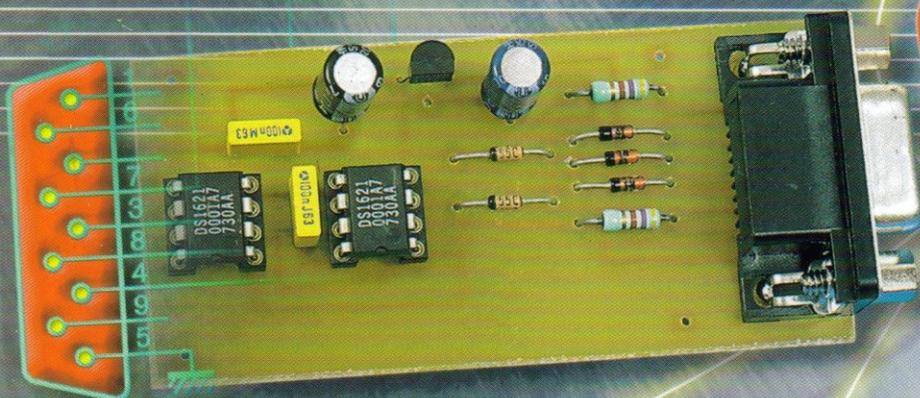


**INTERFACES
ET DÉVELOPPEMENTS
PC**

INTERFACES ET DÉVELOPPEMENTS PC

Réalisez vous-même
15 CARTES

d'applications avec tous
les programmes et les PCB
Compatibles tout PC



**CD ROM
INCLUS**



- **Thermomètre pour Windows®**
- **Jeux sur Minitel®**
- **Interface RS232 pour téléphone portable**
- **Programmateur d'arrosage**
- **Décodeur DTMF**
- **Bus I2C sur le port parallèle**
- **etc.**



T 3271 - 7 H - 35,00 F - RD



SOMMAIRE

ELECTRONIQUE PRATIQUE

H.S. N° 07 - NOVEMBRE 2000
I.S.S.N. 0243 4911

PUBLICATIONS GEORGES VENTILLARD

S.A. au capital de 5 160 000 F
2 à 12, rue Bellevue, 75019 PARIS
Tél. : 01.44.84.84.84 - Fax : 01.44.84.85.45
Internet : <http://www.eprat.com>
Principaux actionnaires :
M. Jean-Pierre VENTILLARD
Mme Paule VENTILLARD

Président du conseil d'administration,
Directeur de la publication : Paule VENTILLARD
Vice-Président : Jean-Pierre VENTILLARD
Assistant de Direction : Georges-Antoine VENTILLARD
Directeur de la rédaction : Bernard FIGHIERA (84.65)
Directeur graphique : Jacques MATON
Maquette : Dominique DUMAS
Couverture : R. MARAI

Avec la participation de : U. Bouteville, J.P. Duval,
A. Garrigou, P. Gueulle, M. Laury, B. Lebrun,
M. Luczak, P. Mayeux, P. Morin, P. Oguic, A. Reboux,
D. Rey, A. Sorokine, Ch. Tavernier.

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

Directeur de la diffusion et promotion :
Bertrand DESROCHE
Responsable ventes :
Bénédictine MOULET Tél. : 01.44.84.84.54
N° vert réservé aux diffuseurs et dépositaires de presse :
0800.06.45.12

PGV - Département Publicité :
2 à 12 rue de Bellevue, 75019 PARIS
Tél. : 01.44.84.84.85 - CCP Paris 3793-60
Directeur commercial : Jean-Pierre REITER (84.87)
Chef de publicité : Pascal DECLERCK (84.92)
E Mail : lehpub@le-hp.com
Assisté de : Karine JEUFFRAULT (84.57)

Abonnement/VPC : Voir nos tarifs en page intérieure.
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.
• Pour tout changement d'adresse, joindre 3, 00 F et la der-
nière bande.

Aucun règlement en timbre poste.
Forfait photocopies par article : 30 F.
Distribué par : TRANSPORTS PRESSE
Abonnements USA - Canada : Pour vous abonner à
Electronique Pratique aux USA ou au Canada, commu-
niquez avec Express Mag par téléphone :
USA : P.O.Box 2769 Plattsburgh, N.Y. 12901-0239
CANADA : 4011boul.Robert, Montréal, Québec, H1Z4H6
Téléphone : 1 800 363-1310 ou (514) 374-9811
Télécopie : (514) 374-9684.
Le tarif d'abonnement annuel (11 numéros) pour les USA
est de 49 \$US et de 68 \$cnd pour le Canada.
Electronique Pratique, ISSN number 0243 4911, is published 11
issues per year by Publications Ventillard at P.O. Box 2769
Plattsburgh, N.Y. 12901-0239 for 49 \$US per year.
POSTMASTER : Send address changes to Electronique Pratique,
c/o Express Mag, P.O. Box 2769, Plattsburgh, N.Y., 12901-0239.



«Ce numéro
a été tiré
à 58 500
exemplaires»

INTERFACES PC

- 05 Edito
- 06 Introduction : Les convertisseurs AN/NA
- 19 Utilisation du CDRom
- 86 Tina Pro
- 88 Kit Velleman K8016
- 94 Afficheur LCD sur port parallèle

Les 15 cartes à réaliser

- 20 Thermomètre pour Windows®
- 26 Interface RS232 pour téléphone portable
- 30 Testeur de télécommande IR
- 34 Implémentation d'un contrôleur de souris par le port série
- 40 Décodeur DTMF
- 44 Jeux sur minitel®
- 50 Contrôleur de moteurs pas à pas sur le port série
- 58 Programmateur d'arrosage
- 62 Interface écran et clavier sur port parallèle
- 68 Girouette électronique
- 74 Capacimètre piloté par liaison RS232
- 80 Bus I2C sur le port parallèle
- 92 Parafoudre

Projet MULTIMEDIA

- 17 Montage de commande de perceuse à PIC 16F84 avec simulation logicielle, uniquement sur le CDRom

Infos OPPORTUNITÉS

16

Windows et Minitel sont des marques déposées



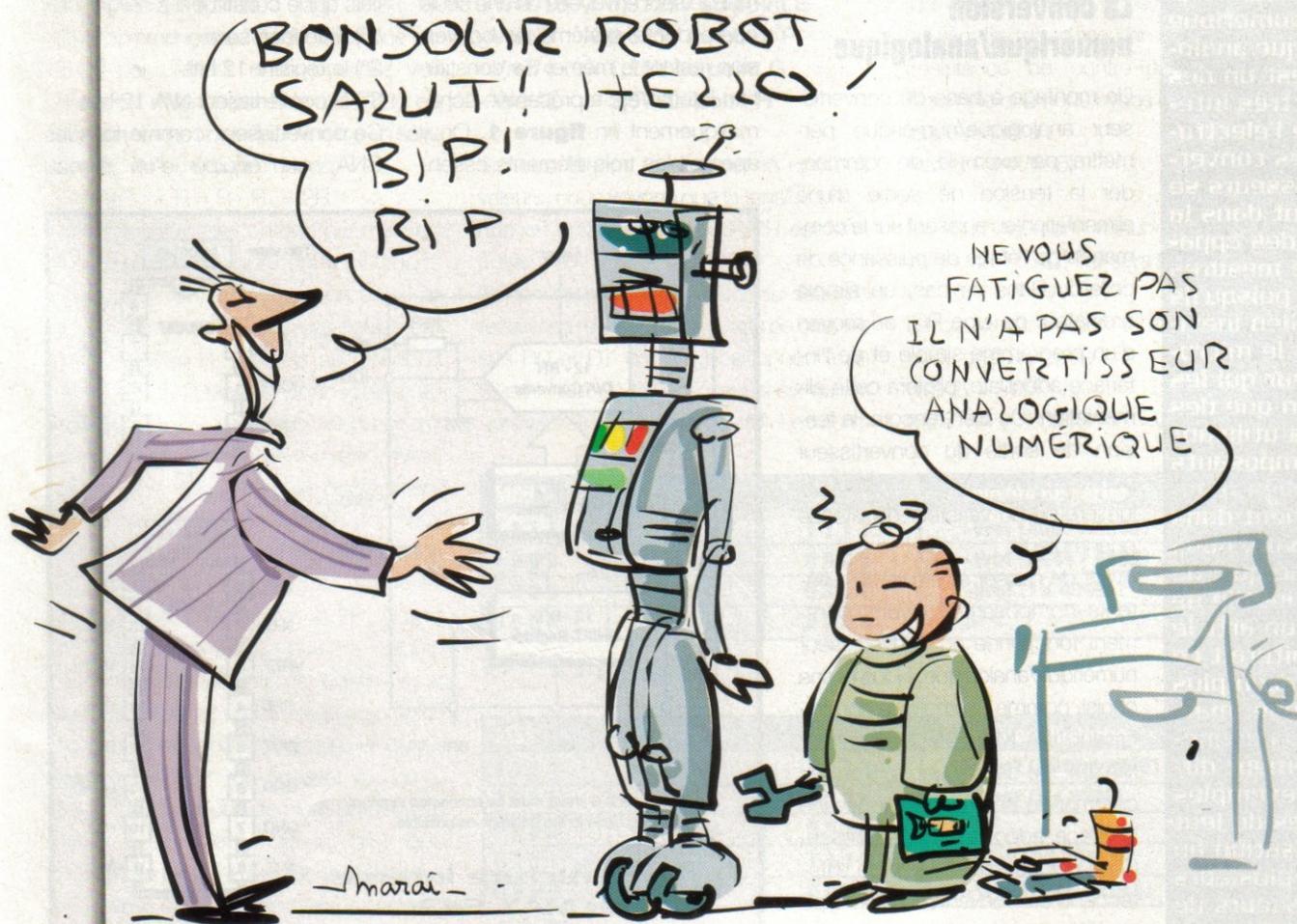
La fin de l'année 2000 approche et, déjà, les nouveautés informatiques apparaissent dans les boutiques spécialisées à des tarifs toujours plus attrayants. Les Noël se suivent et se ressemblent au moins sur un point : Le PC flambant neuf que vous avez acheté il y a un ou deux ans est déjà dépassé et les promotions de cette nouvelle fin d'année vous donnent des idées de cadeaux. Alors pourquoi ne pas faire d'une pierre deux coups ?

En effet, votre matériel informatique obsolète peut encore vous rendre de bons et loyaux services. Les nombreux montages qui vous sont proposés dans INTERFACES PC sont conçus pour fonctionner sur la plupart des ordinateurs. Pas besoin d'une machine de "course" pour exploiter, par exemple, l'interface I2C pour port parallèle qui vous est présenté dans ce numéro. Il en va de même pour le décodeur DTMF ou le capacimètre à liaison RS232.

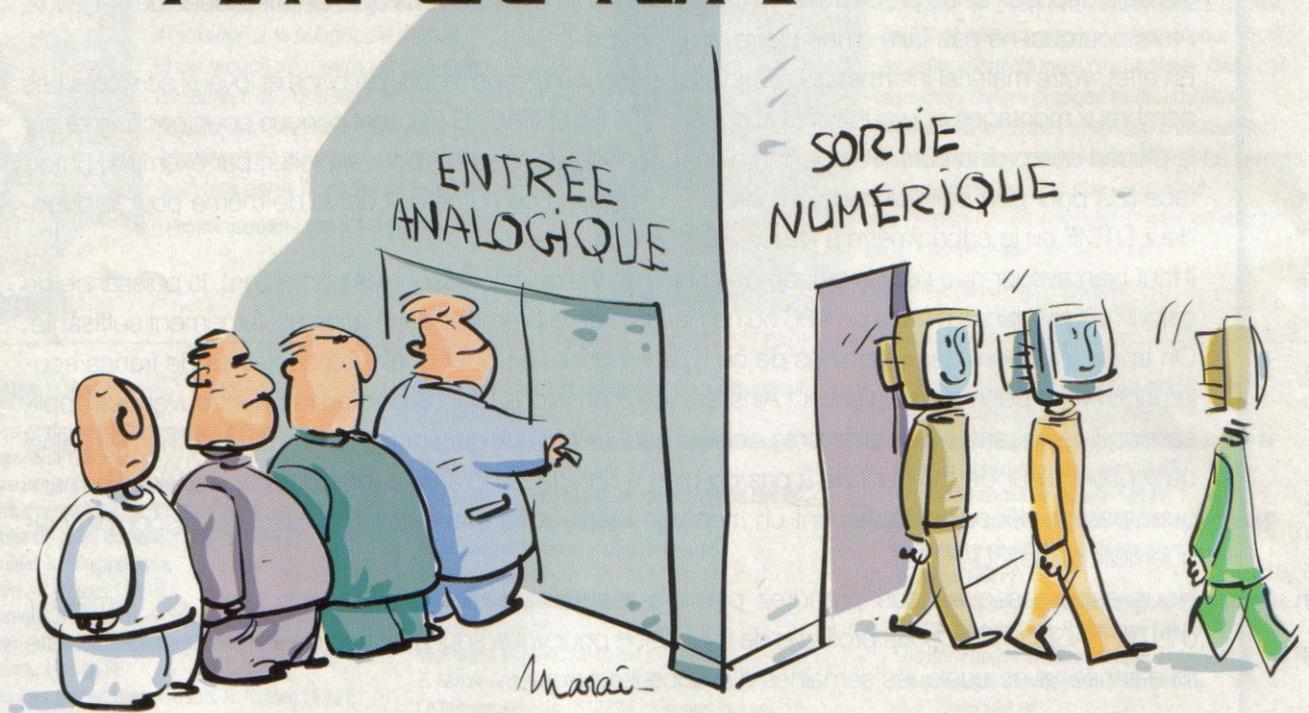
Il faut bien avouer que pour la plupart des applications qui intéressent les particuliers, la puissance de calcul des machines à base de 486 ou de Pentium de la première génération est largement suffisante. On trouve d'ailleurs des machines de ce type en occasion pour quelques centaines de francs seulement. Alors, pourquoi s'en priver ! Ainsi équipé, nous vous faisons confiance pour trouver des applications intéressantes aux différents convertisseurs AN/NA qui sont publiés dans ce numéro ainsi qu'au contrôleur de moteur pas à pas ou bien à l'interface écran/clavier sur port parallèle. Ne manquez pas de découvrir également un montage inédit et sa présentation multimédia disponible uniquement sur le CD-ROM.

Nous espérons que vous prendrez plaisir à réaliser les montages décrits dans ce numéro 7 d'INTERFACES PC et nous profitons de l'occasion pour vous souhaiter nos meilleurs vœux pour cette fin d'année, avec quelques semaines d'avance il est vrai.

