 1**. En effet, sous réserve de pouvoir poser les différentes équations qui régissent le fonctionnement du système à étudier, il est

toujours possible de revenir au schéma bloc de la figure 1 au moyen de plusieurs regroupements. Les regroupements en question ne sont bien souvent qu'une facilité d'écriture mathématique de sorte qu'il n'est pas toujours très simple d'identifier quels sont les paramètres influents du système physique. Cependant, la mise en équation sous la forme de ce schéma bloc permet d'exploiter directement toutes les conclusions de la théorie des asservissements, ce qui en vaut la peine.

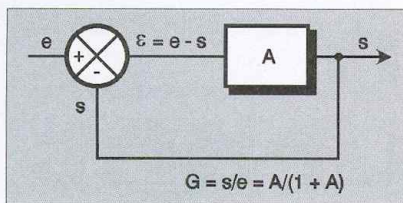
On distingue deux types d'utilisations principales des asservissements linéaires : l'asservissement de position (système suiveur) et la régulation. Dans le cas d'un asservissement de

position, le retour est dit « unitaire », c'est à dire que la grandeur de sortie est directement comparée à la consigne appliquée en entrée (voir la figure 1). Dans le cas d'un système régulé, la grandeur de retour est obtenue par un bloc de traitement (capteur de vitesse, pont diviseur, etc.) qui ramène la grandeur de sortie dans une unité comparable avec la consigne (voir la **figure 2**).

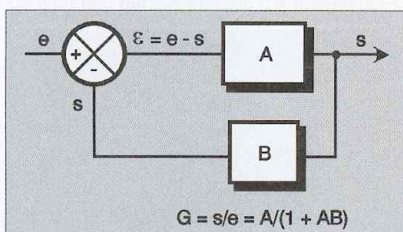
Moyennant un remaniement du schéma bloc de la figure 2, on démontre que l'étude d'un système régulé s'apparente à l'étude d'un asservissement de position suivi d'un traitement de mise à l'échelle, comme cela apparaît sur la **figure 3**. Dans ce cas, la grandeur de retour «  $r$  » est souvent abstraite car elle n'existe qu'au travers des équations mathématiques, ce qui perturbe souvent les jeunes esprits qui cherchent un sens concret aux notions étudiées (où placer la sonde de l'oscilloscope, par exemple, pour visualiser la grandeur de retour ?).

Au vu de l'équation reportée sur la figure 1, on constate que le gain d'un système bouclé à retour unitaire «  $G$  » tend vers 1 lorsque l'amplification «  $A$  » est élevée. Pour les autres systèmes, le gain tend vers  $1/B$  (figures 2 et 3). La contre réaction introduite dans un système bouclé permet donc de s'affranchir de la valeur exacte de l'amplification «  $A$  ». Que l'étage amplificateur soit réalisé avec des transistors (dans ce cas il est assez facile d'obtenir  $A = 10$  à  $100$ ) ou avec un amplificateur opérationnel ( $A = 100\ 000$  et plus) le gain du système bouclé restera très proche de 1 (ou  $1/B$ ). Lorsque l'on étudie un système bouclé, on s'intéresse à deux paramètres contradictoires : la précision de l'asservissement et sa stabilité. En effet,

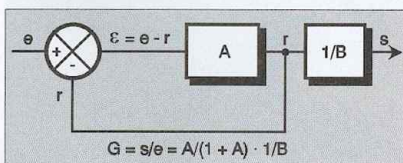




**1** **Asservissement de position (la consigne est directement comparée à la sortie, le retour est unitaire)**



**2** **Cas général d'un système régulé (la sortie n'est pas de même nature que la consigne)**



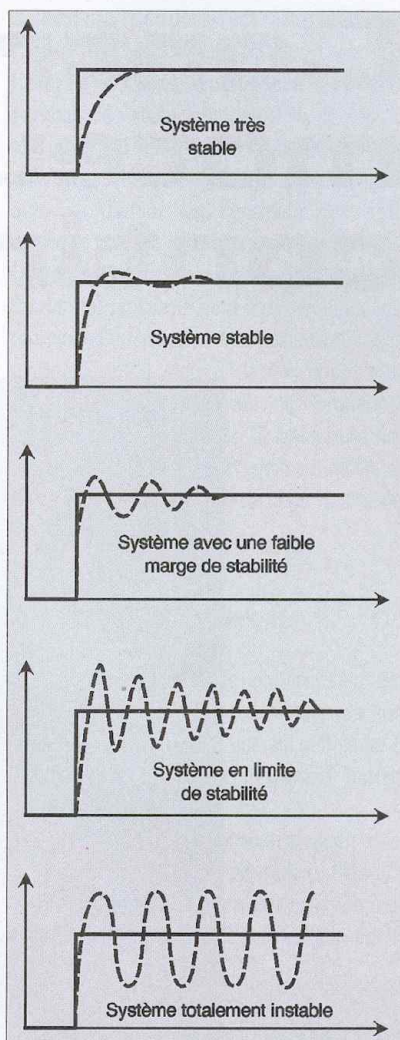
**3** **L'étude d'un système régulé est comparable à celle d'un asservissement de position**

pour obtenir une bonne précision de l'asservissement, il faut maintenir (e) aussi faible que possible. La solution consiste à augmenter l'amplification du système, mais ce faisant, on augmente également le risque d'instabilité comme le précise le critère de Nyquist. On peut comprendre ce critère de façon intuitive si l'on tient compte de la composition spectrale de tout signal électrique. En admettant que le système étudié reste linéaire sur toute la plage de fonctionnement envisagée, on peut déterminer son comportement en étudiant sa réponse pour chaque fréquence du spectre (la réponse totale étant la somme de toutes les réponses pour chaque fréquence individuelle). Pour étudier la stabilité du système, il suffit donc d'étudier le gain et la phase du système en fonction de la fréquence du signal d'entrée. Si pour une fréquence donnée on constate que le signal de sortie est déphasé de 180° tandis que le système possède un gain supérieur à 1, alors le système sera instable. En effet, à cette fréquence particulière, un infime couplage du signal de sortie sur le signal d'entrée transformera le sys-

tème en oscillateur ! Pour rendre un système stable, on ajoute généralement un réseau de compensation (un simple filtre R/C peut suffire) dans la boucle. Dans la pratique, il ne faut pas se contenter de rendre le système théoriquement stable. Il faut prévoir une marge de fonctionnement pour assurer une stabilité correcte du système. On parle de marge de phase ou de marge de gain. Ces termes viennent des différentes représentations de la réponse du système en fréquence sur lesquelles on mesure le critère de stabilité (diagramme de Bode ou diagramme de Nyquist).

La marge de gain est définie par la formule  $20 \text{ Log}(A)$ , le gain A étant calculé à la fréquence à laquelle la phase fait 180°. La marge de phase se mesure à la fréquence pour laquelle le gain du système vaut 0 dB. Ces notions sont illustrées sur la **figure 4**. Dans la pratique, on considère qu'un système est suffisamment stable si sa marge de gain est de -10 dB à -15 dB tandis que la marge de phase recherchée se situe entre 40° et 45° environ.

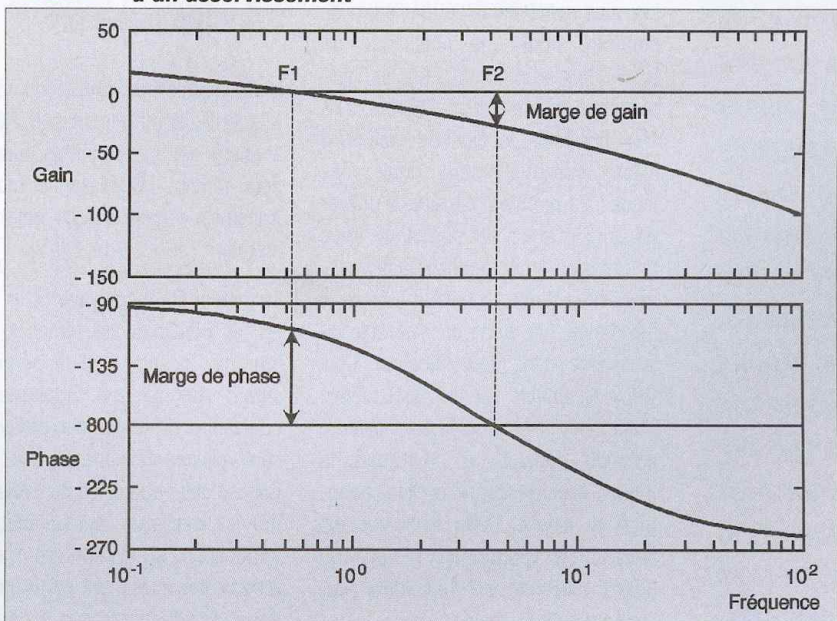
On constate facilement l'intérêt de ces notions si l'on étudie la réponse du système à une impulsion appliquée sur le signal d'entrée. Les courbes de la **figure 5** illustrent la réponse de plusieurs systèmes, en partant d'un système parfaitement stable et en progressant jusqu'à un système totalement instable.



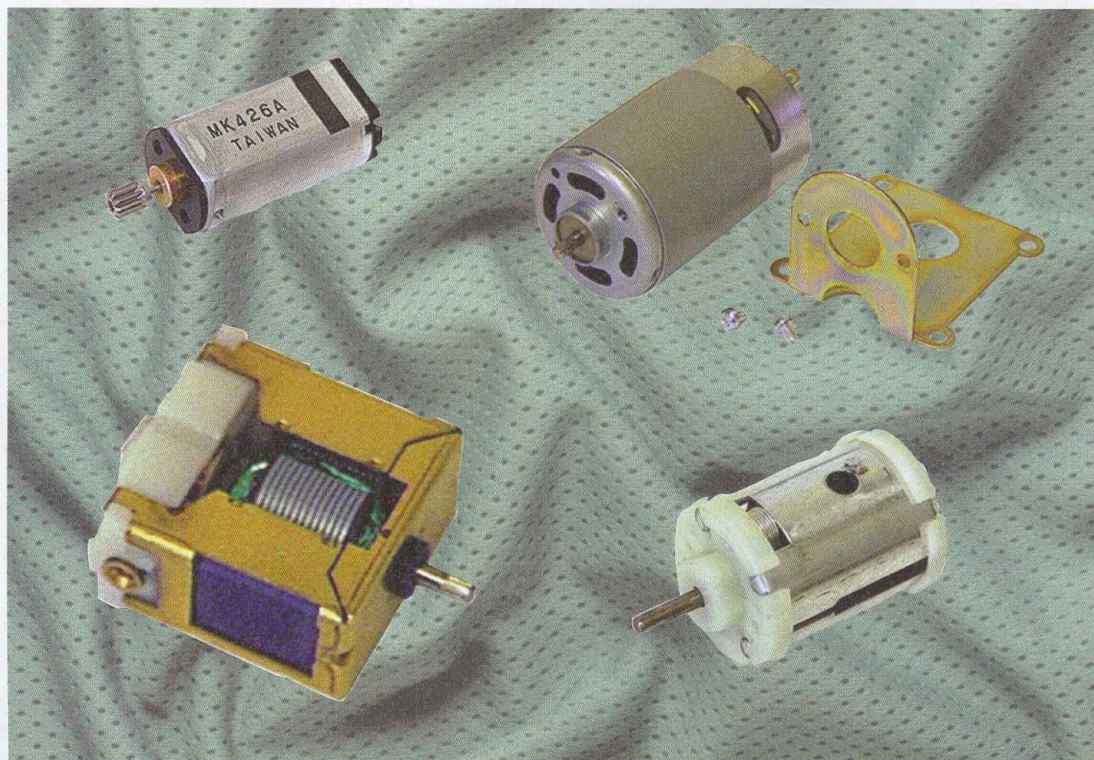
**5** **Réponse à une impulsion en fonction de la marge de stabilité du système**

P. MORIN

**4** **La marge de gain et la marge de phase définissent la stabilité d'un asservissement**



# Contrôle d'un moteur à courant continu par MLI



*Les moteurs à courant continu existent sous une grande variété de tailles, de prix et de qualités. Les aimants permanents utilisés dans ces moteurs permettent d'obtenir, maintenant, des modèles de très hautes performances. En robotique, ce sont bien évidemment ces modèles qui sont recherchés pour diminuer la consommation sans perte de puissance.*

Les plus petits modèles font 4mm de diamètre pour une puissance de 0,1 W. Pour un robot de 1 kg, on pourra utiliser des moteurs de 25 mm, 3 W. Pour contrôler ce type de moteur, il existe essentiellement deux méthodes. La première consiste à utiliser un amplificateur de puissance avec une tension proportionnelle à la vitesse désirée. Cette technique montre, à l'évidence, une perte considérable de puissance dans l'amplificateur. Ceci est inacceptable sur un robot mobile dont l'énergie est limitée. La meilleure solution consiste à commuter le moteur avec un rapport cyclique adapté à la vitesse. Cette technique est aussi mieux adaptée aux moteurs de qualité moyenne que l'on trouve plus communément.

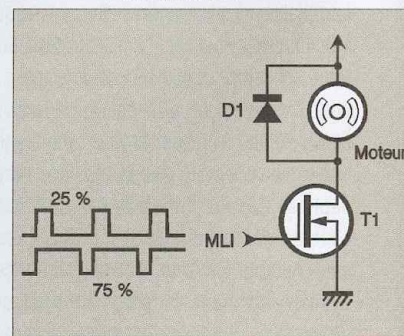
## Principe (figure 1)

La modulation de largeur d'impulsion MLI (Pulse Width Modulation, PWM en anglais) est la technique employée pour réduire la puissance moyenne transmise à une charge comme un moteur.

Le moteur étant équivalent à un circuit LR, sa constante de temps  $L/R$  doit être plus grande que la période du signal qui lui est appliqué. Les constantes de temps des moteurs CC sont voisines de 1 ms environ, ce qui impose des fréquences de 1 kHz minimum. L'inertie du moteur intègre les impulsions et le courant dans le moteur est quasiment constant. Pour éviter les vibrations de la carcasse du

moteur aux fréquences audibles, on choisit souvent des fréquences supérieures à 20 kHz. Cette technique autorise les démarrages à bas régime.

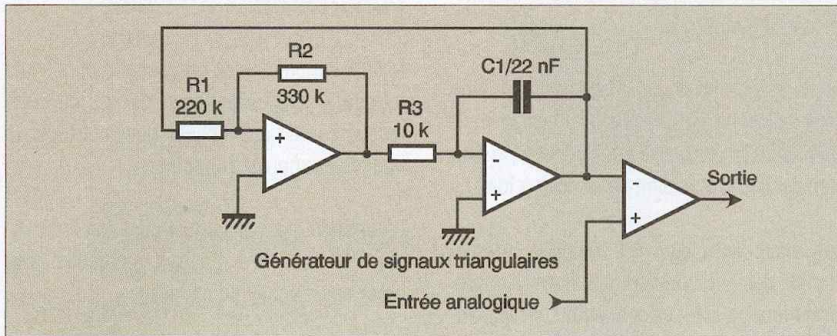
**1** Les signaux faisant varier la vitesse d'un moteur CC



## MLI analogique (figure 2)

Il existe différentes techniques pour obtenir cette modulation. Les plus simples sont entièrement analogiques et utilisent des amplificateurs opérationnels ou des circuits de type 555.

On peut, par exemple, transformer une commande de vitesse analogique en la comparant avec un triangle. Le résultat est un signal de type MLI dont la période est définie par le



**2** Schéma analogique pour générer des signaux MLI

triangle et le rapport cyclique (rapport entre la largeur de l'impulsion et la période) est la durée pendant laquelle la commande est supérieure au triangle.

Une autre solution est d'utiliser un 555 en monostable, déclenché par un autre qui définit la période ou encore d'utiliser des composants spécialisés pour alimentation à découpage. Ces dernières solutions sont toujours d'actualité. Mais dès que l'application devient complexe, les microcontrôleurs sont incontournables et proposent de nouvelles solutions pour générer les signaux nécessaires. C'est plus précisément l'objet de cet article.

## Le microcontrôleur

Les microcontrôleurs ont révolutionné le monde des composants électroniques et les applications qui s'y rapportent. C'est encore plus vrai en robotique mobile, où il faut réduire le nombre de composants en accroissant les capacités.

On a donc inventé des techniques de programmation pour générer des signaux de type MLI pour contrôler des moteurs de tous types et en particulier à courant continu.

Pratiquement toutes ces techniques utilisent une ressource interne importante qu'est l'interruption. On paramètre le microcontrôleur pour obtenir une interruption régulière du

programme afin de traiter notre contrôle des moteurs. Cette dernière technique, quoique convenable, n'est pas sans poser des problèmes dans certains cas.

## MLI numérique

Il existe plusieurs types de programmations pour obtenir le même résultat, mais chacune a ses avantages et ses inconvénients.

durée d'une interruption est calculée pendant celle-ci, en fonction du rapport cyclique souhaité. Le problème apparaît pour les valeurs proches des valeurs limites 0 % et 100 %, car le temps passé dans l'interruption peut être supérieur à la durée du signal à appliquer au moteur. On considère que cette méthode permet de générer des signaux de rapport cyclique compris entre 10 % et 90 %. Ce dernier problème peut être résolu en restant dans l'interruption et en calculant l'autre partie du signal. Alors, il devient possible de générer des signaux MLI de fréquences 20 kHz maximum (quartz 20 MHz) avec une précision de 0 à 100 % en utilisant moins de 25 % du temps. L'écriture de cette routine est longue et fastidieuse.

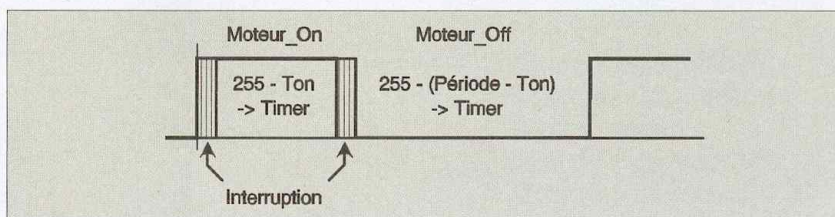
## 2<sup>ème</sup> méthode (figure 4)

Cette première méthode montre déjà des limites d'utilisation. Dans beaucoup de cas, une précision sur 4 ou 6 bits est suffisante et permet d'augmenter la fréquence, critère plus intéressant.

La deuxième méthode consiste à incrémenter un compteur à chaque interruption. Puis une variable contenant la vitesse est ajoutée à ce compteur. Si le résultat dépasse la capacité

## 1<sup>ère</sup> méthode (figure 3)

La première consiste à modifier la ligne affectée au moteur à chaque interruption. La



**3** Le signal change à chaque interruption

### Listing 1 : méthode n° 1, en assembleur pour PIC

Routine MLI (méthode n° 1) vitesse entre 10 et 90 % à fréquence 2 kHz

```

Btfsc      Moteur_pin      ; test si la ligne est à 0
Goto       MLI_bas
MLI_haut   ; génère le temps haut
movf       MLI_vitesse,W   ; transfère la vitesse dans W
sublw     .255             ; 255 - W
movwf     TMRO             ; pour durée de l'état haut
Moteur_On ; ligne moteur à 1
goto      MLI_fin
MLI_bas   ; génère le temps bas
movf       MLI_vitesse,W   ; transfère la vitesse dans W
addlw     .5               ; 255 - ( 250 - W )
movwf     TMRO             ; pour durée de l'état bas
Moteur_Off ; ligne moteur à 0
MLI_fin
    
```

du compteur, le bit de retenue est positionné puis la ligne du moteur est mise à 1 et inversement. La période du signal est égale à la précision, multipliée par la durée entre deux interruptions.

Une addition supplémentaire est nécessaire pour atteindre les 100 %, voir listing.

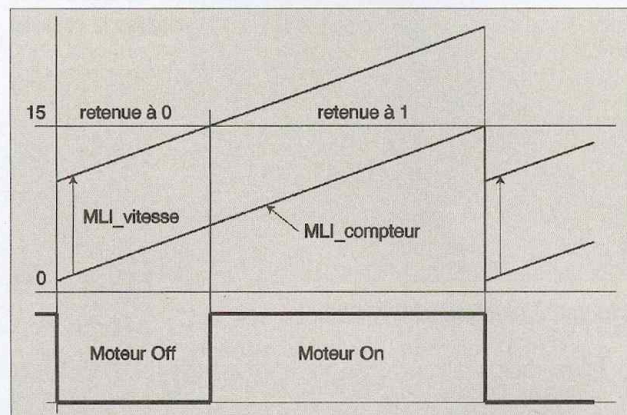
### 3<sup>ème</sup> méthode (figure 5)

Cette dernière méthode est peu employée. Pourtant, elle n'est pas plus compliquée que la précédente et permet de générer des signaux de fréquence plus élevée.

À chaque interruption, on ajoute la valeur de la variable vitesse dans un compteur et on utilise la retenue pour activer le moteur. Cette technique appelée modulation de la fréquence des impulsions MFI (Pulse Frequency Modulation, PFM en anglais) permet de multiplier par 5 au plus la fréquence du signal et d'augmenter la précision par rapport à la méthode précédente.

### MLI enfoui (figure 6)

L'évolution et la demande ont fait que l'on trouve maintenant des microcontrôleurs



**4** Un compteur est incrémenté à chaque interruption

### Listing 2 : méthode n° 2, en assembleur pour PIC

Routine MLI (méthode n° 2) répétée toutes les 50 µs

```

movlw 0x10 ; précision sur 4 bits
addwf MLI_compteur,F ; fréquence : 1,2 kHz
btfsc STATUS,Z ; test si égale à 0
addwf MLI_compteur,F ; addition supplémentaire
movf MLI_compteur,W ; transfère le compteur dans W
addwf MLI_vitesse,W ; ajoute la variable vitesse
btfsc STATUS,C ; teste la retenue
Moteur_On ; retenue à 1 -> ligne moteur à 1
btfss STATUS,C ; teste à nouveau la retenue
Moteur_Off ; retenue à 0 -> ligne moteur à 0
    
```

### Listing 3 : méthode n° 3, en assembleur pour PIC

Routine MFI (méthode n° 3) répétée toutes les 50 µs

```

movf MFI_vitesse,W ; transfère la vitesse dans W
addwf MFI_compteur,F ; ajoute la vitesse au compteur
btfsc STATUS,C ; teste la retenue
Moteur_On ; retenue à 1 -> ligne moteur à 1
btfss STATUS,C ; teste à nouveau la retenue
Moteur_Off ; retenue à 0 -> ligne moteur à 0
    
```

équipés, en interne, de structures pour générer des signaux MLI.

Fini la programmation, il suffit de paramétrer à l'initialisation la période et la précision et de changer le rapport cyclique à la volée, la synchronisation s'occupant de la mise à jour.

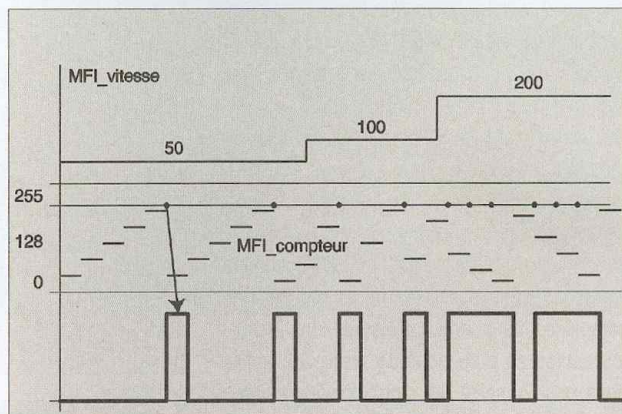
Ces microcontrôleurs ont simplifié la commande des moteurs en robotique. Le programmeur pouvant se concentrer sur le comportement de son jouet (Électronique Pratique n° 290).

### Conclusion

Malgré l'apparition de microcontrôleurs à MLI interne, il existe beaucoup de cas où cette ressource doit être utilisée pour autre chose que la commande de moteurs.

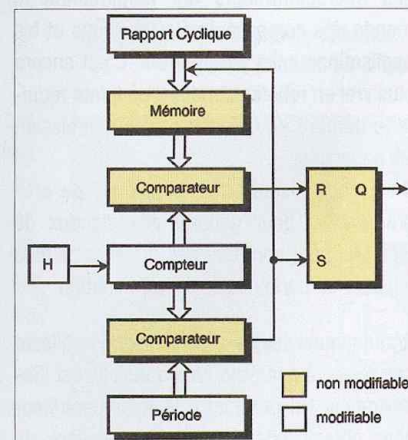
C'est dans ces cas là qu'il peut être intéressant de connaître d'autres méthodes pour créer des signaux de ce type.

F. GIAMARCHI  
giamarchi@iut-nimes.fr

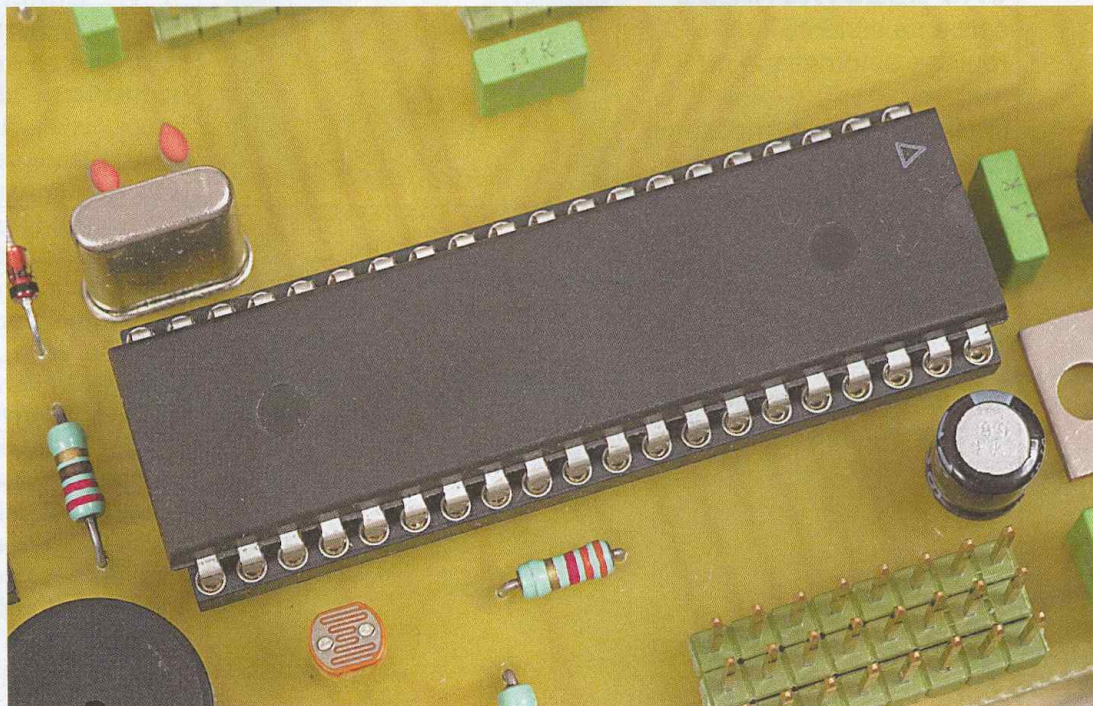


**5** La modulation de la fréquence des impulsions

### 6 Structure d'un générateur de signaux MLI pour microcontrôleur



# Initiation aux PLC ou automates programmables



*Les PLC, (Programmable Logic Controller) ou Automates Programmables Industriels (API), existent depuis de nombreuses années. Surtout utilisés dans l'industrie pour la commande de processus, ils subissent une vulgarisation qui permet au grand public de les utiliser dans diverses applications, que ce soit par exemple la domotique ou la robotique.*

## Généralités

Les PLC sont, si l'on peut dire, des ordinateurs spécialement conçus aux applications d'automatismes et peuvent se comparer à nos ordinateurs personnels mais ne possèdent ni écran, ni clavier. Ils nécessitent cependant un outil de programmation, entre autres un PC, sur lequel est installé un langage de programmation graphique très simple à assimiler : le Relay Ladder Diagram.

La **figure 1** montre un exemple de symboles graphiques utilisés par cet outil de programmation. En fait, une personne ne connaissant même pas les bases de l'électronique est capable, après une courte adaptation, de programmer un PLC.

Si l'on compare un PLC et les lan-

gages de programmation classiques (BASIC, C, assembleur, ...), on s'aperçoit que les PLC sont multitâches, c'est-à-dire que plusieurs actions peuvent être accomplies simultanément.

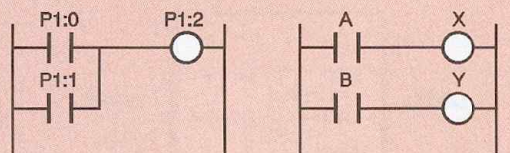
Dans les autres langages dont les programmes comportent une suite d'instructions, le fonctionnement est séquentiel. Les tâches sont accomplies les unes à la suite des autres et ce, une seule à la fois.

Les PLC utilisent pour leurs applications le principe de temps de cycle afin de synchroniser correctement le fonctionnement des entrées et des sorties entre elles. Le dessin donné en **figure 2** montre ce principe. Le temps de cycle est d'une durée fixe, durée pendant laquelle le programme est lu de la première ligne à la dernière ligne et ce,

quelle que soit sa longueur. Comme on le remarque sur la figure, c'est durant cette période fixe que les entrées et les sorties sont mises à jour après une auto inspection.

Comme les microcontrôleurs que nous connaissons, les PLC contrôlent et assurent le fonctionnement de différents types de périphériques (voir **figure 3**) à la différence près qu'ils remplacent les relais dans les équipements d'automatismes et qu'ils peuvent contrôler différents équipements sur des lignes de production.

Les entrées peuvent comporter des capteurs de proximité, de température, de lumière, tandis que les sorties activent des moteurs, des vannes, etc., et ce bien sûr, en fonction du programme qui est inscrit dans leurs mémoires. Les applications possibles



Exemple de Relay Ladder Diagram

Les deux circuits travaillent de façon indépendante sans interférence.

## 1 Symboles graphiques du programme ladder

des PLC sont nombreuses et c'est pour cette raison que le grand public comme les industriels peuvent les utiliser. On peut citer comme exemples l'automatisation de robots mobiles ludiques ou la régulation de la température d'une habitation en fonction de chacune des pièces de celle-ci, applications intéressant directement l'amateur.

## Structure générale d'un PLC

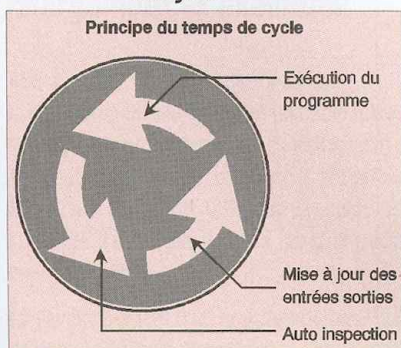
La structure simplifiée d'un PLC est donnée en **figure 4**. On y distingue :

- Le processeur qui se charge de l'exécution du programme
- Le système d'entrées/sorties qui assure la liaison avec l'extérieur
- La mémoire qui stocke les informations

Il existe différents types de mémoires qui peuvent être utilisées :

- La RAM en lecture et en écriture qui perd son contenu en cas de coupure d'alimentation
- La ROM qui ne supporte que la lecture
- L'EEPROM dont le contenu est effacé électriquement. Les données peuvent être modifiées en cours de fonctionnement et subsistent après une coupure d'alimentation
- La FLASH ROM, similaire à l'EEPROM mais dont la méthode de fabrication est différente

## 2 Principe du temps de cycle



## Les différents types de relais

Pour leur fonctionnement, les PLC utilisent différents types de relais. Ils peuvent être de trois sortes :

- Les relais externes connectés aux entrées et aux sorties
- Les relais internes
- Les relais se trouvant uniquement en mémoire

Afin de rendre plus concret ce qui vient d'être énoncé, nous avons choisi un modèle commercialisé pour le grand public et les professionnels. On peut, sans trop se tromper, dire que c'est un échantillon représentatif des PLC.

**M** → relais mémoire (support mémoire) : ce type de relais n'existe que dans la mémoire du PLC et ne peut avoir directement accès sur l'extérieur. Sa fonction est de garder l'information.

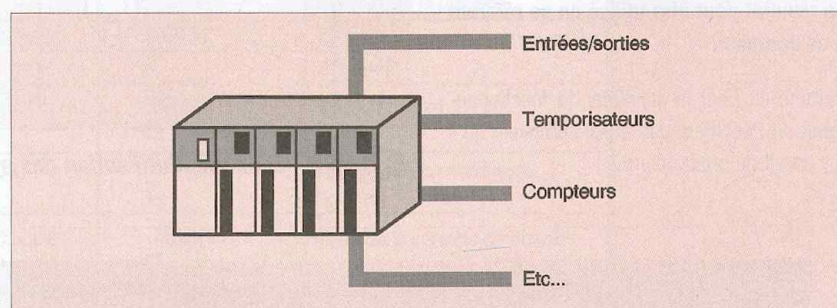
**K** → relais sauvegardés (keep relay) : ils ont un rôle identique au précédent relais. La valeur qui y est stockée est conservée après coupure de l'alimentation. On les utilise si l'information doit être conservée hors tension.

**F** → relais spéciaux (special relay) : c'est une fonction interne du PLC, opérateurs internes ou informations temporisées, comme par exemple, répéter ON et OFF toutes les secondes.

**T** → temporisateurs (timers) : c'est ce type de relais qui contrôle les temporisations qui peuvent être ON ou OFF. La base de temps est de 10 ou 100 ms.

**C** → compteurs (counter) : Ce relais compte un nombre d'impulsions. Si les impulsions comptées sont plus courtes que le temps de cycle, c'est le compteur rapide CNT qui s'en charge.

**D** → données (data) : ce type de relais stocke les données d'opération sous la



## 3 Types de périphériques connectables

Les relais de ce PLC sont classés par des lettres. Hormis les relais K, tous les relais sont à 0 à la mise sous tension ou après un RESET.

Le texte donné ci-dessous, en gras énumère les différents relais ainsi que certains autres composants du PLC :

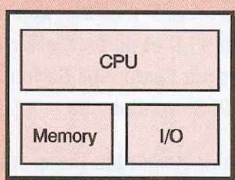
**P** → relais entrées/sorties (I/O relay) : ils sont directement connectés aux entrées et aux sorties. Les entrées peuvent prendre l'état haut ou bas (1 ou 0, ON ou OFF) et sont connectés sur les ports utilisés en entrée. Les sorties peuvent commander des organes électriques.

forme d'un mot de 16 bits. Plusieurs instructions du PLC stockent l'information dans ce domaine.

**S** → relais pas-à-pas (step controller relay) : relais utilisé pour le fonctionnement séquentiel. Le PLC fonctionnant sur le principe du temps de cycle, ce relais permet de résoudre simplement le fonctionnement en séquentiel qui pose parfois des problèmes.

**CH** → affichage LCD : c'est le buffer utilisé pour l'affichage LCD. Si on écrit un code ASCII dans ce domaine, il sera ensuite affiché sur l'écran LCD. Si on écrit un

## Structure générale d'un PLC



La CPU lit les instructions du programme, les interprète et les exécute, les entrées sorties sont chargées de faire la liaison avec l'extérieur, tandis que la mémoire est chargée de stocker les informations.

## 4 Structure schématique d'un PLC

blanc (20H), le caractère est effacé.

G → affichage SGN : c'est le buffer pour l'affichage 7 segments. Si on écrit un code ASCII, il apparaît sur l'afficheur.

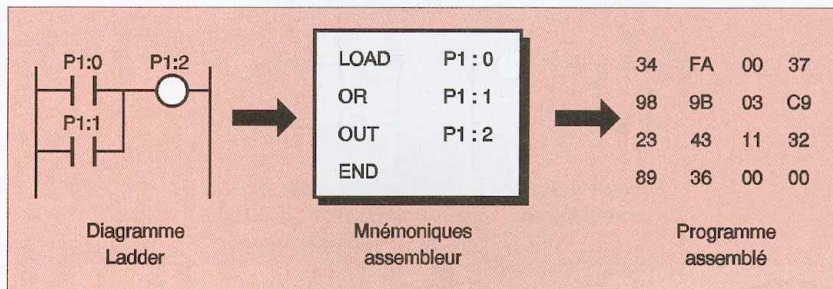
AD → stockage du résultat d'une conversion A/D : c'est dans ce domaine qu'est enregistré le résultat d'une conversion A/D.

Le résultat peut être utilisé en se référant à ce domaine.

CNT → compteur rapide : c'est dans ce domaine qu'est sauvegardée la valeur du compteur rapide.

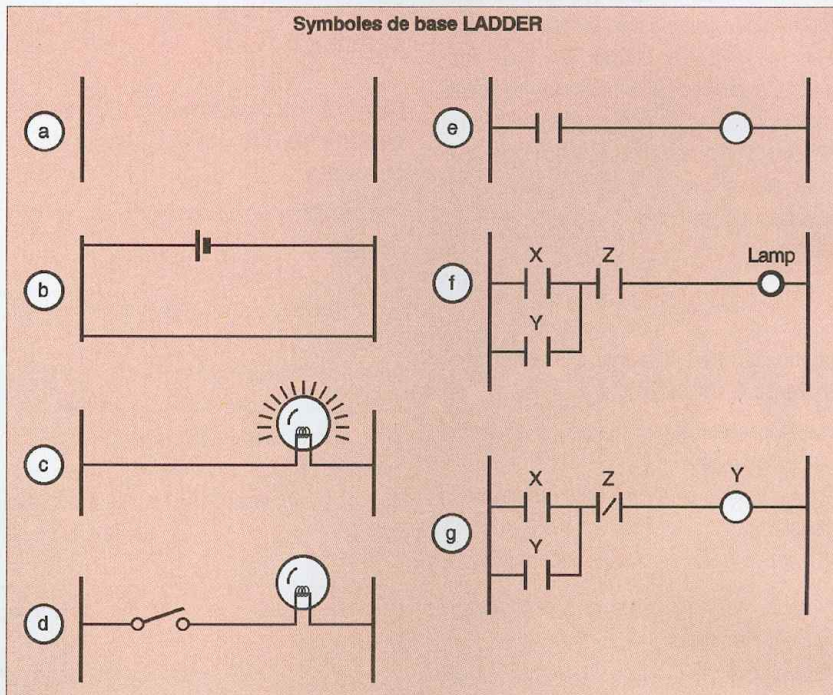
Le résultat peut être utilisé en se référant à ce domaine.

Comme on peut le constater, le fonctionnement du PLC n'est pas aussi complexe qu'il n'y paraît de prime abord.

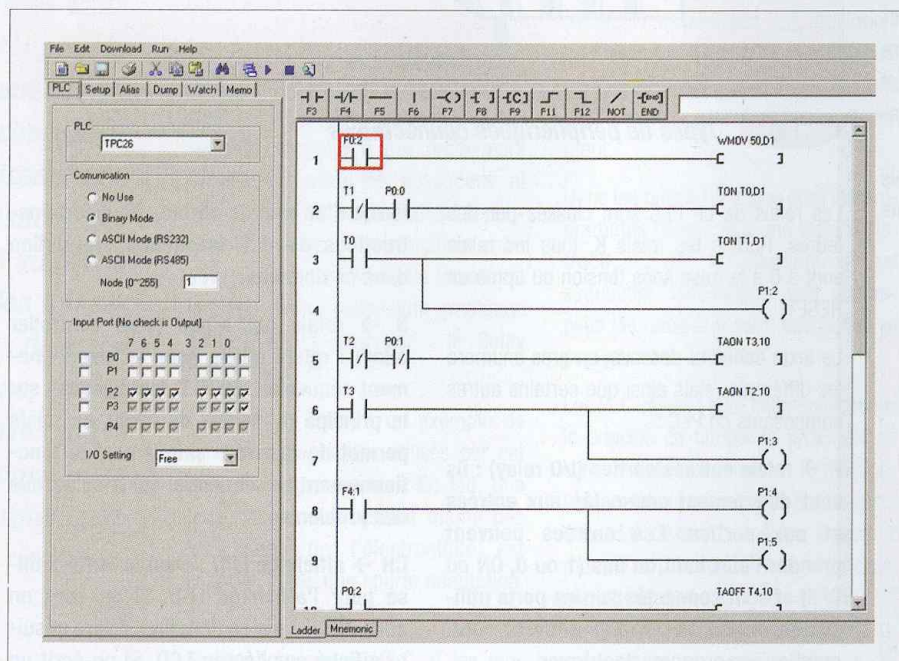


## 6 Processus de programmation d'un PLC

### Symboles de base LADDER



## 7 Exemple d'utilisation des symboles du ladder



## 5 Vue d'écran sous Windows™ d'un programme

## Instructions de base

Nom	Instruction	Symbole	Description
Load	LOAD		Fonction qui commence au point de contact A, NO
Load Not	LOAD		Fonction qui commence au point de contact B, NF
And	AND		Interface série de contacts
And Not	ANDN		Interface série entre NO et NF
Or	OR		Interface parallèle
Orn	ORN		Interface parallèle entre contacts NO et NF
And Stack	ANDS		Interface AND entre blocs
Or Stack	ORS		Interface OR entre blocs
Output	OUT		Sortie, résultat d'une fonction
Not	NOT		Inverse le résultat de la fonction qui la précède
Step sequential set	STEPSET	STEPSET S0:2 	Sortie séquentielle d'un contrôle pas à pas
Step Output	STEPOUT	STEPOUT S0:1 	Sortie d'un pas (LIFO)
Master control set	MCS	MCS 0 	Commence le contrôle de boucles
Master control clear	MCSCCLR	MCSCCLR 0 	Termine le contrôle de boucles

## 8 Instructions de base du ladder

### Le LADDER, chargement et exécution

Afin d'utiliser un PLC, il est nécessaire d'utiliser un programme ladder et un câble reliant la platine de programmation du composant à l'ordinateur. C'est en principe à l'interface RS232 ou à un port USB que ce câble est connecté. Le PC est de plus en plus utilisé pour cette tâche.

Le développement d'un programme pour PLC se déroule en trois étapes :

- Écriture par symboles du programme
- Compilation et téléchargement
- Vérification du fonctionnement

La **figure 5** représente une vue d'écran de l'écriture d'un programme. Le ladder nécessite trois étapes pour effectuer le téléchargement :

- 1 - Compilation du dessin ladder en mnémoniques.
- 2 - Transformation en code par l'assembleur.
- 3 - Téléchargement de l'ordinateur vers le PLC à l'aide du câble.

Le processus est arrêté si une erreur de pro-

grammation est présente. Il suffit simplement de la corriger. Cette suite d'opérations est représentée en **figure 6**.

### Les symboles de base du LADDER

Nous allons maintenant voir que les symboles du ladder sont d'une grande simplicité. La **figure 7** donne deux exemples :

- En a : le symbole ladder consiste en deux lignes de départ.



- En b : ces deux lignes représentent l'alimentation.
- En c : une ampoule qui est connectée à ces deux lignes s'allume.
- En d : en ajoutant un interrupteur, on alimente cette ampoule ou on la place hors tension.
- En e : c'est un diagramme de symboles à relais car le circuit est dessiné par symboles.
- En f : quand plusieurs symboles sont placés ainsi, il s'agit de conditions. X et Y forment un OU et Z forme un ET avec X ou Y. Pour que l'ampoule soit alimentée, il faut soit Z et X, soit Z et Y.
- En g : si X est appuyé, Y se colle car la condition ET avec Z est à ON. Ensuite, Y (OU avec X) maintient le circuit sur ON. Un appui sur Z a pour conséquence le passage de la sortie à OFF.

Nous donnons pour conclure cette brève initiation, trois tableaux représentant en **figure 8** et **9** les instructions de base du ladder et en **figure 10** les commandes de comparaison.

Nous aurons l'occasion, dans un prochain numéro, de revenir sur les PLC, mais cette fois, en abordant le côté pratique.

P. OGUIC

Differential	DF		Front montant (différentiel)
Differential Not	DFN		Front descendant (différentiel)
Set Output	SETOUT	SETOUT P0:0 	Met une sortie à 1
Reset Output	RSTOUT	RSTOUT P0:1 	Met une sortie à 0
Save Status	SAVES	Pas de symbole	Sauve l'état de la fonction en cours
Read Status	RDS	Pas de symbole	Lit l'état sauvegardé
End	END	END 	Fin du programme

### Temps et compteurs

On Timer Base 10 ms	TON	TON T1, 100 	Relais tempo valeur maxi 32767 soit 327,67 s. Quand elle est validée, la tempo démarre, la sortie passe à ON en fin de tempo, sinon reste à OFF
Off Timer Base 10 ms	TOFF	TOFF T1, 100 	Relais tempo valeur maxi 32767 passe à 1 dès validation, se maintient à 1 après dévalidation pendant le temps programmé
On Timer Base 10 ms	TAON	TAON T1, 100 	Base 0,1 s max 3276,7 secondes Fonct. Idem TON
Off Timer Base 10 ms	TAOFF	TAOFF T1, 100 	Idem TOFF mais base de 100 ms Max 3276,7 secondes
Up Counter	CTU	CTU C1, 100 	Compteur (maxi 65535). Augmente de 1 à chaque fois que l'entrée passe à 1, remis à 0 par le reset. La sortie passe à ON quand la valeur programmée est atteinte
Down Counter	CTD	CTD C1, 100 	Décompteur, utilisé pour décompter depuis la valeur programmée (65535 maxi). À chaque fois que l'entrée passe à 1, la valeur diminue de 1. La valeur est réinitialisée à la valeur programmée par le reset

### Commandes de comparaisons

Classification	Format de l'instruction	Description
Commandes de comparaisons Mots 16 bits	=, S1, S2	Dans le cas où S1 = S2, le point de connexion passe à 1. Il compare une valeur de 16 bits
	>, S1, S2	Si S1 > S2, le point de connexion passe à 1
	<, S1, S2	Si S1 < S2, le point de connexion passe à 1
	>=, S1, S2	Si S1 >= S2, le point de connexion passe à 1
	<=, S1, S2	Si S1 <= S2, le point de connexion passe à 1
	<>, S1, S2	Si S1 <> S2, le point de connexion passe à 1
Commandes de comparaisons Mots long 32 bits	D=, S1, S2	Dans le cas où S1 = S2, le point de connexion passe à 1. Il compare une valeur de 32 bits
	D>, S1, S2	Si S1 > S2, le point de connexion passe à 1
	D<, S1, S2	Si S1 < S2, le point de connexion passe à 1
	D>=, S1, S2	Si S1 >= S2, le point de connexion passe à 1
	D<=, S1, S2	Si S1 <= S2, le point de connexion passe à 1
	D<>, S1, S2	Si S1 <> S2, le point de connexion passe à 1

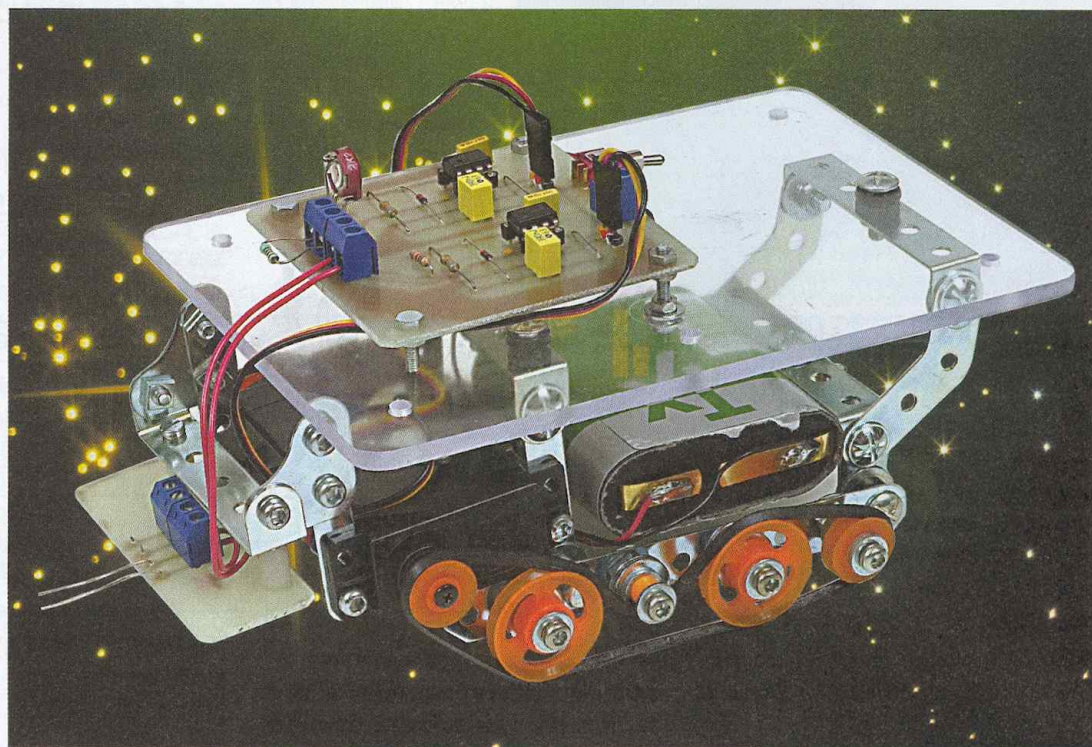
9

Suite des instructions de ladder

10

Les commandes de comparaison

# Mise en œuvre d'un perceptron



**Inventé il y a presque 50 ans, le perceptron est le plus simple des réseaux de neurones et par conséquent le plus facile à réaliser avec un microcontrôleur 16F84.**

Le perceptron est un dispositif identificateur de configuration. Il peut ainsi indifféremment reconnaître des configurations créées par la réunion d'états logiques (0 ou 1) de capteurs, tout comme des images qui ne sont que des configurations de points allumés ou éteints.

La structure typique du perceptron, illustrée sur les figures 1 et 6 n'est pas très différente de celle d'une porte logique.

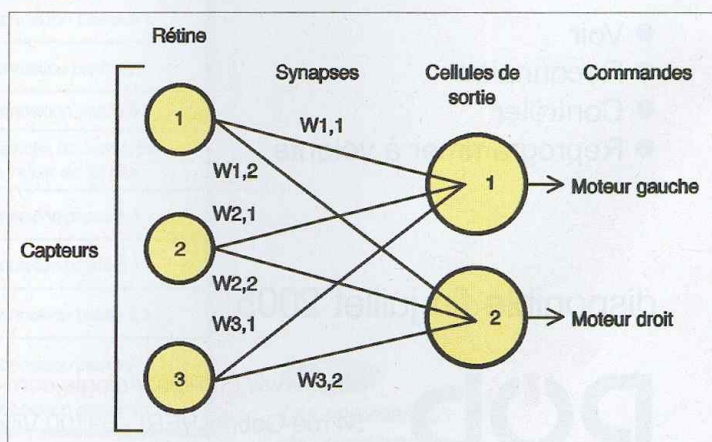
Elle comporte une couche d'entrée, appelée usuellement rétine, formée d'un certain nombre de cellules qui reçoivent l'information logique (0 ou 1) à analyser à leur niveau et qu'elles transmettent par l'intermédiaire de liaisons pondérées excitatrices ou inhibitrices, appelées synapses, aux cellules de la couche de sortie qui intègrent l'ensemble de ces stimulations pour décider de leur propre niveau de sortie logique (0 ou 1). On

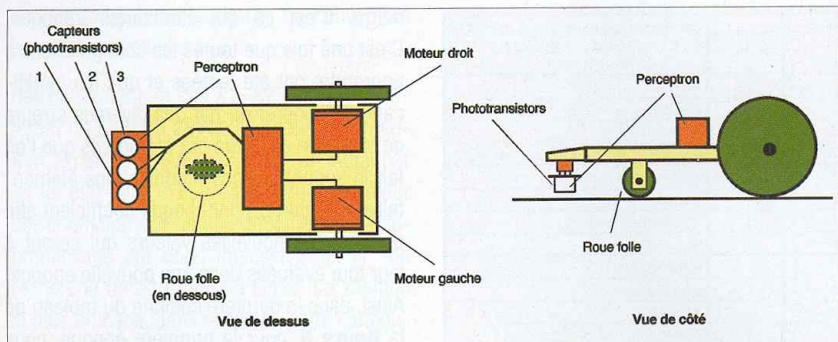
notera que les cellules de sortie sont indépendantes entre elles, il sera donc possible d'étudier le cas de chacune séparément.

Les synapses sont caractérisées par un coefficient d'intensité synaptique, positif ou négatif, noté  $W$ . Dans la

mesure où tous les liens sont potentiellement d'intensité synaptique différente, on utilise une notation indexée. Ainsi, le coefficient  $W$  associé au lien établi entre la troisième cellule de la rétine et la première cellule de la couche de sortie sera noté  $W_{3,1}$ ,

## 1 Perceptron servant à la commande du robot suiveur de ligne





## 2 Robot suiveur de ligne

comme on peut le voir sur l'exemple de la **figure 1**.

Cet exemple décrit un robot suiveur de ligne équipé d'un perceptron dont la rétine ne possède que trois cellules (des phototransistors) et dont les deux cellules de sorties commandent l'activation des moteurs de propulsion droit et gauche (voir **figure 2**). Toute la question est de déterminer la valeur des coefficients  $W$  des différentes synapses qui vont permettre d'associer aux circonstances extérieures qui se présentent au robot les réponses (ou les comportements) que l'on attend de lui et, dans ce cas précis, sa capacité à suivre une ligne tracée sur le sol sans s'en écarter avec en supplément, le choix de le faire tourner à droite en cas de croisement (voir **figure 3**). On a récapitulé dans le tableau de la **figure 4** les correspondances recherchées entre l'état des capteurs d'entrée (les cellules de la rétine) et l'état logique attendu au niveau des cellules de sortie du perceptron.

Pour connaître la réponse d'une cellule de sortie en fonction de la configuration présente au niveau de la rétine, on calcule son

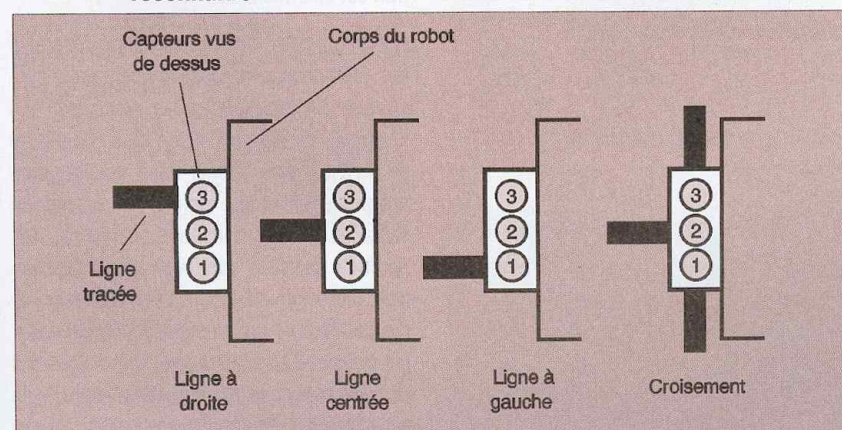
niveau d'activation (noté  $A$ ). Ainsi, pour la cellule 1 de la couche de sortie de la figure 1, la formule sera :

$$A_1 = X_1 \times W_{1,1} + X_2 \times W_{2,1} + X_3 \times W_{3,1}$$

où  $X_1$ ,  $X_2$  et  $X_3$  sont les états logiques (0 ou 1) où se trouvent les cellules de la rétine,  $X_1$  pour la cellule de rétine 1,  $X_2$  pour la cellule de rétine 2, etc... À priori, les valeurs des coefficients synaptiques  $W$  sont des valeurs réelles quelconques. On remarquera au passage que lorsqu'une cellule de la rétine est inactive ( $X=0$ ), cette cellule n'a pas d'influence sur la cellule de sortie, ce qui impliquera, comme nous le verrons, que seules les synapses liées aux cellules de rétine actives verront leur coefficient  $W$  modifié lors de l'apprentissage.

Une fois le niveau d'activation  $A$  de la cellule de sortie calculé, il suffit de le comparer au seuil d'activation que l'on a fixé, qui est le même pour toutes les cellules de sortie. Dans la majorité des cas et notamment dans cet article, ce seuil d'activation  $S$  est tout simplement zéro. Ainsi, si  $A_1$  est strictement supérieur à  $S=0$ , alors l'état de la cellule de sortie sera 1 sinon ce sera 0.

## 3 Les quatre configurations que le perceptron doit apprendre à reconnaître



À l'état initial, il est souvent convenu de tirer les coefficients synaptiques  $W$  au hasard mais il est beaucoup plus simple, surtout lorsqu'il est question de réaliser le perceptron avec un microcontrôleur, de prendre tous ces coefficients nuls au départ. Bien entendu, dans ces conditions, il est peu probable que le perceptron puisse réaliser la moindre association correcte entre les configurations d'entrée et les états de sortie désirés. C'est ici que sert réellement l'analogie avec les véritables neurones puisque c'est en ajustant les valeurs des différents coefficients  $W$  d'intensité synaptique, par apprentissages successifs, que l'on parviendra à obtenir le résultat recherché et ainsi, bien qu'il soit quasiment impossible de prévoir à l'avance quelles seront les valeurs qui conviendront, cela parce que l'on ne possède pas d'information a priori sur la fonction logique à remplir, le perceptron sera capable de trouver la solution sans jamais avoir à connaître cette fonction.

Rétine			Action à effectuer	Cellules de sortie	
Niveau de sortie des capteurs 1	2	3		Commande des moteurs gauche	droit
0	0	1	Tourne à droite	Actif 1	Inactif 0
0	1	0	Avance tout droit	Actif 1	Actif 1
1	0	0	Tourne à gauche	Inactif 0	Actif 1
1	1	1	Tourne à droite	Actif 1	Inactif 0

## 4 Table servant à l'apprentissage

Comme nous l'avons dit précédemment, les deux cellules de sorties (voir **figure 1**) sont indépendantes, nous nous concentrerons donc sur la cellule de sortie n°1, dont toute la procédure d'apprentissage est exposée sur la **figure 5**.

La loi d'apprentissage du perceptron s'appelle règle de Widrow-Hoff et son principe peut être compris en se rappelant l'observation faite peu avant : seules les cellules de rétines actives ( $X=1$ ) voient leur coefficient synaptique  $W$  modifié en cas d'erreur et cette modification s'opère de la façon suivante : si la cellule de sortie répond 0 au lieu de 1, son niveau d'activation  $A$  est trop bas et les coefficients  $W$  concernés seront augmentés d'une valeur  $N$ . Si la cellule de sortie répond 1 au lieu de 0, son niveau d'activation  $A$  est trop élevé et les coefficients  $W$  concernés seront diminués de la même valeur  $N$ .

Epoque	configuration d'entrée			Valeur des coefficients synaptiques			Réponse de la cellule de sortie 1 (moteur gauche)	Réponse attendue	Correction élémentaire à apporter aux coefficients synaptiques en fin d'époque
	X1	X2	X3	W1,1	W2,1	W3,1			
1	0	0	1	0	0	0	$0*0+0*0+1*0=0 \rightarrow 0$	1	Ajouter 1 à W3,1
	0	1	0	0	0	0	$0*0+1*0+0*0=0 \rightarrow 0$	1	Ajouter 1 à W2,1
	1	0	0	0	0	0	$1*0+0*0+0*0=0 \rightarrow 0$	0	Aucune
	1	1	1	0	0	0	$1*0+1*0+1*0=0 \rightarrow 0$	1	Ajouter 1 à W1,1 W2,1 et W3,1
2	0	0	1	1	2	2	$0*1+0*2+1*2=2 \rightarrow 1$	1	Aucune
	0	1	0	1	2	2	$0*1+1*2+0*2=2 \rightarrow 1$	1	Aucune
	1	0	0	1	2	2	$1*1+0*2+0*2=1 \rightarrow 1$	0	Retrancher 1 à W1,1
3	1	1	1	1	2	2	$1*1+1*2+1*2=5 \rightarrow 1$	1	Aucune
	0	0	1	0	2	2	$0*0+0*2+1*2=2 \rightarrow 1$	1	Aucune
	0	1	0	0	2	2	$0*0+1*2+0*2=2 \rightarrow 1$	1	Aucune
	1	0	0	0	2	2	$1*0+0*2+0*2=0 \rightarrow 0$	0	Aucune
	1	1	1	0	2	2	$1*0+1*2+1*2=4 \rightarrow 1$	1	Aucune

## 5 Étapes de l'apprentissage des configurations par la cellule de sortie n° 1

En cas de bonne réponse, les coefficients d'intensité synaptique W restent inchangés. Dans le cadre de cet article, on prendra N=1 (N est appelé coefficient d'apprentissage). D'autres valeurs n'ont de sens que lorsque l'on décide de rendre ce paramètre variable en cours d'apprentissage, ce qui ne sera pas le cas ici.

Les modifications à opérer sur les coefficients synaptique W sont notées dans la dernière colonne du tableau de la figure 5 (par exemple : ajouter 1 à W2,1) mais ces modifications ne sont pas effectuées immédiatement. Les coefficients W gardent la même valeur (ici 0) lors des tests des quatre configurations que la cellule de sortie doit recon-

naître. C'est ce qui caractérise l'époque. C'est une fois que toutes les configurations à apprendre ont été testées et que les modifications à opérer du fait des diverses erreurs de la cellule de sortie ont été notées que l'on fait la somme de ces modifications élémentaires à apporter pour chaque coefficient afin d'obtenir de nouvelles valeurs qui seront à leur tour évaluées dans une nouvelle époque. Ainsi, dans la dernière colonne du tableau de la figure 5, pour la première époque, nous pouvons faire le bilan suivant :

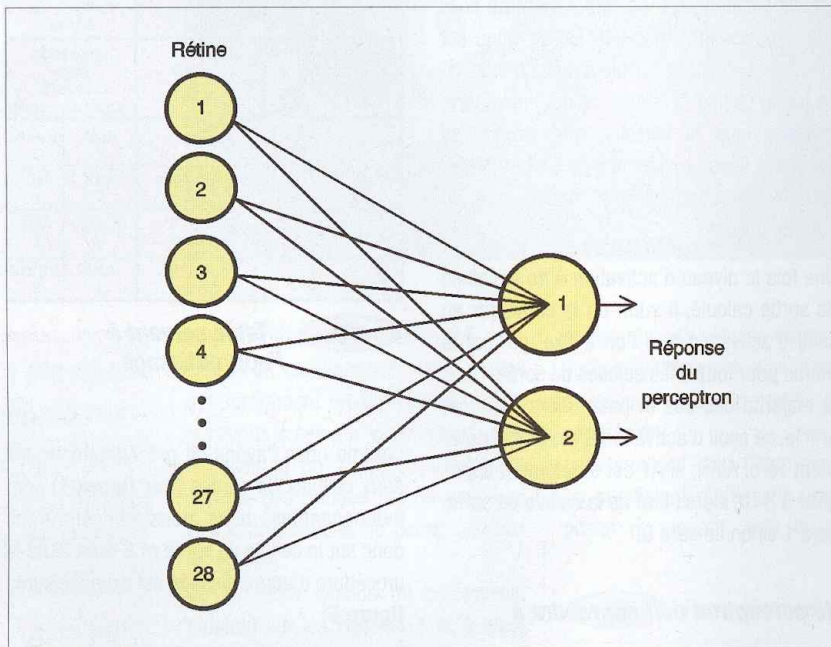
W1,1 = 0 doit être incrémenté de 1 donc W1,1 = 1

W2,1 = 0 doit être incrémenté de 1+1=2 donc W2,1 = 2

W3,1 = 0 doit être incrémenté de 1+1=2 donc W3,1 = 2

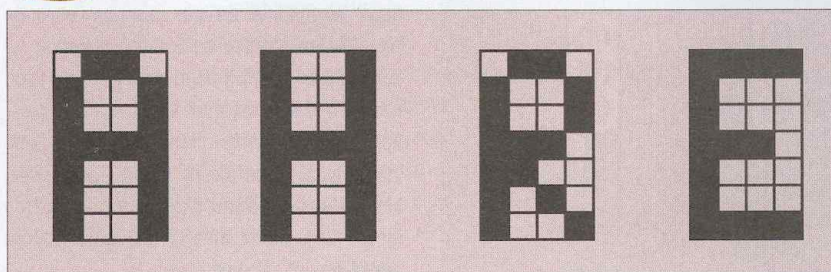
Ces valeurs sont reportées dans le tableau dans la colonne « valeur des coefficients synaptiques » pour l'époque 2.

Il suffit alors de reprendre toute la procédure



## 6 Structure du perceptron identificateur de lettres

## 8 Les quatre lettres que le perceptron apprend à identifier



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28

## 7 Rétine du perceptron en forme de matrice de 4 x 7 cellules

de test précédente jusqu'à ce que la cellule de sortie ne commette plus d'erreur.

À cet instant, son apprentissage est terminé et les valeurs des coefficients W définitives. Une fois que les deux cellules de sortie ont appris à reconnaître les quatre configurations d'entrée décrites dans la figure 3, le perceptron est opérationnel.

Avant de passer à un exemple plus général, il faut faire deux remarques concernant le perceptron. En premier lieu, le perceptron est incapable de différencier ce qu'il connaît de ce qu'il n'a pas appris, toute configuration présentée en entrée donne un résultat en sortie et si cette configuration n'a pas fait partie de sa liste d'apprentissage, la réponse sera tout simplement imprévisible. Dans le cas du robot suiveur de ligne, ceci peut advenir en cas de chevauchement de la ligne par deux capteurs simultanément. Le plus simple est d'ajouter cette configuration à la liste et

de reprendre l'apprentissage de l'ensemble de cette nouvelle liste.

Deuxièmement, le cas  $X1=X2=X3=0$  est différent de tous les autres cas pour le perceptron. En effet, le niveau d'activation induit par un tel état vaut toujours zéro. Aucune valeur des coefficients  $W$  ne peut modifier cela, l'apprentissage est donc impossible ici si l'on recherche un état haut en sortie. La solution la plus efficace, dans le cas où cette configuration est réellement importante, consiste à ajouter (avant l'apprentissage) une quatrième cellule de rétine établissant un lien  $W4,1$  avec la première cellule de sortie et  $W4,2$  avec la seconde cellule de sortie tel que l'état logique  $X4$  de cette quatrième cellule de rétine soit toujours 1. Ainsi, on évite une configuration d'entrée où tous les états logiques sont bas et le perceptron n'a plus de difficulté à apprendre.

Le second perceptron que nous allons étudier est en fait une généralisation du précédent. Sa structure est décrite sur la **figure 6**. Il possède 28 cellules de rétine (ici encore des phototransistors), liées à deux cellules de sortie. Les cellules de la rétine sont disposées en une matrice 4 x 7 comme l'indique la **figure 7**.

Le rôle de ce perceptron est de reconnaître 4 lettres qu'il a précédemment apprises. On trouvera sur la **figure 8** les quatre lettres à identifier (dessinées par exemple sur des calques ou des transparents) et sur le tableau de la **figure 9** la réponse attendue en fonction de la lettre présentée sur la rétine.

La procédure d'apprentissage est semblable en tous points à celle de l'exemple précédent. On présente successivement chaque lettre à apprendre sur la rétine et l'on détermine la réponse des cellules de sortie en calculant d'abord leur niveau d'activation selon la formule :

$$A1 = X1 \times W1,1 + X2 \times W2,1 + \dots + X27 \times W27,1 + X28 \times W28,1$$

pour la première cellule de sortie et

$$A2 = X1 \times W1,2 + X2 \times W2,2 + \dots + X27 \times W27,2 + X28 \times W28,2$$

pour la seconde, puis en comparant ces niveaux  $A1$  et  $A2$  au seuil  $S=0$  pour déterminer leur état logique (0 ou 1).

S'il y a des erreurs, on utilise la règle de Widrow-Hoff pour calculer les corrections élémentaires à opérer sur les coefficients  $W$  des cellules de rétine alors actives. Une fois les quatre lettres présentées, l'époque en cours est achevée et l'on fait la somme des

Lettre	Réponse du perceptron	
	cellule 1	cellule 2
A	0	0
H	0	1
R	1	0
E	1	1

## 9 Table qui servira à l'apprentissage

corrections à faire pour chaque coefficient afin de déterminer sa nouvelle valeur, puis l'on reprend toute la procédure jusqu'à ce que le perceptron ne commette plus d'erreur.

Le lecteur trouvera sur la **figure 10** un tableau illustrant l'évolution des coefficients  $W$  d'intensité synaptique entre les cellules de la rétine et les cellules de sortie en fonction des époques (Le coefficient d'apprentissage  $N=1$ ). On remarque que le perceptron n'a besoin que de trois époques pour savoir identifier les quatre lettres A, H, R et E. Sur la **figure 11** est présenté un exemple de schéma de réalisation électronique de ce perceptron.

## 10 Modification de la valeur des coefficients synaptiques durant l'apprentissage

Valeur du Coefficient synaptique entre:	Cellule de sortie					
	1		2		3	
Cellule de la rétine	1	2	1	2	1	2
1	0	0	1	2	0	2
2	0	0	2	1	1	-1
3	0	0	2	1	1	-1
4	0	0	1	2	0	2
5	0	0	2	2	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	1	1	-1	-1
9	0	0	2	2	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	1	1	-1	-1
13	0	0	2	2	0	0
14	0	0	2	2	1	1
15	0	0	2	2	1	1
16	0	0	0	1	-2	0
17	0	0	2	2	0	0
18	0	0	1	0	1	-1
19	0	0	0	0	0	0
20	0	0	0	1	-2	0
21	0	0	2	2	0	0
22	0	0	0	0	0	0
23	0	0	1	0	1	-1
24	0	0	0	1	-2	0
25	0	0	2	2	0	0
26	0	0	1	1	1	1
27	0	0	1	1	1	1
28	0	0	2	2	0	0
Epoque	1	2	3			

tron avec l'usage d'un simple PIC16F84. Le multiplexage de la matrice 4 x 7 de la rétine est réalisé à l'aide des décodeurs/multiplexeurs 74LS138 et 74LS153 IC1 et IC2. Une telle rétine demande un éclairage infrarouge alimenté en courant continu (un éclairage pulsé dérouterait le programme) très puissant.

Une matrice de 28 leds infrarouges placée à 1 cm des phototransistors convient (chaque led étant alimentée sous 5 V / 470  $\Omega$ ). Les lettres pourront être dessinées en noir sur du papier calque et seront glissées dans la fente entre les deux matrices posées sur la rétine.

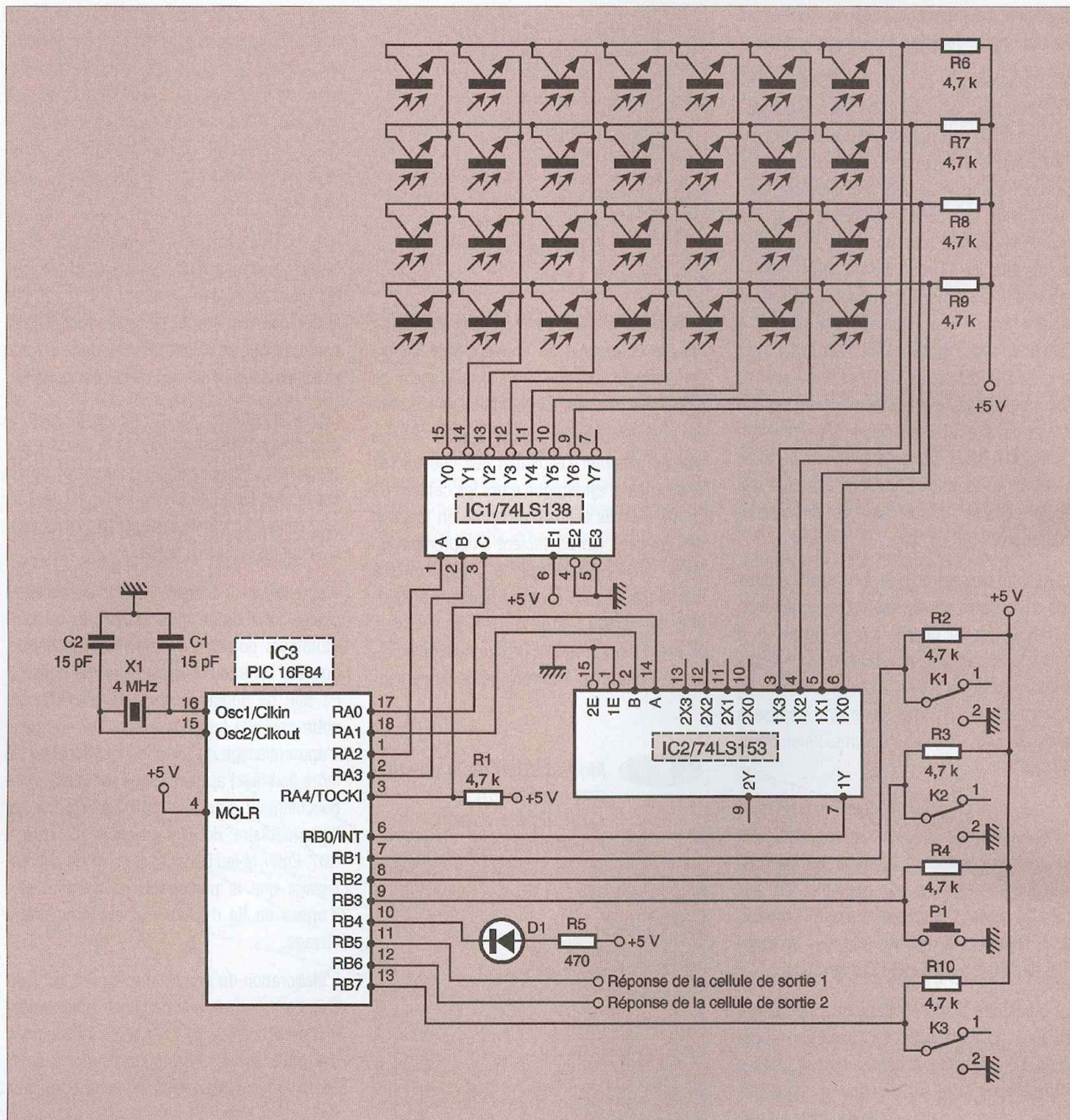
Les interrupteurs K1 et K2 sont sous le contrôle de l'utilisateur lors de la phase d'apprentissage et indiquent au perceptron quelle est la réponse attendue en sortie. K1 sert de référence pour la cellule de sortie 1 et K2 pour la seconde cellule de sortie.

Bien entendu, à chaque changement de lettre placée sur la rétine, les positions de ces commutateurs doivent être modifiées en conséquence. On appuiera ensuite sur le poussoir P1 afin de demander au perceptron de lire cette nouvelle lettre, que l'on soit en phase d'apprentissage ou en mode identificateur de lettre (lorsque l'apprentissage est terminé). Le basculement d'un mode à l'autre se fait par l'intermédiaire de l'interrupteur K3 relié à RB7. Enfin, la led D1 indique en mode apprentissage que le perceptron ne commet plus d'erreur en fin d'époque et est donc prêt à l'usage.

L'élaboration du programme du PIC est facilitée par les choix que nous avons fait précédemment :  $S=0$ ,  $N=1$  et  $W=0$  initialement, car nous avons ainsi la certitude que les coefficients synaptiques  $W$  seront toujours des entiers signés.

Le tableau de la **figure 10** nous montre que leurs valeurs restent faibles après apprentissage, aussi est-il possible de choisir de les représenter par des entiers signés (complément à 2) sur un octet, soit de -128 (1000 0000) à +127 (0111 1111). Ils seront stockés dans l'EEPROM, 28 coefficients  $W$  étant associés à chaque cellule de sortie.

Les sections les plus importantes du programme sont indiquées sur le synoptique de la **figure 12** : la routine REPONSE détermine la réponse d'une cellule de sortie en fonction de la lettre posée sur la matrice et cette routine sert autant durant la phase d'apprentissage que durant celle de reconnaissance. Elle cal-



**II** Schéma de principe du perceptron identificateur de lettres

cule le niveau d'activation sur 16 bits de façon à être sûr d'éviter tout débordement et donne son résultat (cellule de sortie active ou non) sous la forme de l'état du bit 7 de 0x0F.

(Les valeurs indiquées en gras entre parenthèses correspondent à la seconde cellule de sortie).

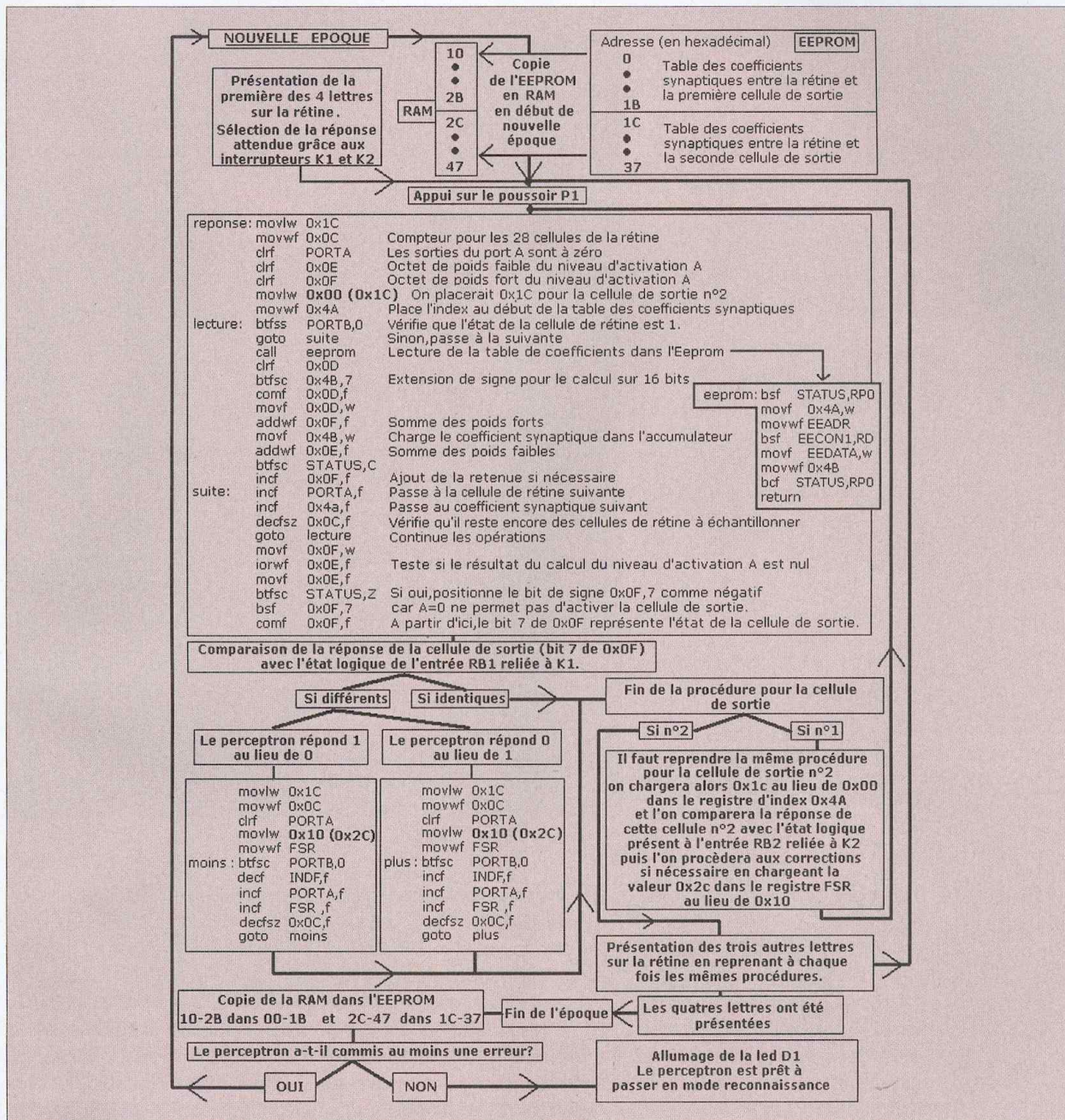
Ce perceptron peut en fait apprendre à reconnaître plus de quatre lettres, simplement il ne peut les classer qu'en quatre catégories puisque les cellules de sorties ne peuvent

définir que quatre états différents. Opérer ainsi un classement des 26 lettres de l'alphabet, de dessins ou d'autres signes est donc possible.