La rédaction



Les convertisseurs A/N et N/A



La conversion analogique/numérique et numérique/analogique est un des domaines très intéressants de l'électronique. Les convertisseurs se rencontrent dans la plupart des appareils de mesures modernes puisqu'ils sont le lien inévitable avec le microprocesseur qui les pilote. Bien que des montages utilisant ces composants soient publiés régulièrement dans notre revue, nous avons pensé qu'il ne serait pas inutile de proposer un article qui traiterait ce sujet de façon plus théorique, mais simple. Nous donnerons également différents exemples pratiques de leur utilisation au moyen de plusieurs convertisseurs de types différents

La conversion numérique/analogique

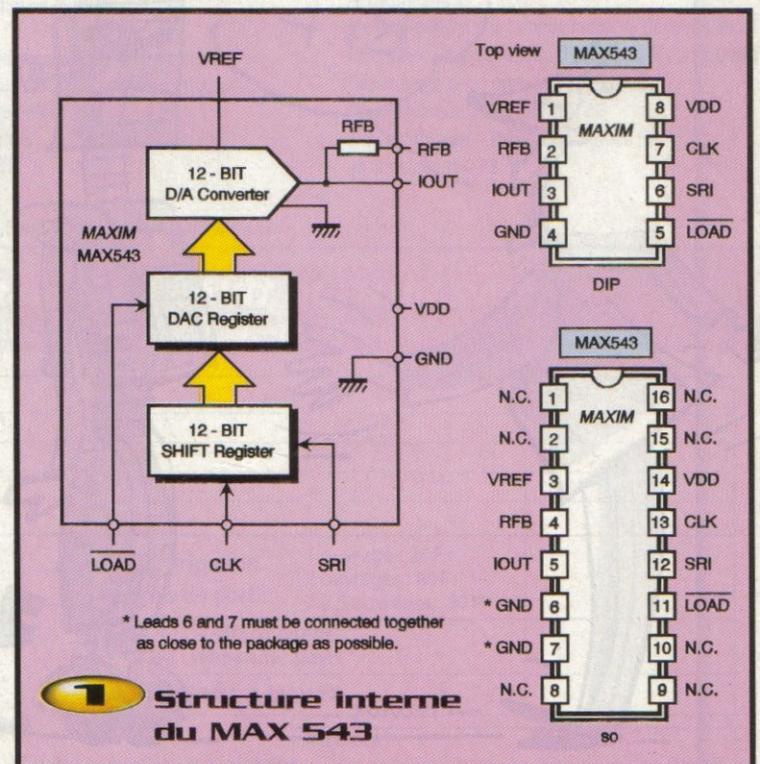
Un montage à base de convertisseur analogique/numérique permettra, par exemple, de commander la tension de sortie d'une alimentation en agissant sur la commande de l'étage de puissance de celle-ci. Dans ce cas, un simple ordinateur de type PC, au moyen d'un programme simple et de l'interface adéquate, pilotera cette alimentation. Ou bien, encore, la tension de sortie du convertisseur pourra commander un variateur de lumière ou un variateur de vitesse pour moteur.

Avant de passer à l'étude de différents composants, voyons comment fonctionne un convertisseur numérique/analogique. Nous avons choisi, comme exemple, le convertisseur N/A MAX543 fabriqué par MAXIM. Ce dernier est de type par commande série et possède une interface adéquate haute vitesse. C'est uniquement cela qui le différencie d'un convertisseur de type parallèle où les données de com-

mande sont envoyées en une seule opération, le système de conversion restant le même. Sa constitution interne est représentée schématiquement en **figure 1**. On y aperçoit les trois éléments essen-

tiels qui le constitue :

- 1°) l'interface série
 - 2°) le registre 12 bits
 - 3°) le convertisseur N/A 12 bits
- Ce convertisseur, comme tous les CNA, est équipé d'un réseau



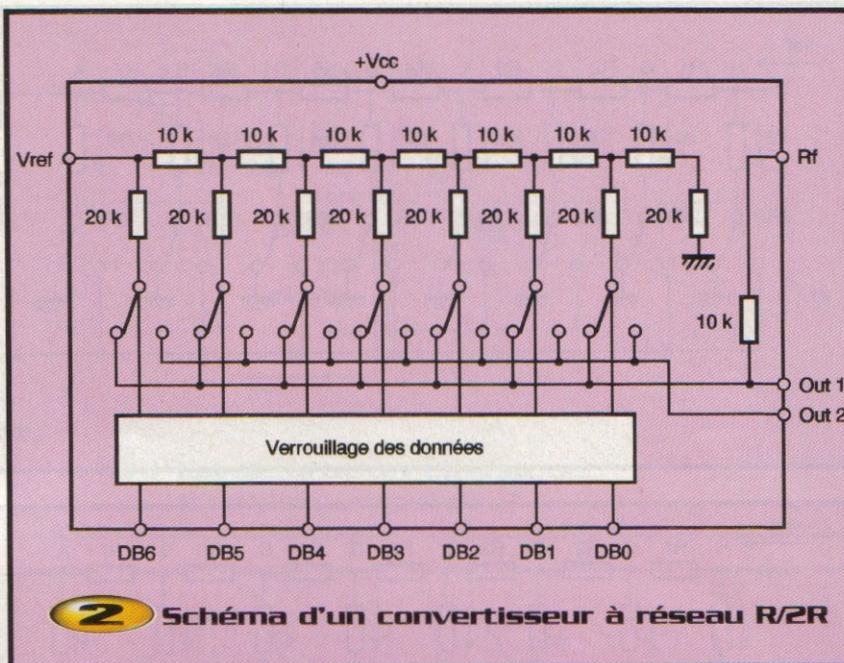
nommé R-2R dont le schéma est donné en **figure 2**. C'est ce type de réseau qui est l'élément essentiel d'un convertisseur. Il est représenté de façon simplifiée en **figure 3** (8 bits). Dans cette configuration, la valeur des résistances est secondaire. Ce qui est fondamental, c'est que les résistances 2R soient égales à 2 x R et, donc, posséder un rapport de deux. C'est cette condition qui détermine la précision du composant. C'est pour cette raison que les résistances constituant ce réseau sont ajustées au laser.

Le fonctionnement en est très simple. Huit commutateurs permettent de connecter les résistances 2R soit à la masse (bit à 0), soit à l'entrée d'un amplificateur opérationnel (bit à 1). Il est indispensable d'alimenter le réseau résistif par une tension très stable. Cette tension appelée Vref sera une tension fournie par une diode zéner de référence.

Voyons le schéma de la figure 3. Les 8 bits des entrées de commande étant positionnés à un niveau bas (0), les résistances 2R sont toutes connectées à la masse. L'une des résistances du réseau est définitivement reliée à la masse. Il s'agit de 2R₇. Si nous commençons l'étude électrique du circuit par celle-ci, nous pouvons voir qu'elle est mise en parallèle avec 2R₇, ce qui nous donne une valeur R, mise en série avec R₇. Ainsi, nous obtenons :

$$(2R_7 // 2R_7) + R = R + R_7 = 2R$$

Cet élément résistif obtenu par mise en parallèle/série de plusieurs résistances est ensuite mis à nouveau en parallèle avec la résistance 2R₆, mais également en série avec R₆. Et cela se reproduit jusqu'aux résistances connectées au point H. C'est à cette broche du circuit intégré que la tension de référence (Vref) est injectée. En ce point, le courant rencontre un diviseur de tension par deux et la résistance 2R₀ est donc traversée par un courant égal à Iref/2. La résistance R₁ l'est également et 2R₁ draine un courant égal à Iref/4 puisque celui-ci rencontre à nouveau un pont diviseur par deux. Cela se reproduit jusqu'au point B où Iref ne possède plus qu'une valeur égale à 1/128^{ème} du courant d'origine. Ce dernier subit une dernière division par deux et sa valeur n'atteint plus que 1/256^{ème} lorsqu'il traverse la résistance 2R₇. Ainsi, si Vref a pour valeur une tension de +5V, nous obtiendrons les



valeurs suivantes selon l'un des huit commutateurs activés :

- Vref/256 = 0,0195312V au point A
- Vref/128 = 0,0390625V au point B
- Vref/64 = 0,078125V au point C
- Vref/32 = 0,15625V au point D
- Vref/16 = 0,3125V au point E
- Vref/8 = 0,625V au point F
- Vref/4 = 1,25V au point G
- Vref/2 = 2,5V au point H

Si nous faisons la somme de ces huit valeurs, nous voyons que la tension obtenue en sortie du réseau R/2R est bien égale à 5V.

Reportons-nous maintenant au schéma représenté en **figure 4**. Dans ce cas, les bits D0 et D7 ont été positionnés à un niveau haut (1) tandis que les six autres sont laissés à un niveau bas (0). Les com-

mutateurs SW₀ et SW₇ ont donc commuté les résistances 2R₀ et 2R₇ à l'entrée de l'amplificateur opérationnel. Un courant égal à Iref/2 + Iref/256 est alors présent à l'entrée de cet AOP. Celui-ci, monté en amplificateur inverseur, voit son entrée - (moins) portée au potentiel de la masse (qui est une masse virtuelle). La totalité du courant transite donc par la résistance R (résistance de contre-réaction). Dans l'exemple que nous avons décrit, ce courant est égal à :

$$(Iref/2) + (Iref/256)$$

Donc, si Vref a une valeur de 5V et R une valeur de 10 kΩ, Iref sera égale à :

$$5 / 10 = 0,5 \text{ mA}$$

Le courant traversant la résistance aura donc une valeur de :

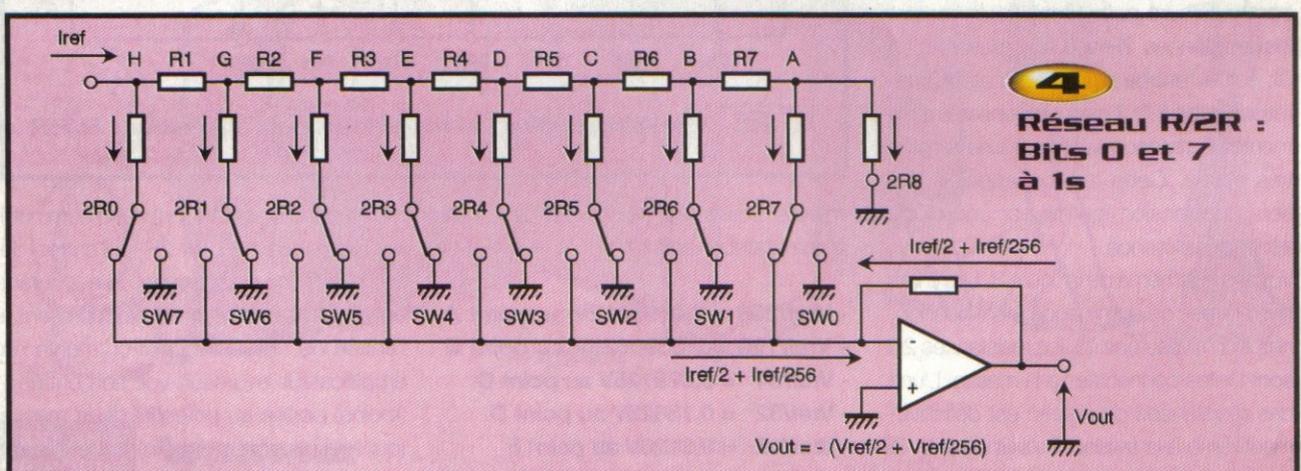
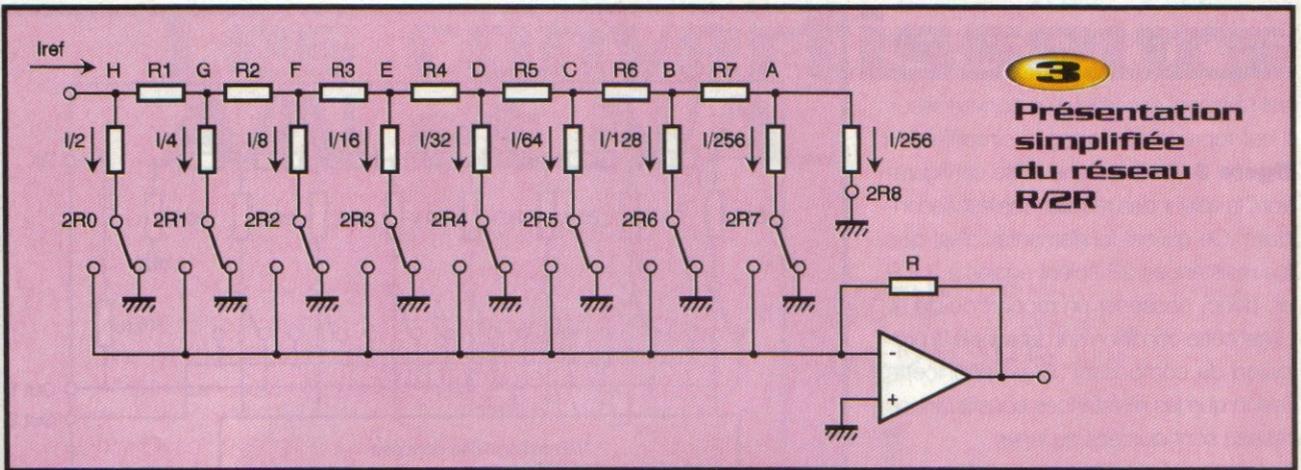
$$(Iref/2) + (Iref/256) = 0,25 \text{ mA} + 0,0019 \text{ mA} = 0,2519 \text{ mA}$$

MSB	LSB		tension de sortie
1 1 1 1	1 1 1 1	1 1 1 1	-Vref (4095 / 4096)
1 0 0 0	0 0 0 0	0 0 0 0	-Vref (2048 / 4096) = -Vref / 2
0 0 0 0	0 0 0 0	0 0 0 1	-Vref (1 / 4096)
0 0 0 0	0 0 0 0	0 0 0 0	0

Tableau 1

MSB	LSB		tension de sortie
1 1 1 1	1 1 1 1	1 1 1 1	+Vref (2047 / 2048)
1 0 0 0	0 0 0 0	0 0 0 0	+Vref (1 / 2048)
1 0 0 0	0 0 0 0	0 0 0 0	0
0 1 1 1	1 1 1 1	1 1 1 1	-Vref (1 / 2048)
0 0 0 0	0 0 0 0	0 0 0 0	-Vref (2048 / 2048)

Tableau 2



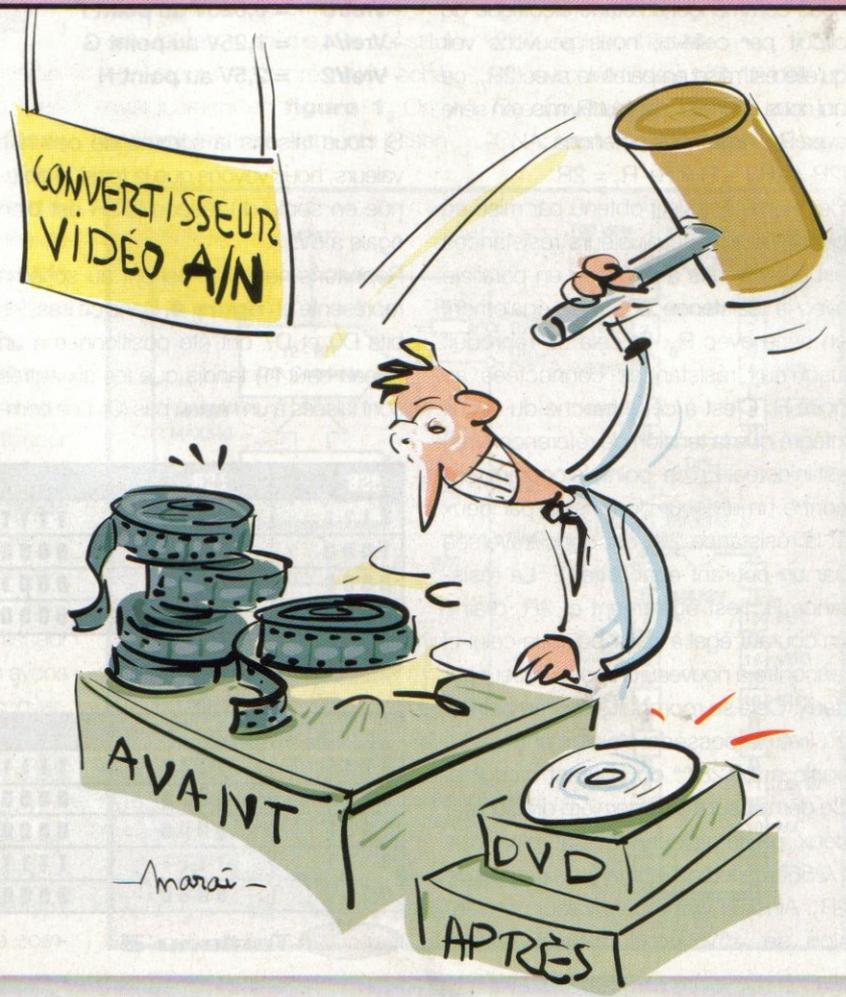
La tension de sortie du convertisseur sera donc de :
 $0,2519 \times 10 = 2,519V$

Application pratique du MAX543

Le schéma représenté en **figure 5** donne la configuration classique du MAX543 afin d'obtenir une tension de sortie unipolaire. Il est à remarquer que l'amplificateur opérationnel de sortie peut être remplacé par un composant plus courant, par exemple les LF351 ou LF356. La tension V_{ref} sera générée par une diode zéner de référence. La TL431CLP, de faible coût, fera parfaitement l'affaire.