De même, en ajoutant par exemple 4 autres cellules de sorties, jusqu'à 64 catégories peuvent être définies. L'apprentissage risque cependant d'être long dans la mesure où chaque époque doit contenir la totalité de ce qu'un perceptron doit savoir reconnaître puisqu'il est impossible de procéder par simple ajout de nouvelles connaissances ce qui ris-

querait de faire oublier les données apprises auparavant.

Les limites à partir desquelles le perceptron ne parvient plus à apprendre ne sont pas faciles à fixer a priori, surtout que vient s'ajouter une autre difficulté : il existe quelques combinaisons qu'un perceptron ne peut pas apprendre, ainsi, un réseau à deux cellules de rétine et une cellule de sortie est incapable d'apprendre la fonction XOR (OU exclusif), mais cela reste un risque avant tout théorique et d'autant plus faible lorsque le



## 12 Synoptique de la routine d'apprentissage

nombre de cellules de rétine est important et que les figures apprises deviennent complexes et diverses.

Le principal avantage du perceptron, par rapport à un simple logiciel d'identification point à point d'images mémorisées, est qu'il demande moins de mémoire pour fonctionner et répond au moins aussi rapidement puisque l'évaluation de la réponse ne demande qu'une simple addition de coefficients synaptiques.

Ainsi, pour classer en 4 groupes les 26 lettres

de l'alphabet, le perceptron aura besoin de  $2 \times 28 = 56$  octets correspondants aux coefficients synaptiques  $W$  associés aux deux cellules de sorties alors que la mémorisation de 26 lettres matricées  $4 \times 7$  demande, elle,  $4 \times 7 \times 26$  bits soit 91 octets en théorie.

Notons que plus le nombre de cellules de sortie du perceptron est important, plus le rapport lui est favorable.

Par contre, une procédure d'apprentissage est beaucoup plus longue qu'une simple mémori-

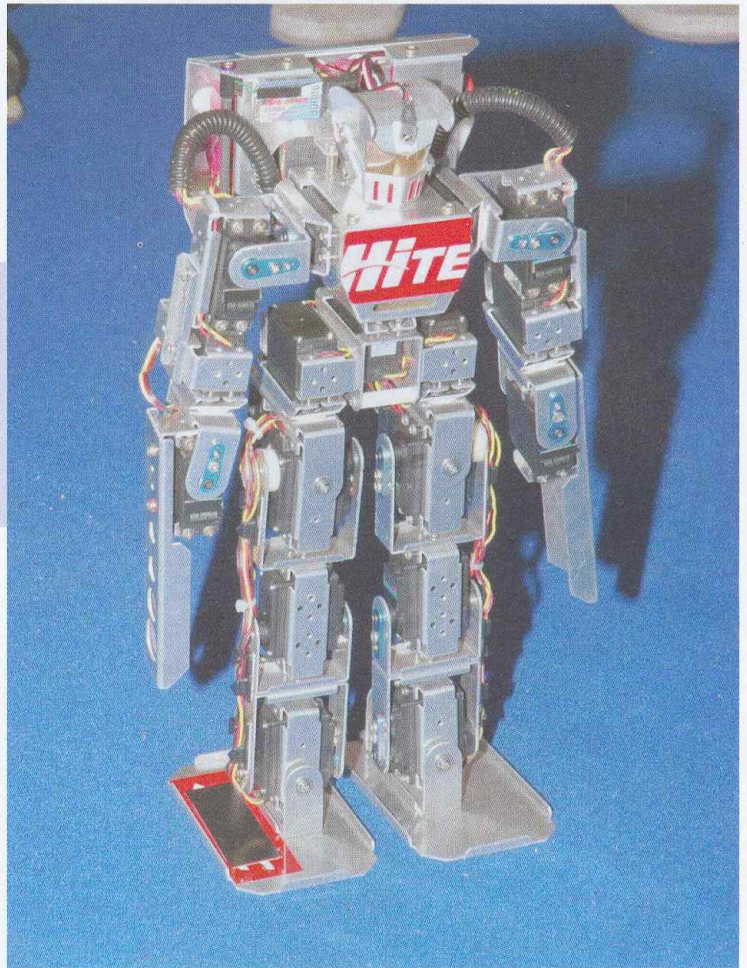
sation de données par stockage direct et, de plus, le perceptron ne peut restituer les signes qu'il a appris contrairement à une simple mémoire graphique.

Le lecteur curieux d'en savoir davantage trouvera de nombreux compléments sur Internet ou bien dans les ouvrages consacrés aux réseaux de neurones.

D. VIACAVAL

## Robot Hitec

*21 servomécanismes pour le robot Hitec. Il n'a pas encore de nom mais vous pourrez l'acquérir en kit...*



**À force de voir ses servomécanismes utilisés pour la réalisation de robots, Hitec a mis à son catalogue un servomécanisme spécialisé : le HSR-5995TG. Il utilise les dernières technologies en matière de servo : pignons en titane et électronique numérique. Les premiers permettent d'obtenir un couple encore plus important associé à une longue durée de vie.**

Il atteint 30 kg/cm avec un temps de réaction de 0,12 seconde. Sa sortie en acier est montée sur un double roulement à billes. La course a été portée à 180° tandis qu'un palier a été installé sur la face opposée à la sortie traditionnelle. La gestion numérique du servo permet d'ajuster ses paramètres comme le point central et les extrémités, la largeur de la bande morte ainsi que la vitesse. Hitec a aussi mis à son catalogue des éléments de montage qui permettront de l'installer facilement dans un robot.

Pour illustrer la qualité, la rapidité et la puissance de ce servomécanisme, Hitec a conçu un robot présenté au dernier salon du modèle réduit et mobilisant 21 de ces servos. Les membres du robot alignent les servos montés dans des étriers d'aluminium. Le palonnier métallique de sortie du servo est vissé sur la pièce à commander. Chaque jambe a 4 servos, le pied en a 2. Chaque bras en utilise 4 et

le dernier sert à la rotation du torse, ce qui nous fait bien 21 axes. La tête est solidaire du torse et reçoit un capteur infra-rouge de télécommande.

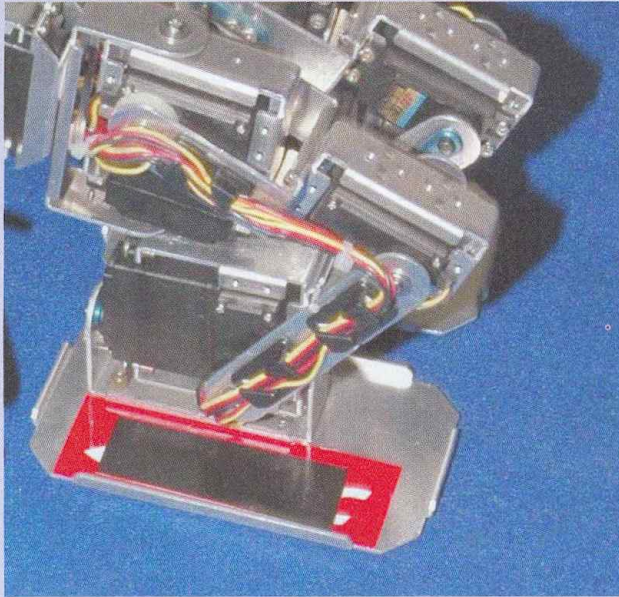
L'appareil est alimenté par une batterie au lithium dont les 2 éléments apportent les 7,4 V nécessaires à l'obtention du maximum de couple des servos. Batterie et électronique s'installent dans le dos de la machine.

Le microcontrôleur se programme par ordinateur, des séquences de commandes peuvent être enregistrées et rappelées par la télécommande afin d'animer le robot d'une façon plus simple que de programmer les 21 servos un à un, ce que l'on pourra faire dans un second temps, une fois les instructions acquises. Hitec a pour le moment présenté un prototype, le développement d'une version commerciale est en cours avec un prix qui se situerait aux environs de 1000 euros, chiffre qui sous entend l'utilisation de servomécanismes plus écono-

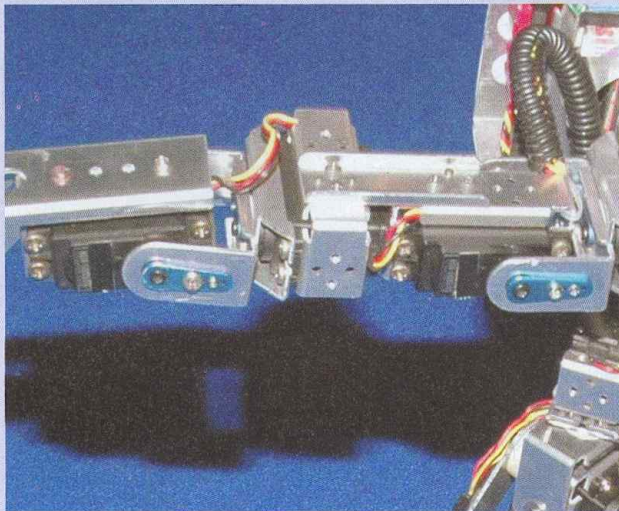
miques que ceux utilisés pour le prototype. On attend aussi un gyroscope qui permettrait à l'engin de bénéficier d'une meilleure stabilité, il lui arrive parfois de tomber, mais il sait se remettre debout tout seul. Il lui sera alors possible de monter un escalier... Cette absence de gyroscope lui donne parfois un équilibre incertain lors de la marche. Il sera également intéressant de pouvoir ajouter des capteurs au robot pour lui donner de l'autonomie. Le principe même de la commande directe des servos et leur réversibilité impose une alimentation permanente lorsqu'on désire le voir maintenir une position debout. Une fois l'alimentation coupée, les articulations deviennent souples et les forces de gravité reprennent le dessus. Les quelques photos montrent le robot en opération. Dommage, nous n'avons pas la vidéo !

E. LEMERY

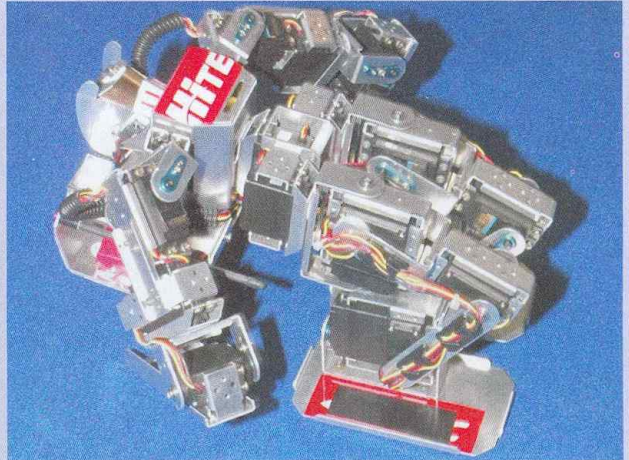




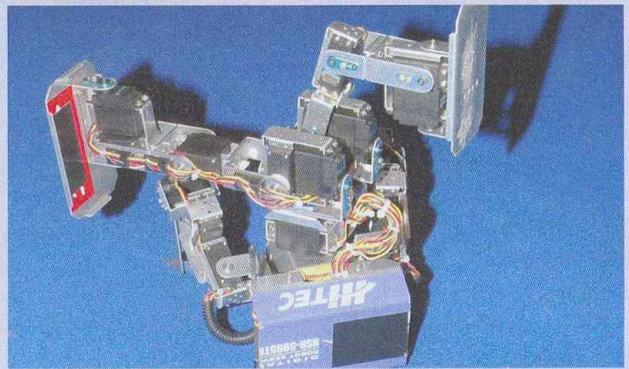
Détail de l'articulation du genou du robot humanoïde. L'articulation du pied a deux axes donc deux servos



Détail du bras, on voit ici la transmission du mouvement par les palonniers métalliques des servos. Les éléments d'aluminium entourent les servos



Vous voyez, je me relève tout seul...



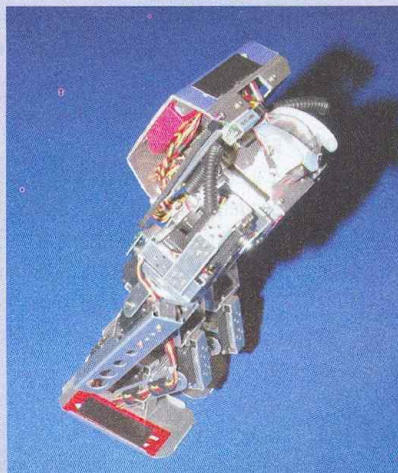
Merci pour la musique, la danse hip hop ne me fait pas peur...



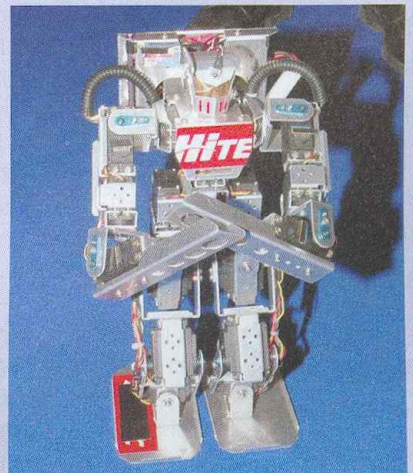
Un robot d'envergure...



Zut, le robot est tombé... Il n'a pas de gyroscope mais il bouge encore...

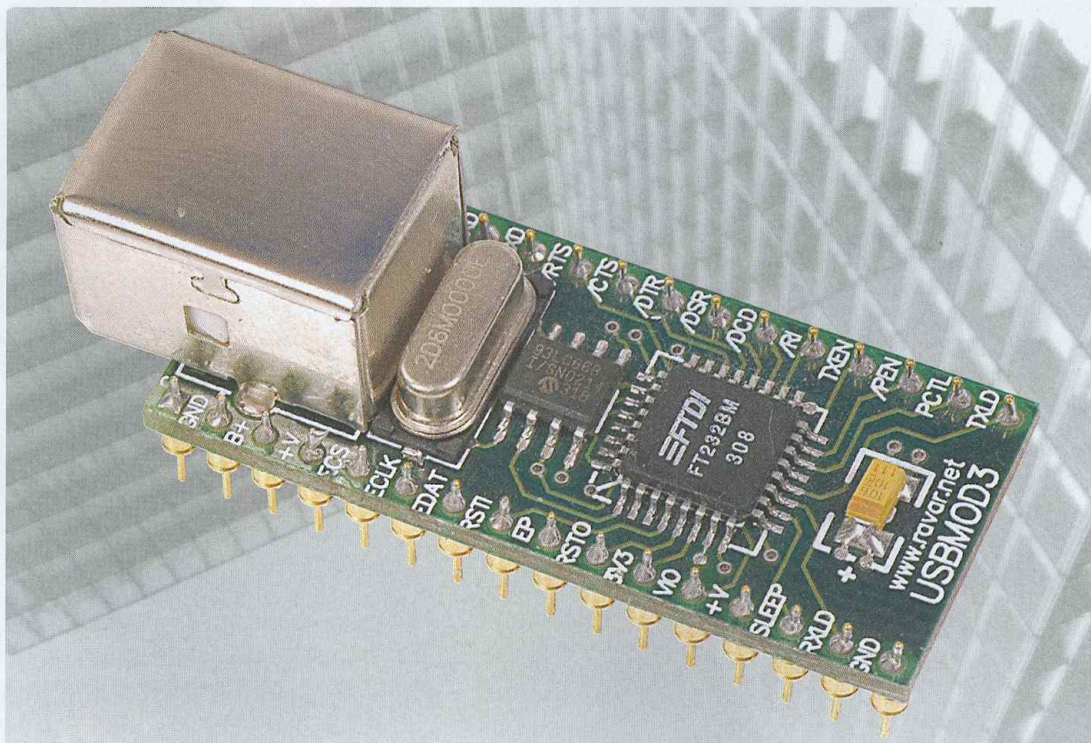


Un petit rétablissement et c'est fait... Je suis remis sur mes pieds



Ça y est, j'ai fini, alors, on m'applaudit ?

# Personnalisation d'un montage USB



**Les modules de conversion USB-RS232 tels que l'USBMOD1 ou l'USBMOD3 offrent une possibilité de personnalisation à travers la présence d'une EEPROM externe pour l'USBMOD1 et interne au module pour l'USBMOD3. Cet article propose de voir comment une telle personnalisation se réalise et quel en est l'intérêt.**

Lors de la connexion d'un montage USB au PC, il y a une phase dite d'énumération au cours de laquelle le montage envoie au PC un certain nombre d'informations pour décrire le montage, le type d'alimentation et régler plusieurs paramètres concernant le protocole de communication USB. Toutes ces informations sont figées dans une mémoire du montage USB. Lorsque l'on réalise un montage USB à base de 68HC908JB8, elles seront placées dans la mémoire Flash (qui contient aussi le programme utilisateur) du microcontrôleur. Il incombe au concepteur le renseignement précis et exhaustif de l'ensemble de ces données.

Dans le cas d'un montage à base du circuit FT232AM ou BM, comme le sont par exemple les modules USBMOD1/3, un jeu de données par défaut est fourni par le circuit lui-même, de sorte qu'il puisse s'intégrer dans un maximum de réalisations sans autre

forme de procès. Cette description par défaut peut s'avérer réductrice. Heureusement, FTDI a eu la bonne idée d'ajouter au FT232 la possibilité d'aller lire dans une EEPROM externe quelques informations de configuration personnelle en remplacement des standards fournis.

Pour illustrer notre propos, on utilisera ici l'USBMOD3 qui présente le gros avantage d'avoir cette EEPROM intégrée sur le module... et pour le même prix que l'USBMOD1.

Il y aura nécessité de bien maîtriser la séquence installation / désinstallation des pilotes, ce que nous ferons dans un premier temps. On commencera par télécharger les dernières versions des deux types de drivers proposés pour le système d'exploitation :

Sur le site [www.elexol.com](http://www.elexol.com), aller à la rubrique Support > Downloads > USBMOD3&4. Dans la partie consacrée à l'USBMOD3, télécharger :

- Le driver VCP (Virtual COM Port) qui

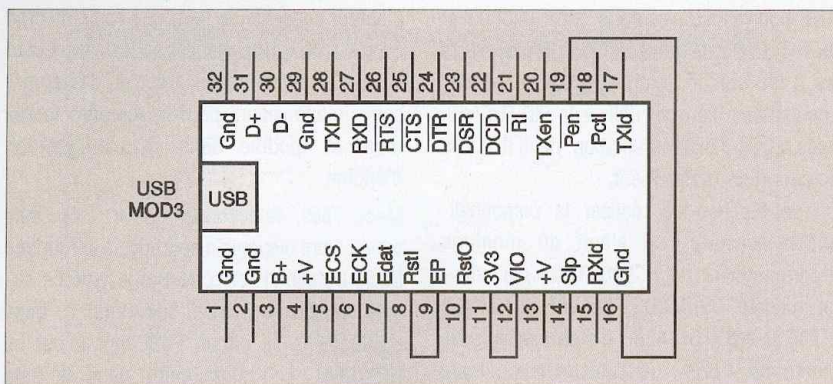
permet d'utiliser le module USB comme un port RS232 ordinaire.

- Le driver D2XX, un driver qui permet d'utiliser le module comme un "vrai" montage USB.

En fait, Elexol a simplement créé un lien de redirection vers [www.ftdi-chip.com](http://www.ftdi-chip.com), le site du concepteur du circuit FTDI232.

Une fois le téléchargement terminé, on se retrouve avec deux fichiers de type archive ZIP. L'auteur vous conseille de décompresser le premier dans un dossier C:\ftdi\vcpl\ et le second dans C:\ftdi\d2xx\). Ces dénominations et emplacements n'ont rien d'obligatoire mais permettront de mieux suivre l'ensemble des procédures que l'on utilisera. Les procédures qui suivent ont été réalisées sous Windows XP Service Pack 2.

Commençons par décrire la démarche à suivre pour une installation standard.



## 1 Câblage minimum de l'USB MOD3

### L'installation standard

Avant de réaliser l'installation, il est bon de vérifier la configuration de Windows en ce qui concerne l'acceptation ou non de pilotes certifiés et l'acceptation ou non d'une connexion à Windows Update pour la recherche des pilotes. Windows XP étant très sécurisé, il peut arriver qu'une installation échoue pour des raisons de "sécurité". L'auteur pense qu'il est préférable de rester maître du système et donc d'éviter les procédures automatiques. Dans Panneau de Configuration > Système > Matériel > Windows Update, sélectionner "Me demander de rechercher un pilote sur Windows Update..." et dans Signature Pilote, sélectionner "Me demander de choisir..."

Connecter un montage à base d'USBMOD3, par exemple le programmeur pour HC908QT/QY. (Remarque : contrairement au module USBMOD1, l'USBMOD3 n'est pas détecté par le PC si on le connecte directement à l'USB : certaines broches du module doivent être connectées entre elles pour que la détection ait lieu (figure 1).

Le système d'exploitation va détecter un nou-

veau périphérique à base d'USBMOD3 et va réclamer les pilotes. On a alors la possibilité d'aller les chercher avec Windows Update : refuser en sélectionnant "Non, pas pour cette fois". Puis, faire le choix "Installer à partir d'une liste..." ; ensuite sélectionner "Rechercher le meilleur pilote dans ces emplacements", cocher "Inclure cet emplacement dans la recherche" et préciser l'emplacement : ici c:\ftdi\vcpl. Comme il y a deux fichiers pilotes à installer, on devra repreciser une deuxième fois cet emplacement. Mais avant il faut franchir l'avertissement de sécurité concernant les pilotes non validés en cliquant sur "Continuer".

Le montage est prêt à fonctionner sur le nouveau port série virtuel qui apparaît.

C'est ainsi que la grande majorité des utilisateurs de ces modules les utilisent.

Pourtant, cette installation "par défaut" peut présenter plusieurs inconvénients.

Tout d'abord, pour un montage alimenté par l'USB, l'intensité réservée sur le hub est limitée à 100 mA. Cette option choisie par le fabricant assure en effet une compatibilité avec tous types d'ordinateurs (PC de bureau alimenté par le secteur ou PC portable ali-

menté par sa batterie). Cette valeur de 100 mA peut dans certains montages être insuffisante, alors qu'il est possible de soutirer jusqu'à 500 mA d'un port USB (si le hub qui supporte ce port est alimenté par le secteur).

Par ailleurs, la connexion d'un deuxième montage à base également d'un module USBMOD3 engendre au niveau du système la création d'un nouveau port série virtuel déclaré de la même façon (dans Panneau de configuration > Système > Gestionnaire de périphériques > Ports (COM et LPT) : USB Serial Port (COMx) (voir figure 2)

Il serait quand même préférable d'avoir une dénomination personnalisée de façon à pouvoir identifier sans ambiguïté les numéros de ports qui ont pu être attribués aux différents montages (voir figure 6). Tout ceci est possible en personnalisant l'EEPROM intégrée au module USBMOD3. (Pour un module USBMOD1, ce qui suit n'est possible que si on a connecté une EEPROM externe au module puisqu'il en est dépourvu).

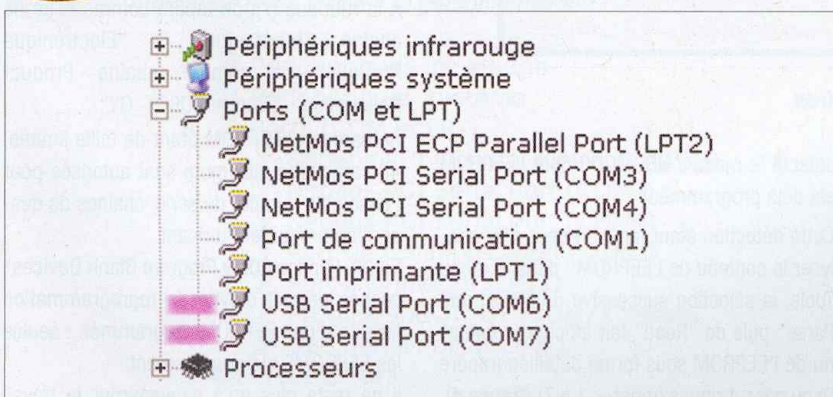
FTDI fournit un logiciel qui permet de lire, d'effacer et de programmer l'EEPROM du module : MPROG. À l'heure où ces lignes sont écrites, c'est la version 2.3 qui est en cours. On ira la télécharger sur le Net : en partant du site Elexol et en suivant comme précédemment le Support > Downloads > USBMOD3&4, cliquer sur Software Utilities. On récupère un fichier ZIP à décompresser, puis on pourra installer MPROG. Ce programme est utilisable si le module est associé au pilote direct : le driver D2XX. Si le module est associé au driver VCP (Virtual COM Port), ce qui est certainement le cas, alors il est nécessaire d'en faire la désinstallation avant de lancer MPROG.

### Désinstallation du driver

Déconnecter tous les montages USB utilisant un circuit de type FTDI232

Lancer le logiciel FTCClean.exe qui, comme son nom l'indique, est là pour faire le ménage. Il faut lui donner le VID et le PID : on garde le VID de FTDI et on entre 6001 dans la boîte de texte du PID (figure 3). Un clic sur le bouton "Clean System" lance la désinstallation. Le logiciel rappelle qu'il faut déconnecter tous les montages, demande ensuite confirmation de la désinstallation, puis une nouvelle boîte de dialogue demande "Do you want to cancel driver uninstillation ?" ("Voulez-vous annuler la désinstallation du driver ?"), ce à quoi il faut répondre par un clic sur "No".

## 2 Ambiguïté sur l'identification des deux modules connectés



## Reconnexion du module

Le système d'exploitation détecte un nouveau matériel USBMOD3 et réclame le pilote nécessaire à son fonctionnement : cela prouve que la désinstallation s'est faite convenablement. On indiquera l'emplacement du pilote : attention, cette fois ci dans C:\ftdi\d2xx\ dans notre exemple.

Après installation, un petit tour dans le Gestionnaire de périphériques nous montrera qu'il n'y a plus de port série virtuel ajouté mais qu'à la rubrique Contrôleur de bus USB, on a un matériel FTDI FT8U2XX. Un clic droit pour en obtenir les Propriétés : dans l'onglet Détail, la rubrique n° d'identification de l'instance de périphérique fait apparaître la chaîne de caractères :

USB\VID\_0403&PID\_6001\ suivi d'un code constitué de deux lettres (EL pour Elexol) et de six caractères alphanumériques. La connexion d'un deuxième module USBMOD3 donnerait la même chaîne de caractères mais un code alphanumérique différent. Ce code permet donc de distinguer sans ambiguïté ces deux modules qui ont le même couple VID (0403) et PID (6001). Leur utilisation simultanée est donc possible.

FTDI pourrait être compris entre 0x0000 et 0xFFFF. FTDI a fourni à l'auteur de cet article les 8 PID compris entre 0xEA28 et 0xEA2F. Les lecteurs pourront utiliser le PID 0xEA28, mais à titre d'expérimentation, dans le cadre de cet article uniquement.

On se propose de réaliser la personnalisation suivante : il s'agit du montage Programmeur de HC908QT\_QY qui utilise un module USBMOD3. Le VID = 0x0403 (FTDI), le PID = 0xEA28. Le montage sera alimenté par l'USB, la consommation reste inférieure à 100 mA. La chaîne de description fabricant : "Electronique Pratique", la chaîne de description produit : "Prog.HC908QT\_QY". Lancer MPROG.

Dans le menu Device, sélectionner "Scan" : MPROG recherche alors tous les montages USB associés au pilote direct D2XX. Si on a sauté l'étape précédente ou si on n'a pas bien associé le module à ce nouveau driver, alors MPROG ne détecte aucun module. Dans le cas contraire, la fenêtre d'état (repère 9 sur la **figure 4**) indiquera : "Number of blank devices : 0" (il n'a pas trouvé de montages USB n'ayant pas d'EEPROM (exemple module USBMOD1) ou ayant une EEPROM vierge) et "Number of programmed devices : 1" (il a

L'auteur conseille de faire une copie d'écran du résultat obtenu (touches Alt + Imp. Ecran du clavier) et de sauvegarder ou d'imprimer cette image, pour pouvoir remettre facilement le module dans sa configuration d'origine.

Menu Files, sélectionner "New" : la seule option alors disponible se trouve à la rubrique Basic Details (1) et concerne le type de circuit. S'agissant d'un montage à base d'USBMOD3, le circuit FTDI monté est un FT232BM. Le sélectionner dans la liste déroulante. Les autres options deviennent alors accessibles : on commencera par définir le couple VID/PID. La liste déroulante propose trois options : le couple VID/PID par défaut (0x0403 et 0x6001), "FTDI supplied PID" c'est-à-dire le VID 0x0403 de FTDI et le PID attribué, "own VID & PID" quand on dispose de son propre couple VID/PID. C'est la deuxième option que l'on choisira en prenant EA28 comme PID.

Au niveau de la configuration de l'alimentation (rubrique 2), on gardera l'option Bus Powered (le montage sera alimenté par l'USB) avec une réserve de courant maximum de 100 mA. (Remarque : on peut réserver au-delà de cette valeur de 100 mA et ce jusqu'à 500 mA mais cela entraîne quelques contraintes supplémentaires sur le montage, contraintes que nous ne détaillerons pas dans cet article).

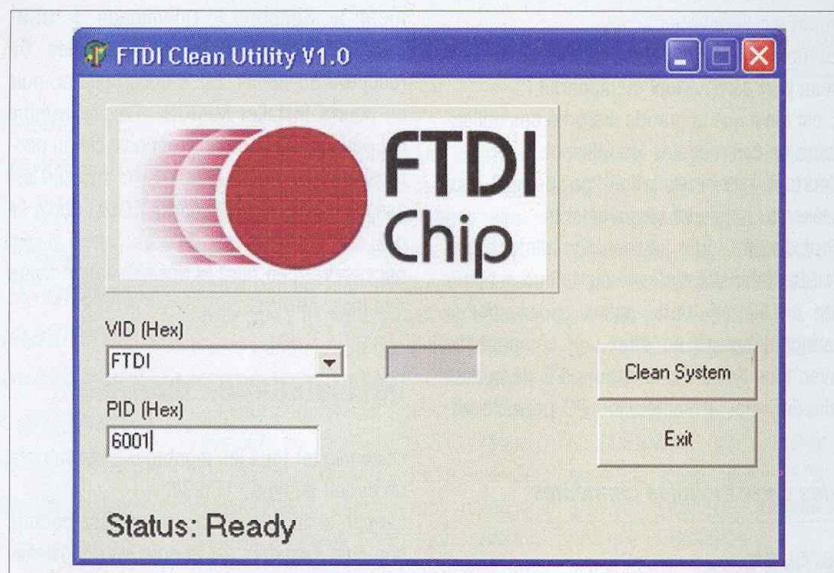
À la rubrique (3) : Numéro de série USB, on pourra changer les deux lettres de préfixe : on a pris ici EP (initiales d'Electronique Pratique). On pourrait choisir par nous-mêmes le reste du numéro de série en cochant "Use fixed serial number", mais il est préférable de laisser le logiciel en fabriquer un lui-même. À la rubrique (4) on sélectionnera dans la liste déroulante USB 2.0 car le module USBMOD3 est compatible USB 2.0. Les autres options resteront décochées. Idem pour celles des rubriques (5) et (6).

À la rubrique (7), on tapera comme texte de chaîne Manufacturer : "Electronique Pratique" et comme chaîne Product Description : "Prog.HC908QT\_QY".

Remarque : l'EEPROM étant de taille limitée, 48 caractères maximum sont autorisés pour l'ensemble numéro de série, chaînes de description produit et fabricant.

En (8), l'option "Only Program Blank Devices" cochée permet d'éviter la reprogrammation fortuite d'une EEPROM programmée : seules les EEPROMS vierges le seront.

Il ne reste plus qu'à sauvegarder le travail



### 3 Nettoyage de la base de registres

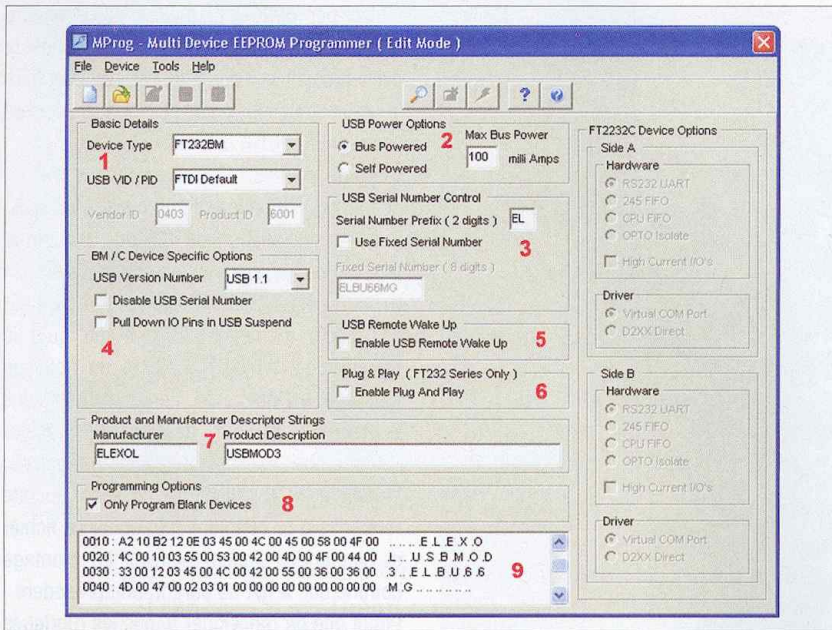
## Personnalisation du montage

La personnalisation prend tout son intérêt si on possède un couple valide VID/PID pour son montage.

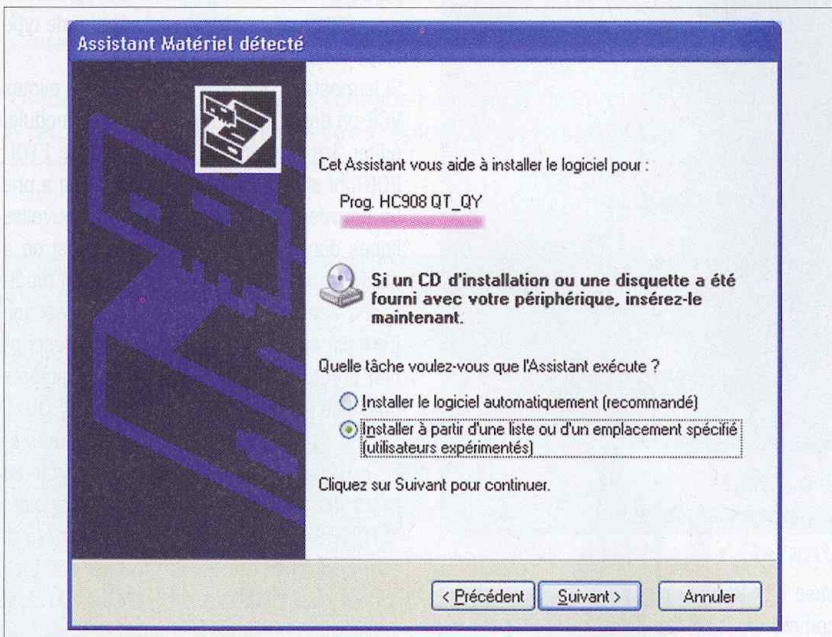
La société FTDI peut en fournir. Le VID sera celui de FTDI : (0x0403). Le PID fourni par

détecté le module USBMOD3 dont l'EEPROM est déjà programmée).

Cette détection étant faite, on peut aller analyser le contenu de l'EEPROM : dans le menu Tools, la sélection successive de "Read and Parse" puis de "Read" fait afficher le contenu de l'EEPROM sous forme détaillée (repère 9) ou par rubriques (repères 1 à 7) (**figure 4**).

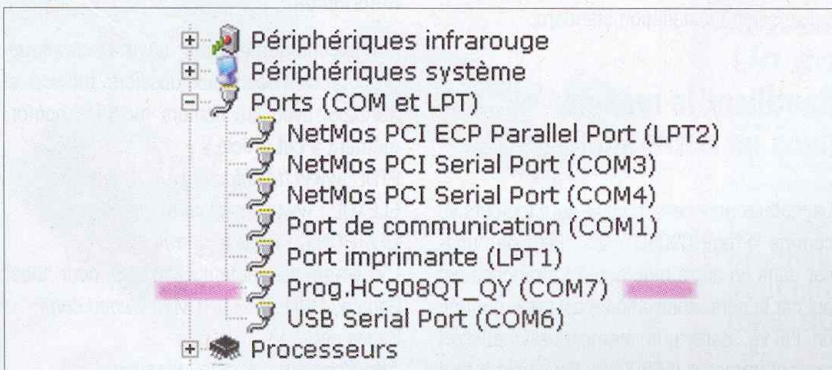


#### 4 Analyse de l'EEPROM d'un USBMOD3



#### 5 Le montage est détecté

#### 6 Ambiguïté levée grâce à la personnalisation



dans un fichier d'extension ".ept" (par exemple "hco8qty.ept") : cela se fait dans le répertoire Template du logiciel. Lors de la phase de programmation, c'est le contenu de ce fichier qui est envoyé vers l'EEPROM..

Deux nouvelles icônes sont alors accessibles (également dans le menu Device) : on clique sur celle permettant l'effacement (Erase) puis sur celle permettant la programmation. La fenêtre d'état (9) indique alors la réussite de chacune de ces étapes.

Le numéro de série généré par algorithme est obtenu dans cette fenêtre d'état (9) une fois la programmation terminée. En prendre note : cela pourra être l'occasion de faire une nouvelle copie d'écran. À ce stade, la personnalisation du module est réalisée... mais il reste encore un petit travail à faire.

### Personnalisation des pilotes

Une tâche qui fait, à tort, bien souvent peur... Pas de crainte à avoir, on ne touche pas au système mais seulement à quelques fichiers de type texte. Dans le pire des cas le montage n'est pas reconnu.

C'est d'ailleurs le cas maintenant : si on déconnecte puis reconnecte le montage, Windows détecte un nouveau montage (figure 5) (remarque : c'est la chaîne de description du produit, placée dans l'EEPROM ("Prog.HC908QT\_QY") qui apparaît pour annoncer le nouveau montage détecté), mais si on essaie d'indiquer un pilote soit dans le répertoire ftdi\vc\ soit dans ftdi\d2xx\, l'installation échoue car Windows n'a pas réussi à associer le montage à un pilote. Normal, lors de la précédente installation, le module avait le couple VID/PID 0x0403 / 0x6001 et maintenant c'est 0x0403 / 0xEA28.

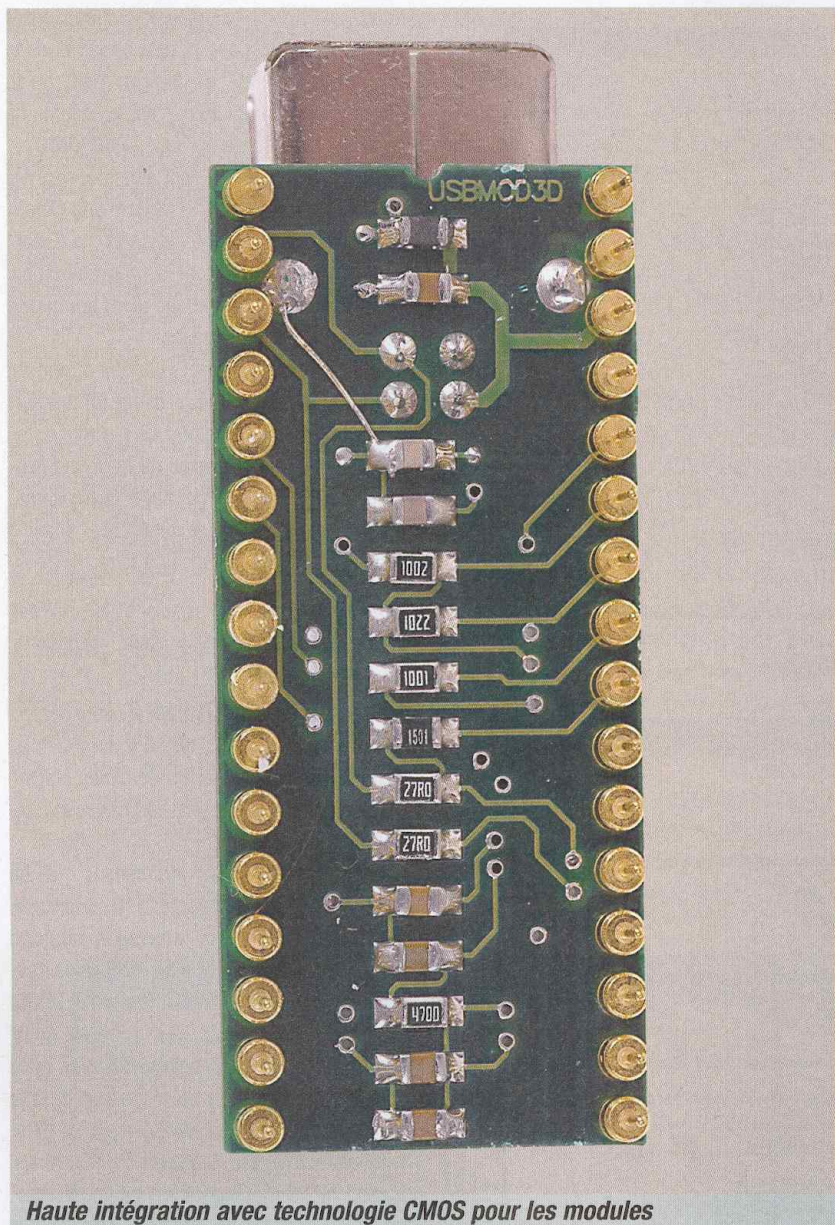
Pour que le montage puisse être reconnu par Windows, il faut ajouter dans les fichiers "\*.inf" des pilotes les références à ce nouveau couple VID/PID.

Dans le dossier ftdi\vc\, ouvrir avec un éditeur de texte (Bloc Notes) le fichier "FTDI-BUS.inf".

Dans la section [FtdiHw], recopier en fin de section une des lignes de cette section mais en changeant le PID en EA28.

Recommencer ce travail dans la section [ControlFlags] qui suit, puis dans la section [Strings] en fin de fichier : là, il faudra changer le PID et la chaîne "USB Serial Converter" en "Prog.HC908QT\_QY"

Lors de la réalisation de ces changements, il faudra être très attentif pour ne pas faire



**Haute intégration avec technologie CMOS pour les modules**

d'oubli ou d'erreur de frappe. Remarque : c'est cette chaîne de caractères qui sera affichée dans le gestionnaire de périphérique et non pas celle placée dans l'EEPROM, il est donc préférable de les rendre identiques.

Le fichier FTDIPORT.inf sera à modifier selon le même principe dans les sections [FtdiHw] et [Strings]. Bien sûr, on sauvegardera dans ce répertoire les fichiers modifiés (sans changer leurs noms).

Reconnectons le montage. Après détection, on peut lui indiquer le dossier ftdi\vcpl comme dossier contenant les pilotes à utiliser. Cette fois-ci, l'installation se fait sans problème. Le gestionnaire de périphériques fait apparaître à la rubrique Ports le nom du montage personnalisé avec le numéro de port qui lui a été attribué. Un autre montage à

base d'USBMOD3 non personnalisé reste vu comme un "USB Serial Port" (voir **figure 6**). Après reprogrammation de l'EEPROM, on pourra faire la désinstallation du driver direct avec FTCClean (PID 6001) de façon à pouvoir réutiliser un module USBMOD3 non personnalisé selon l'installation standard.

### Réutiliser le module dans un autre montage

Le module ainsi personnalisé sera toujours vu comme "Prog.HC908QT\_QY". Peut-on l'utiliser dans un autre montage ? La réponse est oui car la personnalisation s'est faite, comme on l'a vu, dans une mémoire effaçable et reprogrammable (EEPROM). Pour cela il faut

passer par MPROG. Le module doit donc être associé au pilote d2xx. Il faut recommencer le cycle présenté au cours de cet article et faire la désinstallation pour ce module déclaré avec un VID/PID = 0x0403 / 0xEA28.

C'est de nouveau FTCClean qui va entrer en action, on lui donnera cette fois le PID EA28. Cette désinstallation ne fait que supprimer les pilotes VCP associés au couple VID/PID = 0x0403 / 0xEA28, l'EEPROM gardant bien sûr ce couple mémorisé. Ce qui fait qu'à la reconexion, Windows détecte un nouveau montage et demande l'emplacement des pilotes. Si on lui désigne le dossier C:\ftdi\d2xx\, l'association ne se fera pas car le fichier FTD2XX.INF ne possède aucune référence à ce couple. Il faut éditer ce fichier et ajouter les lignes relatives au montage comme on l'a fait au paragraphe précédent. Plutôt que de renseigner toutes les rubriques proposées par MPROG, on pourra préférer rappeler une configuration enregistrée : menu File > Open et choisir le fichier de type ".ept" dans le dossier Templates.

Si le montage doit être utilisé avec les pilotes VCP, on devra, avant reconexion du module, éditer les fichiers "FTDIBUS.inf" et FTDIPORT.inf situés dans C:\ftdi\vcpl. Si on a pris un nouveau PID, il faudra créer de nouvelles lignes dans les sections adéquates, si on a gardé le même PID il faudra seulement modifier la ligne correspondante dans la section [Strings] en donnant à la chaîne de description produite la même suite de caractères que celle placée en EEPROM avec MPROG.

Dans le cadre d'une application commerciale, le concepteur devra bien sûr fournir au client les pilotes modifiés : soit les fichiers "FTDIBUS.inf" et FTDIPORT.inf s'il utilise la connexion USB comme un port série virtuel, soit le fichier FTD2XX.INF dans le cas où c'est le pilote direct qui est utilisé.

Bonnes personnalisations !

**V. LE MIEUX**

#### Références :

(En téléchargement sur [www.electronique-pratique.com](http://www.electronique-pratique.com) les deux dossiers ftdi\vcpl et ftdi\d2xx avec les fichiers modifiés conformément à cet article.)

FTDI : [www.ftdichip.com](http://www.ftdichip.com)

ELEXOL : [www.elexol.com](http://www.elexol.com)

LEXTRONIC : [www.lextronic.fr](http://www.lextronic.fr)

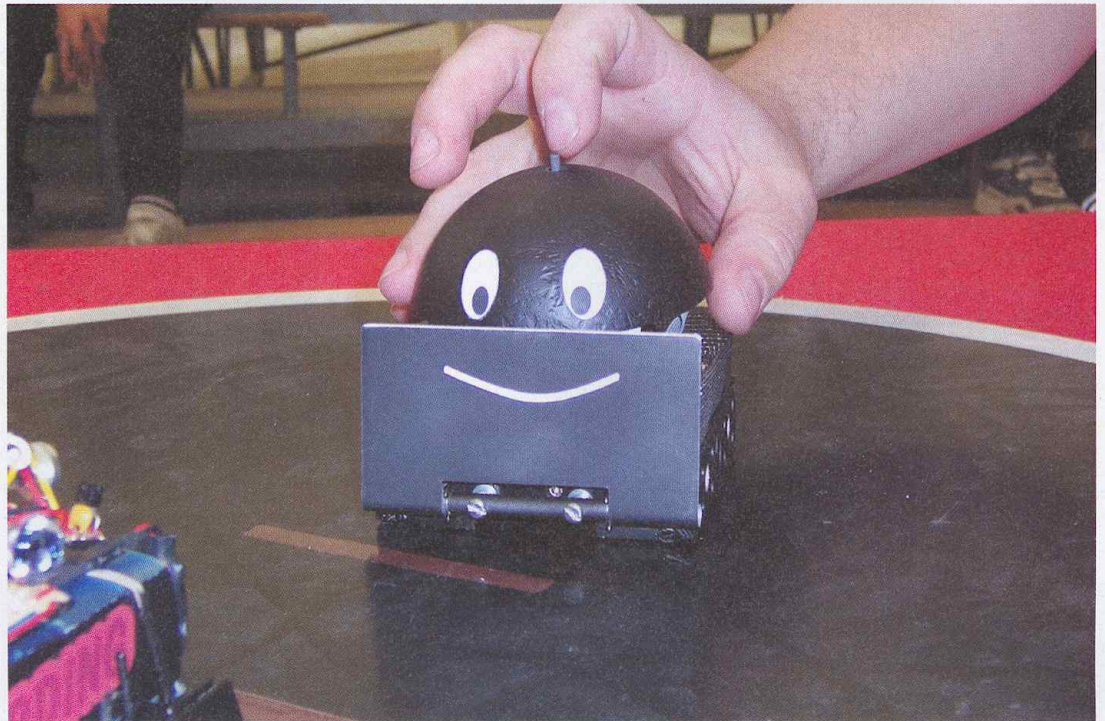
L'ouvrage de l'auteur : "L'USB pour tous" Editions DUNOD/ETSF ([www.dunod.com](http://www.dunod.com))

Pages perso :

<http://home.nordnet.fr/~vlemieux/>

# Tournoi de mini-sumos de Nîmes

*Pour la première fois, un tournoi de robot mini sumos a eu lieu en France. Cette discipline de la robotique est très recherchée en Amérique et au Japon. C'est d'abord une occasion unique pour tous de participer à un concours de robotique quand on n'a plus l'âge d'être étudiant. C'est ensuite une catégorie de tournoi de robotique très spectaculaire. En effet, le but étant de pousser son adversaire en dehors des limites d'un cercle, on peut imaginer les combats acharnés pour soulever et repousser l'autre.*



*Yemo, robot à chenilles, très redoutable malgré son sourire*

## Déroutement

Ce tournoi a eu lieu à Nîmes les 2 et 3 avril, dans le cadre du printemps des enfants, manifestation qui a accueilli 40.000 personnes cette année, il y avait beau temps. Deux catégories étaient présentes. Les enfants de 9 ans à 14 ans ont pu apprendre à réaliser un petit robot sumo filoguidé et concourir. Mais la catégorie reine était celle des robots autonomes qui a accueilli 15 participants de tous les coins de France.

Organisé par le Lion's Club de Nîmes, cette manifestation avait le soutien de la ville de Nîmes, des médias locaux et de nombreux partenaires financiers. Le tournoi s'est déroulé sur deux jours et plusieurs démonstrations autour de la robotique étaient présentées au

public. Un robot cheval, très recherché des enfants, un robot joueur d'échec, plusieurs petits robots qui se baladaient en se croisant sur une grande table ou encore un robot à clavier musical.

## Le samedi

Les participants ont pu découvrir l'arène avec des gradins pour recevoir le public et les tables de jeux (Dohyo), il s'agit de cercles noirs de 77 cm avec une bordure blanche. Une partie du local était réservée pour les ateliers. D'un côté, les enfants suivaient une animation sur vidéo afin de comprendre les différentes parties du petit robot qu'ils allaient construire. De l'autre, les participants au tournoi principal s'installaient. C'est l'occa-

sion de voir les concurrents pour la première fois et de se faire déjà une petite idée sur ses chances de succès. Le poids a été un des plus gros problèmes pour beaucoup de robots. Les 500 grammes ont été difficiles à atteindre et certains concurrents ont percé au maximum leur châssis pour ne pas dépasser ce poids réglementaire. La majorité des robots pesaient entre 490 et 500 grammes. Cette journée a permis de faire un premier classement et défini deux poules à la suite des premières rencontres.

## Le dimanche matin

Cette matinée a permis de reprendre espoir pour certains concurrents qui ont revu leur mécanique ou leur programmation. Mais le sentiment géné-

ral était de vouloir en découdre avec un adversaire pour vérifier et améliorer son robot. Des petites rencontres informelles se sont déroulées sur les tables d'essais en attendant la finale de l'après-midi.

## Les finales

Les finales ont débuté à 14 h pour les enfants et à 15 h pour les robots autonomes. Le nombre des enfants a été tellement important (près de 40 enfants inscrits), qu'il a été très difficile de démarrer la grande finale avant 15h 15. Les gradins étaient pleins et il y avait encore du monde qui souhaitait entrer. De l'avis de tous, les enfants ont montré de réelles compétences pour le maniement et la stratégie des petits robots mini-sumos filoguidés. Enfin la finale tant attendue pour tous les robots autonomes. Des mois de travail pour en arriver là. Chacun de ces robots représente un concentré de technologie et de programmation patiemment élaborée dans un but simple, pousser l'adversaire en dehors des limites du cercle. Enfin, nous allions découvrir quelle serait la meilleure stratégie, cette année. Les combats ont été incroyablement courts pour la majorité et rapidement, certains robots se sont détachés du lot. Plus de puissance, meilleure fiabilité, meilleure détection. Les combats entre ces derniers allaient être titanesques. Et effectivement, nous avons assisté à des rencontres où les robots ont essayé de se pousser face à face. Les concurrents encourageaient leur robot à chaque centimètre gagné et l'assistance applaudissait à la fin de chaque combat. La finale opposa deux robots nîmois très différents. Goliath, un gros robot qui démarre avec le nez en l'air, car il possède deux énormes roues qui lui permettent d'avoir beaucoup de couple contre Nîmesis plus rapide. Nîmesis, aura vite fait d'attraper Goliath sur le côté pour le retourner et le pousser.

## Résultats

1 <sup>er</sup> prix	Nîmesis
2 <sup>e</sup> prix	Goliath
3 <sup>e</sup> prix	Nantes

Divers prix ont récompensé les gagnants. Ces prix ont été remis par les responsables du Lion's club. Cette année, le premier prix était un robot hexapode de Microrobot. Tous les enfants qui ont participé ont gardé leur petit

robot afin de s'entraîner pour l'année prochaine et les trois premiers ont aussi reçu des petits robots à construire. Mais indépendamment des résultats, l'ensemble des participants a été conquis par le tournoi et l'ambiance. Ils ont promis qu'ils reviendront pour gagner.

## Bilan

Parmi les différentes manifestations autour de la robotique en France, le tournoi de robots mini-sumos de Nîmes est une nouvelle démonstration du caractère pédagogique de cette science. Le tournoi est reconduit pour l'année prochaine, toujours à Nîmes, début avril 2006. Il est possible que d'autres démonstrations y soient incluses, pour des robots suiveurs de lignes ou des robots labyrinthe.

G. GIAMARCHI

## Liens Internet :

[www.geii.iut-nimes.fr/fg/sumo.htm](http://www.geii.iut-nimes.fr/fg/sumo.htm)

Site officiel du tournoi avec des vidéos et photos :

[www.fsi.co.jp/sumo-e/index.html](http://www.fsi.co.jp/sumo-e/index.html)

Site officiel japonais

[www.robot-maker.com](http://www.robot-maker.com)

Forum sur le tournoi de Nîmes

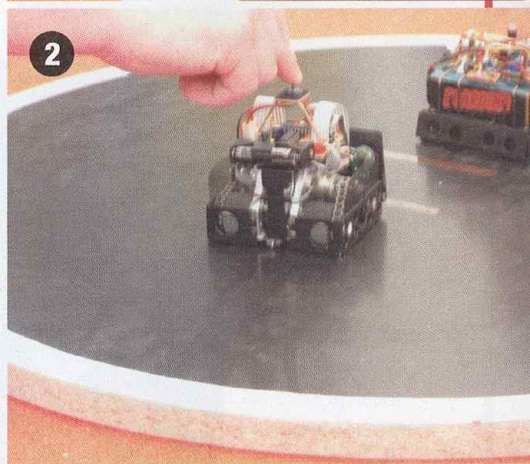
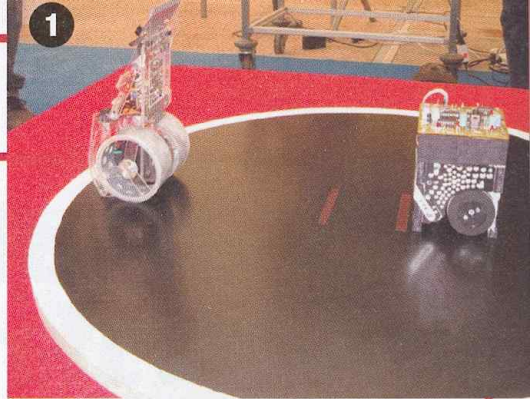
**1**  
*Goliath, à gauche, est un robot qui doit démarrer en appui sur l'arrière pour respecter les dimensions de 10 cm x 10 cm*

**2**  
*Les deux demi-finalistes, Yemo à gauche contre Escap à droite*

**3**  
*Nîmesis face à Nikaia, combat inégal*

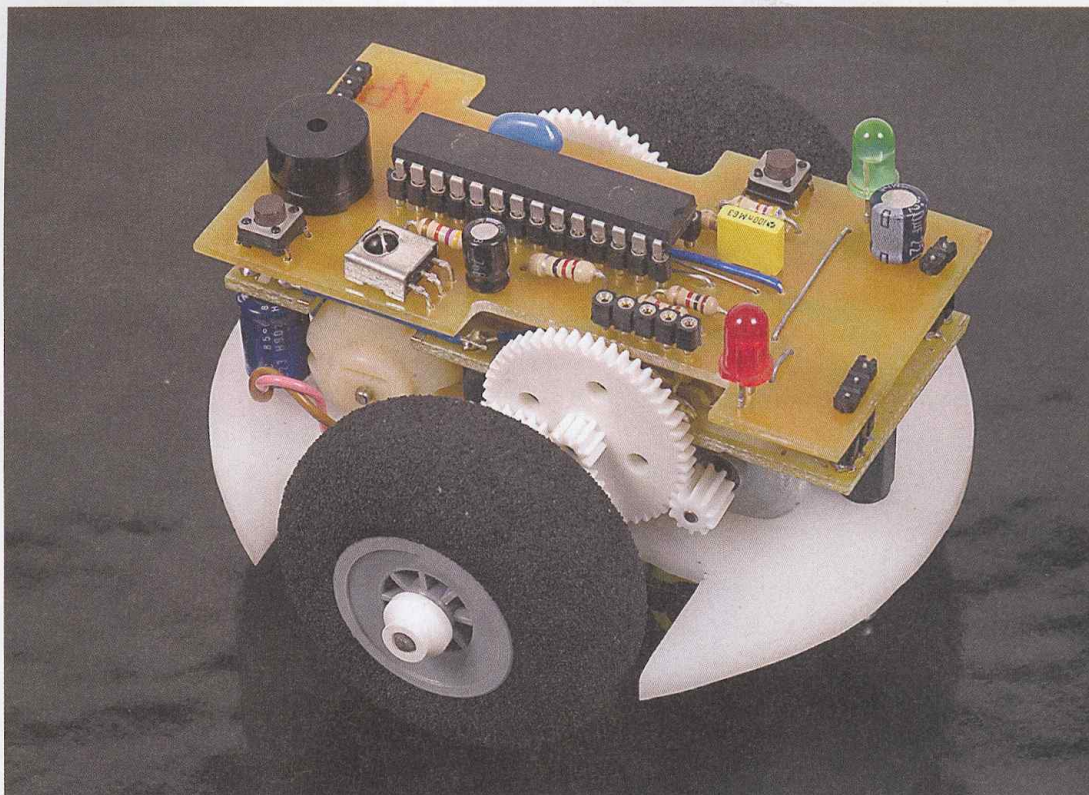
**4**  
*Le grand gagnant, Nîmesis, à la pesée*

**5**  
*L'arène nîmoise prête à admirer un nouveau genre de gladiateurs*





# Robot infrarouge



## Caractéristiques

Pour tester le comportement de sa dernière réalisation en matière de robot mobile autonome, il est parfois utile de lui opposer un robot plus intelligent. Dans le cadre des concours de robotique, il est utile d'avoir un adversaire rusé. C'est pour répondre à ce besoin qu'a été créé ce robot piloté par un PC. Pour supprimer le fil à la patte, on utilise une liaison infrarouge et pour ne pas se compliquer la vie avec un quelconque protocole codé, on utilise un dialogue série RS232. La partie mécanique doit être simple et bon marché. L'interface de puissance choisie n'utilise que des composants discrets. La carte de contrôle utilise un microcontrôleur connu. L'interface infrarouge est aussi des plus simples. L'alimentation est réalisée avec 4 accus de 1,2 volt. Il n'est pas utile de créer un dialogue

dans les deux sens, seule une liaison PC vers robot est réalisée. Un petit logiciel écrit en C ( Builder C++) permet le contrôle du robot, mais il est aussi possible d'utiliser HyperTerminal, logiciel inclus dans Windows.

## La mécanique

La mécanique est constituée d'une plaque de plexi de 3 mm d'épaisseur, correspondant au châssis du robot. Cette plaque est découpée suivant le plan de la **figure 1**. Sa forme ronde a été choisie pour faciliter ses déplacements devant des obstacles anguleux. Cette pièce reçoit deux vis M3 de 35 mm de longueur avec écrous et entretoises filetées, ces vis servent de roues folles, les écrous et les entretoises seront réglés avec les roues (voir **figure 5**). Elle reçoit également quatre vis M3 de 5 mm pour tenir l'axe

des roues (**figure 4**). Deux engrenages blancs sont insérés dans l'axe, puis deux roues et deux bagues d'arrêt. Les roues sont solidaires des engrenages par deux vis plastique de 3 mm dont la tête est coupée après coup. Pour cela, il faut percer deux trous de diamètre 2,5 mm sur les deux engrenages (voir **figures 2 et 3**).

Le reste de la mécanique, le deuxième jeu d'engrenages et les moteurs font partie de l'interface de puissance.

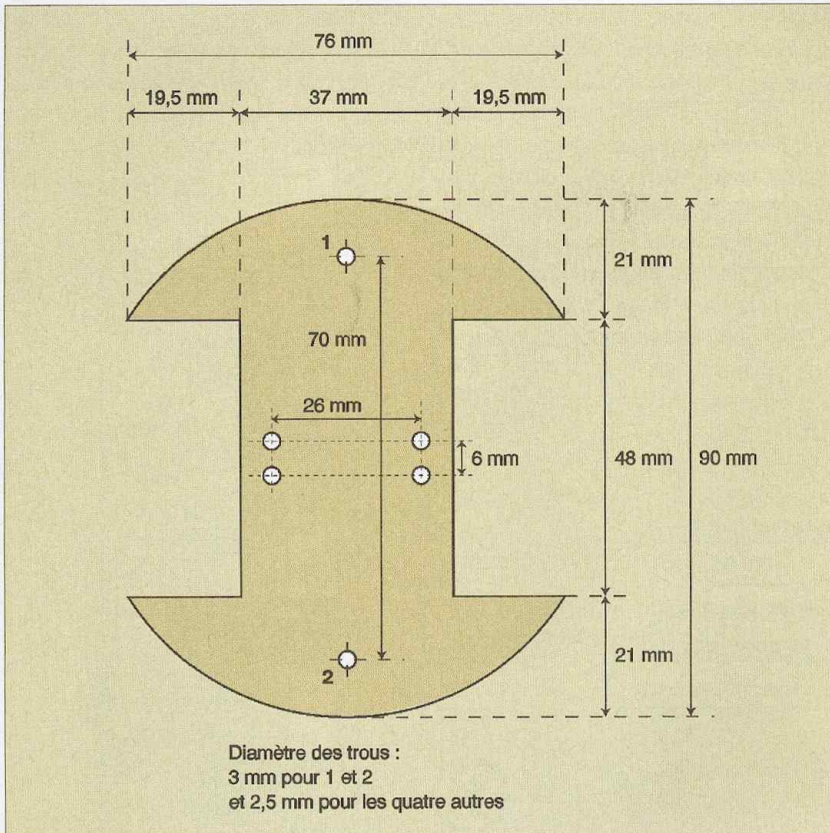
## L'interface de puissance (figure 6)

La particularité de cette interface est que les moteurs sont directement soudés sur le circuit imprimé. Cela permet d'éviter des fils de liaisons vers les moteurs. De même, ces liaisons étant très courtes, on réduit les parasites qui sont importants avec les moteurs utili-

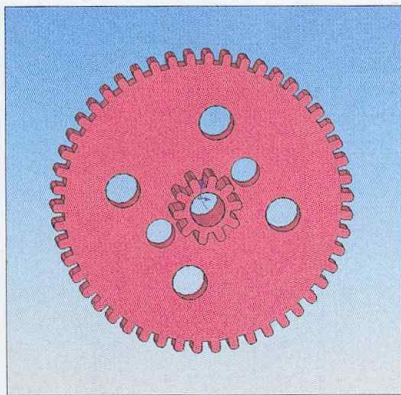
*Les robots autonomes ont parfois besoin d'un partenaire pour les tester. C'est dans cette optique que ce robot piloté par infrarouge a été développé. Une base mécanique simple, une électronique réduite, une programmation succincte et un pilotage par PC par le biais d'une liaison série sans fil.*

## Nomenclature

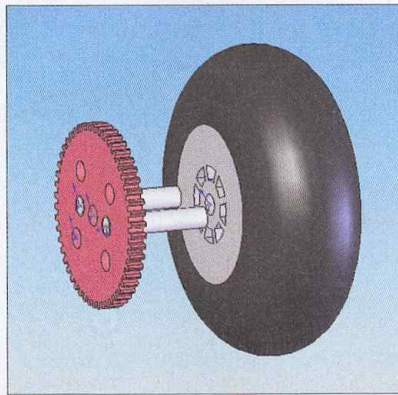
**Pièce n°1**  
 2 vis M3 35 mm  
 2 écrous M3  
 4 vis M3 6 mm  
 4 vis M3 10 mm plastique  
 2 entretoises filetées M3 15 mm  
 1 axe 3 mm 90 mm de longueur  
 1 sachet d'engrenages (module 0,5)  
 réf : 29 77 04 22 (Conrad)  
 2 roues 45 mm  
 réf : 21 73 79 22 (Conrad)



### 1 Le plan de découpe du châssis

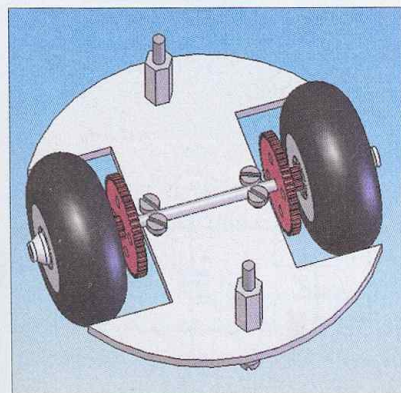


### 2 Le perçage des engrenages

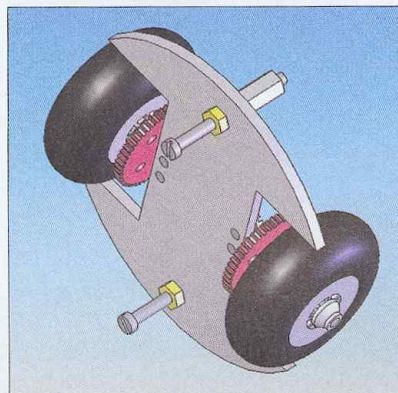


### 3 L'engrenage et la roue sont solidaires l'un de l'autre

### 4 Le châssis est fini



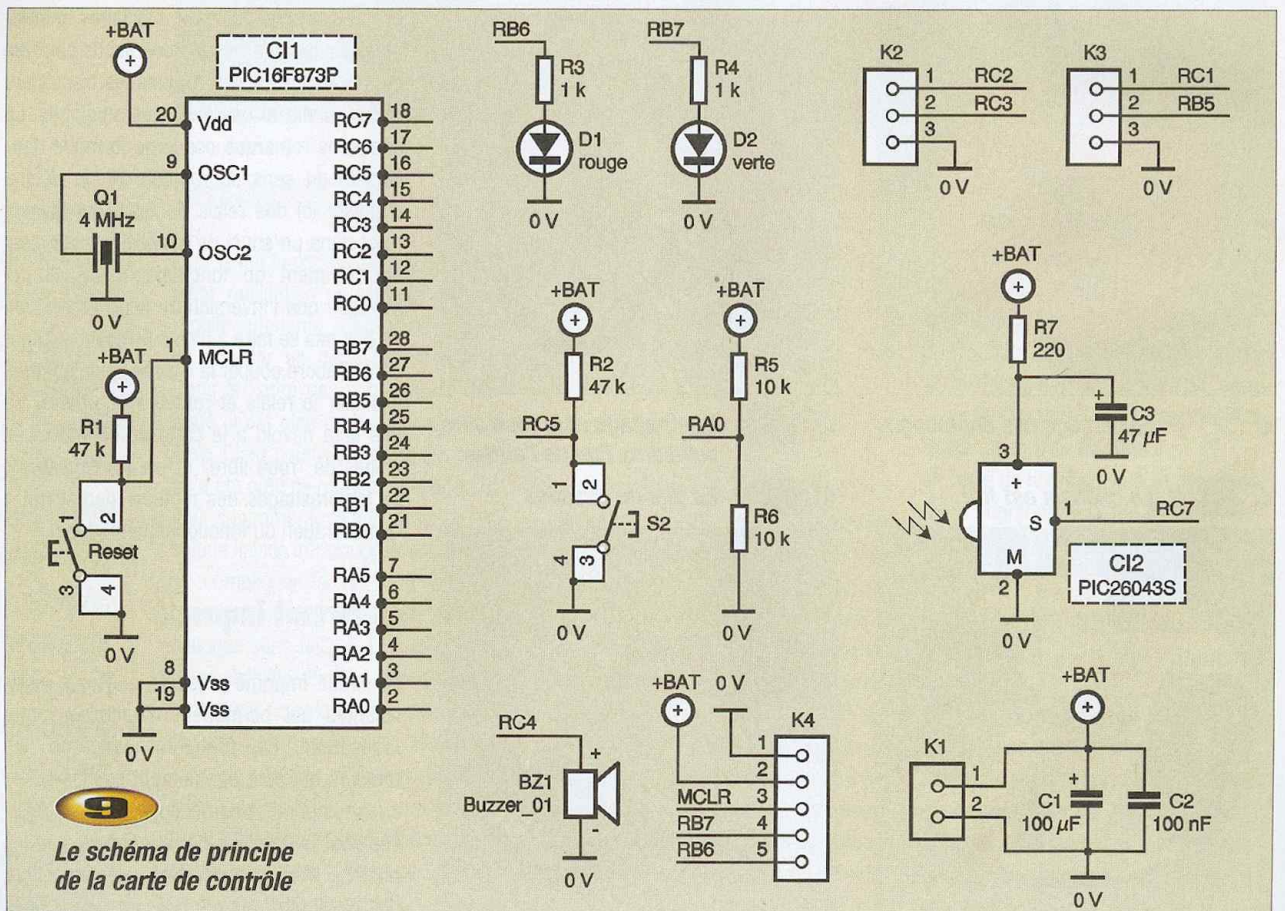
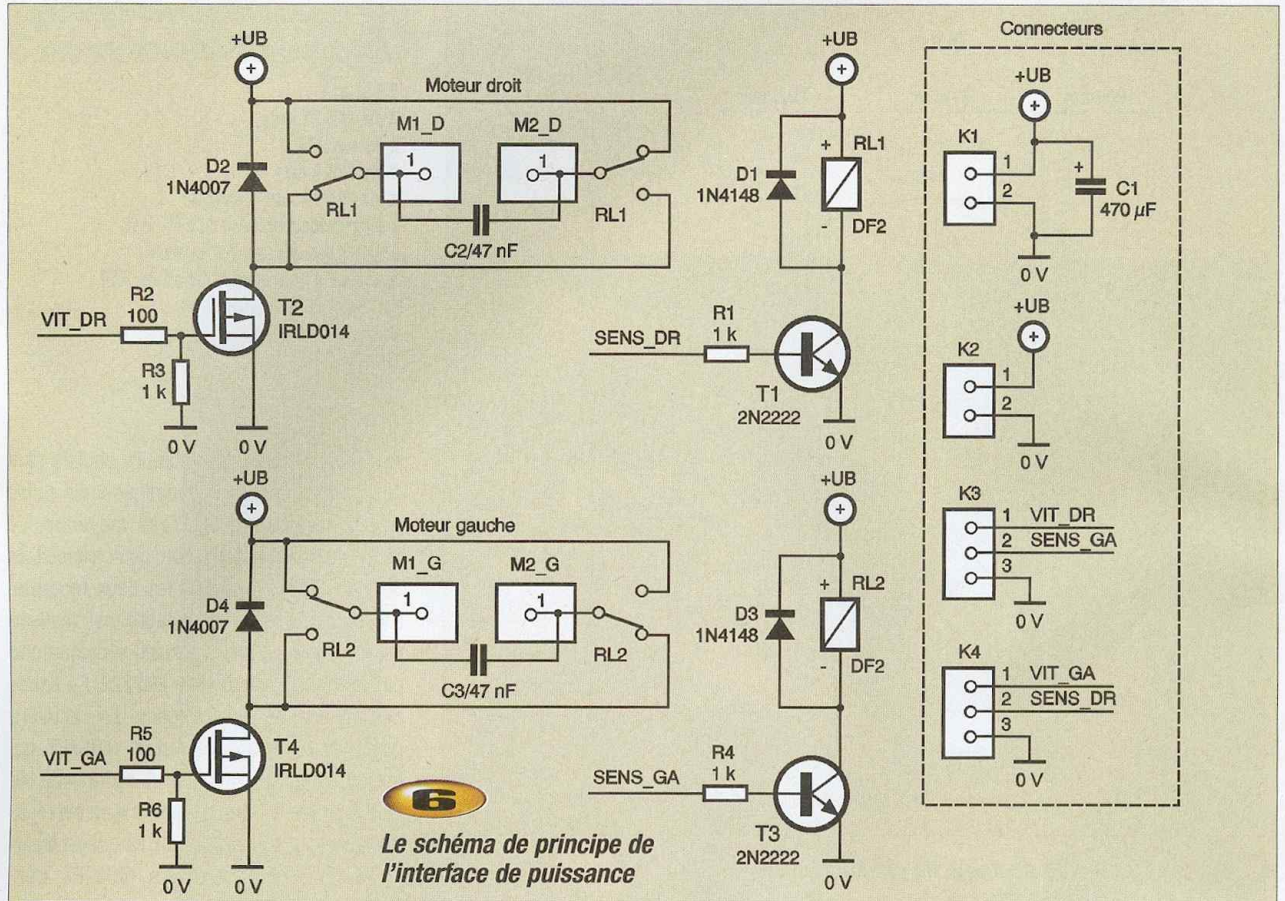
### 5 La vue de dessous



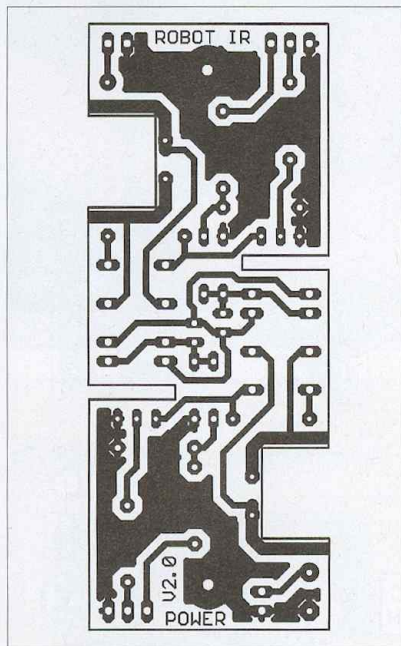
sés. Effectivement, les moteurs choisis sont des petits moteurs économiques de faible rendement et de qualité très moyenne. Le schéma est assez classique, deux transistors de type MOSFET pilotent les deux moteurs. L'inversion du sens de rotation est réalisée par un relais de faible puissance commandé par un transistor bipolaire 2N2222. Ce schéma appelle deux remarques. La première concerne les deux transistors MOSFET qui doivent être des modèles à commande de type logique. En effet, l'alimentation est réalisée avec 4 accus de 1,2 volt, ce qui est insuffisant pour des transistors MOSFET classiques. Le choix d'utiliser des transistors MOSFET plutôt que des bipolaires permet d'utiliser toute la tension des accus pour les moteurs sans en perdre dans les transistors comme c'est le cas avec des bipolaires. La deuxième remarque concerne le mode d'inversion du sens de rotation. Si on décide d'utiliser ici des relais de faible puissance, c'est dans un souci de simplicité mais c'est au détriment du fonctionnement. Ce qui implique que l'inversion du sens de rotation ne doit pas se faire à pleine puissance. Il faut donc d'abord couper la vitesse jusqu'à 0 puis basculer le relais et remettre la vitesse. Ce rôle sera dévolu à la carte de contrôle. Les diodes de "roue libre" et les condensateurs de déparasitage des moteurs participent à l'amélioration du fonctionnement global.

## Le circuit imprimé

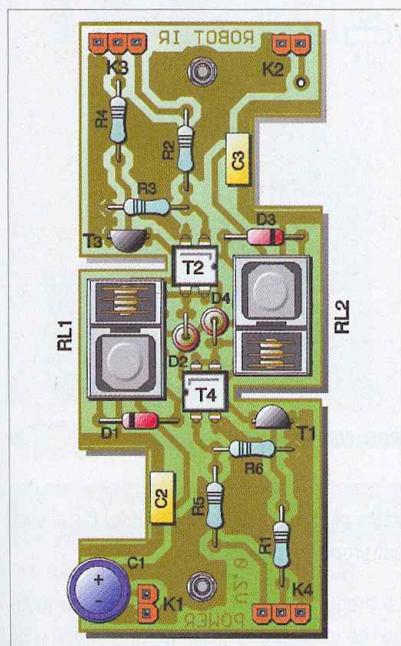
Le circuit imprimé présente quelques particularités qui permettent d'optimiser notre réalisation. La première concerne les moteurs qui sont directement soudés sur le circuit imprimé. Une découpe doit être réalisée pour permettre cela (voir **figure 7** et les photos). Mais une découpe supplémentaire doit aussi être réalisée pour fixer deux tiges



de laiton de 3 mm de diamètre. Chacune de ces tiges supporte un engrenage réducteur blanc, indispensable pour que notre robot ait une vitesse réaliste. La compacité de l'ensemble présente d'autres particularités que nous allons voir.



**7** Tracé du circuit imprimé



**8** Implantation des composants

Il faut vérifier que les découpes sont assez larges pour permettre un repositionnement des moteurs après coup. On doit coller en premier les deux tiges de laiton côté compo-

sants puis souder l'ensemble des composants à l'exception de toutes les résistances et des condensateurs C2 et C3 qui seront soudés côté cuivre. Attention aux risques de court-circuit avec les moteurs. Attention, tous les connecteurs sont soudés côté cuivre sauf le connecteur d'alimentation K1. Il y a encore 5 liaisons volantes à réaliser côté cuivre. Ces liaisons supplémentaires n'ont pu être réalisées côté composants en raison d'un risque de court-circuit avec les moteurs. Une fois le circuit imprimé fini, il faut placer les deux moteurs avec leurs deux pignons, les deux engrenages et l'ensemble sur le châssis. L'ensemble est fixé par deux écrous. On peut maintenant souder les deux moteurs de manière à ce que les engrenages et les pignons présentent un jeu réduit.