Trois signaux sont nécessaires pour la commande de chargement des données dans le convertisseur :

- 1°) SRI qui est l'entrée des données sur 12 bits. Le MSB (Most Significant Bit, bit de poids le plus important) sera chargé en premier,
- 2°) CLOCK INPUT qui est l'entrée du signal d'horloge. Un coup d'horloge est



nécessaire à la mémorisation de chaque bit,

3°) LOAD/ qui est le signal permettant l'envoi des 12 bits au convertisseur proprement dit.

Le diagramme de la **figure 6** donne la chronologie des différents signaux.

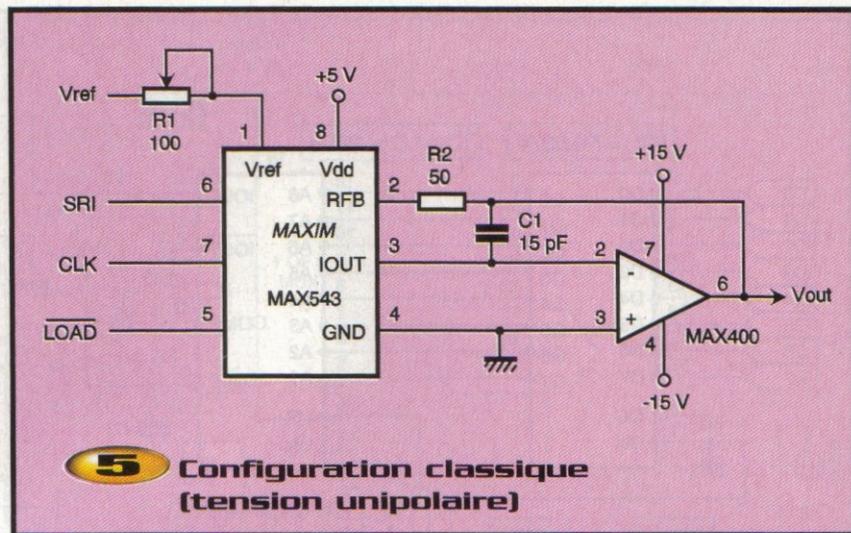
Le tableau donné ci-dessous nous donne la tension de sortie du convertisseur en fonction de la valeur des 12 bits envoyés sur son entrée (**tableau 1**) :

Le schéma de la **figure 7** nous donne la manière de procéder afin d'obtenir en sortie du MAX543 une tension bipolaire. La même tension de référence est nécessaire. la résistance ajustable R_1 permet de régler le 0V lorsque la donnée de 12 bits 1000 0000 0000 est envoyée au convertisseur. Le **tableau 2** donne les tensions de sortie en fonction des données :

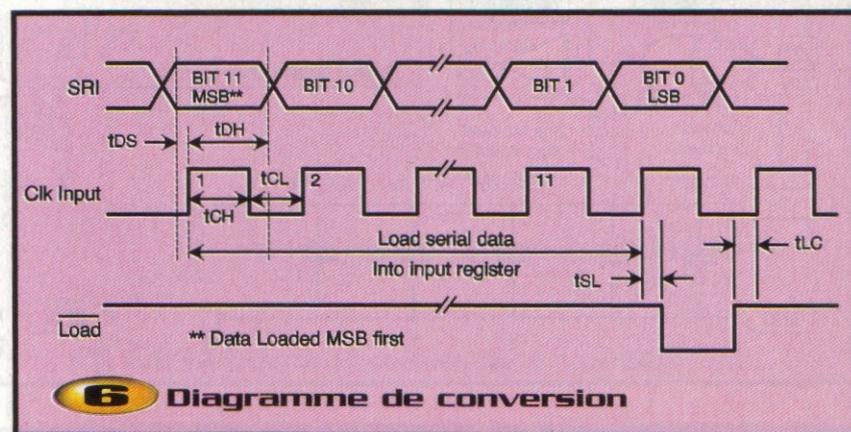
Le dessin donné en **figure 8** donne un autre exemple d'application, mais cette fois en utilisant un convertisseur parallèle, le DAC0808, composant plus courant mais d'une résolution de huit bits. Le principe de fonctionnement est le même que le montage précédent. Il nécessite une tension de référence qui est générée également par une diode zéner de précision alliée à un amplificateur opérationnel. Ce montage est conçu pour fonctionner sur le bus ISA d'un ordinateur de type PC.

La conversion analogique/numérique

La conversion analogique/numérique est essentiellement utilisée pour l'acquisition de données par un microprocesseur (ou ordinateur). Il permet de concevoir des appareils de mesures pilotés par un programme informatique, appareils présentant une précision excellente selon le type de convertisseurs utilisés (8 bits, 10 bits, 12 bits et plus). Ce type de conversion utilise également un réseau R/2R ainsi qu'un convertisseur numérique/analogique et une tension de référence. Le schéma interne de ce type de composant est représenté en **figure 9**. La plupart de ces convertisseurs utilisent une technique appelée conversion par approximations successives. Le principe de fonctionnement est très simple : huit interrupteurs (pour un convertisseur 8 bits) sont actionnés à tour de rôle. La conversion débute par la



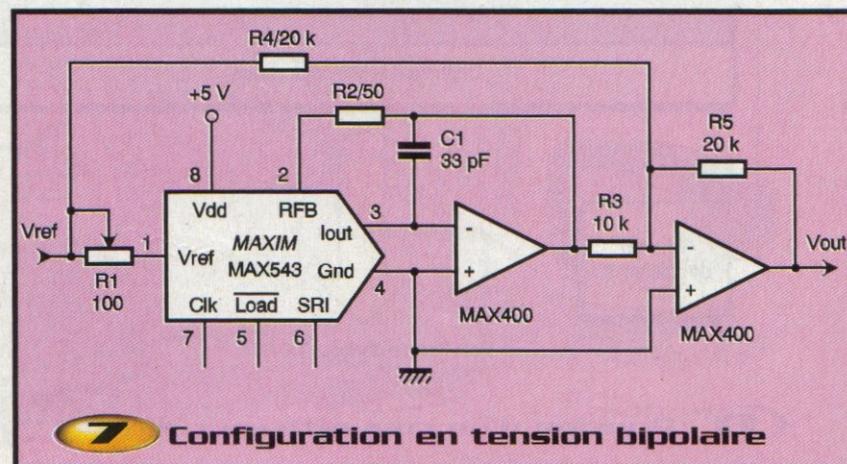
5 Configuration classique (tension unipolaire)



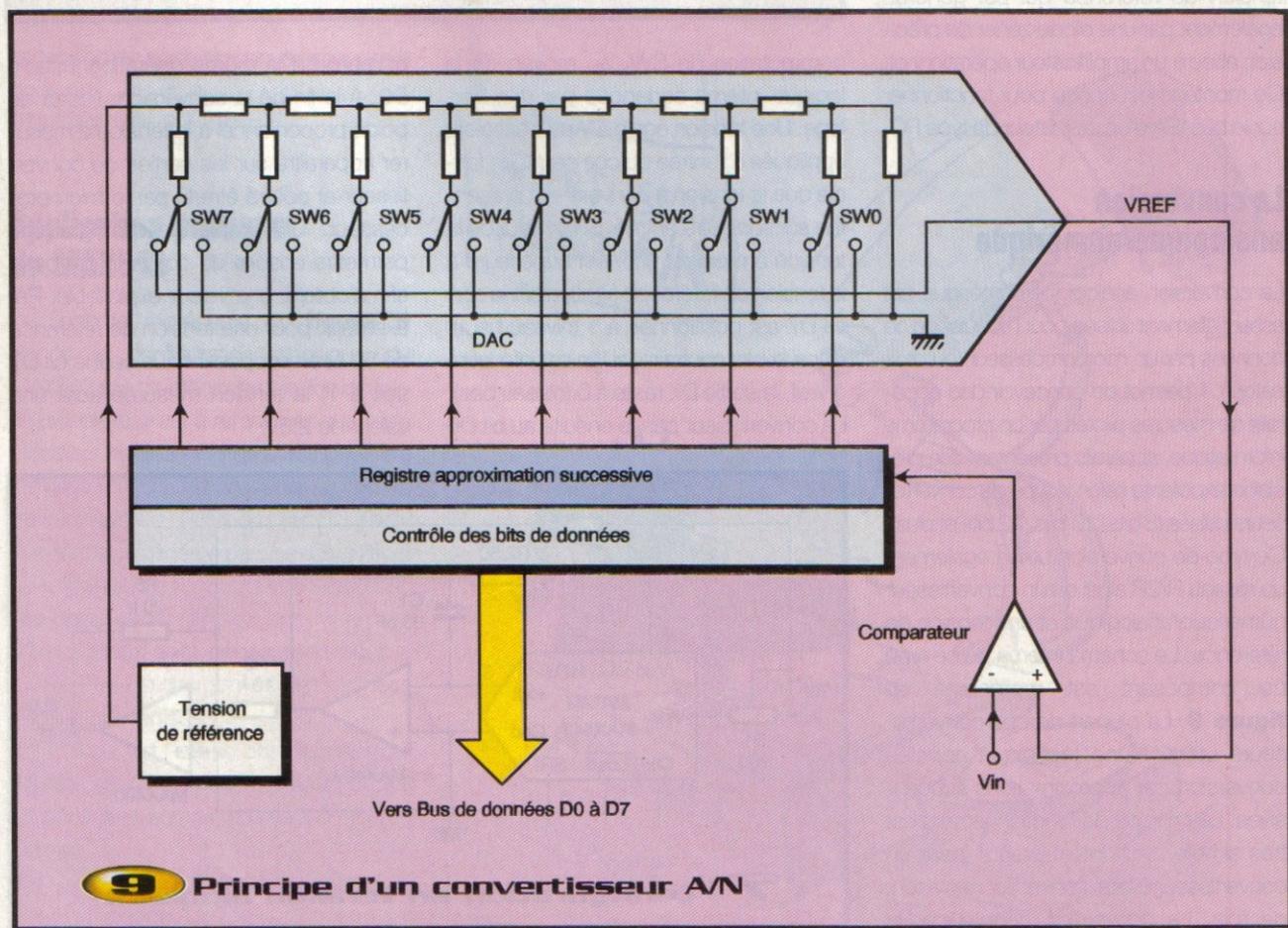
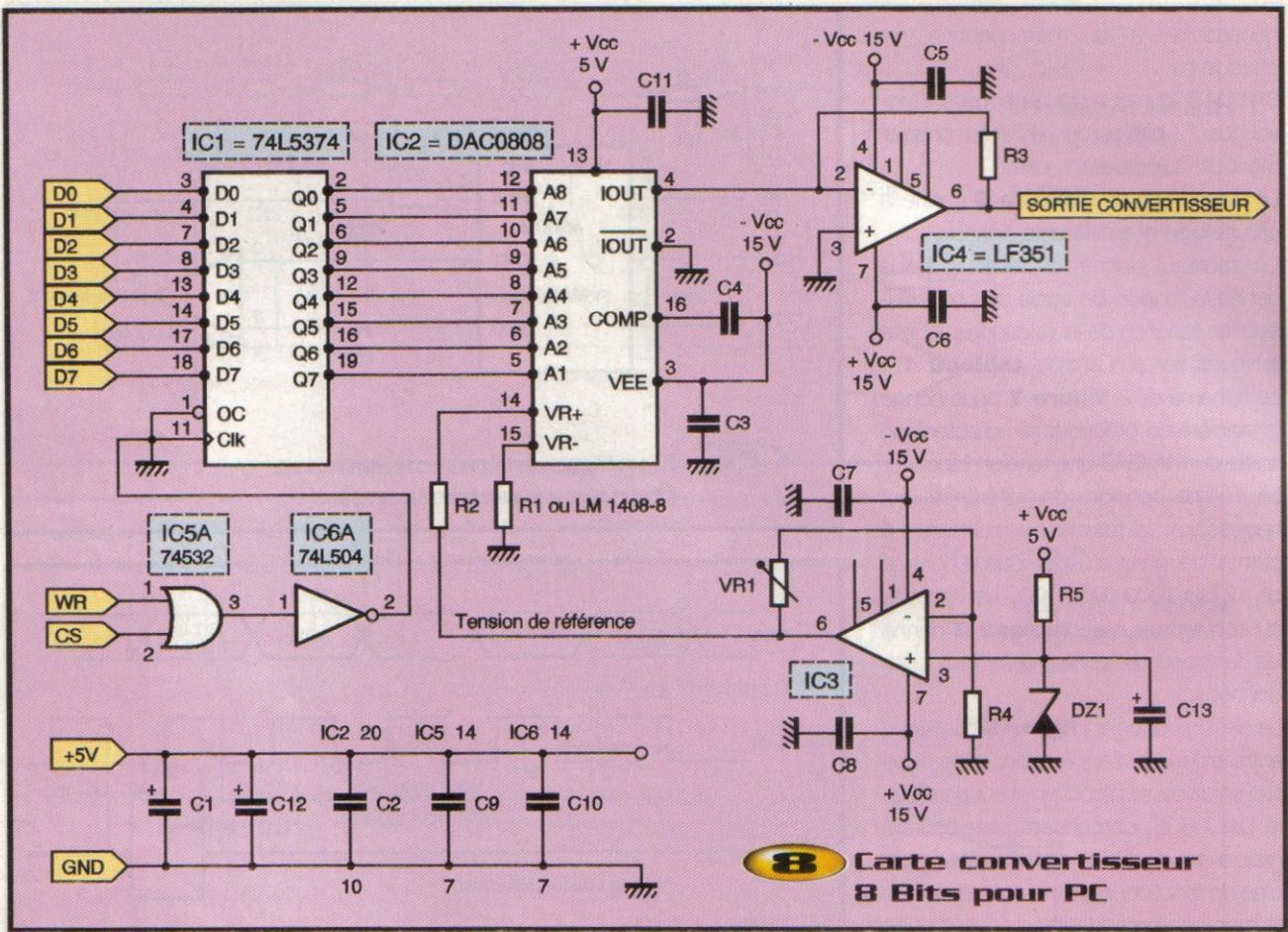
6 Diagramme de conversion

commutation de SW_7 au moyen de la logique interne cadencée par une horloge. Une tension égale à $V_{ref}/2$ est alors appliquée à l'entrée du comparateur, tandis que la tension à convertir est présentée sur son autre entrée. Si la valeur de la tension à mesurer (V_{in}) est supérieure à la tension de référence (V_{ref}), alors la sortie D7 est positionnée à 1 (niveau haut). Dans le cas contraire, si V_{in} est inférieure à V_{ref} , la sortie D7 reste à 0 (niveau bas). Le convertisseur passe ensuite au bit D6

et reproduit la même opération jusqu'à D0. A la fin de la conversion, l'octet de poids proportionnel à la tension à mesurer apparaîtra sur les sorties du convertisseur et pourra être lu par le microprocesseur. Un simple petit programme permettra ensuite de convertir cet octet afin d'obtenir une valeur exploitable. Par exemple, pour une tension de référence de 5V et en supposant que seul le bit D7 soit à 1, la tension mesurée aura une valeur de 2,5V.