## Nomenclature

R1, R3, R4, R6 : 1 k $\Omega$  (marron, noir, rouge, or)  
 R2, R5 : 100  $\Omega$  (marron, noir, marron, or)  
 C1 : 470  $\mu$ F (radial)  
 C2, C3 : 47 nF  
 D1, D3 : 1N4148  
 D2, D4 : 1N4007  
 T1, T3 : 2N2222  
 T2, T4 : IRLD014  
 RL1, RL2 : Relais 2RT  
 K1, K2, K3, K4 : barrette sécable femelle.  
 2 moteurs : Mabushi type 130  
 réf : 24 44 06 22 (Conrad)  
 2 tiges : laiton  $\varnothing$  : 3 mm x 15 mm  
 2 engrenages blancs  
 2 pignons

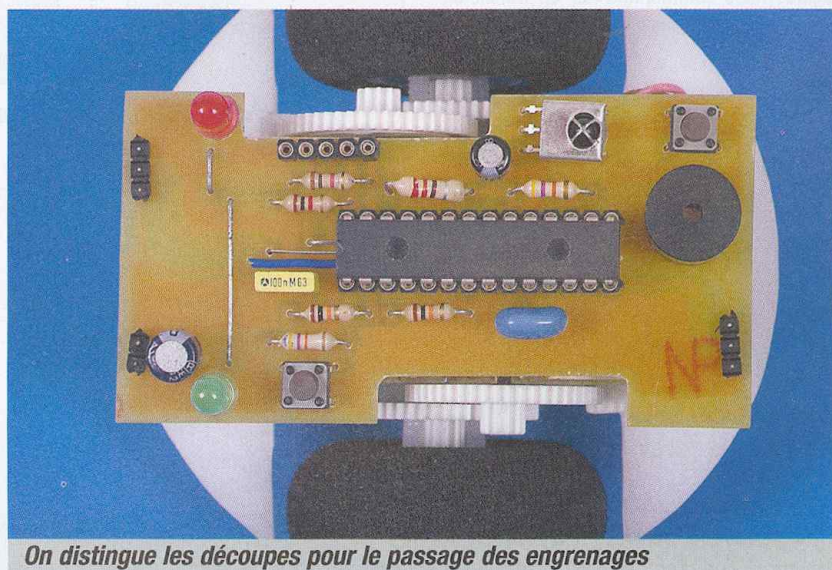
## La carte de contrôle (figure 9)

La carte de contrôle utilise un PIC16F873. Ce composant est le grand frère du célèbre

PIC16F84. Notre sélection s'est faite sur les différents PICs disponibles, de coût réduit et facilement programmables. Il aurait été plus simple de choisir un composant plus petit en raison du faible nombre d'entrées/sorties nécessaires à ce projet. Un PIC16F627 ou mieux un PIC16F88, aurait simplifié le circuit imprimé. Mais la programmation de ce dernier n'est certainement pas encore bien maîtrisée par la plupart des lecteurs, malgré le très bon article de J.P. Mandon sur le numéro de mars dernier. Ce n'est que partie remise. Peu de composants autour du PIC, quelques résistances pour alimenter deux LEDs ou pour donner des niveaux aux poussoirs de Reset et S2, un buzzer, deux condensateurs et un résonateur. Des connecteurs de liaisons avec la carte interface de puissance K1, K2 et K3. Et surtout un photo module infrarouge pour recevoir les ordres transmis par le PC. Ce composant possède deux sorties spécialisées pour générer des signaux PWM pour piloter les moteurs. Un pont diviseur R5, R6 permet de vérifier la tension sur la batterie pour couper le robot en cas de baisse de tension. Le connecteur K4 sert à la programmation ICSP, c'est à dire sans sortir le composant de son support.

## Le circuit imprimé (figure 10)

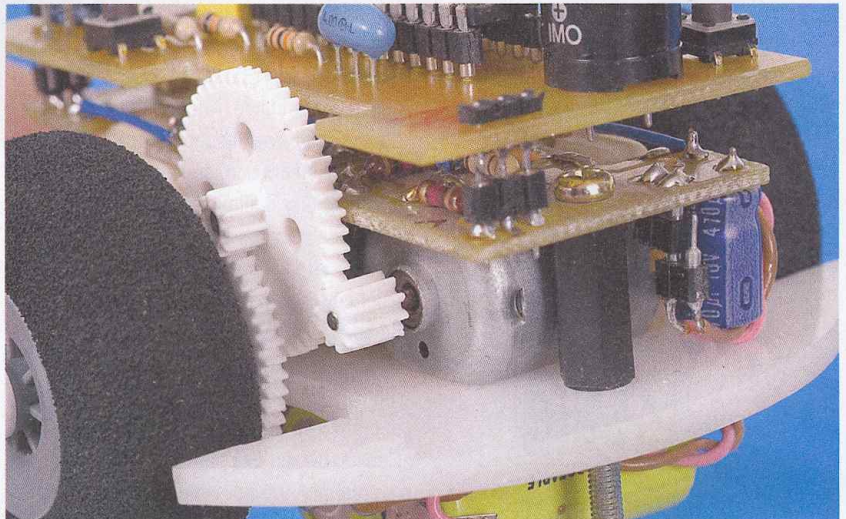
Là encore, le circuit imprimé possède une forme non conventionnelle. Deux découpes permettent le passage des engrenages de la carte de puissance. Le circuit imprimé comporte 5 straps (voir photo). Les connecteurs doivent permettre de s'insérer dans ceux de la carte de puissance. Pour cela, il faut utiliser des connecteurs mâles à l'exception de



K4 qui sera un modèle femelle et soudé côté composants pour ce dernier.

## Nomenclature

R1, R2 : 47 k $\Omega$  (jaune, violet, orange, or)  
 R3, R4 : 1 k $\Omega$  (marron, noir, rouge, or)  
 R5, R6 : 10 k $\Omega$  (marron, noir, orange, or)  
 R7 : 220  $\Omega$  (rouge, rouge, marron, or)  
 C1 : 100  $\mu$ F (radial)  
 C2 : 100 nF  
 C3 : 47  $\mu$ F (radial)  
 D1 : led rouge  
 D2 : led verte  
 Q1 : résonateur 4 MHz  
 C11 : PIC16F873P  
 C12 : PIC26043, TSOP1838  
 K1, K2, K3 : barrette sécable mâle  
 K4 : barrette sécable femelle  
 1 buzzer  
 Support 28 broches



Position des engrenages (voir texte)

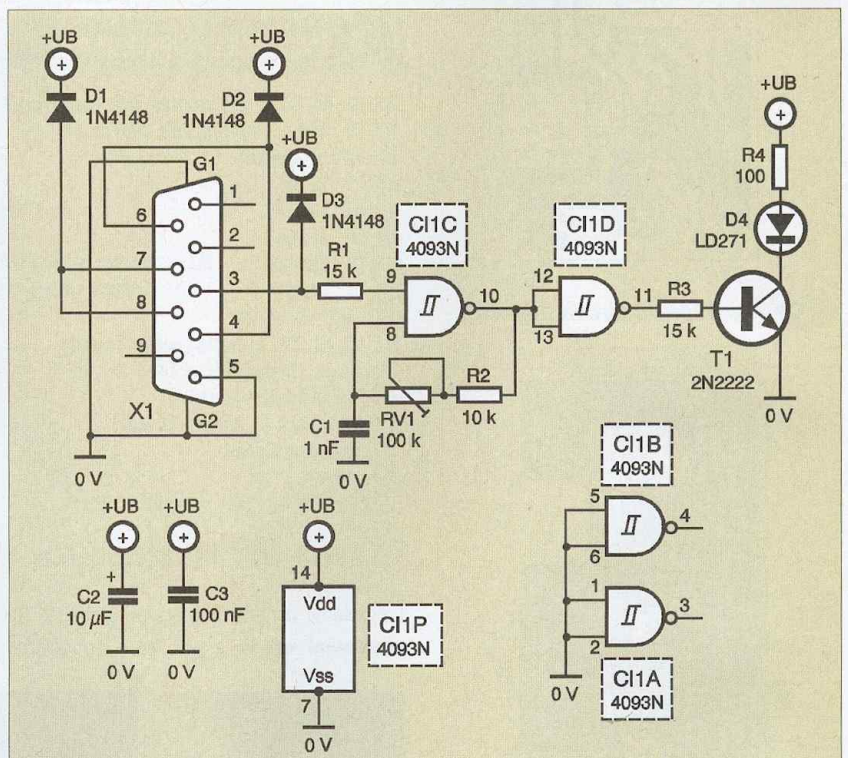
## L'interface infrarouge (figure 12)

Ce montage permet de dialoguer entre un PC et notre robot d'exercice. Pour simplifier le schéma et réduire la consommation, le dialogue ne fonctionne que dans un seul sens, du PC vers le robot. L'alimentation est tirée des lignes de la liaison série RS232. Les lignes RTS, DTR et même la ligne de dialogue TXD sont utilisées pour cela. Protégé par une résistance, le signal active un générateur de signaux carrés dont la fréquence est de l'ordre 38 kHz.

Cette fréquence devra être réglée par la résistance variable RV1 pour s'adapter au photo module utilisé par la carte de contrôle. Pour disposer d'une puissance d'émission suffisante, la diode d'émission n'est pas reliée directement à la sortie de la porte logique mais commandée par un transistor. La résistance R4 limite la distance d'émission mais il n'est pas conseillé de la diminuer. De toute manière, il est important que la liaison infrarouge soit directe entre l'interface infrarouge et le robot. La vitesse de dialogue ne devra pas dépasser 1200 bauds, pour une qualité de signal optimale. La portée se dégrade vite si les longueurs d'onde de l'émetteur et du récepteur sont très différentes. Plusieurs essais ont été nécessaires pour trouver une association correcte, dans l'exemple une LD271 avec un PIC26043.

## Le circuit imprimé (figure 13)

Ce montage ne présentant pas de difficultés, il sera réalisé en quelques minutes. Pour le réglage, on cherchera le maximum de portée.



12 Schéma de principe de l'interface infrarouge

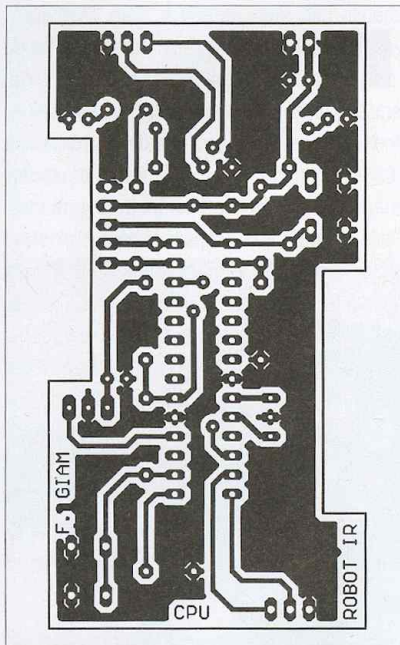
## La programmation

Le comportement du robot doit répondre aux ordres envoyés par le PC. Afin de ne pas compliquer le dialogue, les commandes sont des lettres minuscules tapées au clavier : (e) pour avancer, (d) pour droite, (s) pour gauche, (x) pour reculer, (q) pour stopper, (o) pour accélérer, l pour ralentir. Il est donc parfaitement possible d'utiliser l'interface de Windows, HyperTerminal.

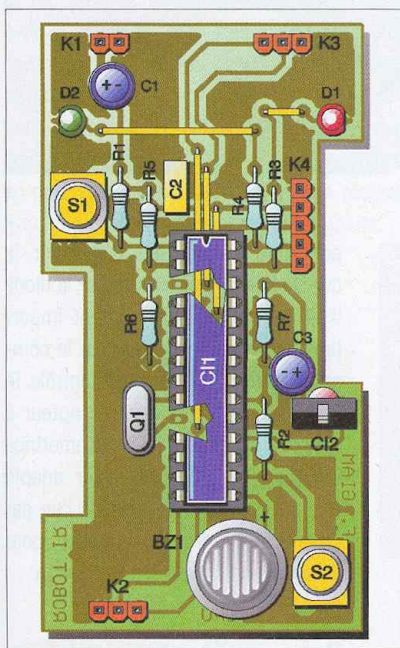
Mais pour plus de confort, une petite interfa-

ce de contrôle, écrite avec Builder C++, vous est proposée figure 15.

Le programme du robot comporte une routine de dialogue principale -qui attend une commande par la liaison série à 1200 bauds. Puis une routine d'interruption de l'ordre de la milliseconde traite le clignotement des LEDs rouge et verte, ainsi que le traitement des vitesses. Les deux LEDs clignotent à chaque fois que l'on inverse le sens de rotation d'un des moteurs. En marche arrière, on ajoute un bip cadencé. À chaque fois que l'on



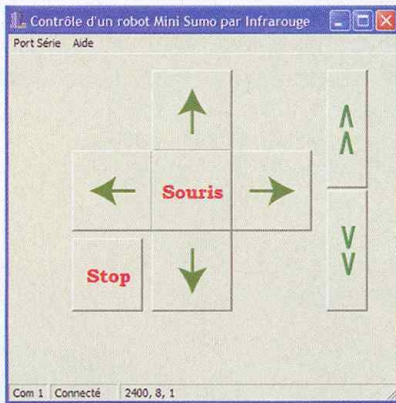
**10** Tracé du circuit imprimé



**11** Implantation des composants

## Nomenclature

- R1, R3 : 15 k $\Omega$  (marron, vert, orange, or)
- R2 : 10 k $\Omega$  (marron, noir, orange, or)
- R4 : 100  $\Omega$  (marron, noir, marron, or)
- RV1 : 100 k $\Omega$  (horizontal)
- C1 : 1 nF
- C2 : 10  $\mu$ F
- C3 : 100 nF
- D1, D2, D3 : 1N4148
- D4 : LD271 (led infrarouge)
- Q1 : 4093
- X1 : Connecteur DB9 femelle soudé  
Support 14 broches

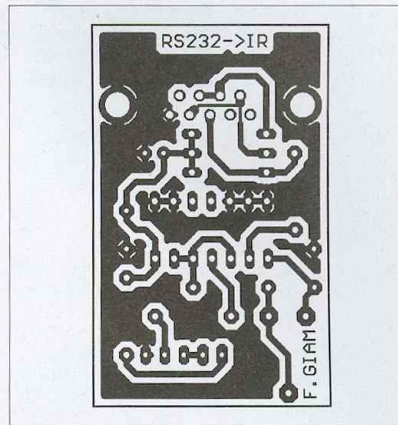


**15** Vue d'écran

donne un nouvel ordre, la vitesse est coupée et repart de zéro pour protéger les relais. Les changements de vitesse se font progressivement. Les vitesses en marche avant et les autres sont mémorisées à chaque commande stop du robot. Ces vitesses mémorisées sont restaurées à chaque nouvelle mise sous tension du robot. Le bouton poussoir S2 n'est pas utilisé dans le programme, mais il était prévu initialement d'ajouter un pare-chocs.



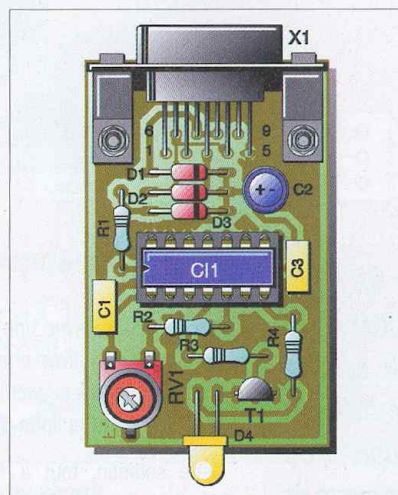
Mise en place des accumulateurs



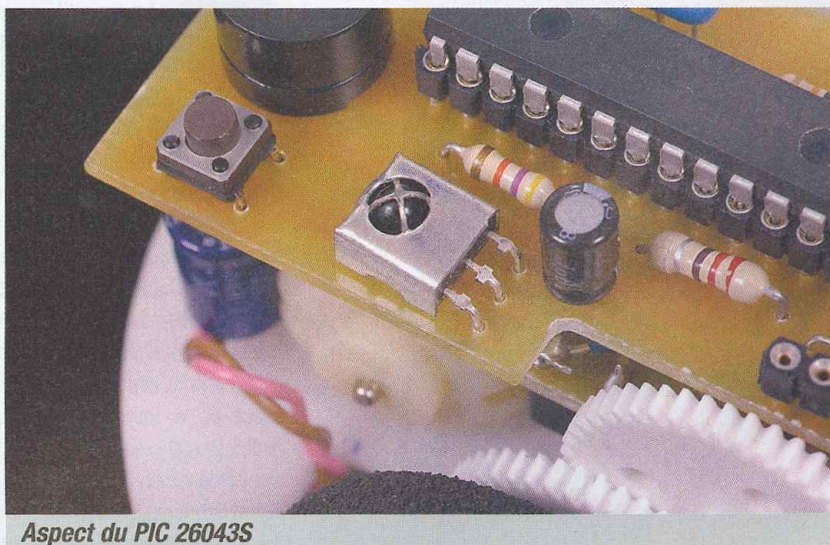
**13** Tracé du circuit imprimé

Les différents fichiers sont disponibles sur le site de la revue.

F. GIAMARCHI  
giamarchi@iut-nimes.fr  
www.geii.iut-nimes.fr/tg

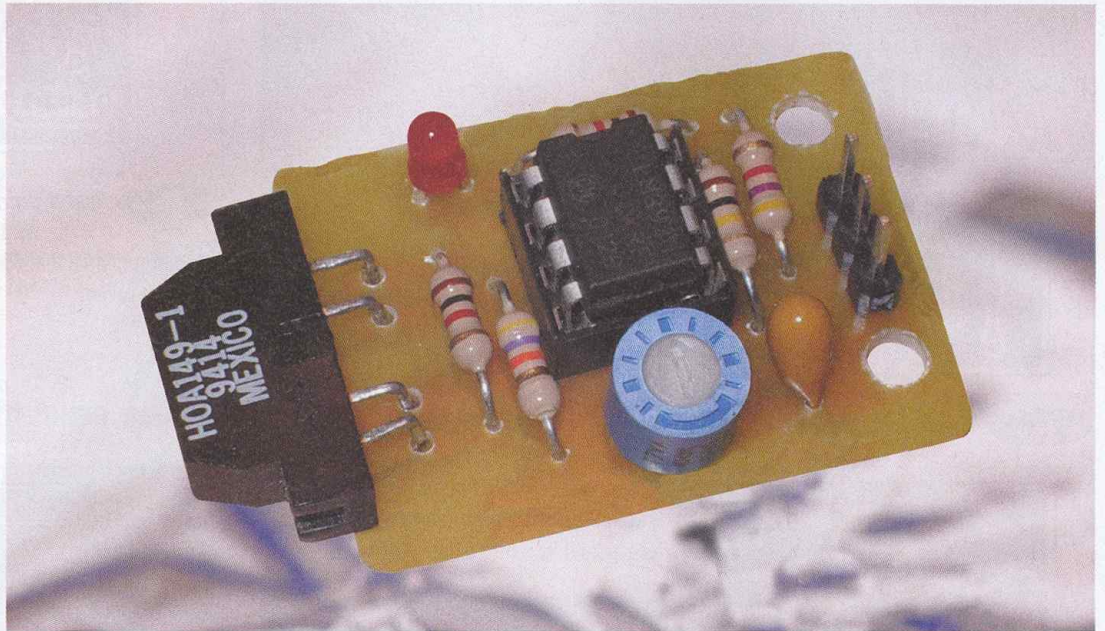


**14** Implantation des composants



Aspect du PIC 26043S

# Détecteur de ligne

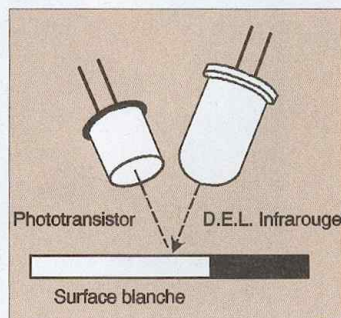


*Un détecteur de ligne est un accessoire classique sur les robots qui utilisent des lignes noires ou blanches pour se repérer. Nous vous proposons un montage très simple délivrant une information logique fiable.*

## La problématique

Souvent utilisé avec une électronique sommaire, il délivre une information proportionnelle à la quantité de lumière réfléchie par la ligne (**figure 1**).

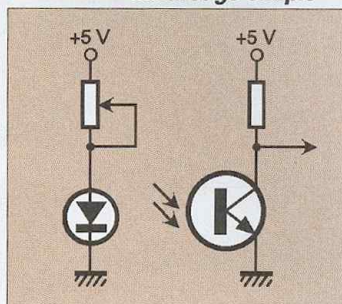
Cette solution, tout à fait correcte, nécessite une entrée logique pour détecter la transition entre le blanc et le noir (**figure 2**) ou une entrée analogique pour détecter l'approche de cette transition. Si l'on souhaite réaliser un robot qui doit suivre une ligne



**2** La transition entre le noir et le blanc doit être parfaitement détectée

peut être corrigé en jouant sur la quantité de lumière émise par la diode infrarouge. Un autre élément important à prendre en compte est la compatibilité avec une carte de contrôle. Et pour finir, l'utilisation d'un capteur à réflexion qui intègre la diode émettrice infrarouge, le phototransistor adapté et dont le positionnement de l'un par rapport à l'autre est optimisé pour détecter une cible à 5 mm environ.

**1** Le schéma d'utilisation d'un capteur infrarouge simple



avec une bonne précision, ce schéma est suffisant en utilisant une entrée analogique et à condition d'écrire un programme qui réalise un étalonnage avant d'utiliser la valeur lue. Mais il est souvent possible de se contenter d'une entrée logique pour informer le robot de la transition noir/blanc. Le problème qui se présente alors est la concordance entre la transition et l'information logique délivrée.

Le seuil de basculement du transistor

## Schéma électronique

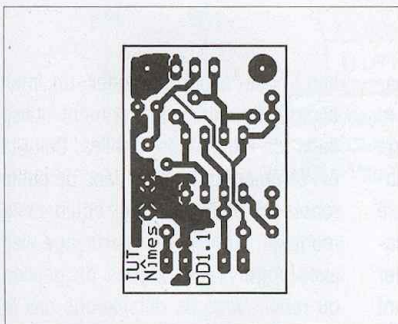
Malgré sa simplicité, le schéma choisi permet de répondre aux différents points critiques (**figure 3**). Hormis le capteur, l'élément central est un circuit intégré très classique comportant deux comparateurs à sortie à collecteur ouvert. Ce point est indispensable, on le verra plus loin.

Le capteur à réflexion est un modèle classique qui pourra être remplacé par n'importe quel autre modèle, du moment que les pattes sont en ligne. En général, les correspondances sont

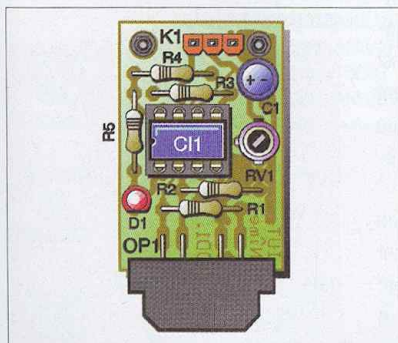
respectées mais il y a toujours des exceptions. Le capteur est polarisé par une résistance fixe, de même que le phototransistor. Le seuil de déclenchement sera réglé par une résistance ajustable. Les deux tensions sont comparées avec un circuit intégré LM393, comparateur double. On réalise une légère hystérésis avec la résistance R3 afin de s'affranchir des petites variations dues aux poussières. La compatibilité est obtenue facilement avec la résistance de charge du collecteur ouvert R4. Ce dispositif permet de réaliser l'adaptation avec des entrées dont les seuils ne sont pas connus à l'avance. On complète le montage avec un indicateur lumineux pour permettre le réglage en local du montage. La led rouge s'éclairera lorsque le module sera sur une surface blanche.

## Réalisation (figures 4 et 5)

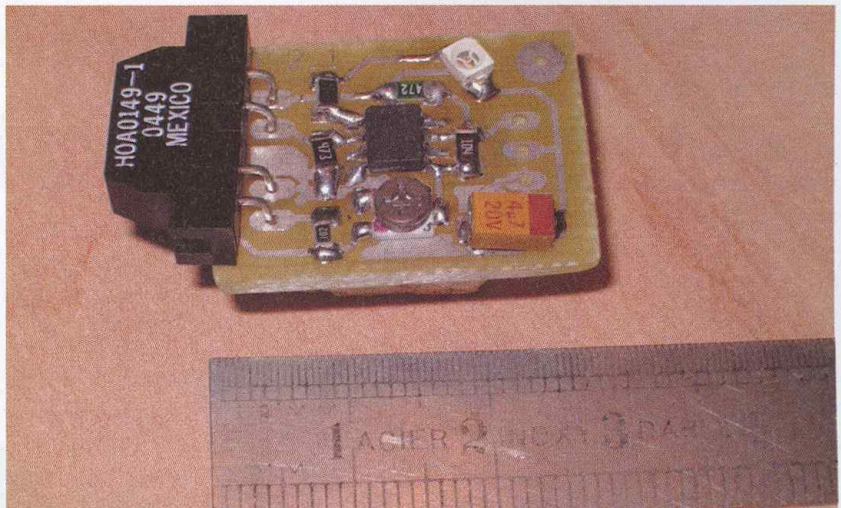
Ce montage ne présentant pas de difficulté, il sera réalisé en quelques minutes. Dans l'ordre, souder les 5 résistances, le support du circuit intégré, le condensateur, la résistance ajustable, la led rouge 3mm et pour finir le capteur à réflexion qui doit être soudé au préalable. Le connecteur K1 sera réalisé avec une barrette sécable mâle à 3 points.



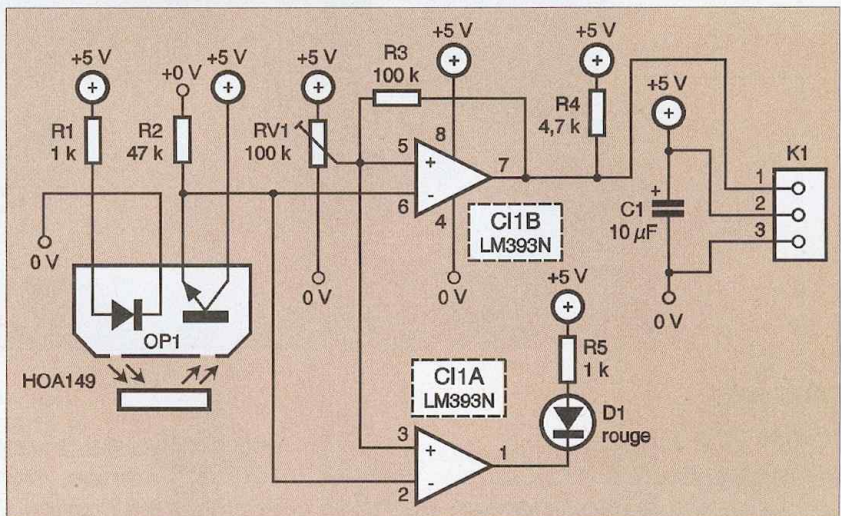
**4** Le circuit imprimé obtenu est le plus petit possible



**5** Implantation des éléments



*Si vous trouvez que l'ensemble prend encore trop de place, vous pouvez le réaliser en CMS. Mais cela ne devient intéressant que si on déporte le capteur*



**3** Le schéma électronique est très simple

## Test

Sur une feuille blanche, placer un bout de ruban adhésif noir. Régler la résistance variable RV1 à mi-course, placer le circuit intégré sur son support et alimenter le montage par le connecteur JP1. Régler la résistance variable pour obtenir un allumage de la led rouge lorsque vous passez du noir au blanc. Tester le réglage en reculant le détecteur. En l'absence d'un fonctionnement normal, vérifier le sens d'implantation du capteur à réflexion et du circuit intégré.

## Caractéristiques

Tension de fonctionnement : 3 à 12 V  
Consommation : 6,7 mA (sous 5 V)

Distance de détection : 3 à 30 mm (suivant éclairage ambiant)

Compatible microcontrôleur

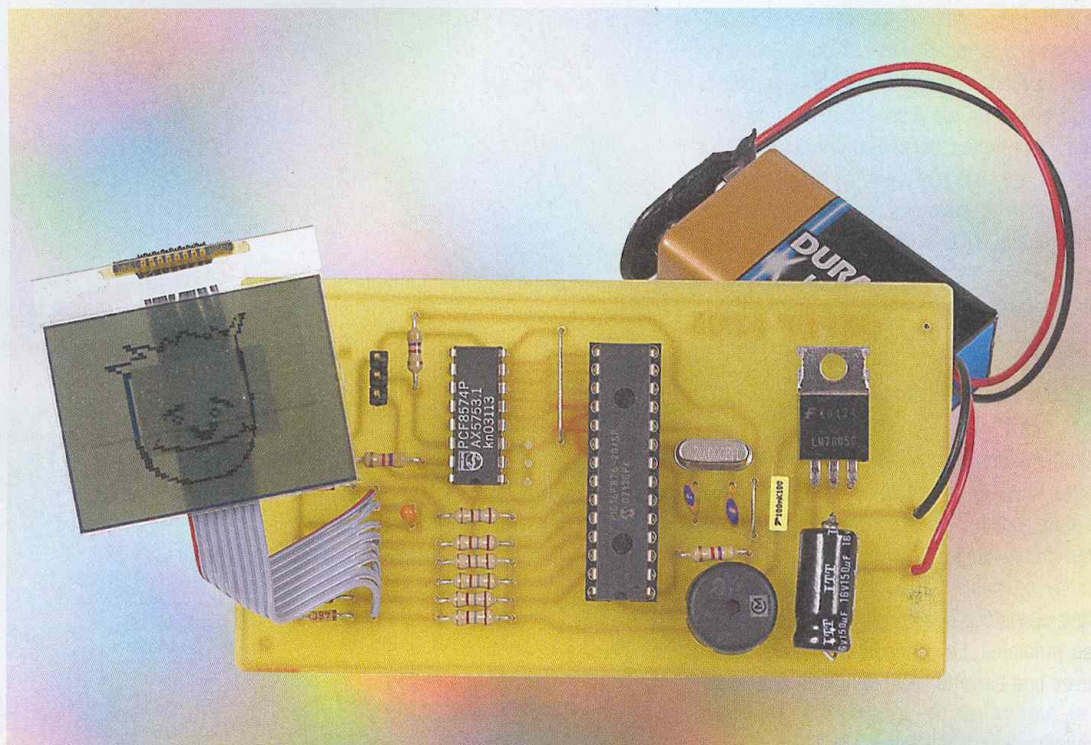
F. GIAMARCHI  
giamarchi@iut-nimes.fr  
www.geii.iut-nimes.fr/fg/

## Nomenclature

- R1, R5 : 1 kΩ (marron, noir, rouge, or)
- R2 : 47 kΩ (jaune, violet, orange, or)
- R3 : 100 kΩ (marron, noir, jaune, or)
- R4 : 4,7 kΩ (jaune, violet, rouge, or)
- RV1 : 100 kΩ (horizontal bas profil)
- C1 : 10 µF
- D1 : led rouge (3 mm)
- CI1 : LM393N
- OP1 : HOA149, HOA1405
- JP1 : barrette sécable mâle 3 points
- Support 8 broches



# Donnez un air à vos robots



*Le montage présenté ici vous propose de donner à vos robots la capacité de communiquer une émotion ou une expression grâce à l'image et au son. Cette fonction vous permettra d'améliorer l'interface homme/machine de votre robot et d'augmenter, de cette manière, son interactivité avec l'homme.*

## Description

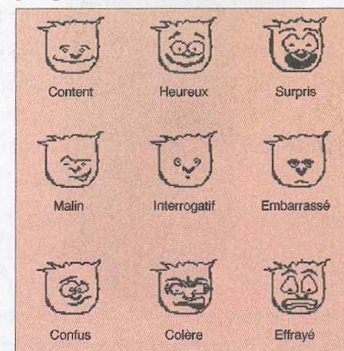
La robotique demande à la fois des compétences en électronique et en mécanique. Cependant, pour les robots destinés à interagir avec l'homme, il est nécessaire de développer une capacité à créer des interfaces hommes/machines. En effet, dans le cas des robots conçus pour directement servir ou distraire l'homme, la fonction d'interface se révèle la plus importante en faisant passer les fonctions électro-mécaniques au second plan. Pour communiquer avec l'homme, l'interface du robot doit exploiter un langage directement compréhensible par tous. L'affichage de textes et d'icônes en fait partie mais fait appel à des pré-requis importants tels que la connaissance du langage ou l'apprentissage de la signification des images. Les émotions des robots ludiques sont souvent matérialisées par l'emploi de

leds de couleurs dont la compréhension n'est pas toujours évidente et demande systématiquement l'usage d'un mode d'emploi. Aussi, la communication la plus basique et comprise par tous reste les expressions du visage. C'est ce que nous allons exploiter dans notre montage en vous donnant la possibilité, à l'aide d'une interface I2c, de déclencher l'affichage d'une expression que vous pourrez faire correspondre aux situations que rencontre votre robot lors de ses déplacements par exemple. Vous aurez ainsi accès à neuf expressions qui couvrent les expressions du visage les plus significatives. La **figure 1** vous présente ces diverses expressions. Afin d'augmenter encore le réalisme de l'expression, ces dernières sont toutes animées et sont également accompagnées d'un son destiné à faciliter davantage la reconnaissance de l'émotion. Pour réaliser cette fonc-

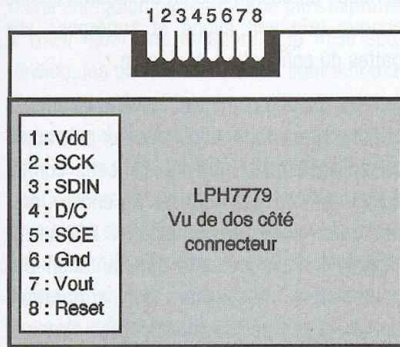
tion, nous allons exploiter un mini écran graphique généralement utilisé dans les téléphones mobiles. Celui-ci est aisément intégrable dans de petits robots et l'interface I2c, équipant le montage, autorise un interfaçage aisé avec votre propre platine de gestion du robot. Nous ne détaillerons pas le



## Les différentes expressions programmables



mini écran exploité ici. Il s'agit du modèle LPH7779 équipé du processeur PCD8544 qui à déjà fait l'objet de montages et de descriptions complètes dans les colonnes de votre magazine. Vous retrouverez son brochage à la **figure 2**.



**2** Le brochage du LPH7779

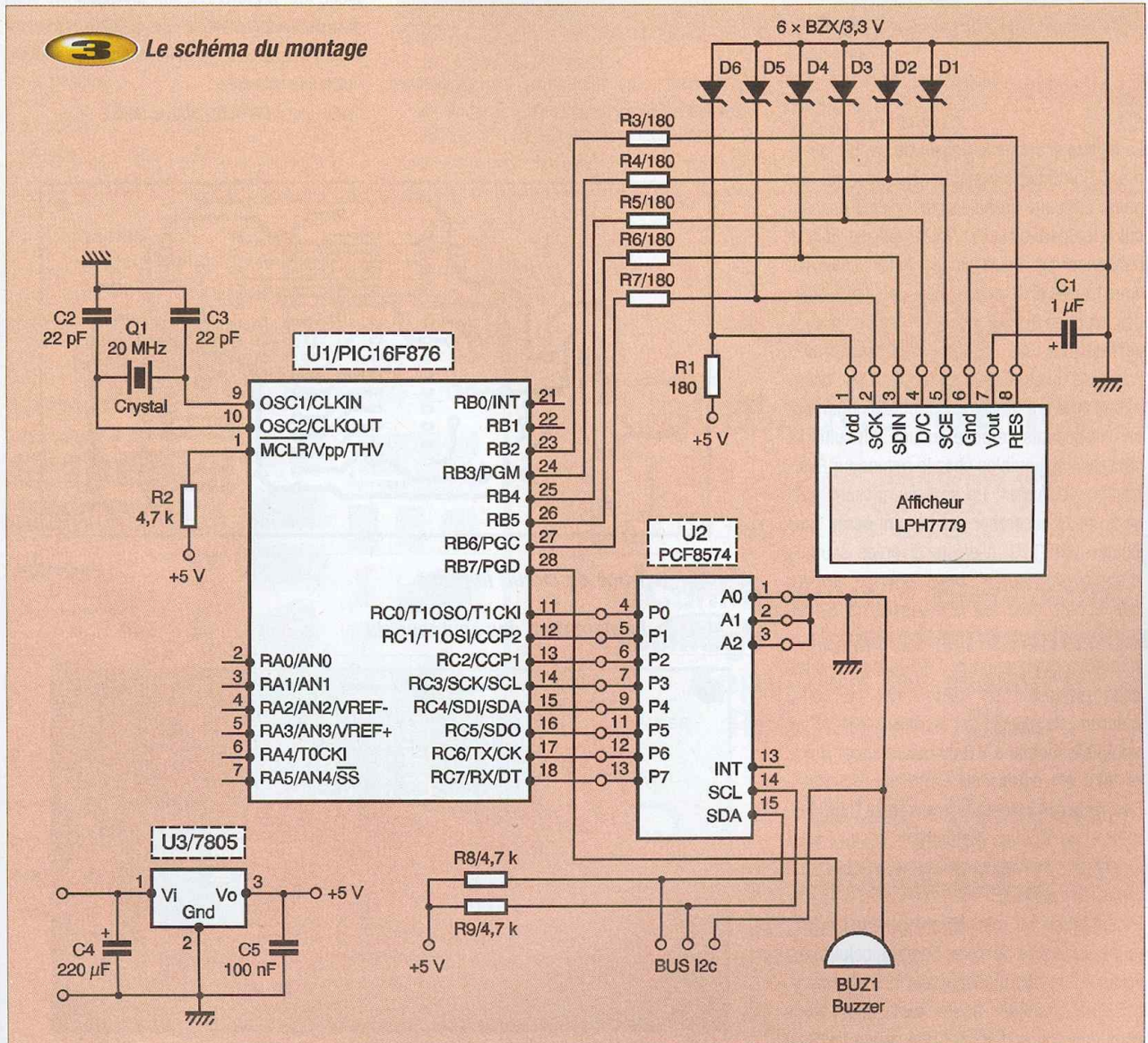
## Schéma de principe

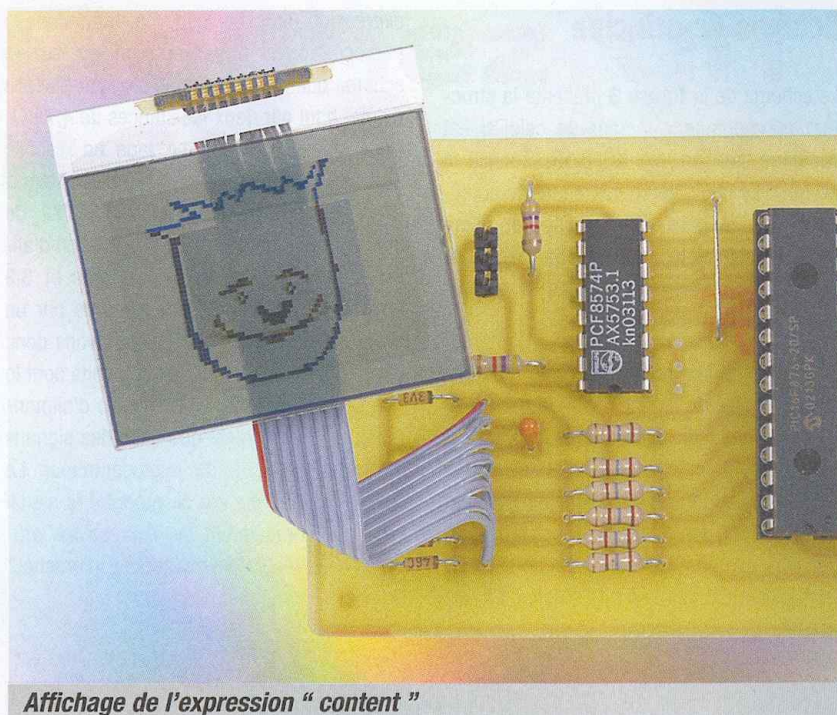
Le schéma de la **figure 3** présente la structure du montage. Le cœur de celui-ci est constitué par un pic 16F876 cadencé à 20 MHz. Le choix de ce modèle a été conditionné par la nécessité de disposer d'une mémoire suffisante, ici 8 Ko, pour permettre le stockage de tous les dessins et du programme de gestion.

L'interface I2c est réalisée à l'aide d'un circuit spécialisé de chez Philips : le PCF8574. Ce circuit met à disposition de tout contrôleur gérant l'I2c 8 ports paramétrables en entrée comme en sortie. Ceux-ci sont directement connectés au port C du Pic. C'est la lecture régulière du port C qui permettra de détecter l'instruction I2c demandant l'affichage d'une

expression stockée dans la mémoire du microcontrôleur. Vous remarquerez sur le schéma que les deux lignes I2c sont placées à l'état haut par deux résistances de 4,7 k $\Omega$ . Dans le cas où votre montage en dispose déjà, vous pourrez ne pas les installer. Le port B est consacré au pilotage du LPH7779. Ce mini écran graphique exige une tension d'alimentation comprise entre 2,7 volts et 3,3 volts. Notre montage étant alimenté par un classique régulateur 7805, nous avons donc utilisé des diodes zéners de 3,3 volts dont le rôle consiste à abaisser la tension d'alimentation de l'écran ainsi que celle des signaux de contrôles issus du microcontrôleur. Le buzzer dont le rôle est de produire le signal sonore accompagnant les expressions affichées est directement connecté à la broche 7 du port B.

**3** Le schéma du montage





Affichage de l'expression "content"

## Réalisation

La **figure 4** donne le dessin du circuit imprimé. Celui-ci est classique mais présente une petite difficulté au niveau des pistes passant entre les pattes du Pic. Vous veillerez donc à l'absence de court-circuit entre celles-ci. Vous trouverez le dessin de 6 pastilles autour des broches 1,2 et 3 du PCF. Celles-ci vous permettront de câbler plus facilement l'adresse du circuit I2c en fonction de l'occupation de votre bus I2c. L'approvisionnement en composants ne pose aucune difficulté, ils sont tous disponibles chez le revendeur Électronique Diffusion. Le seul composant qu'il vous faudra chercher est le mini écran graphique LPH7779. Il existe diverses sources d'approvisionnement. Vous pourrez trouver ces écrans chez les revendeurs de pièces détachées pour mobiles et Internet vous fournira un grand nombre d'adresses de sites vous proposant de commander ce genre d'écran. Rappelons ici que le LPH7779 a équipé le mobile 3310 du constructeur Nokia et qu'il est équipé du contrôleur PCD8544. Ces diverses références devraient vous permettre de trouver rapidement ce que vous cherchez à l'aide de votre moteur de recherche préféré.

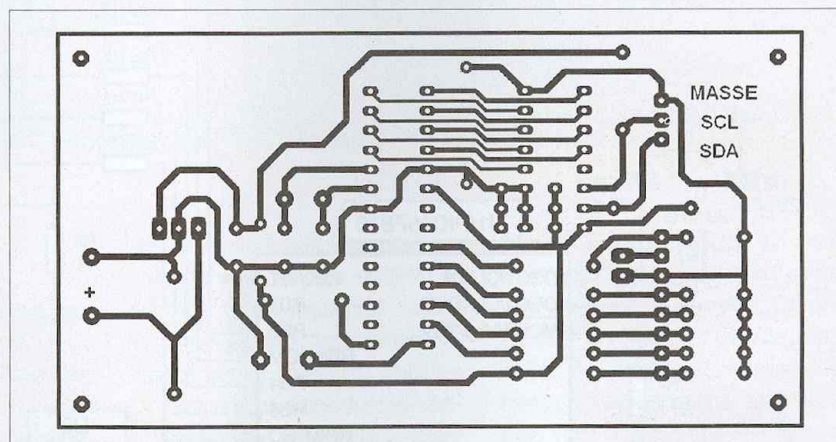
L'installation du mini écran constitue également une étape délicate. En effet, celui-ci est équipé d'un micro connecteur non standardisé. Pour procéder à son installation, nous vous proposons d'utiliser une nappe de fils à

sertir dont vous dénuderez les extrémités pour les enduire d'étain.

Poursuivez en appliquant une goutte d'étain sur chaque patte du connecteur de l'écran en prenant garde à ne pas surchauffer l'écran, n'hésitez pas à faire une pause entre chaque soudure. Une fois cette opération achevée, fixez à l'aide d'un morceau de ruban adhésif la nappe au dos de l'écran de façon à ce que chaque brin soit aligné et superposé aux pattes du connecteur de l'écran.

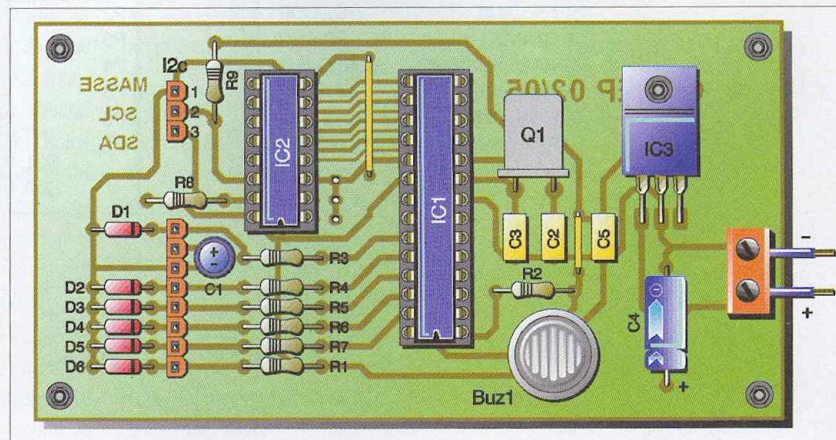
À l'aide d'une pince à bec fin, alignez chaque brin à chaque patte du connecteur puis appliquez simplement votre fer sur cette combinaison afin de la solidariser. Ici encore, prenez garde à ne pas surchauffer l'écran. La **figure 5** vous présente l'implantation des composants ainsi que leur orientation. Commencez classiquement par les straps et les résistances et poursuivez la construction en installant les composants en fonction de leur taille. Les circuits intégrés prennent place de préférence sur un support. Vous pourrez programmer le pic à l'aide du fichier AIR.HEX que vous trouverez sur notre site à l'adresse suivante :

[www.electroniquepratique.com](http://www.electroniquepratique.com).



**4** Tracé du circuit imprimé

**5** Implantation des composants



## Exploitation du montage

Ainsi que nous l'indiquions en début d'article, ce sont les instructions reçues par l'interface I2c qui vont déclencher l'affichage d'une animation permettant de faire exprimer à votre robot une émotion. À la mise sous tension, les ports du PCF 8574 sont automatiquement paramétrés en sortie et reflètent directement l'instruction reçue sur le port I2c. Pour écrire dans le PCF, il faut d'abord préciser l'adresse du circuit intégré, définie par la connexion des broches A0, 1 et 2 sur le circuit imprimé, en envoyant l'octet précisé à la **figure 6** suivi de l'instruction : ici en gras.

Nous ne précisons pas ici les bits spécifiques au bus I2c et vous recommandons, si vous voulez en savoir plus, de vous référer au mode d'emploi du bus I2c disponible sur le site du fabricant Philips. Les bits D0 à D6 peuvent prendre les valeurs décimales suivantes en fonction de l'expression dont vous avez besoin :

- En colère : 30
- Confus : 40

0	1	0	0	A2	A1	A0	0	0	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	----	----	----	---	---	----	----	----	----	----	----	----

### 6 L'adressage du PCD8574

- Ennuyé : 50
- Heureux : 60
- Interrogatif : 70
- Malin : 80
- Effrayé : 90
- Surpris : 100

Lorsque aucune instruction n'est envoyée, l'émotion par défaut correspond à l'état « content ». Le programme gérant le pic se

charge de lire régulièrement l'état du port C de manière à afficher l'état qui lui est demandé et ce, de façon continue. Si vous souhaitez revenir à l'état par défaut, il vous suffit d'envoyer une valeur différente de celles listées plus haut.

G. EHRETSMANN  
<http://g.ehretsmann.free.fr>

## Nomenclature

IC1 : Pic 16F876 20 MHz + support 28 broches tulipes étroit (Electronique Diffusion)  
 IC2 : PCF 8574 port 8 bits pour bus I2c (Electronique Diffusion)  
 IC3 : régulateur 7805  
 C1 : 1 µF/16 V Tantale  
 C2, C3 : 22 pF  
 C4 : 220 µF/16 V horizontal  
 C5 : 100 nF  
 R1, R3 à R7 : 180 Ω (Marron, Gris, Marron)  
 R2, R8, R9 : 4,7 kΩ (Jaune, Violet, Rouge)

D1 à D6 : Diode Zéner BZX 3V3  
 Q1 : quartz 20 MHz  
 Buz1 : transducteur piezo  
 Nappe à sertir  
 Ecran graphique : LPH7779 à 8 broches ou compatible à récupérer ou à commander, en autre, chez  
<http://www.jelu.se/shop.php>  
 Coupleur pile 9 V  
 Pile 9 V

844 pages, tout en couleurs

**Selectronic**  
L'UNIVERS ELECTRONIQUE

Catalogue Général  
**2006**  
☎ 0 328 550 328

[www.selectronic.fr](http://www.selectronic.fr)

B.P. 10050 • 59891 • LILLE Cedex 9 • Fax : 0 328 550 329  
 Magasin de LILLE : 24C de l'Orée du Golf, 16 rue Jules Verne • 59791 Ronchin  
 Magasin de Paris : 11 place de la Nation • 75011 • ☎ 155 258 800 - Métro Nation

**NOUVEAU**

## Catalogue Général

# Selectronic

L'UNIVERS ELECTRONIQUE

Connectique • Electricité  
 Outillage • Librairie technique  
 Appareils de mesure  
 Robotique • Etc.

Plus de 15.000 références

Coupon à retourner à : **Selectronic** B.P 10050 • 59891 LILLE Cedex 9

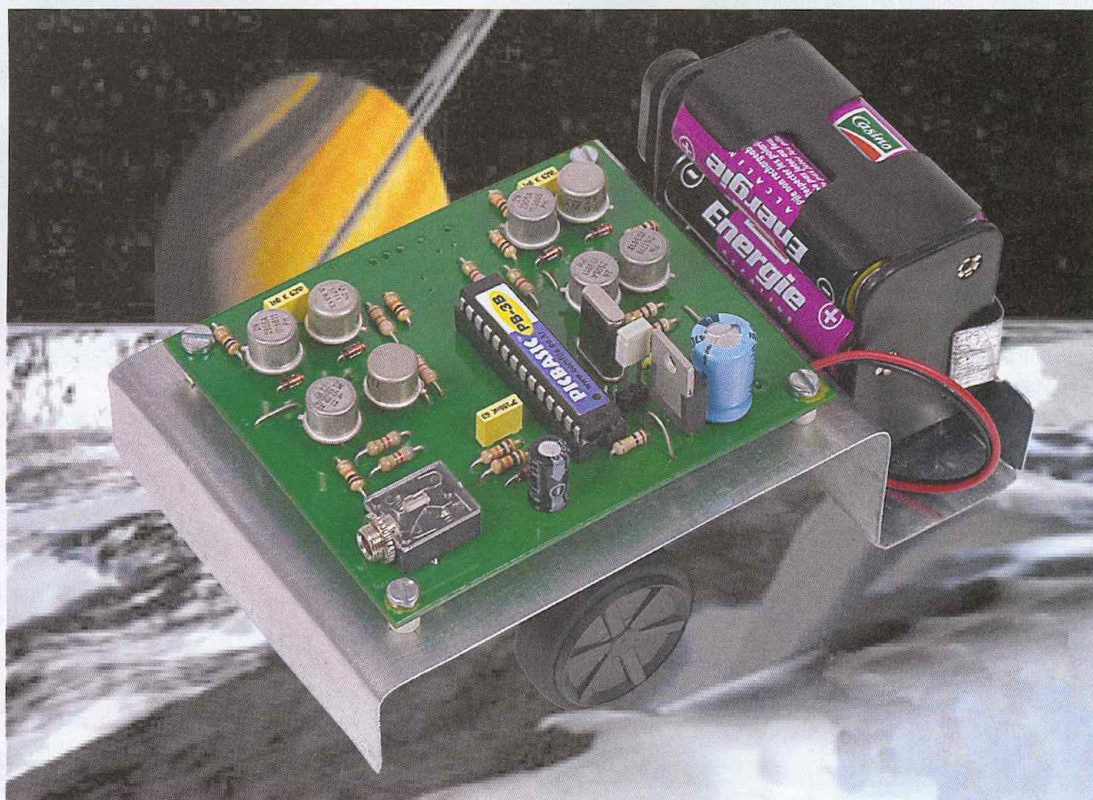
OUI, je désire recevoir le **Catalogue Général 2006 Selectronic** à l'adresse suivante (ci-joint 10 timbres-poste au tarif "lettre" en vigueur) :

EP

Mr. / Mme : ..... Tél : .....  
 N° : ..... Rue : .....  
 Ville : ..... Code postal : .....

"Conformément à la loi informatique et libertés n° 78.17 du 6 janvier 1978, Vous disposez d'un droit d'accès et de rectification aux données vous concernant"

# Mobile à déplacement aléatoire



*Avec cette réalisation, nous vous proposons de mettre en œuvre un double bloc moteur à 2 roues, garnies de pneus gomme. Cette approche simplifie énormément les étapes mécaniques souvent délicates et sources de découragement. Nous aurons surtout l'occasion de découvrir une manière simple de piloter 2 moteurs indépendants à l'aide de ponts en H à transistors.*

Il ne s'agira donc pas à proprement parler d'un robot très sophistiqué, mais plutôt d'une réponse moderne à leur motorisation. Ensuite, il ne sera guère difficile d'exploiter cette base pour y adjoindre quelques capteurs et modifier en conséquence le programme très réduit que nous avons chargé dans la mémoire Flash du célèbre microcontrôleur Pic Basic PB-3B. Une astuce logicielle permettra de choisir parmi 8 modes de déplacements, simplement en indiquant un nombre décimal à 3 chiffres. Il sera donc possible d'enchaîner des figures très particulières, pendant un délai précis et dans un ordre quelconque. Un déplacement sur un trajet précis pourra être mémorisé et reproduit à volonté.

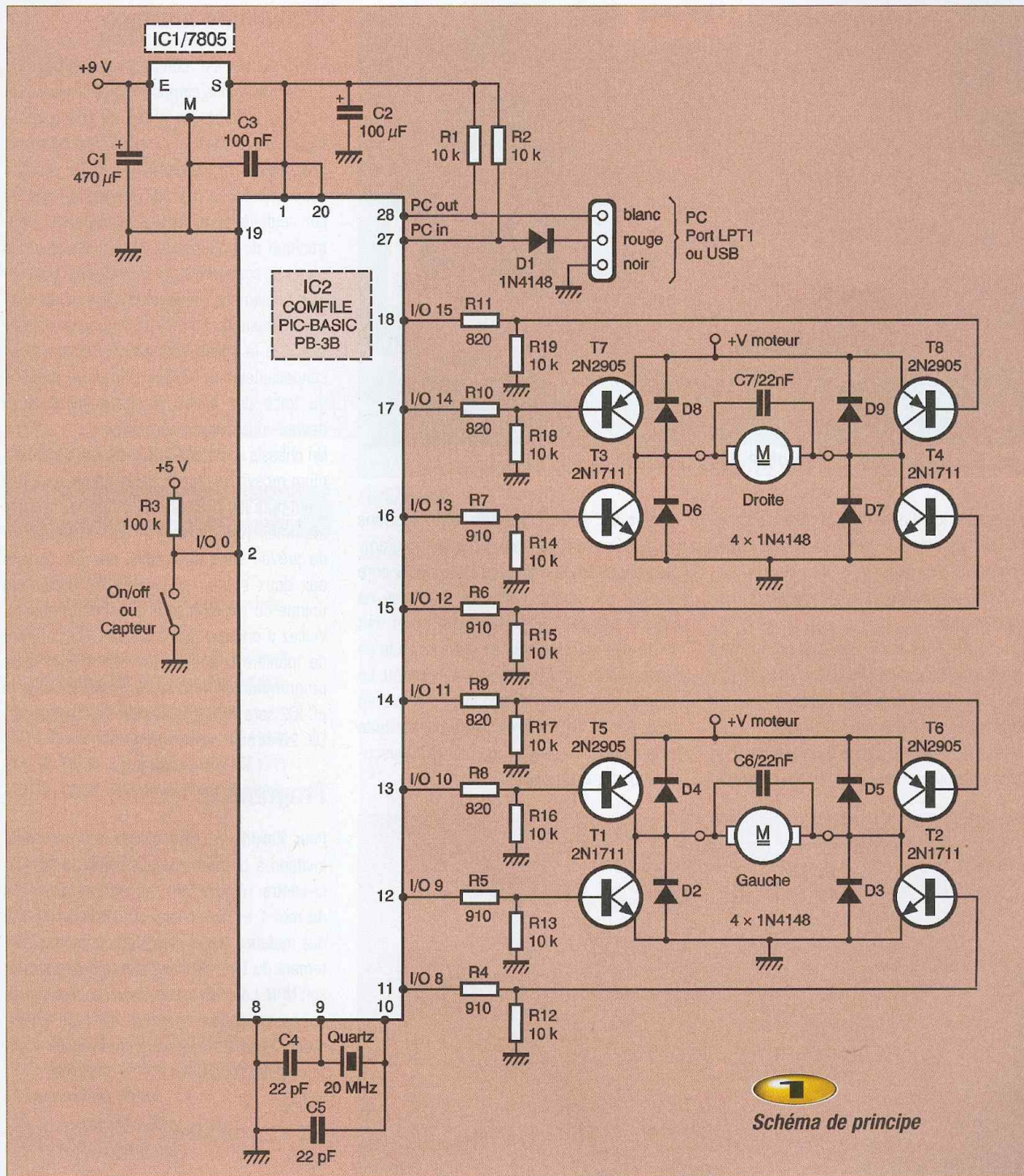
## La commande des moteurs

Nous utiliserons un module mécanique fort compact, comportant 2 minuscules moteurs à courant continu et toute la pignonnerie réductrice de vitesse avec 2 roues à pneus gomme.

Ce produit, fabriqué par Microrobot et disponible chez le distributeur Lextronic, porte la référence MRM-009. La tension nominale préconisée pour les moteurs est de 5 volts, avec une puissance de 0,46 W par élément. La vitesse à vide est de l'ordre de 20 000 tr/mn, avec une consommation de 70 mA. Après réduction, la vitesse des roues est donnée pour environ 354 tr/mn. Il suffit d'appliquer

une tension continue sur les 2 fils d'un moteur à CC pour le voir s'animer. Pour un changement du sens de rotation, il convient bien entendu d'inverser les polarités aux bornes du moteur. Deux solutions sont proposées habituellement : l'utilisation des contacts de 2 relais inverseurs ou un montage statique faisant appel à 2 transistors NPN associés à 2 transistors PNP. On parle dans ce cas de structure en H (voir figure 2).

Cette seconde solution a été retenue, malgré sa complexité de commande, car elle offre une rapidité d'inversion bien supérieure. Par contre, on se doit de noter une légère chute de tension dans les transistors, ce qui oblige parfois à prévoir une alimentation plus généreuse.



## Schéma électronique

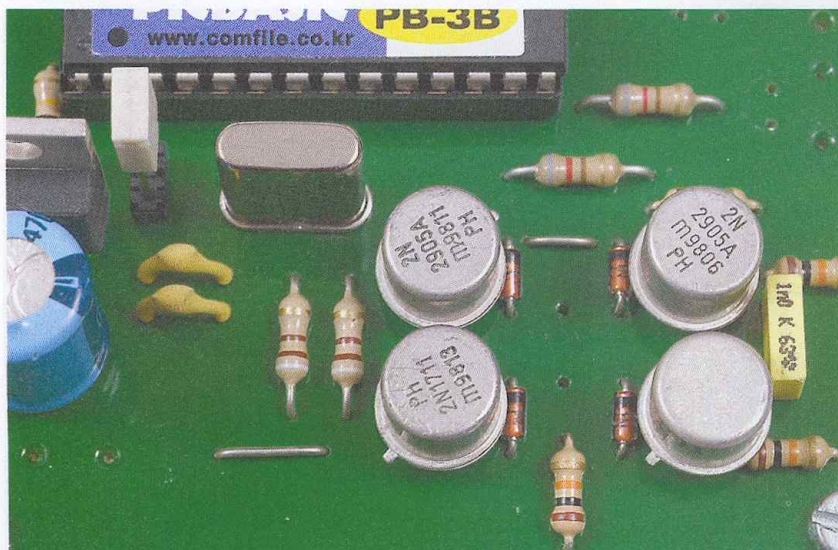
Il est donné sur la **figure 1** et s'articule autour du désormais célèbre microcontrôleur COMFILE Pic Basic PB-3B, déjà présenté maintes fois dans cette revue. Il se trouve que nous allons mettre à contribution 8 sorties spécifiques du  $\mu\text{C}$ , formant un octet, à savoir les ports I/O 8 à I/O 15. Ces 8 valeurs binaires seront individuellement chargées de

pilotez chacune un transistor parmi les 8 nécessaires pour la commande complète des 2 moteurs.

Nous y reviendrons sous la rubrique concernant le programme. Pour chaque moteur, on trouve donc 2 couples de transistors complémentaires. Par exemple, pour le moteur gauche, il faudra commander simultanément T1 et T6 (soit I/O 9 et I/O 11) pour obtenir un sens de marche. L'autre sens nécessitera de

faire appel au couple T2-T5 avec les broches I/O 8 et I/O 10.

Petit rappel : pour rendre passant un transistor de type NPN, il suffira de porter au niveau 1 sa commande de base alors que pour un modèle PNP, la mise au niveau 0 sera requise. Quelques diodes et un condensateur de faible valeur améliorent le fonctionnement de cet ensemble très fiable et peu coûteux. Le fonctionnement du moteur de droite est



Les quatre transistors NPN/PNP, de la structure en "H"

strictement identique. Pour le reste du schéma, une alimentation de 5 volts est nécessaire, à partir d'un bloc de 6 piles de 1,5 volt, avec une régulation et un étage de filtrage fort classique.

A l'aide de 3 fils seulement, on pourra raccorder le montage au PC, via le port imprimante ou une entrée USB (selon le type de cordon et le système d'exploitation).

L'entrée I/O 0 sur la broche 2 sera dotée d'un petit interrupteur ou d'un strap amovible

comme sur notre maquette. Nous pourrions ainsi provoquer une interruption du programme à tout instant. Cette entrée pourra encore recevoir un capteur LDR pour une mise en marche optique. Notre but reste avant tout de dévoiler les subtilités de la commande de puissance des moteurs à courant continu. Le lecteur pourra, en développant sa propre carte, greffer de nombreux capteurs sur cette base mobile, et modifier de pair le minuscule programme proposé.

## Réalisation pratique

Notre circuit est compact et regroupe la quasi-totalité des composants du schéma. On trouvera le tracé des pistes de cuivre sur la **figure 3** à l'échelle et l'implantation des composants sur la **figure 4**. Quelques pastilles supplémentaires ont été prévues au cas où l'on souhaiterait alimenter séparément l'électronique de commande (le Pic Basic) et les moteurs, sous une tension différente. Dans ce cas seulement, on devra OUTER le strap spécial et implanter en plus un régulateur de tension de la valeur souhaitée, avec quelques condensateurs de filtrage. Une étude attentive du tracé des pistes et des pastilles vides devrait vous éclairer davantage.

Un châssis rudimentaire en tôle ou en aluminium recevra le bloc des 2 moteurs, un logement pour les piles et sur le dessus, la carte imprimée (voir photos). Il est indispensable de prévoir sous le châssis, du côté opposé aux deux roues, une pièce de frottement unique ou mieux encore une petite roulette. Veillez à orienter le connecteur jack 3,5 mm de manière à pouvoir insérer le cordon de programmation vers le PC. Bien entendu, le  $\mu C$  IC2 sera monté en dernier sur un support DIL 28 étroit à broches tulipes.

## Programme

Pour l'expliquer plus aisément, nous vous invitons à prendre connaissance du tableau ci-contre récapitulant les entrées sollicitées du mot 1 et les actions obtenues au niveau des moteurs. Nous avons tenu compte directement du type de transistor utilisé, soit PNP soit NPN. La somme des divers poids binaires donne un nombre décimal qui sera finalement utilisé dans le programme avec l'instruction BYTEOUT.

Rappel :

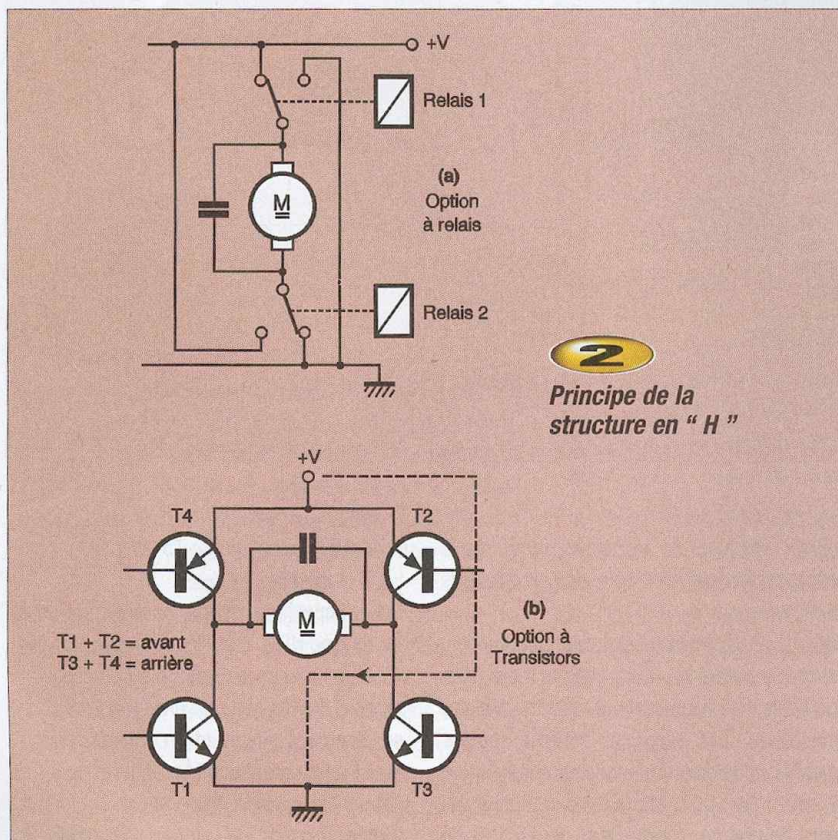
PNP à 1 = non passant    PNP à 0 = passant

NPN à 0 = non passant    NPN à 1 = passant

Voici finalement les quelques lignes de programme Basic réalisant les 8 figures différentes, séparées par un délai de 1 000 ms.

L'ordre des nombres décimaux dans l'instruction TABLE détermine les enchaînements des mouvements du mobile. On pourra aisément obtenir des déplacements particuliers en alternant les mouvements (BYTEOUT, xxx) et leur durée (DELAYxxx).

Nous ne doutons pas que vous saurez exploiter cette réalisation pour y adjoindre de nom-



Et si vous réalisez

Moteur	Sens	PNP	PNP I/O15	NPN I/O14	NPN I/O13	PNP I/O12	PNP I/O11	NPN I/O10	NPN I/O9	Décimal I/O 8
GA	AV	1	1	0	0	1	0	0	1	201
GA	AR	1	1	0	0	0	1	1	0	198
DR	AV	1	0	0	1	1	1	0	0	156
DR	AR	0	1	1	0	1	1	0	0	108
GA+DR	AV	1	0	0	1	1	0	0	1	153
GA+DR	AR	0	1	1	0	0	1	1	0	102
DR	AV	0	1	1	0	1	0	0	1	105
+GA	AR									
GA	AR	1	0	0	1	0	1	1	0	150
+DR	AV									
Poids binaires		128	64	32	13	8	4	2	1	

Tableau de commande

## Nomenclature

### Semi-conducteurs :

IC1 = régulateur intégré 5 volts positifs, 7805, boîtier TO 220  
 IC2 = micro contrôleur COMFILE Pic Basic PB-3B, boîtier DIL 28 étroit (LEXTRONIC)  
 D1 = diode commutation 1N 4148  
 D2 à D8 = diodes commutation 1N 4148  
 T1, T2, T3, T4 = transistors NPN 2N 1711  
 T5, T6, T7, T8 = transistor PNP 2N 2905

### Résistances (toutes valeurs 1/4 de watt)

R1, R2 = 10 kΩ  
 R3 = 100 kΩ  
 R4 à R7 = 910 Ω  
 R8 à R11 = 820 Ω  
 R12 à R19 = 10 kΩ

### Condensateurs :

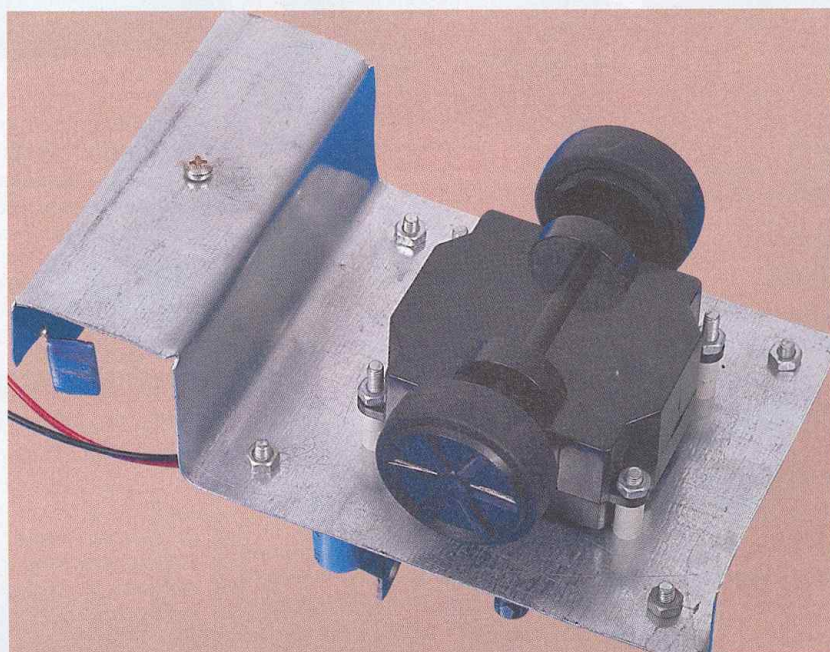
C1 = chimique vertical 470 µF / 25 volts  
 C2 = chimique vertical 100 µF / 25 volts  
 C3 = plastique 100 nF  
 C4, C5 = céramique 22 pF  
 C6, C7 = plastique 22 nF

### Divers :

Double bloc moteur MRM-009 à 2 roues, tension 5 volts (LEXTRONIC)  
 Support à souder DIL 28 étroit à broches tulipes ou 2 supports DIL 14  
 Quartz à fils 20 MHz  
 Connecteur pour 6 piles 1,5 volt  
 Coupleur pression pile 9 volts  
 Connecteur à souder CI, jack 3,5 mm stéréo  
 2 picots à wrapper + strap de shuntage  
 Aluminium, entretoises, visserie

## Programme

```
' robot aléatoire 2005
DIM T as BYTE, N as BYTE, C as BYTE      ' déclaration variables
DEB: C = KEYIN(0,25)                       ' présence du shunt ON/OFF ?
IF C = 1 THEN GOTO DEB                    ' ...cycle de 8 figures si présent
FOR N = 0 TO 7
    T = TABLE(N,201,198,156,108,153,102,105,150)  'équivalent décimal des codes binaires
    BYTEOUT T, T                            ' lecture du mot 1 sur 8 bits
    DELAY 1000                              ' tempo 1 seconde
NEXT N
BYTEOUT 1,0 : DELAY 5000                   ' désactivation des moteurs 5 secondes
GOTO DEB                                   ' retour
```



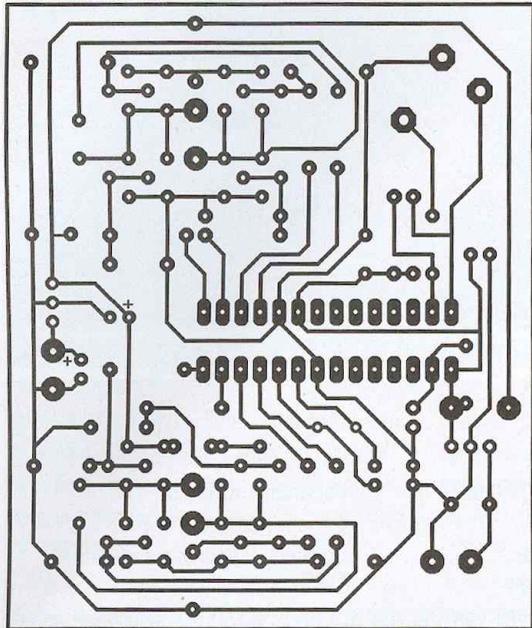
Simplicité du châssis et double bloc-moteur



breuses fonctionnalités ou capteurs divers pour voir enfin s'animer votre propre robot. En tout cas, le  $\mu$ C PB-3B dispose encore de nombreuses entrées, certaines analogiques,

pour donner vie à vos projets les plus ambitieux.

### 3 Tracé du circuit imprimé

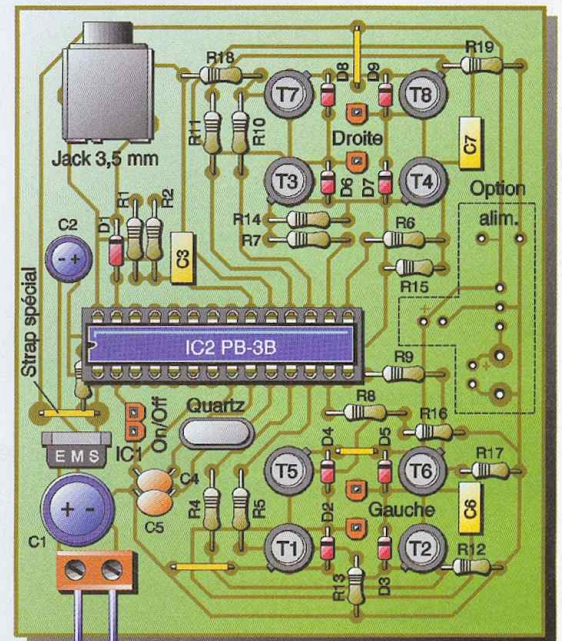


### 4

### Implantation des éléments

G. ISABEL

PC port LPT1 ou USB

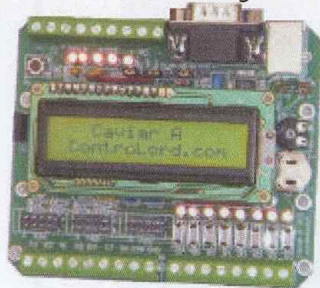


+9 V PIC Masse  
(Bloc de 6 piles 1,5 V)

## Starter Kit Système embarqué Caviar

Gratuit : + compilateur C

Gratuit : + noyau multi-tâche



### Caviar A

AVR ATmega32  
32k Flash  
4 entrées analogiques  
16 entrées/ sorties numériques  
1 relais  
2 moteurs (pour petit robot)  
LCD 2 lignes x 16 caractères  
USB, Rs232  
Horloge temps réel(option)  
Prix : 190 E. Prix de lancement jusqu'à fin mai 2005 : **149 E**



### Caviar B, Caviar C

AVR ATmega32/128  
32k/128k Flash  
8 entrées analogiques  
Entrées/ sorties numériques  
8 relais  
Rs232 x 2  
Horloge temps réel\*  
LCD graphique 128x64 \*  
Clavier 4x4 \*  
8 entrées opto-couplées \*  
3 sorties analogiques \*  
Carte à mémoire CompactFlash  
Prix : à partir de **270 E**

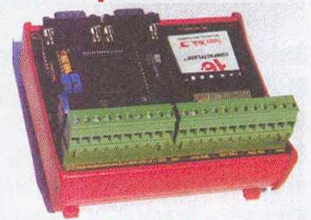
Avec IDE complet :  
Editeur, assembleur,  
Compilateur C (ANSI C)  
Débogueur in situ par USB ou Rs232.  
Point d'arrêt, pas à pas  
Drivers, exemples en assembleur et en C  
Mutex : Noyau multi-tâche

## Data Logger sur CompactFlash



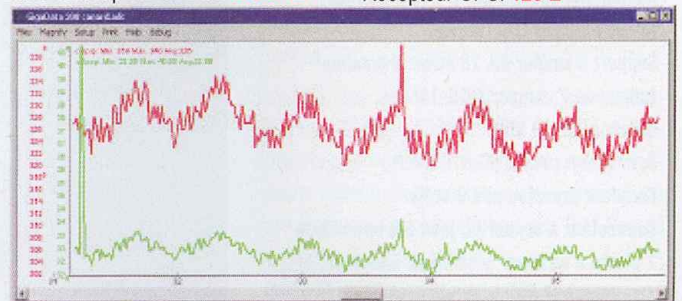
### GigaLog

8 entrées analogiques  
0..2,5V ou 0..20mA ou compteurs.  
Cadence de 1 ms à 24 heures  
Enregistrement des données Rs232 jusqu'à 115200 baud  
Mémoire jusqu'à 2 Gb  
Programmable en C  
Carte seule: **250 E**  
Dans coffret pour rail DIN: **290 E**

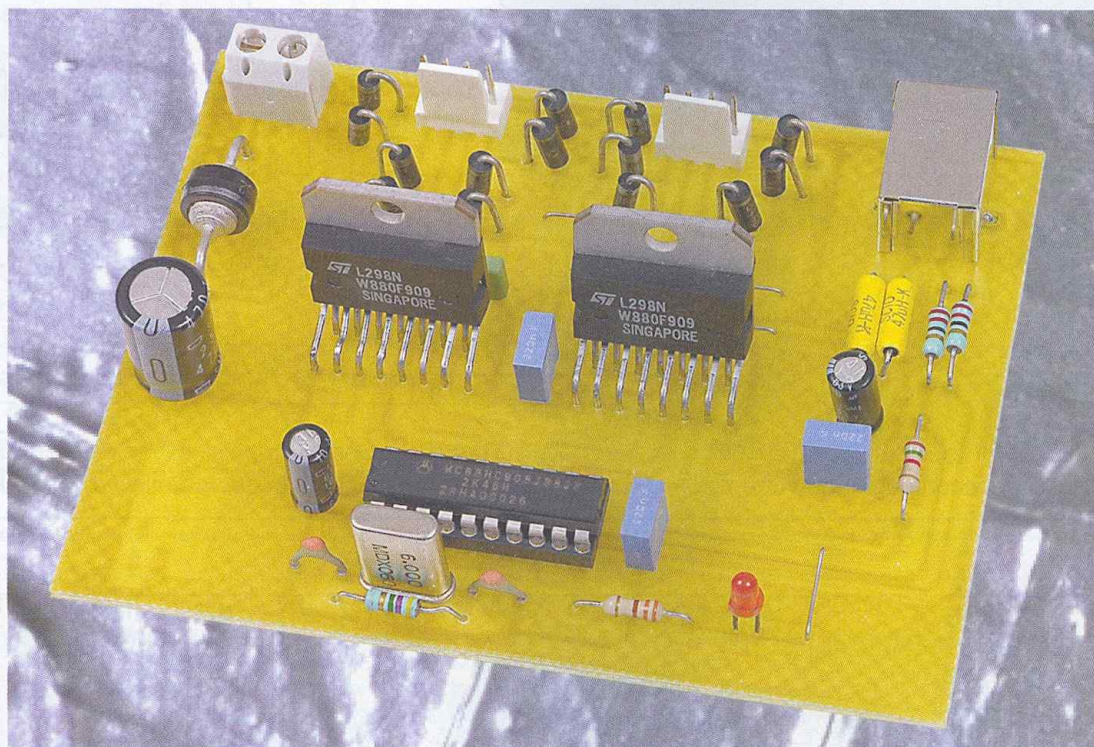


### FlashLog

8 entrées analogiques  
0..2,5V ou 0..20mA  
Consommation zéro entre deux échantillonnages.  
Mémoire jusqu'à 32 Mb  
Configuration, transfert de données, alarme par modem, GSM, SMS.  
Modem GSM + antenne: **275 E**  
Récepteur GPS: **120 E**



# Commande USB pour moteurs pas à pas



*Très appréciés en robotique, les moteurs pas à pas font régulièrement l'objet de toutes les attentions aux travers de petits montages dédiés à leur pilotage. Ce mois-ci, c'est le bus USB qui a été retenu comme organe de liaison avec un PC qui fournira les commandes. Le montage que nous vous proposons dans ces pages permet de piloter simultanément deux petits moteurs bipolaires, ce qui devrait vous ouvrir la voie pour de nombreuses applications.*

## Schéma

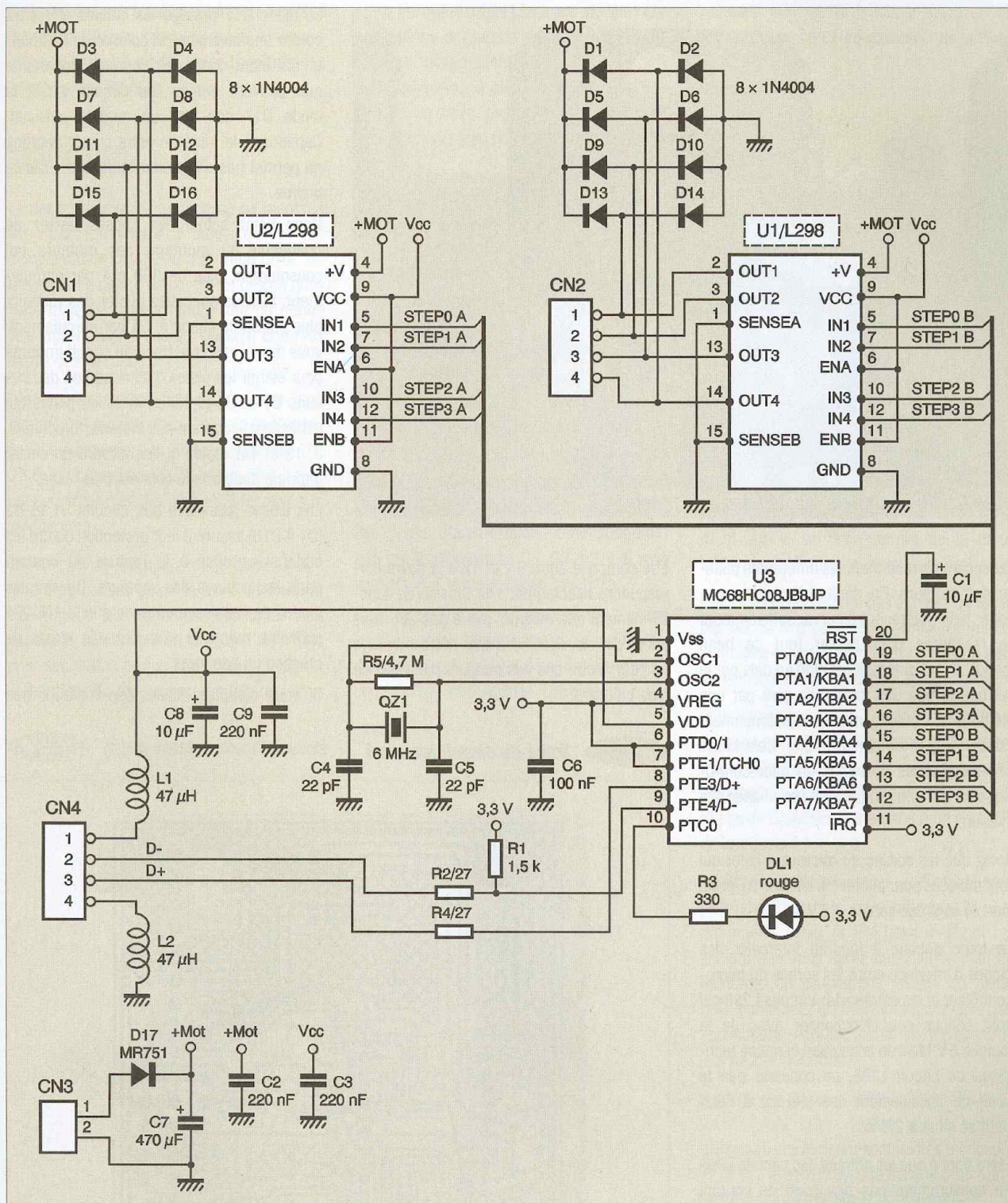
Le schéma de notre montage est reproduit en **figure 1**. On y distingue très aisément le microcontrôleur (U3) et les circuits dédiés au contrôle des moteurs pas à pas (U1 et U2). Une fois n'est pas coutume, nous commencerons la description de ce schéma en abordant les alimentations.