7 Configuration en tension bipolaire

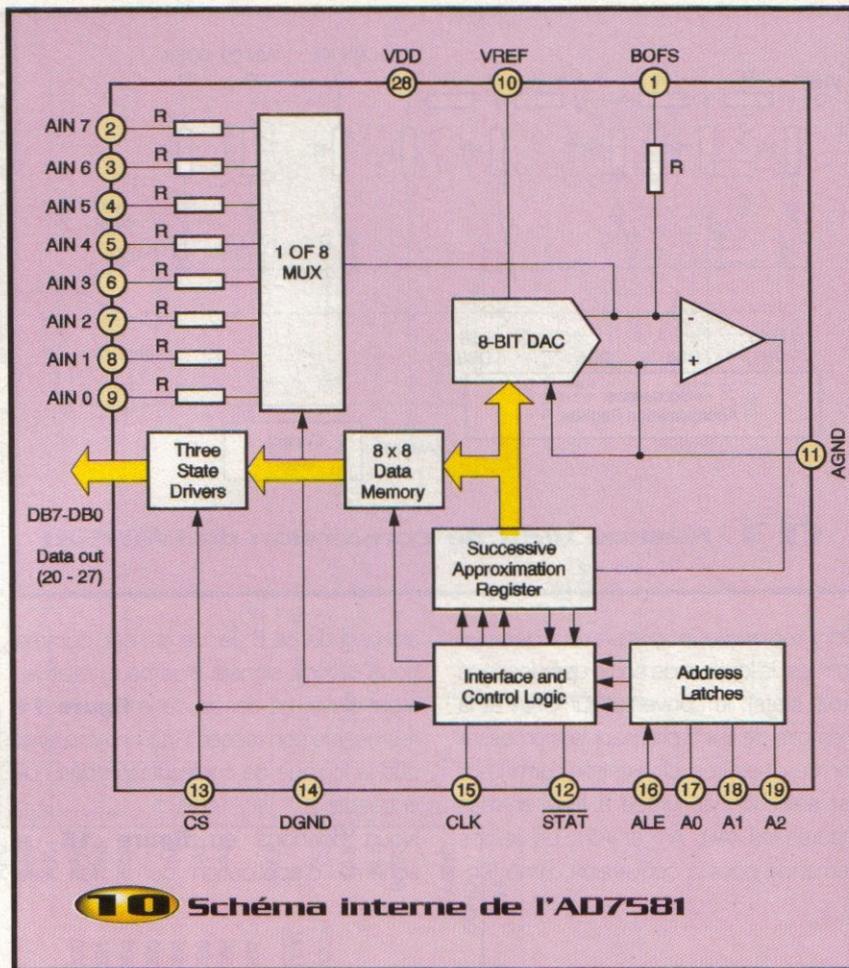


Il est évident que plus le nombre de pas sera élevé, plus la précision sera grande. En effet, pour une tension de référence de 5V et un convertisseur de 8 bits, le pas sera de 0,0195V (5 / 256), alors qu'un convertisseur travaillant en 12 bits aura une résolution de 0,0012V (12 / 4096).

Comme on le voit, la conversion par approximations successives est simple dans son fonctionnement. Mais elle présente également un défaut qui est une relative lenteur puisque plusieurs opérations sont nécessaires pour la lecture d'une tension. Comparé à un CNA qui exécute une conversion en moins de 100 ns dans le pire des cas, un CAN demande plusieurs dizaines de µs pour la même opération (inverse).

Application pratique des CAN

Notre choix s'est porté sur un convertisseur fabriqué par ANALOG DEVICES, le AD7581. Il est très simple d'emploi et présente la particularité de posséder un système sur 8 canaux, ce qui permet de disposer de huit entrées de mesure. Son schéma interne est donné en **figure 10**.



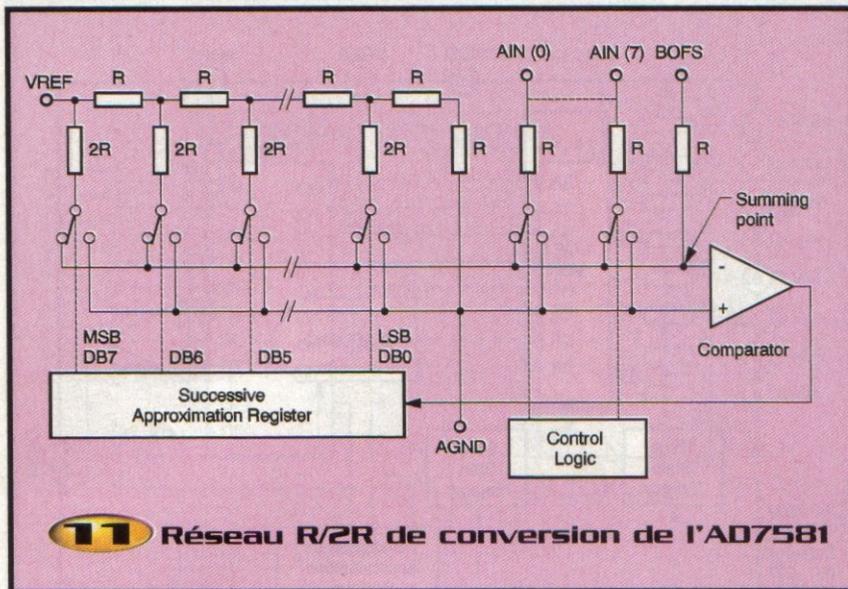
10 Schéma interne de l'AD7581

C'EST VOTRE
CONVERTISSEUR A/N
MADAME DENIS...



Bibliographie

- MAXIM, new releases data book, 1992
- Mesures et PC, P. OGUIC, collection ETSF (DUNOD)
- Interfaces PC, P. OGUIC, collection ETSF (DUNOD)



11 Réseau R/2R de conversion de l'AD7581

On y remarque le multiplexeur des huit entrées, la logique de sortie des données (trois états), le convertisseur CNA et la mémoire de 8 x 8 bits pour la mémorisation des mesures. Une entrée parmi les 8 est sélectionnée grâce à trois entrées d'adresses (A0, A1 et A2). Le temps demandé pour la conversion d'une ten-

sion est de 66,6 μ s, ce qui est, comme nous l'avons signalé plus haut, relativement élevé. Le dessin de la **figure 11** représente son réseau R/2R interne ainsi que la logique de commutation des huit entrées.

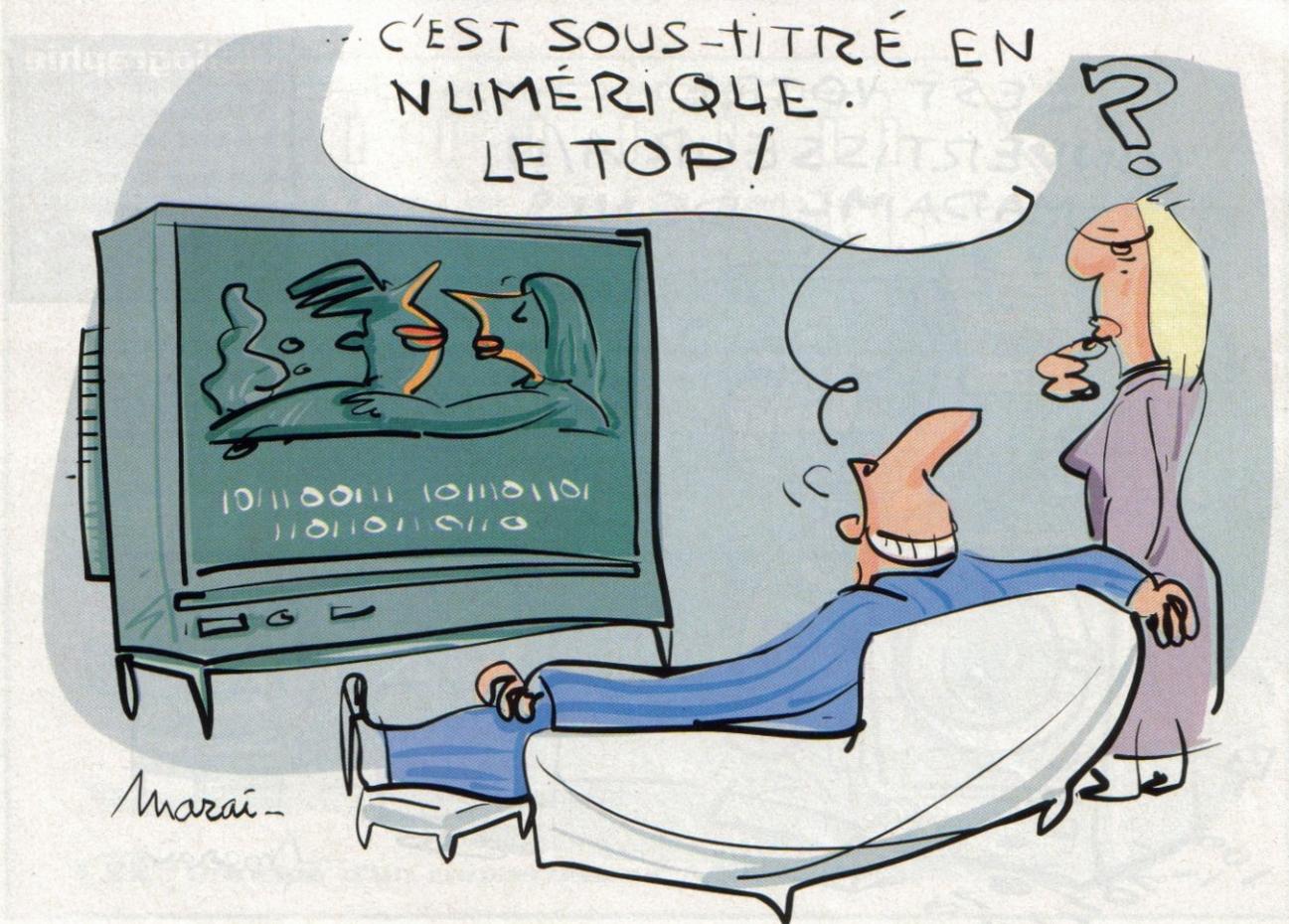
Nous donnons en **figure 12** un schéma d'application qui a fait ses

preuves. Ce montage est prévu pour fonctionner sur le bus ISA d'un ordinateur PC. Le signal d'horloge, qui ne devra en aucun cas dépasser 1,2 MHz, pourra être généré soit par un oscillateur intégré, soit par un circuit de type 74HCT4060 et un quartz d'une valeur de 18 MHz. La sortie du signal se fera sur la broche 7 du circuit intégré.

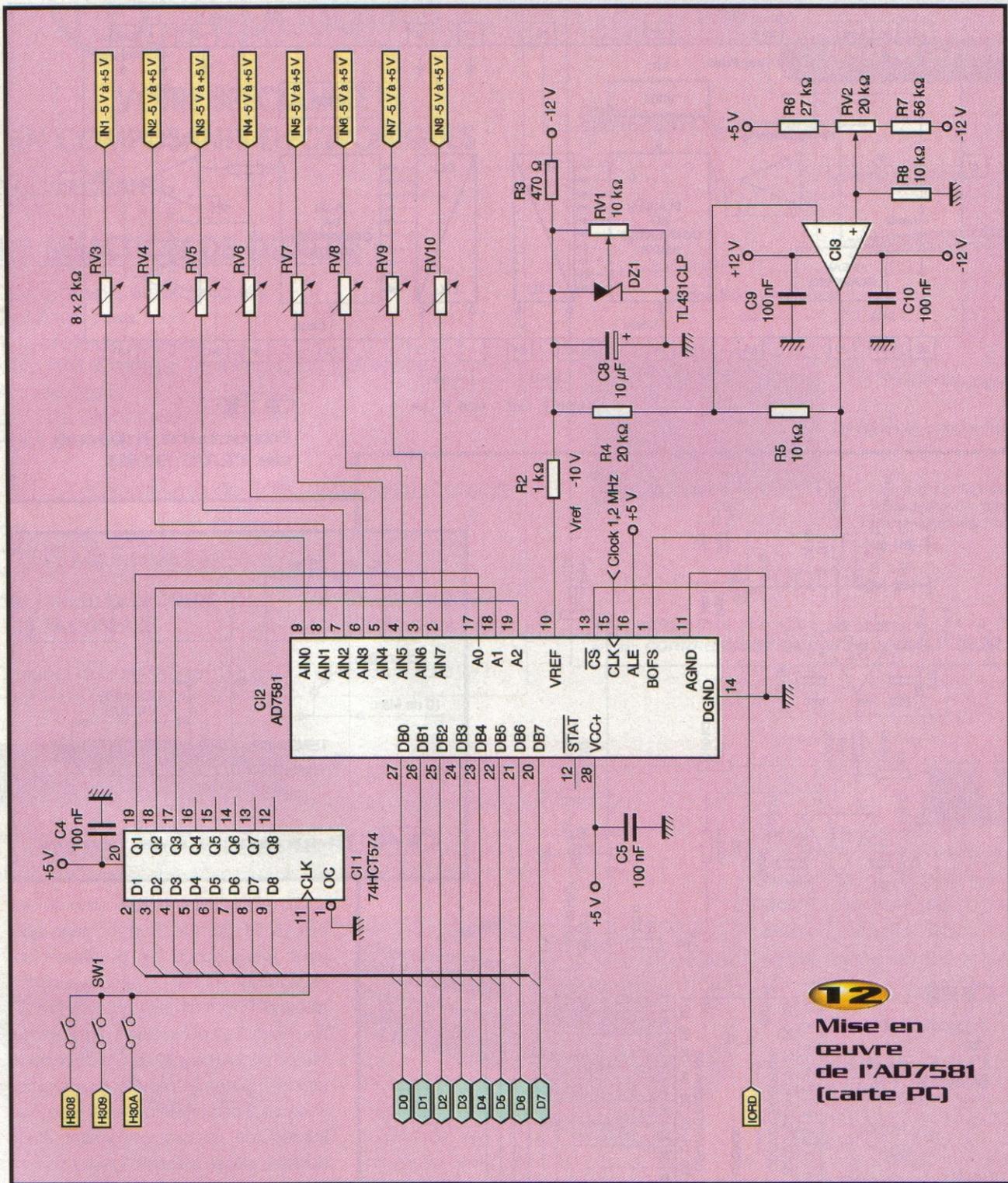
Là encore, nous utilisons une diode zéner de référence TL431CLP, mais alimentée par une tension négative de -12V afin d'obtenir les -10V requis par le AD7581. La résistance ajustable R_{V_2} ainsi que celles placées en série dans les entrées permettent de régler le 0V lorsqu'aucune tension n'est appliquée sur les entrées du convertisseur.