L'alimentation du montage sera fournie en partie par le bus USB. En effet, toute la section logique du montage sera alimentée directement par le PC (ou par un HUB) au travers de la liaison USB. La consommation sur Vcc étant relativement modeste (environ 30 à 40 mA), notre périphérique USB n'aura pas besoin de réclamer au PC de passer sa sortie en courant max (500 mA). Cela arrange bien nos affaires car

sinon, il faudrait prévoir une structure qui permette au microcontrôleur de démarrer les négociations via le port USB pendant que le reste du montage est en mode basse consommation. Notre montage étant relativement simple, nous n'aurons pas besoin de tout cela.

L'alimentation 5 V fournie par le bus USB est filtrée au moyen de L1, L2, C8 et C9. Cela permet de lisser le courant d'alimentation qui circule dans le câble, ce qui limite le rayonnement électromagnétique du cordon USB au strict minimum. Notre microcontrôleur fonctionne avec une tension interne de 3,3 V. Fort heureusement, le microcontrôleur en question (MC68HC908JB8) possède un régulateur interne. Il suffit donc d'alimenter le microcontrôleur en 5 V sur sa patte nommée VDD

(broche 5). La sortie du régulateur interne à besoin d'être filtrée par le condensateur externe C6 (car il est très difficile d'inclure des condensateurs de forte valeur dans les circuits intégrés). Le régulateur interne du microcontrôleur n'est pas conçu pour fournir un courant important au monde extérieur. Il faut donc veiller à ne pas trop consommer de courant sur la sortie VREG du microcontrôleur (broche 4). Dans notre cas, la tension de 3,3 V fournie par le régulateur du microcontrôleur servira tout d'abord à polariser la résistance R1. Cette résistance détermine la vitesse de communication sur le bus USB (vitesse lente dans notre cas). La tension de 3,3 V sert également à alimenter une petite diode Led qui sera pilotée par le port PTC0 du microcontrôleur (broche 10).



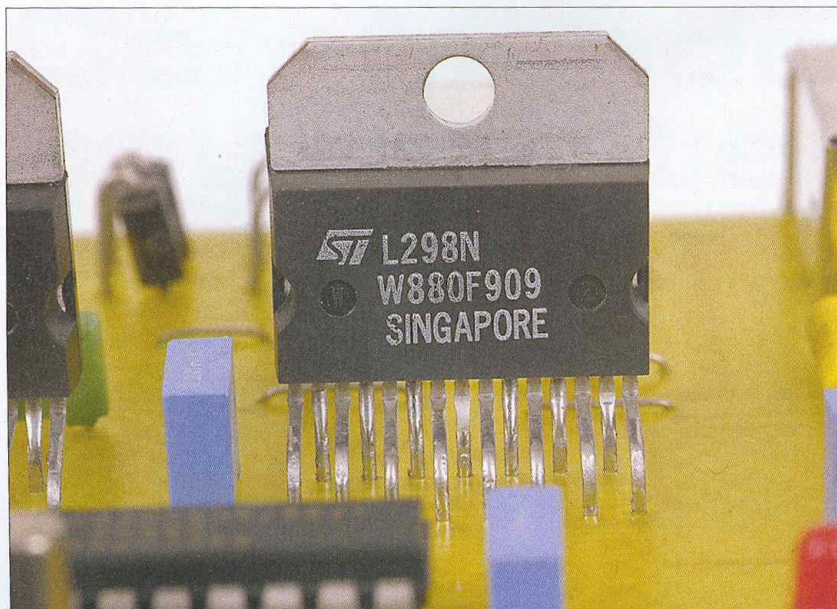
## 7 Schéma de principe

Cette diode s'allumera pour signaler que les moteurs sont en mouvement.

L'oscillateur nécessaire au fonctionnement du montage est intégré dans le microcontrôleur. Il suffit donc de lui adjoindre un quartz (ici 6 MHz) et le tour est joué. C'est simple,

mais il ne faut pas oublier les condensateurs C4 et C5 qui forment le réseau de déphasage nécessaire à l'entretien des oscillations et la résistance R5 qui fixe le fonctionnement de la porte inverseuse intégrée dans U1 dans sa zone linéaire.

Abordons maintenant le pilotage des moteurs. Les bobines des moteurs seront contrôlées par les circuits U1 et U2 qui sont commandés directement par le microcontrôleur. Ce dernier devra gérer simultanément la liaison USB et la progression des moteurs en



Les L298N contrôlent les bobines des moteurs

cadencant correctement les différentes phases des moteurs. En dépit des apparences, notre petit microcontrôleur a suffisamment de ressource pour piloter tout ce beau monde. Il faut dire qu'il est aidé en cela par le fait que le bus USB peut être géré par une méthode de polling avec un cadencement assez lent (10 ms par exemple). Cela laisse largement assez de temps au microcontrôleur pour gérer la progression des phases des moteurs.

Notez que les sorties du microcontrôleur qui sont utilisées pour piloter les circuits U1 et U2 (port A) sont des sorties 3,3 V.

En toute rigueur, il faudrait intercaler des étages d'interface entre les sorties du microcontrôleur et les entrées des circuits L298 qui sont conçus pour fonctionner avec de la logique 5 V. Mais en consultant la notice technique du circuit L298, on constate que le seuil de basculement des entrées à l'état haut se situe à 2,3 V.

Étant donné que les entrées des circuits L298 ne consomment pas beaucoup de courant (moins de 100  $\mu$ A), notre brave petit microcontrôleur n'aura aucune difficulté à fournir des signaux à l'état haut. En revanche, il faut tout de même remarquer que, ce faisant, nous avons fortement réduit la marge de bruit de cette partie du montage. Pour un montage expérimental, cela reste tout de même acceptable.

L'alimentation des moteurs sera fournie par une alimentation externe. La tension pourra

être comprise entre 9 V et 12 V et devra correspondre exactement à la tension de fonctionnement des moteurs pas à pas que vous aurez choisis. Précisons que notre montage ne peut piloter que des moteurs par à pas de type bipolaire.

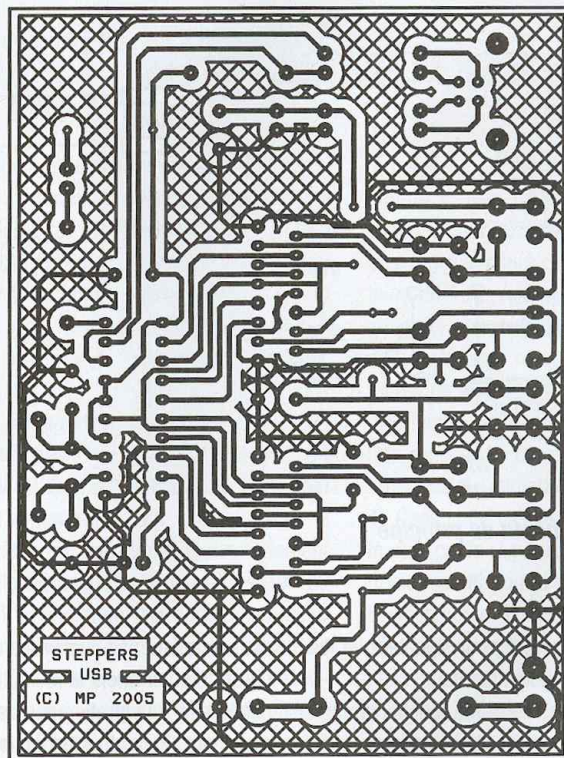
La diode D17 protège les circuits U1 et U2 contre une inversion de polarité de l'alimentation. Étant donné les courants importants que peuvent débiter les circuits L298, la diode D17 doit être un modèle puissant. Cependant, le tracé de notre circuit imprimé ne permet pas d'exploiter toute la puissance prévue.

Dans l'état actuel des choses, évitez de connecter au montage des moteurs qui consomment plus de 300 mA par enroulement. Si vous souhaitez piloter des moteurs plus puissants (jusqu'à 1 A par enroulement) vous devrez revoir le tracé du circuit imprimé pour élargir les pistes d'alimentation des circuits U1 et U2 (broches 4) et les pistes qui véhiculent le courant aux moteurs (broches 2, 3, 13 et 14). Notez qu'en utilisant un circuit imprimé double face cela est plus facile.

Les diodes associées aux circuits U1 et U2 (D1 à D16) assurent leur protection contre les surtensions dues à la rupture du courant dans les bobines des moteurs. De simples diodes de redressement telle que la 1N4004 suffisent, même si elles ont des temps de réaction un peu lents.

Si vous comptez utiliser des moteurs très

## 2 Tracé du circuit imprimé



inductifs (ce qui n'est pas toujours spécifié dans le catalogue des revendeurs) il vaudra mieux utiliser des diodes un peu plus rapides telles que les BYV27.

## Réalisation

Le dessin du circuit imprimé de ce montage est visible en **figure 2** tandis que la vue d'implantation associée est reproduite en **figure 3**. Le circuit ne comporte pas de difficulté majeure mais il faudra tout de même être vigilant aux pistes qui passent entre les pattes des circuits U1 et U2.

Les pastilles seront percées à l'aide d'un foret de 0,8 mm de diamètre, pour la plupart. En ce qui concerne les diodes, les circuits U1, U2 et les connecteurs CN1, CN2, CN3 il faudra percer les pastilles avec un foret de 1 mm de diamètre.

Et enfin, en ce qui concerne la diode D17, il faudra percer les pastilles avec un foret de 1,5 mm de diamètre. Le connecteur USB nécessite également un perçage plus gros pour ses pattes de fixation. Il faudra adapter le diamètre de perçage au modèle que vous

aurez acheté (il existe plusieurs modèles disponibles sur le marché avec de petites différences d'implantation).

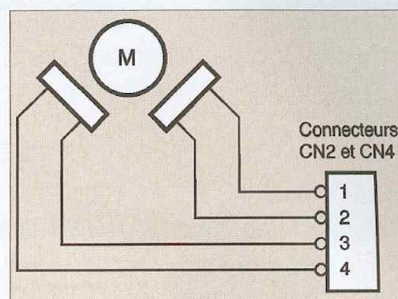
Avant de réaliser le circuit imprimé, il est préférable de vous procurer les composants pour vous assurer qu'ils s'implanteront correctement.

Cette remarque concerne particulièrement le connecteur USB. Les inductances L1 et L2 ne sont pas indispensables. Elles peuvent être remplacées par des straps si le montage est destiné à fonctionner dans un environnement familial (dans ce cas, le câble USB risque de rayonner un peu ce qui pourrait perturber un poste de radio ou un téléviseur situé à proximité du montage).

Pour le reste de l'implantation, il n'y a pas de difficulté particulière. Soyez tout de même attentif au sens des condensateurs et des circuits intégrés. Respectez le découplage des lignes d'alimentation et ne supprimez pas les condensateurs C2, C3 et C9 si vous voulez éviter les mauvaises surprises.

Enfin, vous noterez la présence de 7 straps sur le circuit imprimé qu'il est préférable d'implanter en premier pour des raisons de commodité. Pour câbler correctement vos

moteurs afin qu'ils fonctionnent avec ce montage, vous pourrez vous aider de la **figure 4**.



## 4 Câblage correct des moteurs

Le microcontrôleur U3 sera programmé avec le contenu du fichier « STEPUSB.S19 » que vous pourrez vous procurer par téléchargement sur le serveur Internet de la revue (<http://www.electroniquepratique.com>).

Les lecteurs qui ne sont pas équipés du matériel nécessaire pour programmer ce microcontrôleur pourront se reporter à l'article paru dans Electronique Pratique n°276, page 82.

Pour piloter le montage à partir d'un PC fonctionnant sous Windows à l'aide du programme « STEPPER.EXE » (qui vous sera remis également lors du téléchargement), il ne suffit pas de brancher le montage à un port USB de la machine !

Il faut au préalable installer un pilote de périphérique. Pour cela vous trouverez le programme « usbio\_el\_V151.exe » parmi les fichiers que vous aurez téléchargés pour ce montage. Ce programme installe un pilote USB générique exploitable avec toutes les applications réalisées autour du microcontrôleur MC68HC908JB8.

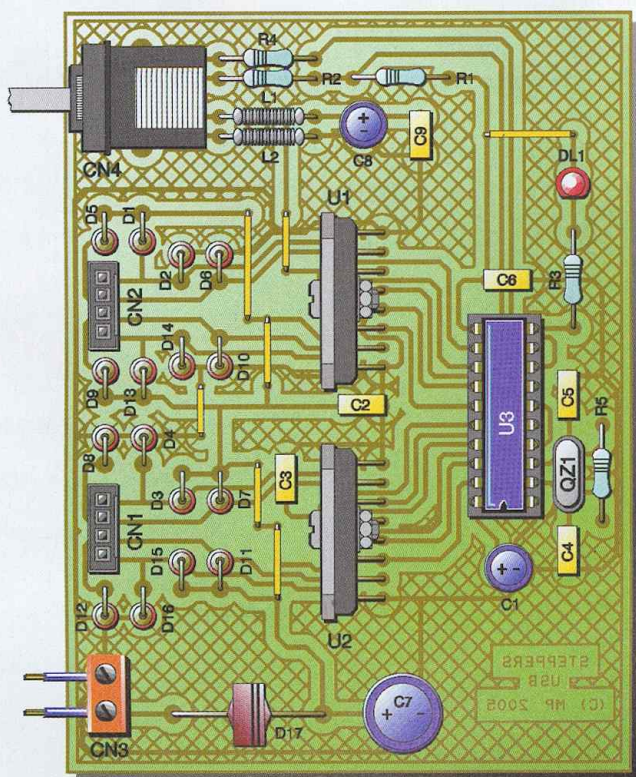
Il s'agit du driver « USBIO Driver Package V1.51 (Light EL) » réalisé par la société Thesycon. Les lecteurs sont invités à consulter le site Internet de cette société pour vérifier si des mises à jour sont disponibles :

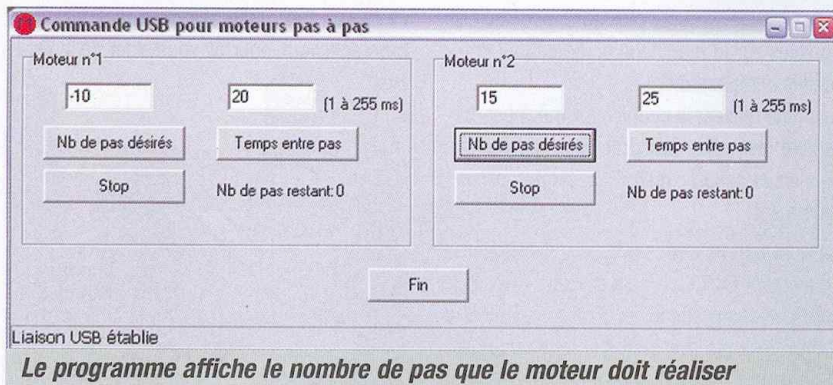
([http://www.thesycon.de/eng/usbio\\_light.shtm](http://www.thesycon.de/eng/usbio_light.shtm)).

Une fois l'installation du pilote USB terminée, il ne vous reste plus qu'à brancher votre montage à l'aide d'un câble USB puis à lancer le programme « STEPPER.EXE » dont l'interface est assez rudimentaire comme vous pouvez le voir sur la **figure 5**.

Notez que si vous utilisez ce montage avec

## 3 Implantation des éléments





**Le programme affiche le nombre de pas que le moteur doit réaliser**

d'autres montages réalisés autour du microcontrôleur MC68HC908JB8, vous pourrez rencontrer des conflits.

En effet, le programme « STEPPER.EXE » ne sait actuellement gérer qu'un seul périphérique USB à la fois (lorsqu'il y a plusieurs montages raccordés au PC, le programme suppose que le 1<sup>er</sup> est le bon).

Si ce comportement vous semble gênant, vous pourrez éventuellement modifier les sources du programme mais il faut une bonne dose de persévérance pour se lancer dans cette aventure (il faut également un compilateur Borland Builder C++).

Sinon, en ce qui concerne l'exploitation du montage, l'utilisation du programme « STEPPER.EXE » est extrêmement simple. Il vous suffit de saisir le nombre de pas que vous voulez imposer au moteur (-127 à +127)

sans oublier de préciser le temps nécessaire à votre moteur pour effectuer un pas (en fonction de sa puissance et de la charge qu'il pilote).

Le temps s'exprime en période de 10 ms (1 pour 10 ms, 2 pour 20 ms, etc.). Généralement, un temps de progression de 10 ms à 20 ms entre les pas est suffisant pour la plupart des moteurs.

Pendant les mouvements, le programme affiche le nombre de pas que le moteur doit encore réaliser.

Si vous avez programmé une vitesse de progression relativement lente (255 ms par exemple) il se peut que vous souhaitiez interrompre le mouvement qui est en cours.

C'est pour cela que le programme comporte un bouton STOP pour chaque moteur. Notez

également que pendant des mouvements lents, vous pouvez ajouter ou soustraire des pas individuellement à chacun des moteurs.

Il suffit de saisir le nombre de pas souhaité et de cliquer le bouton « Nb de pas désiré ».

Le microcontrôleur de votre montage se chargera de faire le bilan du travail qu'il lui reste encore à effectuer.

P. MORIN

## Nomenclature

**CN1, CN2 : Barrette mini-KK, 4 contacts, sorties droites, à souder sur circuit imprimé, référence MOLEX 22-27-2041**

**CN3 : Bornier de connexion à vis, 2 plots, au pas de 5,08 mm, à souder sur circuit imprimé, profil bas**

**CN4 : Embase USB (embase pour connecteur type B)**

**C1, C8 : 10 µF / 25 V, sorties radiales**

**C2, C3, C9 : 220 nF**

**C4, C5 22 µF**

**C6 100 nF**

**C7 : 470 µF / 25 Volts, sorties radiales**

**DL1 Diode LED rouge 3 mm**

**D1, D2, D3, D4, D5, D6, D7, D8 : 1N4004**

**D9, D10, D11, D12, D13, D14 D15, D16 : 1N4004**

**D17 : MR751**

**L1, L2 : Inductance 47 µH**

**QZ1 : Quartz 6 MHz en boîtier HC49/U**

**R1 : 1,5 kΩ 1/4 W 5% (marron, vert, rouge)**

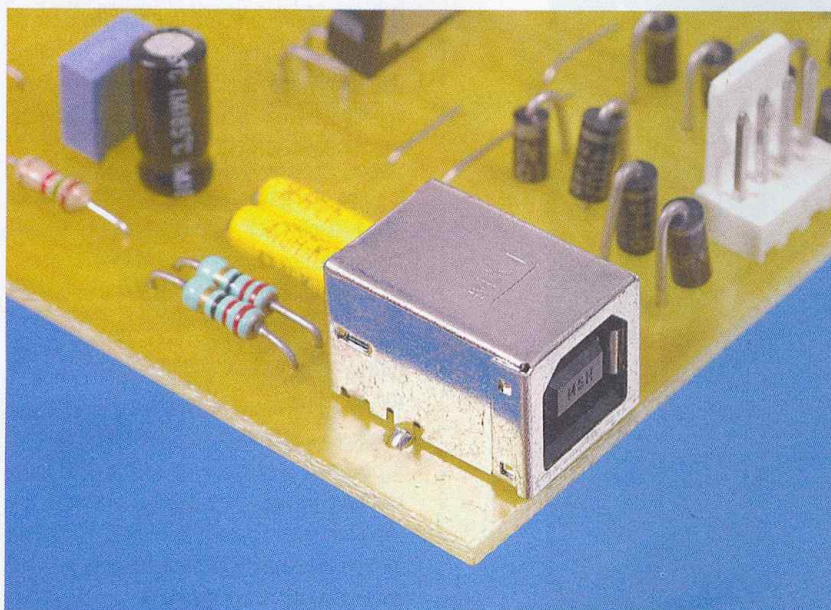
**R2, R4 : 27 Ω 1/4 W 5% (rouge, violet, noir)**

**R3 : 330 Ω 1/4 W 5% (orange, orange, marron)**

**R5 : 4,7 MΩ 1/4 W 5% (jaune, violet, vert)**

**U1, U2 : L298**

**U3 : MC68HC08JB8JP**



**Gros plan sur l'embase USB (CN4)**

# Capteur directionnel



La balise et le détecteur

*Ce montage n'est pas nouveau, nous l'avons en effet conçu il y a précisément 20 ans. Comme l'électronique ne change pas beaucoup, nous le proposons avec quelques modifications.*

## Principe

Ce capteur est destiné à s'installer sur un robot. Il associe deux composants, un émetteur qui sert de balise et un récepteur qui délivre une tension dont la valeur dépend de l'orientation du capteur par rapport à la source lumineuse. La balise envoie sa lumière dans toutes les directions et le robot se dirige vers elle. Nous vous donnons le moyen de fabriquer un capteur mais ce sera à vous d'installer le capteur et d'en tirer des informations pour commander un engin. Le but du capteur directionnel est de délivrer une information qui sera fonction de la position angulaire relative entre le capteur et la balise.

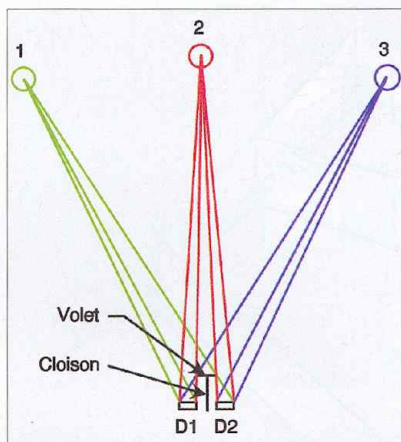
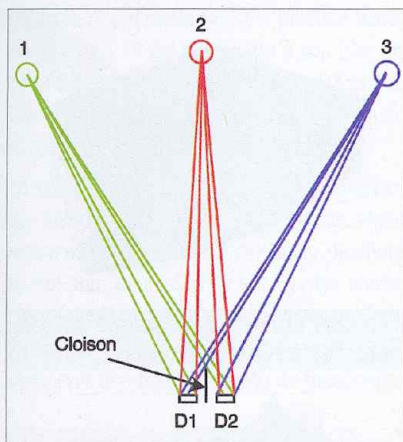
On utilise pour cela un capteur différentiel utilisant deux photodiodes infrarouges montées dans un système optique particulier. La **figure 1** donne le principe différentiel utilisé ici. Lorsque la source est en position 2 par

rapport au détecteur, les deux photodiodes D1 et D2 reçoivent la même intensité lumineuse. Lorsque la source est en 1, la diode D1 est complètement éclairée tandis que la cloison fait de l'ombre. La diode D2 reçoit donc moins de lumière. Si maintenant la balise est en 3, ce sera la diode D1 qui recevra le moins de lumière. Cette configuration et l'éloignement des deux diodes fait que si on déplace légèrement la balise autour de la position 2, les deux photodiodes resteront éclairées identiquement pour une certaine variation angulaire. Une solution consiste à placer les deux diodes côte à côte, ce qui n'est pas toujours facile. Pour notre capteur, nous avons adopté une variante. Elle consiste à placer un petit volet à l'extrémité de la cloison (voir **figure 2**). Le moindre déplacement angulaire de la balise par rapport à la source créera immédiatement une ombre sur l'une des deux photodiodes D1 ou D2. Sur notre figure, aux posi-

tions extrêmes 1 et 3, une seule des diodes est éclairée alors que sans volet, une diode était éclairée et l'autre illuminée sur plus de la moitié de sa surface. Nous reprendrons cette formule en tenant également compte de la géométrie interne et invisible des diodes infrarouges.

Les **figures 3, 4 et 5** donnent les dimensions du capteur réalisé. Il abrite une paire de photodiodes infrarouges BP104 montées tête-bêche avec leurs deux cathodes réunies. Le capteur est réalisé avec du stratifié verre époxy cuivré de 0,3 mm d'épaisseur (ou plus si on veut). Ce matériau cuivré se soude facilement et présente l'avantage de ne pas être traversé par la lumière. La position relative des diverses faces cuivrées facilite le soudage et assure l'isolement des sorties du capteur. Par contre, le cuivre réfléchit la lumière et si on veut éviter les interférences de lumière externe, on recouvrira le cuivre d'une couche de peinture noire mate.

Les diodes de notre capteur ont été soudées sur la plaquette arrière, la gravure est faite suivant la méthode anglaise c'est à dire avec une petite fraise. Nous avons miniaturisé le capteur mais on peut bien sûr l'agrandir, la seule chose à respecter étant la largeur du volet qui sera égale ou supérieure à la distance séparant les deux puces installées dans les photodiodes. Cette distance (**figure 5**), est égale à l'entraxe des diodes moins 2,8 mm. La puce est montée asymétriquement, les deux cathodes (ergots) étant juxtaposées. Compte tenu de l'encapsulation des diodes dans une résine noire transparente aux rayons infrarouges, ces puces sont invisibles. Ce volet a toutefois l'inconvénient de perturber le fonctionnement à très faible distance du capteur. En effet, si la diode est trop proche, elle fait de l'ombre sur les deux capteurs qui conservent néanmoins leur fonctionnement différentiel.



**1** Principe du capteur de base. Un volet placé entre les deux capteurs permet d'avoir une exposition des photo détecteurs qui change avec la position de la source par rapport au capteur

**2** En installant un volet au centre, on rend le système plus linéaire vers le centre

ce permet de s'éloigner des perturbations créées par les sources d'éclairage. Par ailleurs, on est bien au-dessous des fréquences des télécommandes infrarouge qui ne devraient pas trop perturber le système et qui n'émettent généralement qu'un court instant. La fréquence d'émission peut éventuellement être changée en modifiant la valeur d'un condensateur. Cette opportunité permettra d'avoir plusieurs balises en service sur un site, chacune ayant une fréquence associée à sa position particulière. Une reconnaissance de la valeur de la fréquence permettra alors de diriger automatiquement le robot vers l'un de ces points.

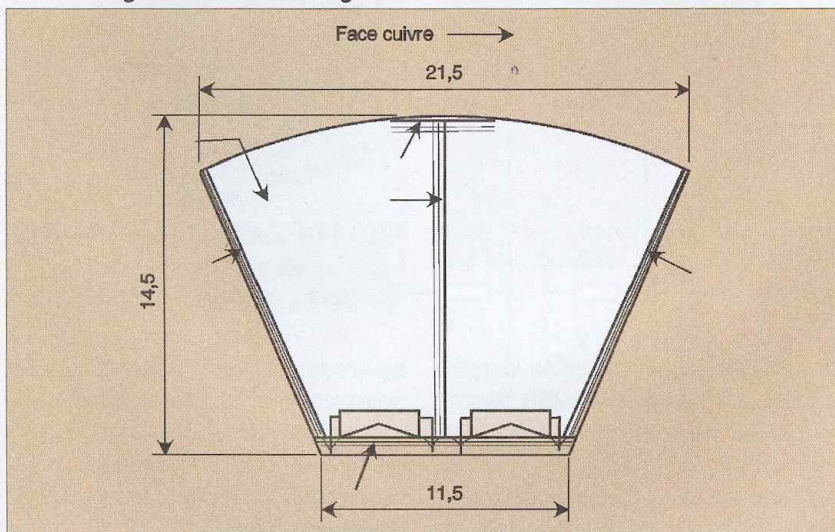
Il est également possible de modifier la fréquence d'émission en changeant la polarisation de la broche 5 du circuit intégré par une

résistance que l'on placera entre cette broche et l'un des points d'alimentation du circuit. La valeur dépend du type de circuit intégré. Pour un NE 555, quelques milliers d'ohms conviennent mais pour le TLC 555, il faut multiplier cette valeur par 20. Le courant est débité par un générateur de courant constitué du transistor T1, de la diode électroluminescente D1 et de la résistance R4. Le transistor doit être capable de débiter un courant relativement important, ce qui explique le choix d'un BC 328 capable de faire passer 1 ampère. La diode D1 sera une diode rouge. En effet, la chute de tension dans les diodes électroluminescentes dépend de la nature du semi-conducteur et, par conséquent de leurs couleurs. On prend une diode standard et non une ultra brillante. Les diodes électroluminescentes émissives sont des classiques LD 271, relativement directives. En utilisant deux de ces diodes, on couvre un angle de plus de 50°. Ces composants génèrent une lumière non cohérente si bien qu'il n'existe aucune interférence entre les deux sources. Le courant de crête est ici de 150 mA, on peut éventuellement augmenter sa valeur. La diode LD 271 admet un courant moyen de 130 mA et le rapport cyclique de la balise est voisin de 1.

## Électronique

L'électronique se compose de deux éléments, un émetteur qui sert de balise et un récepteur. La figure 6 donne le schéma de l'émetteur. Son oscillateur est un TLC 555 ou un NE 555, le TLC 555 a une consommation inférieure mais dans le cas présent, la consommation principale sera due au courant envoyé dans la diode électroluminescente infrarouge, l'emploi d'un 555 classique convient parfaitement. La fréquence de la balise se situe dans une zone tranquille de quelques kilohertz, le choix de cette fréquence

**3** Dimensions de notre capteur, on pourra bien sûr l'agrandir à son gré. Les flèches désignent la face cuivrée

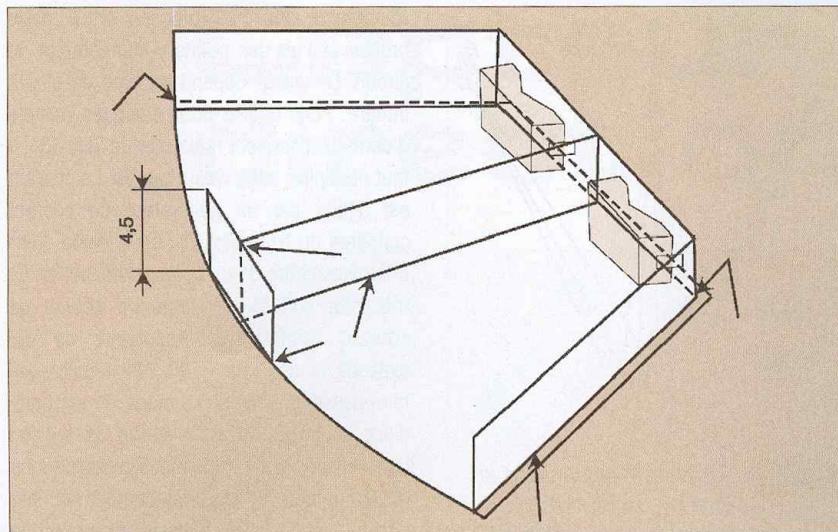


Détail du capteur réalisé en stratifié verre époxy de 0,3 mm

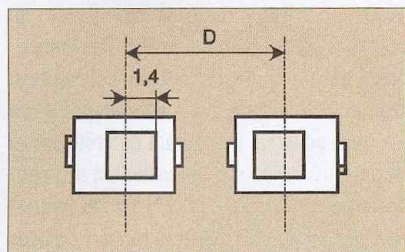
## Récepteur

Le synoptique du capteur est représenté figure 7. Les deux signaux des deux capteurs sont amplifiés après un filtrage destiné à réduire les signaux parasites issu du secteur. Le signal est amplifié. Pour assurer une distance de détection relativement importante avec une plage de distance étendue, on utilise un circuit de commande automatique de gain qui exploite la somme des signaux des deux capteurs. Plus le capteur sera proche de la balise, plus l'amplitude du signal sera importante et plus le gain sera réduit. Le



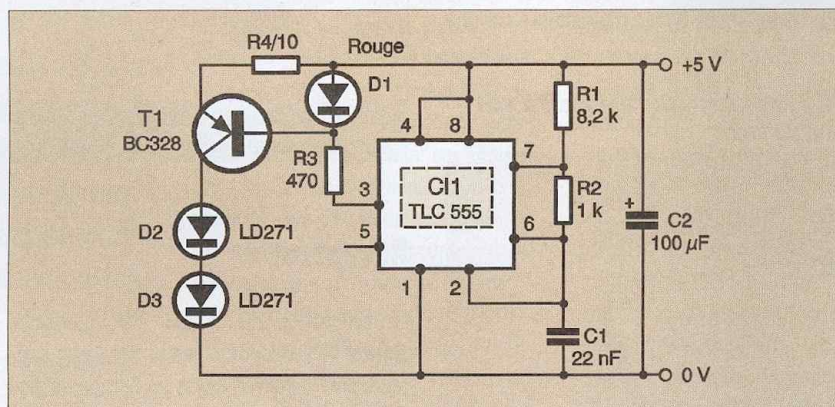


**4** *Vue en perspective du capteur, les flèches montrent les lignes de soudure*



**5** *La largeur du volet sera fonction de l'écartement des capteurs dont nous avons représenté ici la surface sensible cachée par la matière noire de l'enrobage*

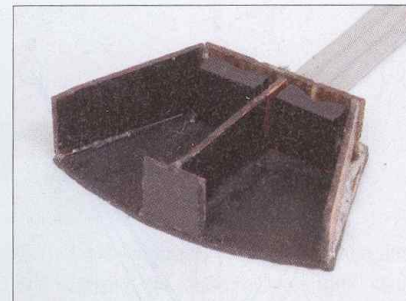
reçoit plus d'énergie que l'autre, la tension diminuera. Le récepteur (**figure 8**) traite le signal électrique capté par les photodiodes. Les résistances R1 et R2 polarisent les diodes en inverse, la variation d'éclairage entraîne une variation de courant convertie en variation de tension par la résistance. Le condensateur C1/C2 transmet la tension à des étages adaptateurs d'impédance en assurant un premier filtrage de type passe-haut. Nous avons installé, derrière, un filtre de type passe-haut à source contrôlée filtre du second ordre ayant une pente de 12 dB/octave. La tension est transmise à un étage amplificateur à gain élevé, la valeur de



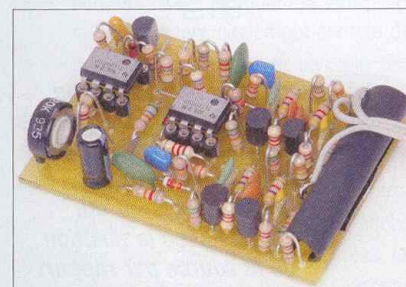
**6** *Schéma de principe de l'émetteur*

second amplificateur mesure la différence de niveau reçue par chacun des capteurs. Si les deux capteurs reçoivent la même énergie, nous aurons une tension donnée. Si le capteur du bas reçoit plus d'énergie, la tension de sortie ira vers le haut, si le capteur du haut

C8 assure un filtrage passe-haut complémentaire. Aux fréquences les plus basses, nous aurons donc un filtrage à 24 dB/octave. Après amplification, le signal est détecté par des diodes classiques. On notera que la polarisation des amplificateurs opérationnels est



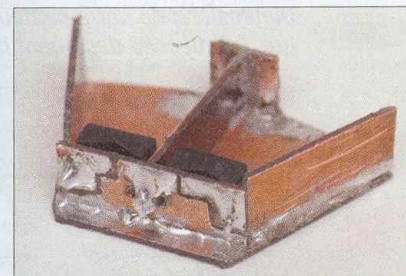
*La face cuivrée a été peinte en noir pour éviter les réflexions*



*Le détecteur et ses amplis*



*Constitution des photo coupleurs, un film adhésif noir empêche la pénétration de la lumière au travers du circuit imprimé. Les boucles des sorties des diodes permettent un réglage de leur position par rapport à la photorésistance (équilibre)*

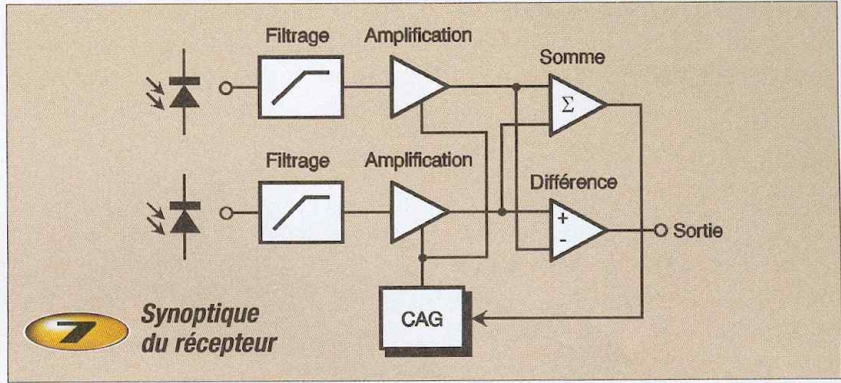


*Vue arrière du capteur, on voit ici la gravure du circuit permettant de connecter les deux photodiodes*

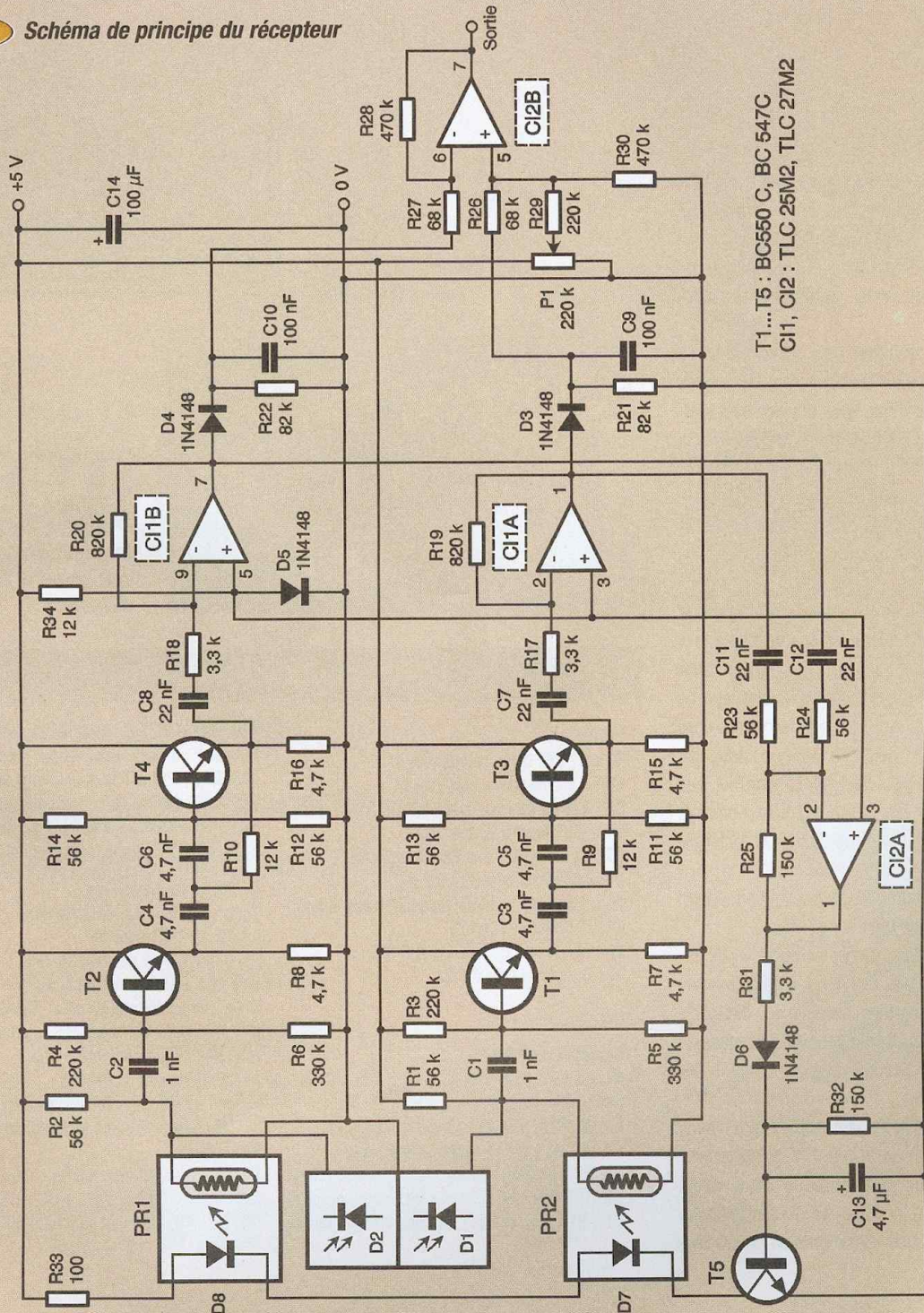
assurée par une diode identique, ce qui assure une compensation automatique de la température. On élimine pratiquement le seuil de détection de la diode. Après filtrage, la tension part d'un côté vers le sommateur C12A et le différentiateur C12B. La somme va être

détectée par la diode D6 et la jonction base-émetteur de T5 qui alimentera à son tour les deux diodes électroluminescentes D7 et D8. Ces diodes sont installées dans un photo coupleur à photorésistance.