Un convertisseur analogique/numérique plus rapide

Il est souvent nécessaire de disposer d'un convertisseur plus rapide, en particulier lorsque les signaux que l'on veut



Marai-



12

Mise en œuvre de l'AD7581 (carte PC)

mesurer fluctuent. La vitesse de conversion doit alors atteindre une vitesse élevée. On doit alors mettre en œuvre des convertisseurs flash qui ne fonctionnent pas selon la technique d'approximations successives. Ces convertisseurs sont équipés d'un nombre important de comparateurs et permettent une conversion immédiate de la tension d'entrée. Il en existe un en particulier, le UVC3130 qui est un convertisseur utilisé en vidéo, mais

qui peut l'être également dans n'importe quelle application.

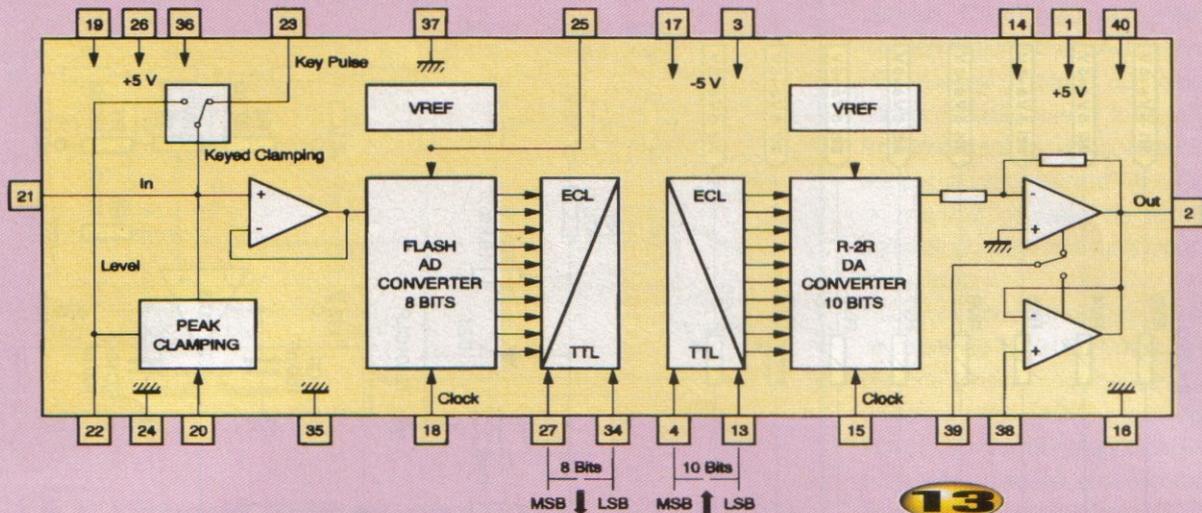
Ce composant, dont le schéma interne est donné en **figure 13**, contient :

- un convertisseur A/D 8 bits pouvant fonctionner à une vitesse supérieure à 30 MHz,
- un convertisseur D/A 10 bits fonctionnant avec un réseau R/2R et par commutations de sources de courant,
- un buffer d'entrée présentant une impé-

dance d'entrée de 20 MΩ à une fréquence d'entrée de 1 kHz et 100 kΩ à une fréquence de 10 MHz,

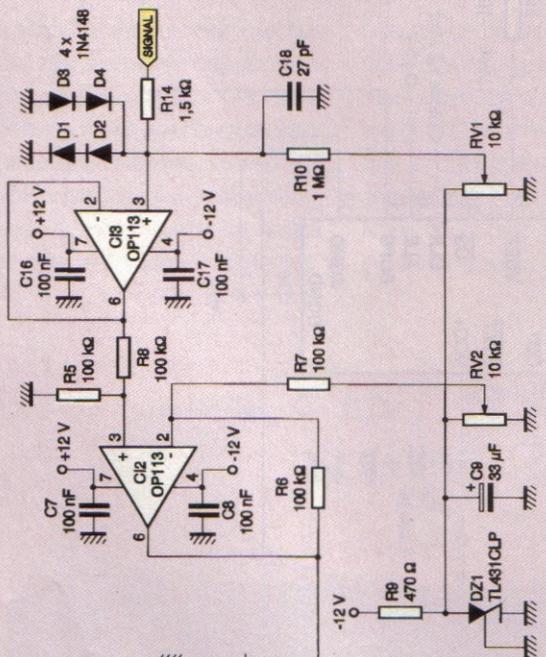
- une référence de tension,
- un préamplificateur de sortie.

Ses entrées et sorties sont compatibles TTL, ce qui permet de l'utiliser dans n'importe quel montage. Le diagramme de la **figure 14** donne les temps nécessaires au circuit afin de convertir une donnée. Enfin, en **figure 15**, est représenté un



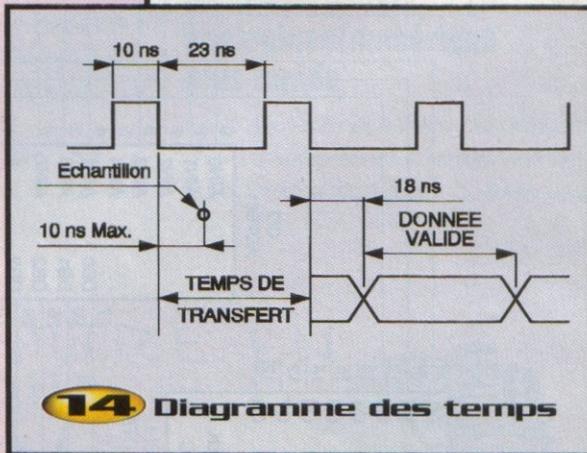
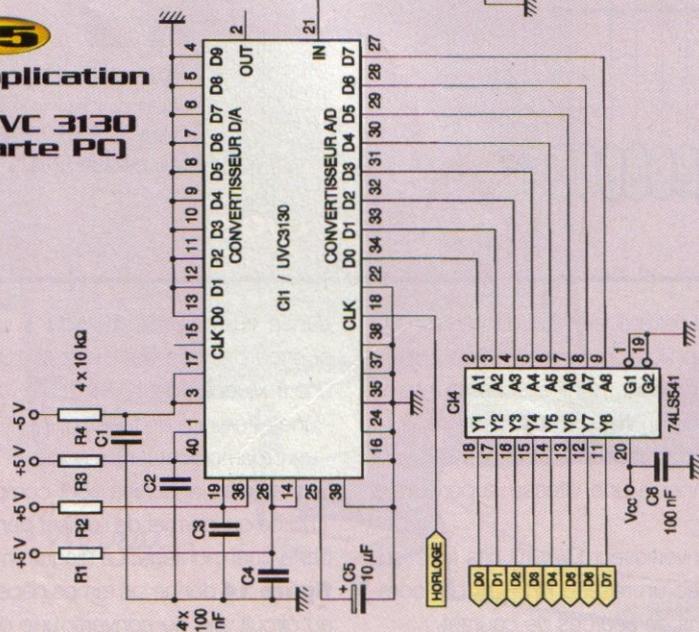
13

Structure interne de l'UVC 3130



15

Application de l'UVC 3130 (carte PC)



14

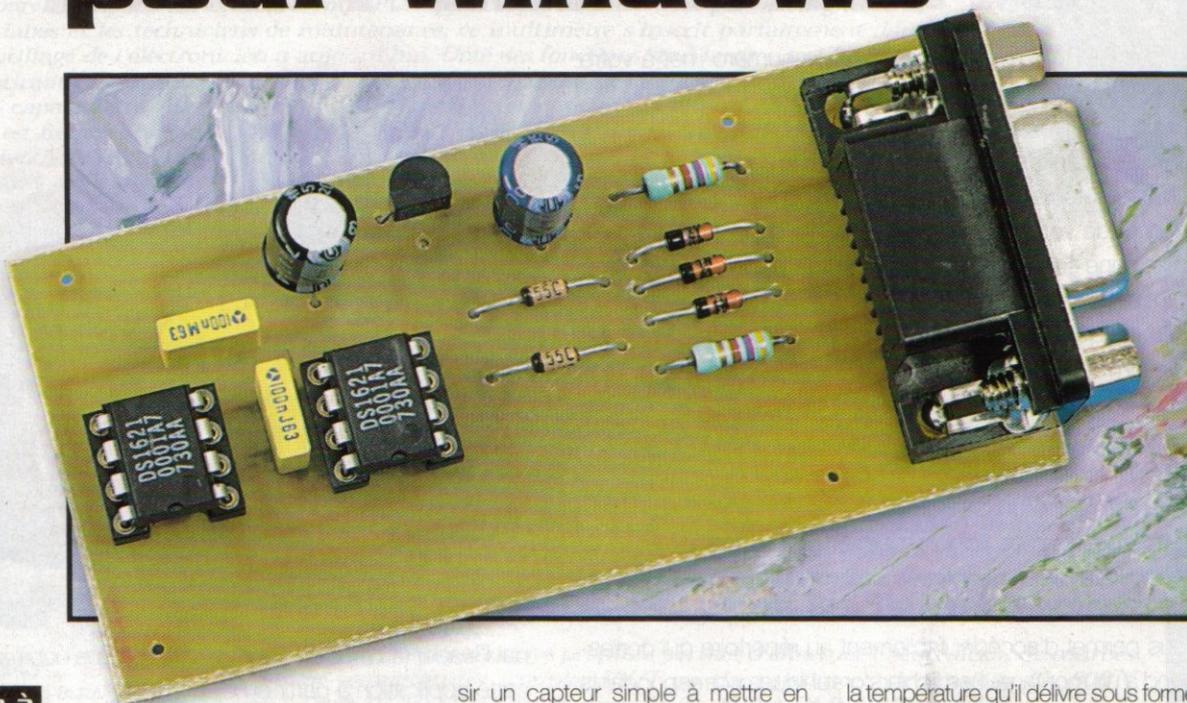
Diagramme des temps

schéma donnant une application du circuit UVC3130. C'est un montage que nous avons conçu afin de réaliser un oscilloscope sur PC. On remarque immédiatement l'extrême simplicité du circuit. Ne nécessitant qu'un signal d'horloge de vitesse choisie pour la mesure de signaux rapides ou non et piloté par un logiciel adéquat, il permet de disposer d'un instrument de mesure performant étant donné la vitesse à laquelle fonctionne son convertisseur analogique/numérique.

Nous voici arrivés au terme de ce court et simple exposé sur les convertisseurs. Nous espérons qu'il permettra à certains de nos lecteurs une meilleure compréhension de leur fonctionnement. Si certains points restaient néanmoins obscurs, nous leur proposons de nous poser des questions via notre service internet (<http://www.eprat.com>).

P. OGUIC

Thermomètre pour Windows



Même s'il arrive à tous les utilisateurs de Windows d'avoir des poussées de fièvre, ce n'est pas pour constater cela que ce thermomètre a été conçu ! Notre montage permet en effet de mesurer la température de deux capteurs distincts, placés où bon vous semble, et d'afficher les résultats de ces mesures dans une fenêtre de Windows ou sous forme réduite dans la barre des tâches, comme c'est le cas par exemple pour l'horloge système.

La mesure de ces températures est entièrement automatique et la valeur affichée est rafraîchie au rythme de votre choix, programmable d'une fois par seconde à une fois par minute. De plus, un fichier texte, contenant la liste de toutes les mesures réalisées, peut être automatiquement enregistré sur le disque dur de votre PC pour exploitation ultérieure.

Ce montage ne nécessite aucune intervention dans le PC, auquel il est associé, car il se connecte sur un de ses ports séries sur lequel il prélève même son alimentation.

Enfin, et avant de découvrir cette petite merveille, précisons que le logiciel qui la pilote est un véritable logiciel Windows, c'est à dire qu'il s'exécute sans problème en tâche de fond alors que vous continuez à travailler avec les applications de votre choix.

Le DS1621 de Dallas

Afin que notre montage soit le plus simple possible, et il l'est comme vous l'avez peut être déjà constaté en examinant son schéma, il fallait choisir

un capteur simple à mettre en œuvre et capable de communiquer facilement avec le PC. Notre choix s'est porté sur le DS1621 de Dallas qui contient, tout à la fois, un thermomètre de précision, un thermostat (non utilisé ici) et une interface pour bus I2C. Les caractéristiques principales de ce circuit peuvent être résumées de la façon suivante :

- Circuit très peu coûteux, proposé en boîtier DIL 8 pattes ou en boîtier CMS.
- Plage de mesure de température de -55°C à +125°C par pas de 0,5°C.
- Température fournie sous forme d'un mot de 8 ou 9 bits permettant de bénéficier d'une précision de 1°C (8 bits) ou de 0,5°C (9 bits).
- Alimentation sous toute tension comprise entre 2,7 et 5,5V avec une consommation inférieure à 1 mA en phase de mesure et à 1 µA en phase de repos.
- Interface I2C normalisée deux fils.
- Thermostat interne programmable (non utilisé ici) avec sortie à drain ouvert.

Si l'on ne souhaite pas faire appel à la fonction thermostat, l'utilisation de ce circuit est fort simple. Il suffit en effet, après l'avoir initialisé en lui envoyant un mot de configuration, de venir lire

la température qu'il délivre sous forme d'un mot de 8 ou 9 bits.

Ce dialogue avec le circuit a lieu au moyen d'un bus I2C dont nous allons rappeler brièvement le principe, car il conditionne une grande partie du schéma et du logiciel.