En présence d'un signal de grande amplitude, les diodes vont s'allumer et les photo résistances verront leur résistance diminuer ce qui aura pour effet de shunter les photodiodes. Leur signal de sortie diminuera. Cette technique d'action dès l'entrée permet de conserver une bonne linéarité du fonctionne-



## 8 Schéma de principe du récepteur



ment. L'étage différentiateur délivre une tension variable dont on peut faire varier la valeur moyenne par le potentiomètre P1.

## Réalisation

L'émetteur ne pose pas de problème, nous l'avons même réalisé sur carte pastillée reprenant approximativement la disposition des composants du circuit imprimé des figures 9 et 11.

La polarité des diodes est donnée par le fil le plus long qui correspond à l'anode du composant, anode recevant le courant du transistor de commande T1.

## Récepteur

Le récepteur est plus complexe à réaliser compte tenu d'une densité relativement importante des composants. Vous devrez réaliser les deux photocoupleurs qui devront idéalement être aussi identiques que possible.

L'idéal serait d'utiliser une photorésistance double, composant particulièrement difficile à trouver aujourd'hui. Vous pourrez éventuellement changer la position relative de la diode par rapport à la cellule pour augmenter ou réduire la quantité de lumière qui frappe la cellule. Une réduction de l'intensité lumineuse entraîne une augmentation de la valeur de la résistance.

Les deux photocoupleurs sont alignés et n'ont pas besoin d'être isolés l'un de l'autre. Nous les avons logés dans la gaine d'un câble fendue dans le sens de la longueur. Le fond du circuit imprimé est recouvert d'un adhésif noir ou peint de cette couleur. On évite ainsi l'entrée de lumière externe. Les extrémités du tube pourront être obturées (résine époxy, colle thermique et peinture noire).

Les photos illustrent cette conception simple et à la portée de tous.

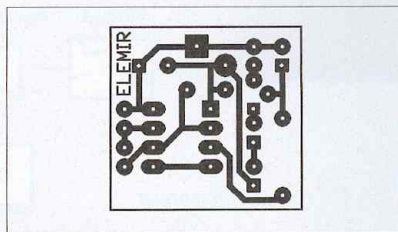
Certains composants sont montés verticalement, nous avons installé des pastilles carrées pour distinguer la cathode des diodes, le pôle positif des condensateurs chimiques, le collecteur des transistors et la borne 1 des circuits intégrés.

Le potentiomètre P1 sera ajusté pour avoir une tension de sortie de 2 V, cette tension variera de part et d'autre de cette valeur lorsque le capteur changera d'orientation par rapport à la balise. Cette tension pourra être

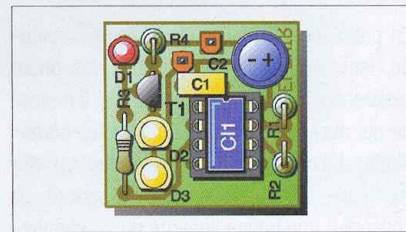
envoyée dans un micro contrôleur qui déterminera l'action à entreprendre pour que le mobile se déplace. Si on a besoin d'une information concernant la fréquence de la balise,

on la recueillera sur la sortie de CI2A, par exemple sur la résistance R25.

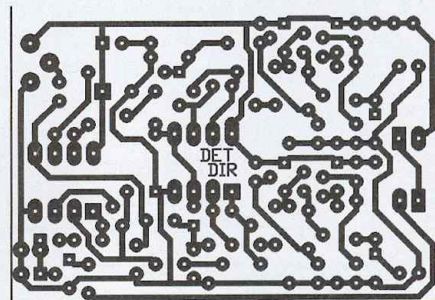
E. LEMERY



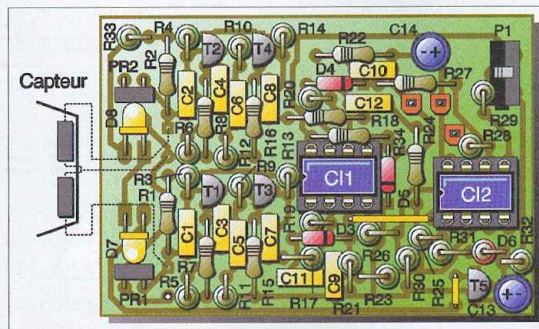
**9** Circuit imprimé de la balise



**11** Implantation des composants de la balise



**10** Circuit imprimé du détecteur



**12** Implantation des composants du détecteur

## Nomenclature

### Émetteur

R1 : 8,2 kΩ (gris, rouge, rouge)  
 R2 : 1 kΩ (marron, noir, rouge)  
 R3 : 470 Ω (jaune, violet, marron)  
 R4 : 10 Ω (marron, noir, noir)  
 C1 : 22 nF, MKT 5 mm (ou céramique)  
 C1 : 100 µF chimique radial 5 V  
 D1 : diode électroluminescente rouge, 3 mm  
 D2, D3 : diode IR LD 271  
 CI1 : Circuit intégré NE ou TLC 555  
 T1 : transistor PNP BC328

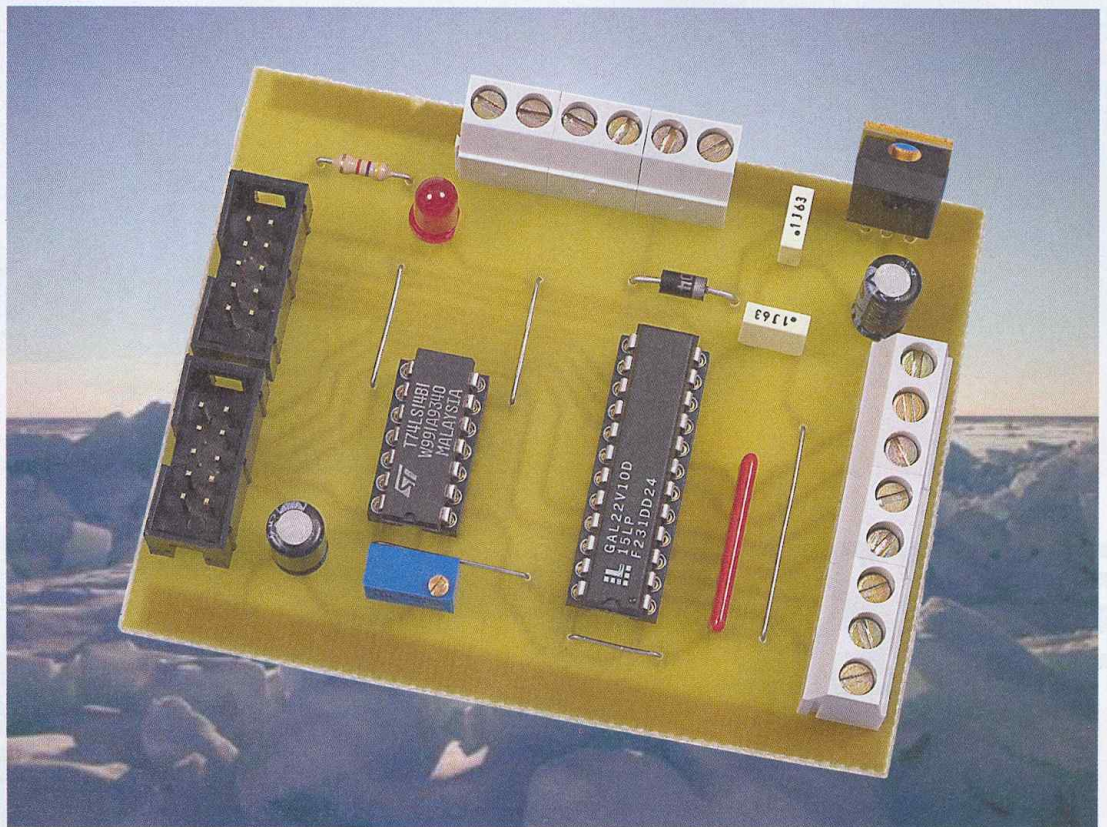
### Récepteur

Résistance 1/4 W, 5%  
 R1, R2, R11, R12, R13, R14, R23, R24 : 56 kΩ (vert, bleu, orange)  
 R3, R4, R29 : 220 kΩ (rouge, rouge, jaune)  
 R5, R6 : 330 kΩ (orange, orange, jaune)  
 R7, R8, R15, R16 : 4,7 kΩ (jaune, violet, rouge)  
 R9, R10, R34 : 12 kΩ (marron, rouge, orange)  
 R17, R18, R31 : 3,3 kΩ (orange, orange, rouge)

R19, R20 : 820 kΩ (gris, rouge, jaune)  
 R21, R22 : 82 kΩ (gris, rouge, orange)  
 R25, R32 : 150 kΩ (marron, vert, jaune)  
 R26, R27 : 68 kΩ (bleu, gris, orange)  
 R28, R30 : 470 kΩ (jaune, violet, jaune)  
 R33 : 100 Ω (marron, noir, marron)  
 C1, C2 : 1 nF céramique  
 C3, C4, C5, C6 : 4,7 nF céramique  
 C7, C8, C11, C12 : 22 nF  
 C9, C10 : 100 nF céramique  
 C13 : 4,7 µF, tantale goutte, 6,3 V  
 C14 : 100 µF chimique radial, 6,3 V  
 P1 : Potentiomètre ajustable vertical 220 kΩ  
 PR1, PR2 : Photorésistance 5 mm  
 D1, D2 : Photodiode BP 104  
 D3, D4, D5, D6 : Diode silicium 1N4148  
 D7, D8 : Diode électroluminescente 3 mm rouge  
 T1, T2, T3, T4, T5 : Transistor NPN/BC550 C, BC239 C  
 CI1, CI2 : TLC 25M2, TLC 27M2, LM 358

# Commande de moteurs par GAL

*Nombre de montages pilotant des moteurs sont proposés avec un traitement des informations confié à un microcontrôleur. Cependant, le choix de ce composant n'est pas toujours justifié, notamment en terme de coût, surtout si le fonctionnement s'avère simple. Nous vous proposons de commander, à partir de quatre contacts nommés CCP, CCN, PPP et PPN, un moteur à courant continu (CC) et un moteur pas à pas (PP).*



Deux contacts par moteur permettent de choisir le sens de rotation (Négatif ou Positif). Un GAL 22V10 se charge de la gestion des données. Le moteur à courant continu est associé à une détection de surintensité SI. Cette détection s'avère très utile pour arrêter le moteur en cas de blocage de son arbre. Deux contacts supplémentaires SH et SB permettront, par exemple, de servir de butée haute et basse et donc d'arrêter aussi le moteur à courant continu. Il est évident que des aménagements du programme proposé seront nécessaires à votre propre utilisation...

## Description du fonctionnement

Le schéma structurel de la carte de gestion des données est représenté sur la **figure 1**.

Le bornier JP4 sert à alimenter la carte en +12 V. Les borniers JP5 et JP6 servent de relais afin d'alimenter les cartes moteurs.

La diode D1 protège des inversions de polarité.

L'alimentation +5 V des circuits intégrés est assurée par le régulateur U3.

La mise sous tension est indiquée par la diode électroluminescente LED1.

Le circuit U2A et les composants C1 et POT1 créent un signal d'horloge nécessaire à la fabrication des quatre signaux de commande du moteur pas à pas (PH1 à PH4).

La vitesse de ce moteur pourra alors être ajustée à l'aide du potentiomètre POT1.

Le réseau de résistances R1 impose que tous les contacts nécessaires à la commande des moteurs s'établissent avec la masse puisque celui-ci a son point commun connecté au +5 V.

## Schémas

Le schéma structurel de la carte d'interface du moteur à courant continu est représenté sur la **figure 2**.

Le circuit U4, un L298 est constitué d'un double pont en H que nous avons mis en parallèle afin de pouvoir disposer du courant

maximum, soit 4A. Son fonctionnement est donné dans le tableau ci-dessous :

Les diodes D2 à D5 sont des diodes de roue libre. L'ensemble des composants C7 à C9, R3 à R7, POT2 et U5 assure la détection de la surintensité. C7, C8, R6 et R7 filtrent et convertissent le courant moteur en tension.

R5 et C9 évitent de détecter la surintensité au

démarrage du moteur. R3 et POT2 règlent le seuil de déclenchement du comparateur à collecteur ouvert U5 associé à R4.

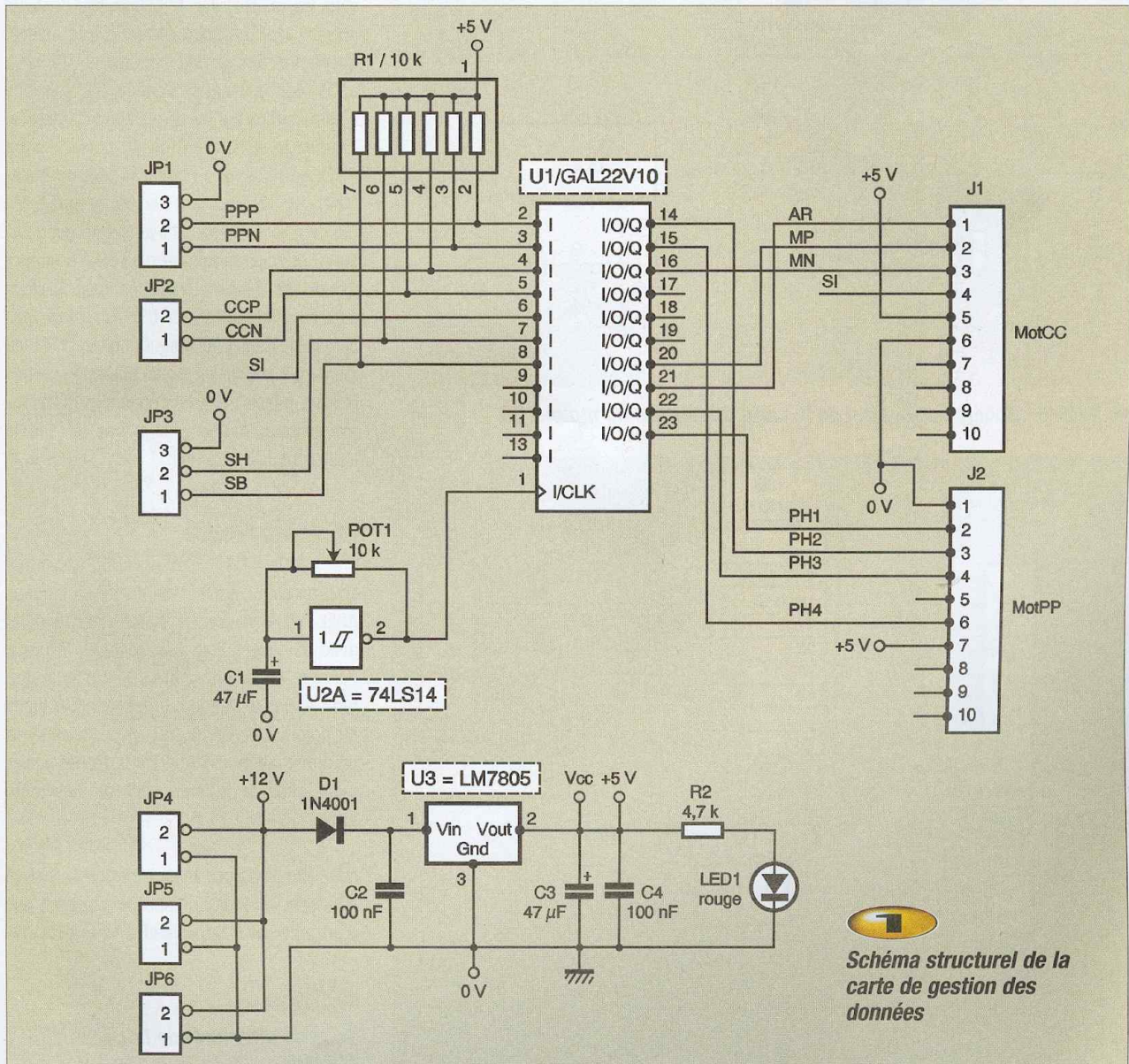
Ce seuil est à régler en fonction du moteur utilisé, la valeur de la résistance R3 pourra donc être modifiée si nécessaire. Nous avons utilisé un moto réducteur 12 V de 4 W.

Le schéma structurel de la carte d'interface du moteur pas à pas de type unipolaire 7,5 ° est représenté sur la **figure 3**.

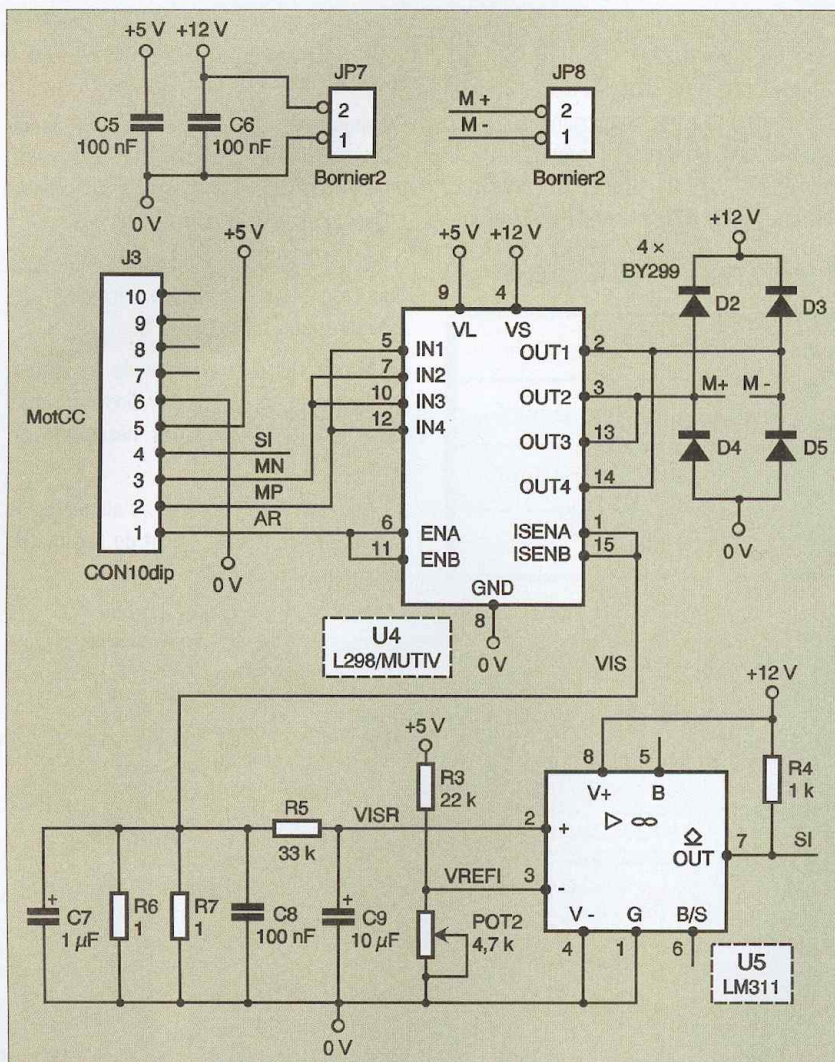
Le circuit U6 est constitué de transistors Darlington protégés par des diodes de roue libre qu'il n'est donc pas nécessaire de rajouter.

Le régulateur U7 permet d'alimenter le moteur en +5 V séparément de la carte de gestion des données.

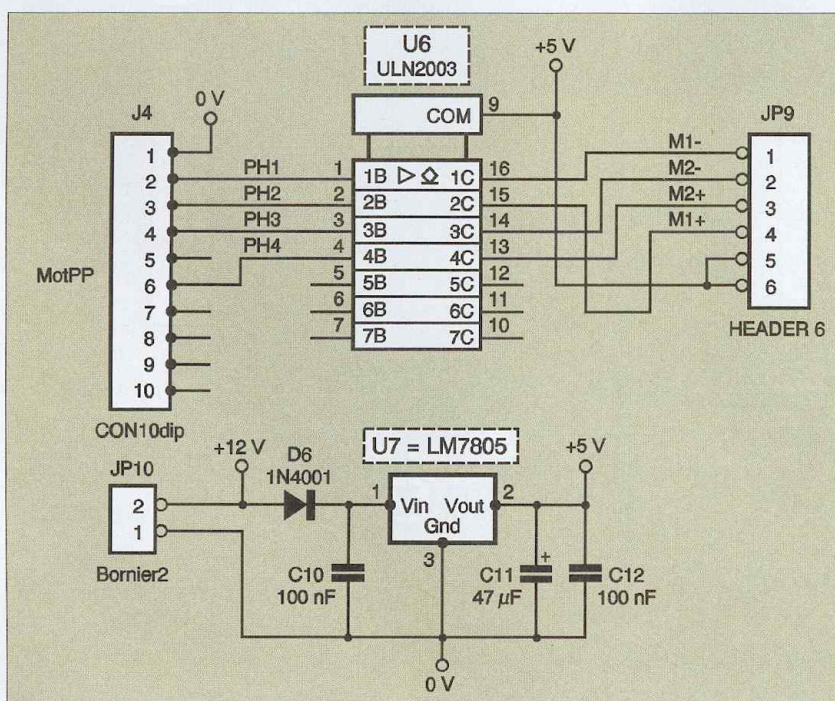
Entrées		Fonction
AR = "1"	MP = "1" ; MN = "0"	Translation sens positif
	MP = "0" ; MN = "1"	Translation sens négatif
AR = "0"	MP = MN = "X"	Arrêt roue libre



**Schéma structurel de la carte de gestion des données**



**2** Schéma structurel de la carte d'interface du moteur CC



## La réalisation

Les schémas d'implantation de nos trois cartes sont donnés figures 5, 7 et 9, les typons sont donnés figures 4, 6 et 8. Les montages sont réalisés sur des circuits imprimés simple face et comportent 11 straps en tout qu'il faudra câbler en premier. Montez ensuite les supports de circuits intégrés puis les composants passifs et finissez par les connecteurs. Attention au montage des régulateurs et du circuit L298 qu'il faudra équiper, si nécessaire, de dissipateurs. Connectez les cartes entre elles à l'aide de nappes dix points pour J1, J3 et J2, J4 puis des câbles multibrins pour les tensions d'alimentation JP5 vers JP7 et JP6 vers JP10. Placez les quatre contacts de commande sur JP1 et JP2 et les deux contacts de butées (si vous les utilisez) sur JP3. Tous ces contacts doivent avoir un point commun à la masse. Connectez JP4 à une batterie de 12 V, la LED1 doit s'allumer. Vérifiez les tensions d'alimentation sur les divers circuits intégrés. Tournez la vis de réglage du potentiomètre POT2 de manière à obtenir la tension maximum sur la broche 3 de U5 (à peu près 800 mV). Comme nous l'avons dit précédemment, ce réglage permet d'arrêter le moteur CC en cas de surintensité et donc, si vous êtes au minimum, le moteur ne pourra pas démarrer. Réglez le potentiomètre POT1 de manière à obtenir un signal d'horloge de fréquence inférieure à la fréquence de décrochement du moteur pas à pas, si non le moteur ne tournera pas.

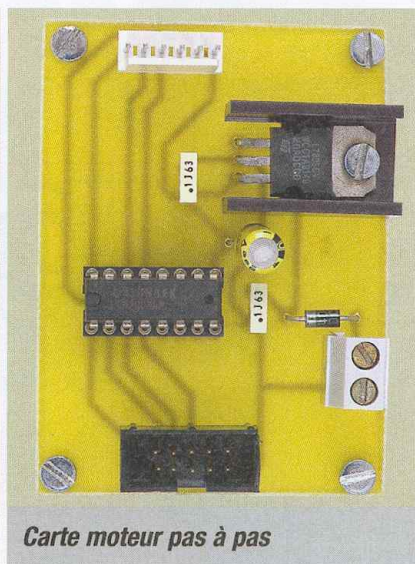
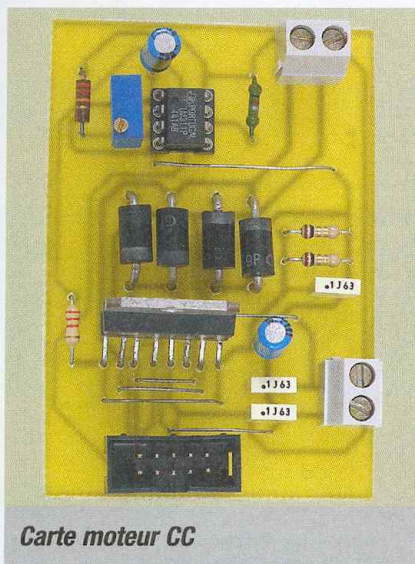
## Le programme

Le programme suivant est donné en langage ABEL. Vous pourrez le saisir et le modifier avec ISPleaver (Voir Électronique Pratique numéro 288 de novembre 2004) puis transférer le fichier Jedec correspondant à l'aide du programme GALBLAST et de la carte de programmation de GAL 22V10 décrite également dans le numéro 288 de novembre 2004.

**N.B. :** Pour utiliser le logiciel Galblast sous Windows XP lorsque vous ne disposez pas d'un port parallèle, il faudra vous procurer



**3** Schéma structurel de la carte d'interface du moteur pas à pas



Carte moteur CC

Carte moteur pas à pas

module cmdmot;  
titre 'commande de deux moteurs';

declarations

"entrées

H,PPP,PPN pin 1,2,3;  
SI,CCP,CCN,SH,SB pin 6,4,5,7,8;

"Sorties

PH1,PH2,PH3,PH4 pin 23,14,22,15 istype 'reg';  
ARPP pin 17 istype 'com';  
MP,MN,ARCC pin 20,16,21 istype 'com';  
IA pin 18 istype 'com'; "test sur-intensité  
sortie = [PH4,PH3,PH2,PH1];  
s0=0; s1=5; s2=9; s3=10; s4=6;

equations

sortie.clk=H; "horloge rotation  
sortie.ar=PPP&PPN; "Reset des bascules rotation  
ARPP=(PPP&!PPN)#(!PPP&PPN); "condition pour tourner (rotation Jd ou Jg)  
MP=!CCP&CCN&SH; "translation vers le haut  
MN=CCP&!CCN&SB; "translation vers le bas  
ARCC=(MP&!IA) # (MN&!IA); "condition pour tourner (translation Jh ou Jb)

" Mémorisation de la surintensité

when (((MP == 1) # (MN == 1)) & (SI == 1)) then IA = 1;  
else when ((MP == 0) & (MN == 0)) then IA = 0;  
else when (((MP == 1) # (MN == 1)) & (SI == 0)) then IA = IA;  
else IA = X.;

"Cycle de rotation de la table (commande moteur PP)

state\_diagram sortie;

state s0: goto s1;  
state s1: if !PPP then s2 else s4;  
state s2: if !PPP then s3 else s1;  
state s3: if !PPP then s4 else s2;  
state s4: if !PPP then s1 else s3;

end cmdmot

## Nomenclature

### Carte de gestion des données

REPÈRE	VALEUR	BOITIER
C1 :	47 µF	Chimique V
C2 :	100 nF	Céramique
C3 :	47 µF	Chimique V
C4 :	100 nF	Céramique
D1 :	1N4001	
J1 :	CON10dip	Droit
J2 :	CON10dip	Droit
JP1 :	Bornier3	
JP2 :	Bornier2	
JP3 :	Bornier3	
JP4 :	Bornier2	
JP5 :	Bornier2	
JP6 :	Bornier2	
LED1 :	LED	Rouge
POT1 :	10 kΩ	Multi tours V
R1 :	10 kΩ	SIP7
R2 :	4.7 kΩ	1/4 W
U1 :	GAL22V10	Dip24 + support
U2 :	74LS14	Dip14 +support
U3 :	LM7805	T092

### Carte d'interface du moteur à courant continu

REPÈRE	VALEUR	BOITIER
C5 :	100 nF	Céramique
C6 :	100 nF	Céramique
C7 :	1 µF	Chimique V
C8 :	100 nF	Céramique
C9 :	10 µF	Chimique V
D2 :	BY299	
D3 :	BY299	
D4 :	BY299	
D5 :	BY299	
J3 :	CON10dip	Droit
JP7 :	Bornier2	
JP8 :	Bornier2	
POT2 :	4,7 kΩ	Multi tours V
R3 :	22 kΩ	
R4 :	1 kΩ	
R5 :	33 kΩ	
R6 :	1 Ω	
R7 :	1 Ω	
U4 :	L298	Multiwatt 15
U5 :	LM311	Dip8 + support

### Carte d'interface du moteur pas à pas

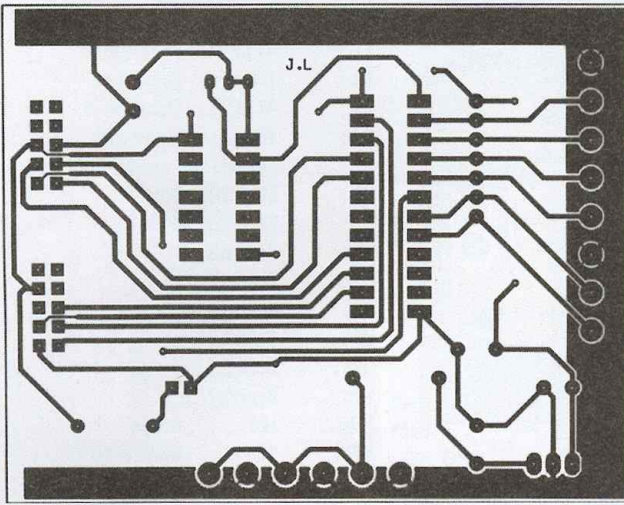
REPÈRE	VALEUR	BOITIER
C10 :	100 nF	Céramique
C11 :	47 µF	Chimique V
C12 :	100 nF	Céramique
D6 :	1N4001	
J4 :	CON10dip	Droit
JP9 :	HEADER 6	
JP10 :	Bornier2	
U6 :	ULN2003	Dip16 + support
U7 :	LM7805	T092

une carte d'extension additionnelle qui se connecte sur le port USB ou PCMCIA. Si vous disposez du port parallèle, il faudra lancer

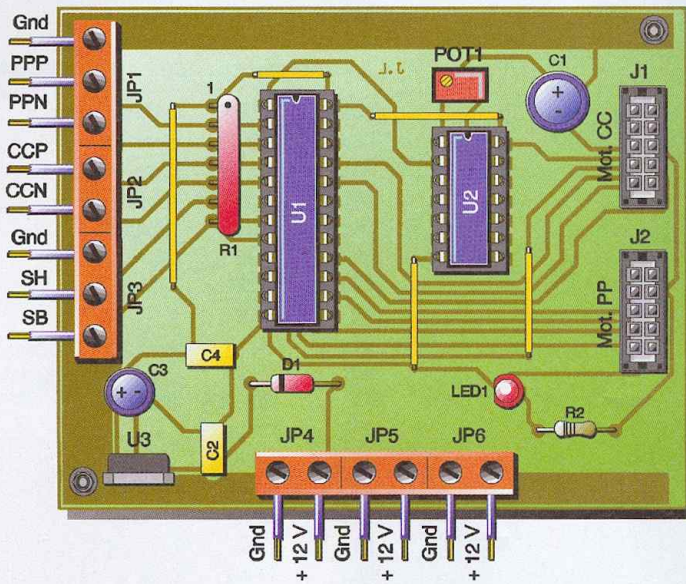
Galblast à partir d'une fenêtre DOS et configurer le port en mode PS/2, bidirectionnel ou AT et non en ECP. Vous devrez alors désins-

taller votre driver d'origine et utiliser un driver du type Direct I/O pour accéder au port.

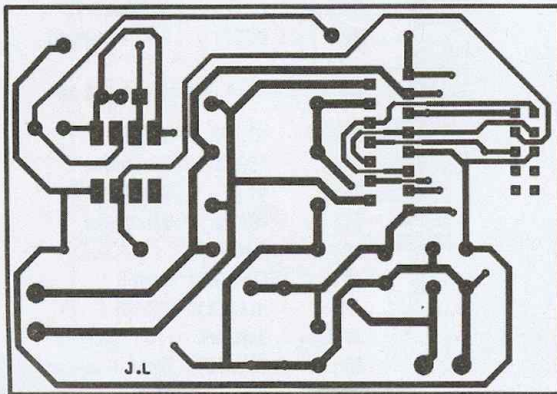
L. JOSSE



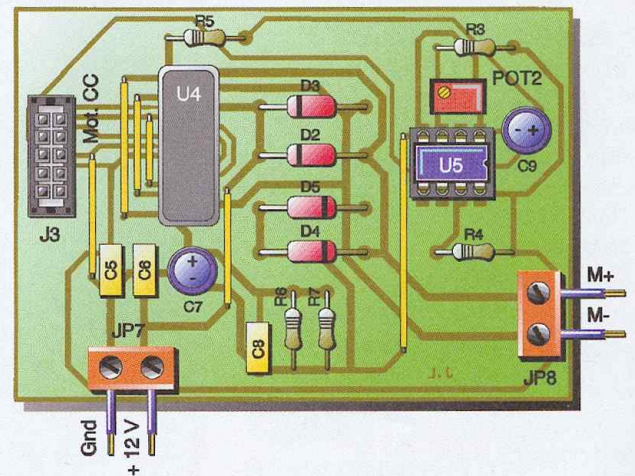
**4** Tracé du circuit imprimé gestion



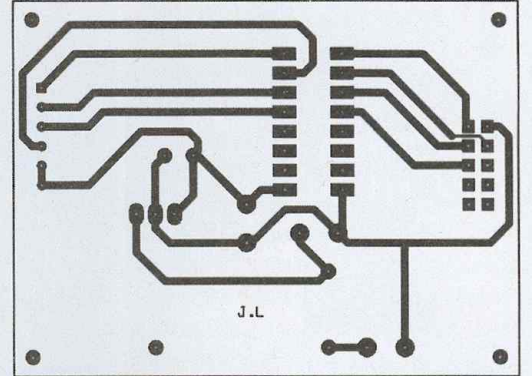
**5** Implantation des éléments



**6** Tracé du circuit imprimé moteur CC



**7** Implantation des éléments



**8** Tracé du circuit imprimé moteur pas à pas

**9** Implantation des éléments