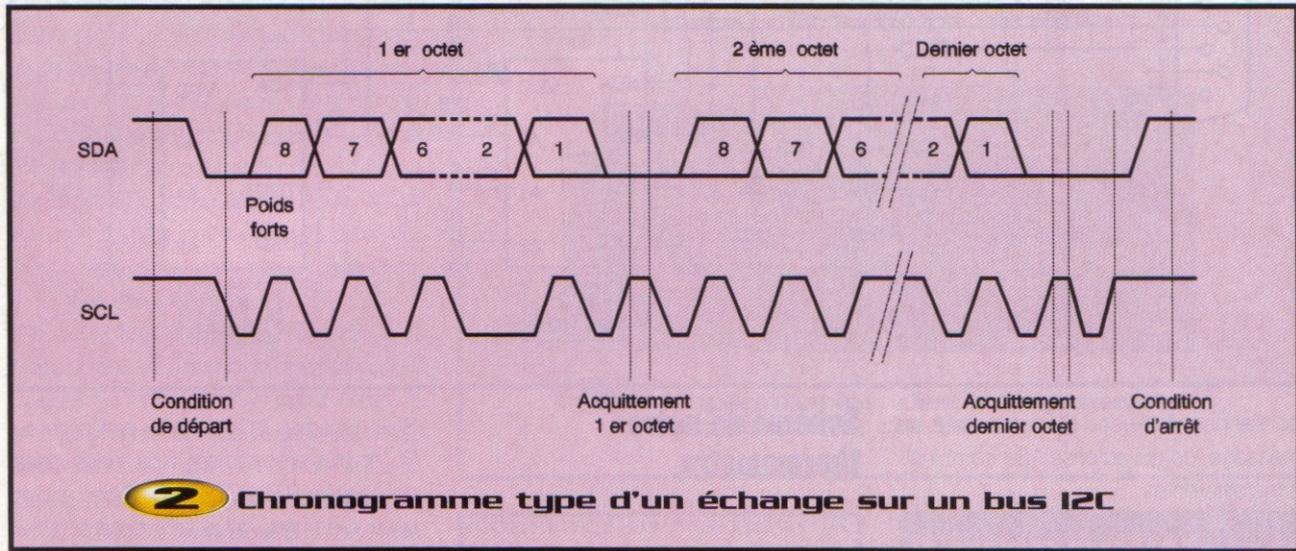
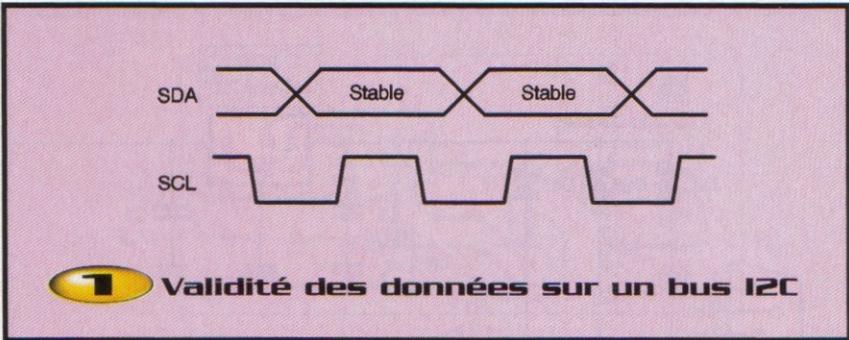
Le bus I2C en quelques mots

Le bus I2C, qui n'utilise que deux lignes de signal (et les masses correspondantes bien sûr), permet à un certain nombre d'appareils d'échanger des informations sous forme série avec un débit pouvant atteindre 100 Kbits/seconde, ou 400 Kbits/seconde pour les versions les plus récentes. Ses points forts sont les suivants :

- c'est un bus série bifilaire utilisant une ligne de données appelée SDA (Serial Data) et une ligne d'horloge appelée SCL (Serial Clock),
- les données peuvent être échangées dans les deux sens sans restriction,
- le bus est multi-maîtres,
- chaque abonné dispose d'une adresse codée sur 7 bits. On peut donc connecter simultanément 128

abonnés d'adresses différentes sur le même bus, sous réserve de ne pas le surcharger électriquement bien sûr,

- un acquittement est généré pour chaque octet de donnée transféré,
- le bus peut travailler à une vitesse maximum de 100 Kbits/seconde (ou 400 Kbits/seconde) étant entendu que son protocole permet de ralentir automatiquement l'équipement le plus rapide pour



s'adapter à la vitesse de l'élément le plus lent, lors d'un transfert,

- le nombre maximum d'abonnés n'est limité que par la charge capacitive maximale du bus qui peut être de 400 pF. Ce nombre ne dépend donc que de la technologie des circuits et du mode câblage employés,
- les niveaux électriques permettent l'utilisation de circuits en technologies CMOS, NMOS ou TTL.

Ceci étant vu, la **figure 1** résume le principe fondamental d'un transfert à savoir : une donnée n'est considérée comme valide sur le bus que lorsque le signal SCL est à l'état haut. L'émetteur doit donc positionner la donnée à émettre lorsque SCL est à l'état bas et la maintenir tant que SCL reste à l'état haut. Comme la transmission s'effectue sous forme série, une information de début et de fin doit être prévue. L'information de début s'appelle ici "condition de départ" et l'information de fin "condition d'arrêt".

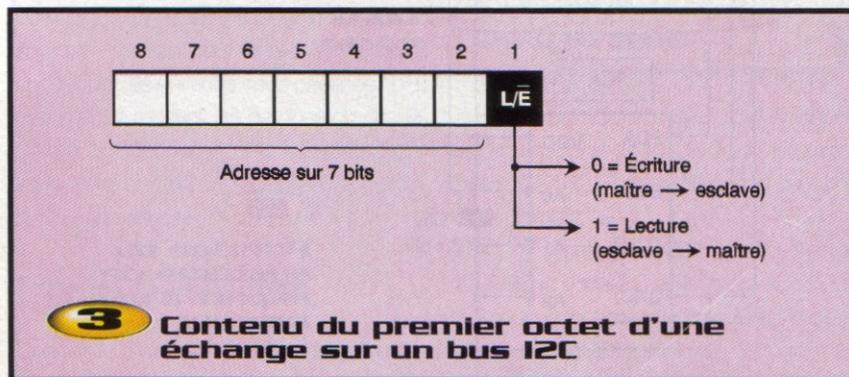
Une condition de départ est réalisée lorsque la ligne SDA passe du niveau haut au niveau bas alors que SCL est au niveau haut. Réciproquement, une condition d'arrêt est réalisée lorsque SDA passe du

niveau bas au niveau haut alors que SCL est au niveau haut.

En outre, bien que nous soyons en présence d'un bus série, les données sont envoyées par paquets de huit octets, même si un octet regroupe en fait huit bits indépendants. Le bit de poids fort est envoyé le premier. Chaque octet est suivi par un bit d'acquiescement de la part du destinataire et l'ensemble du processus fonctionne de la façon présentée **figure 2**. Nous y voyons tout d'abord une condition de départ, générée par le maître du bus à cet instant. Elle est suivie par le premier octet de données, poids forts en tête. Après le huitième bit, l'émetteur, qui est aussi le maître dans ce cas, met sa ligne

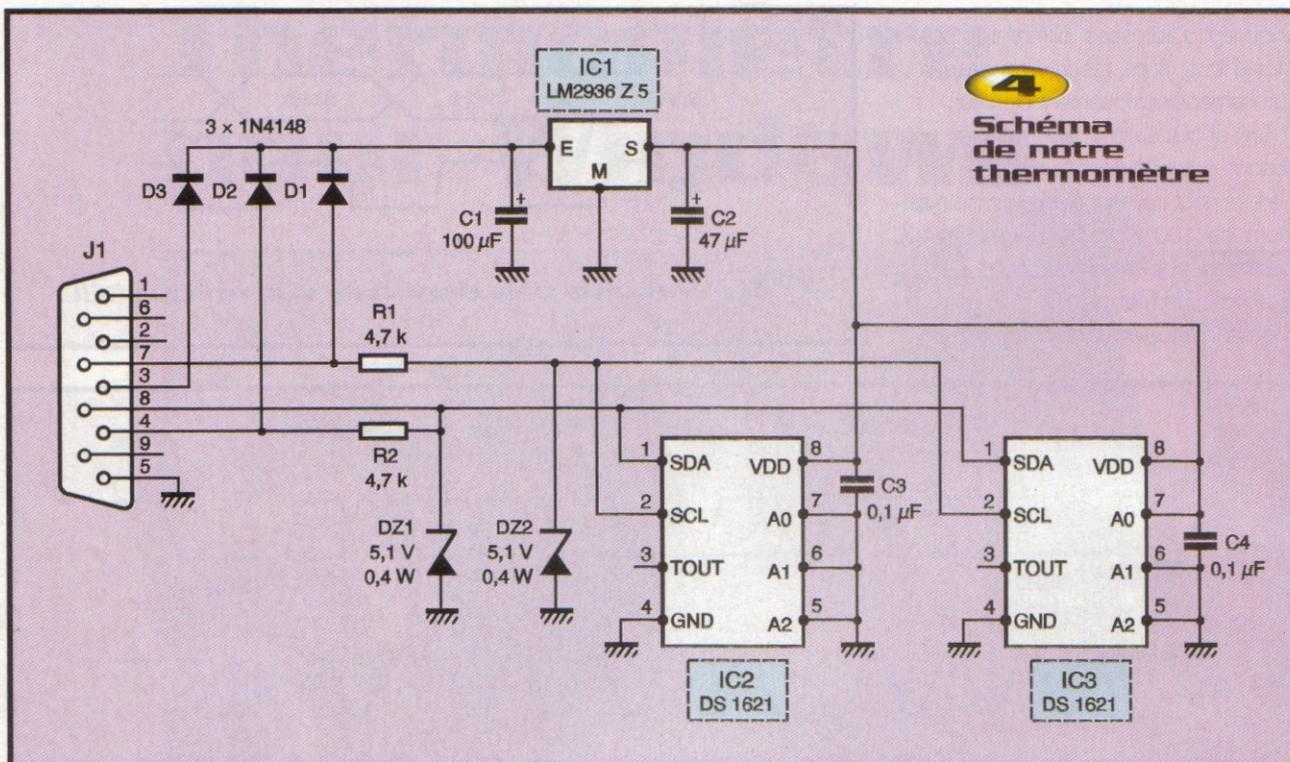
SDA au niveau haut, c'est à dire au repos, mais continue à générer l'horloge sur SCL. Pour acquiescer l'octet, le récepteur doit alors forcer la ligne SDA au niveau bas pendant l'état haut de SCL qui correspond à cet acquiescement, prenant en quelque sorte la place d'un neuvième bit. Le processus peut alors continuer avec l'octet suivant et se répéter autant de fois que nécessaire pour réaliser un échange d'informations complet. Lorsque cet échange est terminé, le maître génère une condition d'arrêt.

Pour compléter cet exposé, examinez la **figure 3** qui montre le contenu du premier octet, toujours présent en début d'échange. Ses 7 bits de poids forts



4

Schéma de notre thermomètre



contiennent l'adresse du destinataire du message, ce qui autorise 128 combinaisons différentes.

Le bit de poids faible, quant à lui, indique si le maître va réaliser une lecture ou une écriture.

En d'autres termes, si ce bit est à zéro le maître va écrire dans l'esclave ou lui envoyer des données.

Si ce bit est à un, le maître va lire dans l'esclave, c'est à dire que le maître va recevoir des données de l'esclave.

Schéma de notre thermomètre

Ce court exposé ne vous a pas transformé en spécialiste de l'I2C mais vous en savez maintenant assez pour découvrir le principe et le schéma de notre thermomètre présenté **figure 4**.

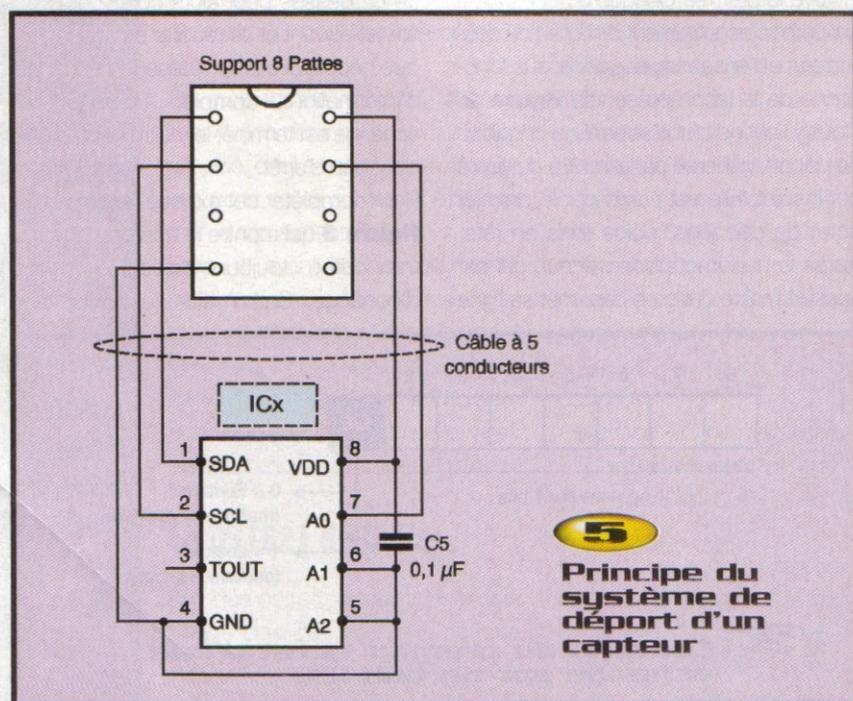
Comme vous pouvez le constater, deux DS1621 peuvent y être utilisés, repérés respectivement IC₂ et IC₃. Ils sont tous deux reliés au bus I2C via leurs lignes SDA et

SCL générées ici à partir du port série du PC et, plus précisément, à partir des lignes de contrôle RTS, CTS et DTR de ce port série. Ce choix, qui peut à priori sembler curieux car l'utilisation de RXD et de TXD aurait pu paraître plus logique, est dicté par le fait que l'on peut facilement manipuler par logiciel l'état de ces lignes, ce qui n'est pas le cas des lignes d'émission et de réception de données qui sont sous le contrôle direct de l'UART du PC.

Comme les niveaux présents sur ces lignes sont aux normes RS232, ils sont réduits à la plage 0 - 5V par les résistances R₁ et R₂ et les zéners DZ₁ et DZ₂. Remarquez que la ligne SDA, nécessairement bidirectionnelle, utilise DTR dans le sens PC vers DS1621 et CTS dans le sens contraire.

Afin de différencier les deux capteurs, leurs adresses sont programmées différemment puisque IC₃ est placé à l'adresse zéro par mise à la masse de ses pattes A0, A1 et A2 alors que IC₂ est placé à l'adresse un par mise au niveau haut de sa patte A0. Si un seul circuit est utilisé, ce que permet le logiciel associé, il doit être placé à l'adresse zéro c'est à dire sur le support de IC₂.

L'alimentation du montage est prélevée sur le port série grâce aux diodes D₁ à D₃ qui chargent le condensateur C₁ lorsque les lignes correspondantes du port sont au niveau haut. Cette tension est ensuite réglée à 5V par IC₁ qui est un régulateur intégré trois pattes à très faible chute de tension



5

Principe du système de déport d'un capteur

et, surtout, à très faible consommation. Ne le remplacez en aucun cas par un classique 78L05, le montage ne pourrait pas fonctionner en raison de la consommation excessive de ce dernier.

Réalisation

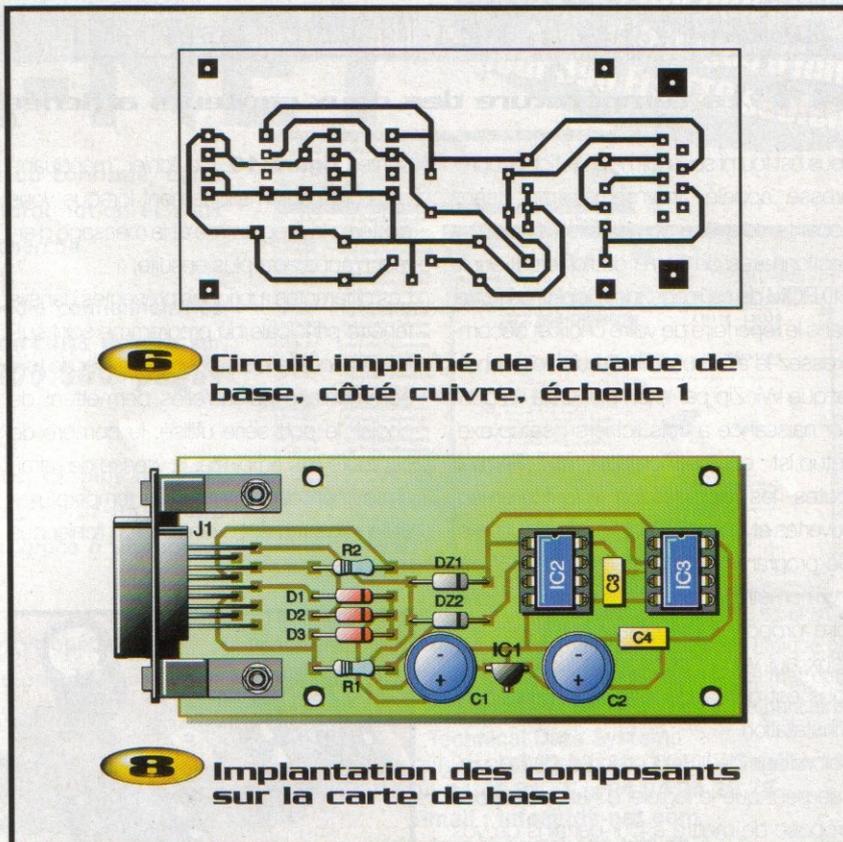
Afin que notre montage soit aussi polyvalent que possible, nous avons prévu deux circuits imprimés. Le premier, ou circuit de base, reçoit l'ensemble des composants de la **figure 4**. Mais, et vous en conviendrez sans doute avec nous, il ne sert à rien d'avoir les deux capteurs IC₂ et IC₃ côte à côte sur le même circuit imprimé ! Nous avons donc dessiné un minuscule circuit imprimé qui ne supporte qu'un capteur, IC₂ ou IC₃, afin de pouvoir placer celui-ci où bon vous semble.

Ce petit circuit imprimé se raccorde à la carte de base au moyen de cinq fils souples soudés sur un support de circuit intégré 8 pattes, faisant office de connecteur mâle, qui s'enfiche alors dans le support de IC₂ ou de IC₃, selon celui de ces circuits que vous voulez déporter. La **figure 5** précise cela aussi bien qu'un long discours.

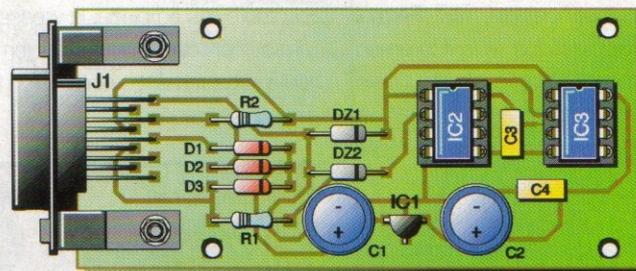
Rien ne vous interdit, bien sûr, de réaliser ce circuit de déport en deux exemplaires afin de déporter vos deux capteurs.

L'approvisionnement des composants ne pose pas de problème particulier mais, au risque de nous répéter, ne remplacez en aucun cas le LM2936Z5 par un autre régulateur 5V car le montage ne pourrait plus fonctionner. En cas de difficulté, sachez que les DS1621 et le LM2936Z5 sont disponibles chez Radiospares pour un prix dérisoire (moins de 100 francs au total).

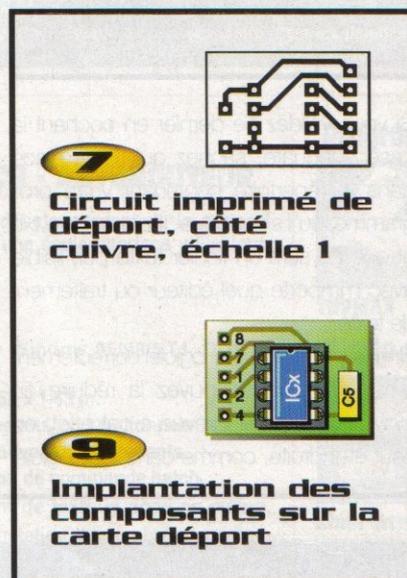
Les circuits imprimés nécessaires sont visibles en **figures 6** et **7** et les plans d'implantation correspondants en **figures 8** et **9**. Commencez l'assemblage par la carte de base en veillant à bien respecter le sens des composants polarisés. Montez des supports 8 pattes aux emplacements de IC₂ et IC₃ ce qui vous permettra d'y enficher les circuits ou le câble de liaison avec le circuit déporté. Ce dernier reçoit un support 8 pattes ou le DS1621 directement soudé, si vous le souhaitez, ainsi qu'un condensateur de découplage. La liaison avec la carte de base est à réaliser avec du câble à 5 conducteurs (câble téléphonique à 3 paires par exemple). L'extrémité de ce câble est sou-



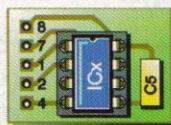
6 Circuit imprimé de la carte de base, côté cuivre, échelle 1



8 Implantation des composants sur la carte de base



7 Circuit imprimé de déport, côté cuivre, échelle 1



9 Implantation des composants sur la carte déport

dée sur un support de circuit intégré à contacts tulipes car c'est sur ce type de support que l'on peut le plus facilement souder des fils dans les trous destinés normalement à recevoir les broches du circuit intégré. Ce support peut ensuite très facilement être enfiché dans celui de IC₂ ou IC₃ sur la carte de base.

Logiciel et utilisation

Une fois le montage terminé et soigneusement vérifié, placez au moins un DS1621

Nomenclature

IC₁ : LM2936 Z 5

IC₂, IC₃ : DS1621

(un ou deux selon utilisation)

D₁ à D₃ : 1N914 ou 1N4148

DZ₁, DZ₂ : zéners 5,1V/0,4W

R*1, R*2 : 4,7 kΩ 1/4W 5%

(jaune, violet, rouge)

C₁ : 100 µF/15V chimique radial

C₂ : 47 µF/15V chimique radial

C₃ à C₅ : 0,1 µF mylar

J1 : connecteur SubD 9 points femelle
coudé à implanter sur CI

2 supports de CI 8 pattes

ou 4 supports de CI 8 pattes

si un capteur déporté

sur le support IC₂ ou sur le circuit de déport relié au support de IC₂. Si vous utilisez les deux DS1621, placez-les comme bon vous semble.

Reliez le montage au port série COM1 ou COM2 de votre PC au moyen d'un câble muni de connecteurs normalisés SubD 9 points et installez le logiciel.

Ce logiciel a été écrit en Visual Basic et

Démarrer



23,0°C [26,5°C]

11 La température des deux capteurs affichée dans la barre des tâches

vous est fourni sous forme d'un fichier compressé appelé "thermometre.zip" (sans accents pour être compatible de tous les gestionnaires de noms de fichiers !) sur le CD ROM de ce magazine. Copiez ce fichier dans le répertoire de votre choix et décompressez-le avec un utilitaire de votre choix tel que WinZip par exemple. Cela va donner naissance à trois fichiers : setup.exe, setup.lst et thermometre.cab. Fermez toutes les applications éventuellement ouvertes et lancez l'exécution de setup.exe. Ce programme installe alors le logiciel du thermomètre. Acceptez les noms de répertoire proposés par défaut ou choisissez ceux qui vous conviennent lorsque cela vous est demandé lors de la procédure d'installation.

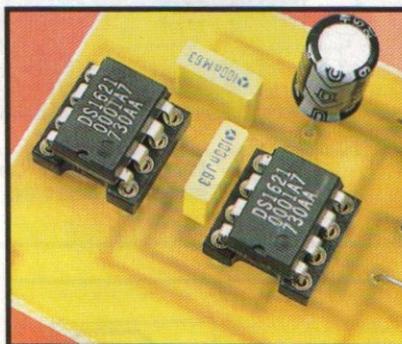
Selon l'âge de votre version de Windows, il se peut que le logiciel d'installation vous propose de mettre à jour certains de vos fichiers système. Acceptez sans crainte mais en respectant scrupuleusement la procédure indiquée. Vous pourrez alors terminer l'installation sans difficulté.

Le programme est alors accessible comme n'importe quel logiciel Windows en cliquant sur son nom (thermometre, bien sûr) disponible dans la liste des programmes du menu "démarrer"

Lors de la première exécution, une fenêtre d'avertissement va s'ouvrir indiquant l'absence du fichier "thermometre.ini". Ignorez-la en cliquant sur le bouton OK qu'elle comporte et vous aboutirez alors à la fenêtre normale d'exploitation du programme pré-

sentée **figure 10**. Ce fichier "manquant" sera créé automatiquement lorsque vous quitterez le programme et le message d'erreur n'apparaîtra plus ensuite.

Les différentes rubriques présentes dans la fenêtre principale du programme sont suffisamment explicites pour se passer de tout commentaire puisqu'elles permettent de choisir le port série utilisé, le nombre de capteurs mis en place, la vitesse de rafraîchissement de l'affichage des températures et l'enregistrement éventuel d'un fichier des mesures réalisées.



Si vous validez ce dernier en cochant la case adéquate, sachez qu'il sera placé dans le répertoire contenant votre programme, qu'il s'appellera "thermometre.txt" et que ce sera un fichier texte pur, lisible avec n'importe quel éditeur ou traitement de texte.

Une fois la fenêtre du logiciel correctement renseignée, vous pouvez la réduire au moyen de la case prévue à cet effet, en haut et à droite, comme dans tout logiciel

Windows qui se respecte. Vous disposerez alors, dans la barre des tâches de votre PC, de l'affichage de la température du ou des deux capteurs comme indiqué **figure 11**. Notez, à ce propos, que la température du capteur 2, c'est à dire de IC₃, est présentée entre parenthèses pour la différencier de l'autre.

Si votre PC est relativement lent ou si vous exécutez des applications très gourmandes en ressources, veillez à ne pas choisir une vitesse de rafraîchissement de l'affichage de la température trop élevée car cela ralentirait de façon importante votre machine. L'exécution complète du logiciel avec interrogation du port série a lieu, en effet, lors de chaque renouvellement de l'affichage.

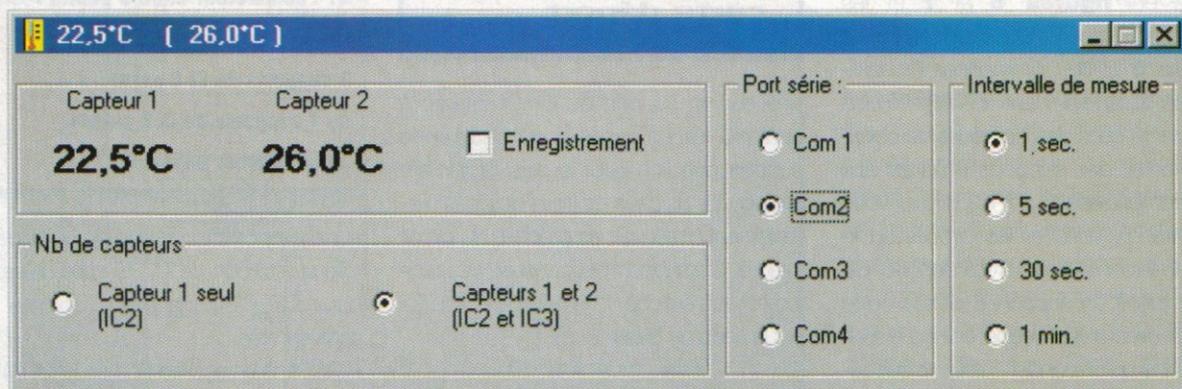
Conclusion

Voici de quoi surveiller d'un seul coup d'œil, à la barre des tâches, la température de votre bureau, de l'unité centrale de votre PC, celle qu'il fait dehors ou bien encore celle de tout autre local ou appareil dans lequel vous aurez jugé bon de placer un capteur déporté et, ce, pour un investissement ridicule et une mise en œuvre très facile.

C. TAVERNIER

Remerciements :

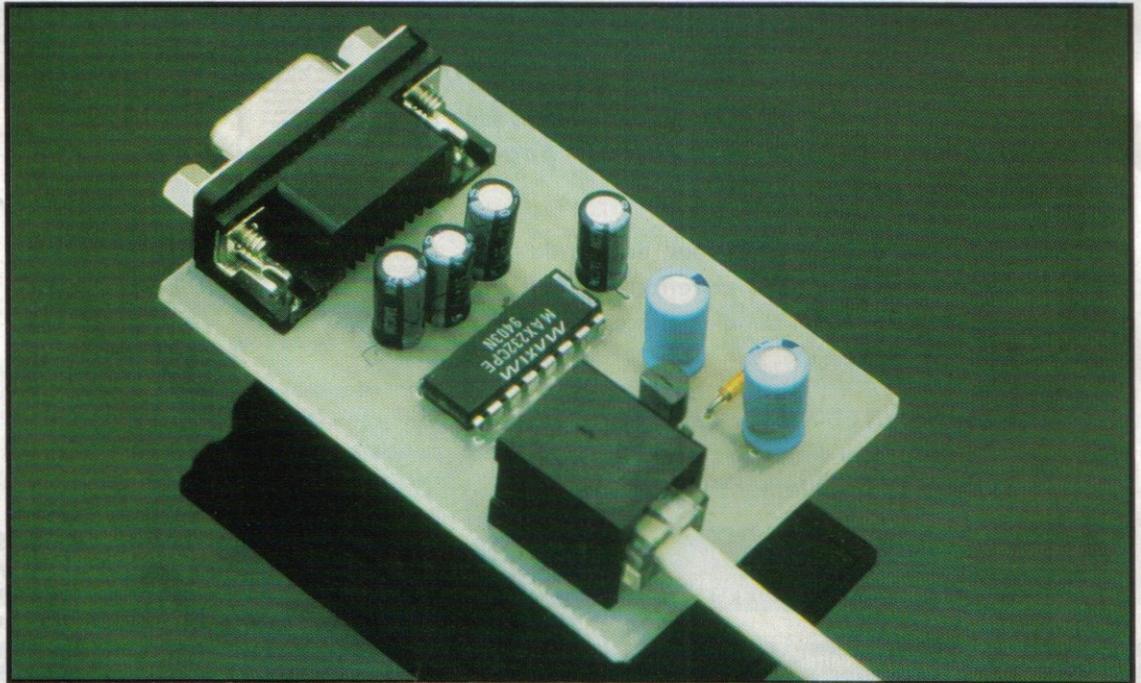
Merci à Claudio Lanconelli et à Alberto Ricci Bitti pour les idées de départ ayant conduit à la réalisation de cette intéressante application



10 La fenêtre principale du logiciel

Interface RS232

pour téléphone portable



Un adaptateur et un cordon

Chaque téléphone portable ou presque constituant un cas particulier, l'approche la plus pratique semble être de construire un module universel branché sur un port série du PC et qui pourra être relié à, virtuellement, n'importe quel téléphone à l'aide d'un cordon assurant une simple adaptation de connectique. Très rarement disponible dans le commerce de détail, la fiche spécifique de chaque modèle pourra souvent être obtenue en "cannibalisant" un accessoire peu coûteux comme un cordon allume-cigares ou un kit piéton.

La partie commune à tous les cas de figure est une version de plus de ce grand classique qu'est l'interface RS232/TTL. L'équivalent, en somme, des "cordons Minitel" qui ont fait, jadis, les délices de nos lecteurs. Bien que basé sur un très courant MAX232, le schéma de la **figure 1** n'en présente pas moins un certain nombre de particularités liées à l'ap-

plication envisagée. Contrairement à bien des schémas concurrents, il s'efforce tout d'abord de se passer d'une quelconque alimentation séparée.

Une tension positive extraite du port RS232 est limitée à 5V par un régulateur à faibles pertes mais, en cas de chute de tension excessive, on pourrait toujours se rabattre sur une pile 9V branchée après suppression de la diode. Une embase pour jack modulaire RJ11 (4 contacts sur 6 positions) rassemble les lignes d'entrée et sortie de données, la masse et le + 5V dont certains téléphones

portables ont besoin pour activer leur port série. Quelques modèles nécessitent également le pontage d'au moins deux broches de leur connecteur d'extension, liaison qui se fera tout simplement dans la fiche spécifique, lors de son montage, au bout du cordon serti sur la fiche RJ11.

Très simple, la réalisation pratique du module adaptateur se fera sur un petit circuit imprimé simple face, conforme au tracé de la **figure 6**. Câblé selon le plan de la **figure 7**, il pourra soit s'enficher directement dans un port COM du PC, soit rejoindre celui-ci par l'intermédiaire d'une rallonge DB9 mâle/femelle courante ou d'un adaptateur DB9/DB25.

La réalisation du câble spécifique se fera soit à partir d'un cordon RJ11 tout fait, soit en sertissant une fiche au bout d'un morceau de câble adéquat. On s'efforcera alors de respecter le sens permettant à la masse de correspondre

**Fiche RJ11 et
fiche GSM spécifique**



Tout téléphone portable normalement constitué possède un connecteur d'extension multi-broches à usage plus ou moins universel. Parallèlement au branchement de toute une variété d'accessoires, il permet presque toujours d'établir une liaison série pour dialoguer avec son processeur central. Moyennant un cordon d'adaptation RS232, le premier PC venu peut alors accéder à toutes sortes de fonctions "secrètes" grâce à des informations et des logiciels parfois disponibles tout simplement sur Internet.

au fil noir et au +5V de passer par le fil rouge, ce qui minimisera les risques d'erreurs aux conséquences potentiellement funestes !

Il restera alors à monter la fiche spécifique à l'autre extrémité de ce câble, après avoir obtenu (par exemple sur Internet) le détail de son brochage.

A titre d'exemple et d'illustration, les **figures 3** et **4** montre comment raccorder le câble sur quelques modèles de fiches correspondant à des téléphones portables courants. Recueillies sur Internet, ces informations sont naturellement reproduites sous toutes réserves, chacun devant bien comprendre qu'il en usera entièrement à ses risques et périls.

Les logiciels

En principe, n'importe quel logiciel de communication (ou "terminal") sous DOS ou Windows devrait pouvoir servir à dialoguer avec n'importe quel téléphone portable relié

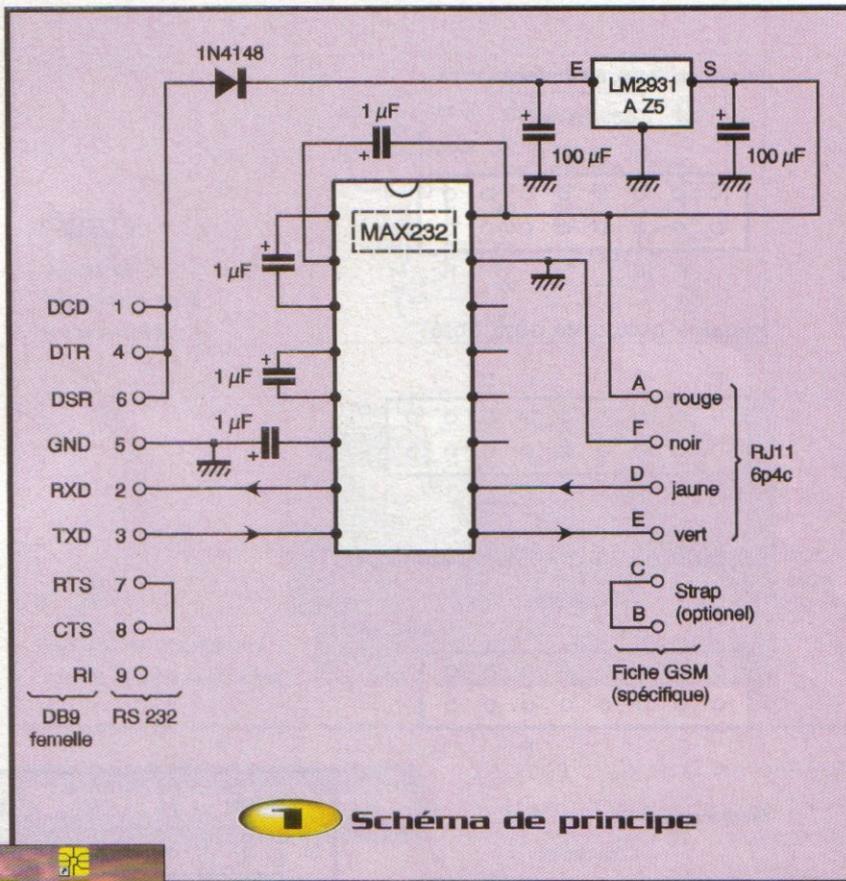
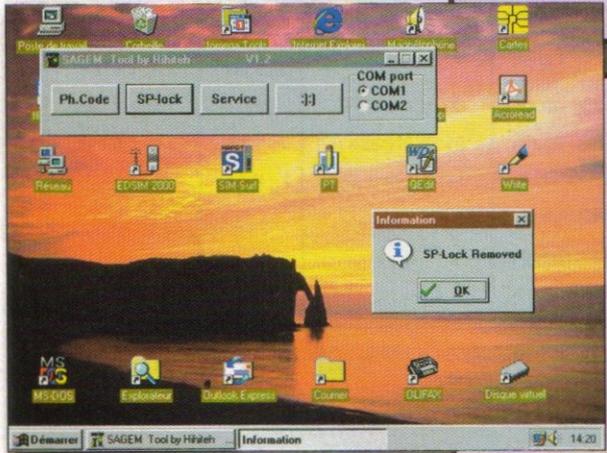


Schéma de principe

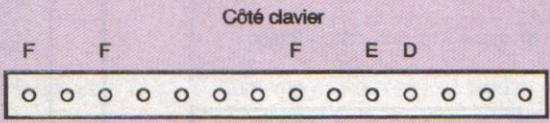


2 Vue d'écran

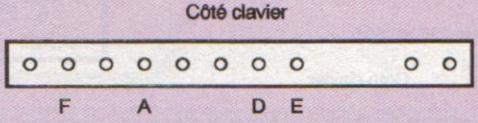
par le dispositif ici décrit. Dans la pratique, les détails des commandes à utiliser ne sont pas toujours disponibles et ne s'inventent certes pas.

En présence d'un Ericsson 388, par exemple, on devrait pouvoir entrer dans le mode "test" en tapant TP1: et recevoir OK en retour. Cela, pourvu que la ligne série soit réglée sur 8 bits de données, 1 bit de stop, pas de parité et, si possible, 115200 bauds. Parmi les commandes disponibles, EERE suivie d'une adresse permettrait d'aller lire en mémoire, et "EWR adresse valeur" d'y écrire. On murmure même que la remise à 00 d'adresses aux alentours de 1587 permettrait de déverrouiller les téléphones sans

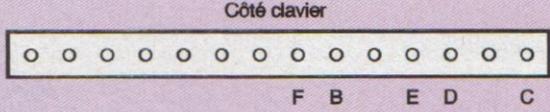
Alcatel one touch (easy, club, max, view) :



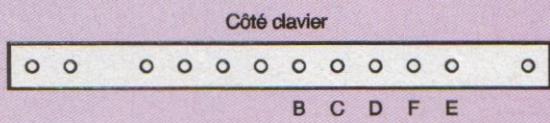
Alcatel one touch bibandes (easy, club, max) :



Ericsson 388 :

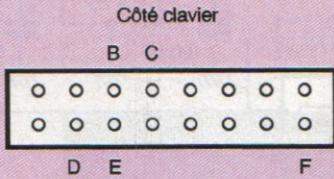


Ericsson 628, 688, 768, 788, 868, 888, A1018s, T10s, T18s, T28s :

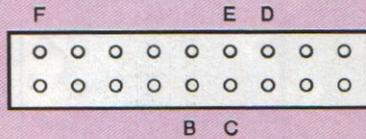


3 Exemple de raccordement

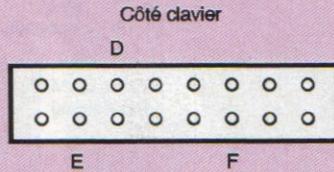
Panasonic G450, G520, G600 :



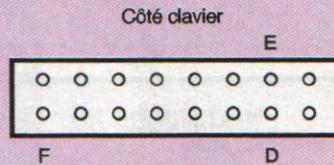
Panasonic GD30, GD50, GD70, GD90 :



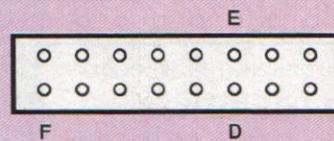
SAGEM RC712, 715, 725, 730, 750, 820, 612, 615 :



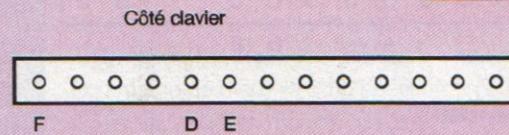
Siemens S6 et S8 :



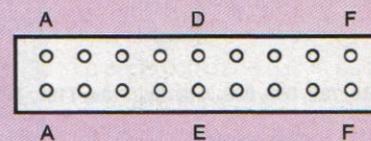
Siemens S10, S15, CMD - X2000 :



Siemens C25 et S25 :



Sony CMD - C5 :



4 Exemples de raccordement suivant le type de GSM

connaître le code normalement fourni par l'opérateur qui en a assuré la vente "subventionnée"...

Le plus souvent, on utilisera cependant ce montage avec des logiciels obtenus sur Internet et spécialisés pour telle ou telle opération (déverrouillage, notamment) sur tel ou tel modèle de portable. Cela, bien évidemment, après avoir pris connaissance des conditions dans lesquelles la chose est permise ou non.

Il faut savoir que des commandes hasardeuses envoyées à un téléphone sous garantie pourraient fort bien rendre celle-ci caduque et que certains contrats interdisent absolument tout déverrouillage par "contournement" du code ad-hoc. Encore faudrait-il que, de leur côté, les opérateurs "jouent le jeu"...

Pour notre part, nous nous bornerons prudemment à fournir un très simple programme permettant de tester le montage

	128	64	32	16	8	4	2	1
AL	B7	B6	B5	B4	B3	B2	B1	B0
000 :	110 bauds		Parité :		Bits de		Longueur :	
001 :	150 bauds		00 : sans (N)		stop :		10 : 7 bits	
011 :	600 bauds		01 : impaire (O)		0 : 1 bit		11 : 8 bits	
100 :	1200 bauds		11 : paire (E)		1 : 2 bits			
101 :	2400 bauds							
110 :	4800 bauds							
111 :	9600 bauds							

Exemple : 10011010 (9Ah ou 154d) (Minitel)

5 Signification de chaque bit



en "bouclant" provisoirement ses bornes D et E. Développé sous Delphi 3 (32 bits), il constitue une bonne occasion de montrer comment écrire une DLL toute simple pour accéder aux ports série. Grâce à TTYLIB.DLL, le programme de test proprement dit (RS232.EXE) est réduit à sa plus simple expression. Son rôle se borne à envoyer, en format 1200 bauds "Télételet", un caractère A ou B à chaque fois que l'on clique sur le bouton correspondant. Si la ligne série est bien bouclée sur elle-même, chaque caractère envoyé doit apparaître aussitôt dans la boîte située à droite des boutons, le port série étant lu à intervalles réguliers (définis par un composant "timer" de Delphi).

Quatre fonctions logicielles sont utilisées pour accéder au port série sur lequel est branché le montage :

- teletel(1) ouvre COM1: en mode Télételet
- sendcom (1,octet) envoie un octet sur COM1:
- getcom(1) lit un octet sur COM1: (à la volée)
- statcom(1) lit l'état (status) de COM1:

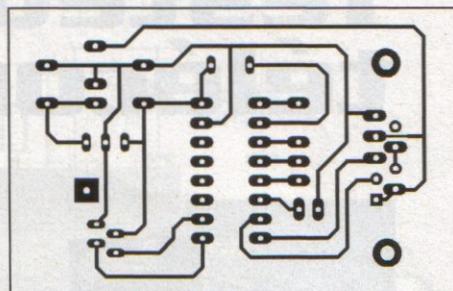
Bien entendu, il suffirait de remplacer 1 par 2 pour pouvoir utiliser COM2: au lieu de COM1:.

Delphi ne disposant pas, en standard, de fonctions de ce genre, il est commode de mettre à contribution les services de l'interruption 20 du BIOS. Cela nécessite toutefois l'écriture de courtes routines en langage machine, qu'il est commode de rassembler une bonne fois pour toutes dans une DLL qui pourra être réutilisée à volonté (et pas seulement sous Delphi, d'ailleurs !)

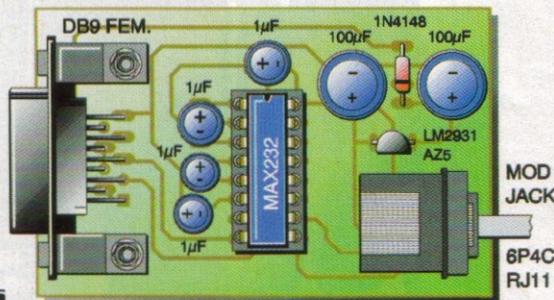
Créer une DLL suppose que le code source commence par "library" au lieu de "unit". A cette condition, la compilation produira un fichier .DLL au lieu de .EXE et les fonctions qu'il contient pourront être utilisées par n'importe quel programme écrit en conséquence. Encore faut-il que chaque fonction de la DLL soit affectée d'un numéro d'ordre ou "index" dont l'outil "aperçu rapide" de Windows permet d'ailleurs l'inspection (par exemple si on a égaré le code source !). Dans le code source de la DLL (ici TTYLIB.DPR), cela se fait très simplement dans la partie "exports" située en fin de listing.

Dans le code source du programme utilisant les services de la DLL (ici TTY.PAS), cela se fait en tête de la partie "implementation" où chaque fonction de la DLL que

6
Tracé
du circuit
imprimé



7
Implantation
des éléments



Nomenclature

- CI₁ : MAX232 (DIP)**
- CI₂ : LM2931 AZ5**
- D₁ : 1N4148**
- C₁ à C₄ : 1 µF/25V radial**
- C₅, C₆ : 100 µF/16V radial**
- Embase DB9 femelle coudée pour CI**
- Embase RJ11 (6p4c) simple**
- Cordon à fiche RJ11**
- Fiche GSM spécifique (voir texte)**
- Pile 9V et son clip (en option)**

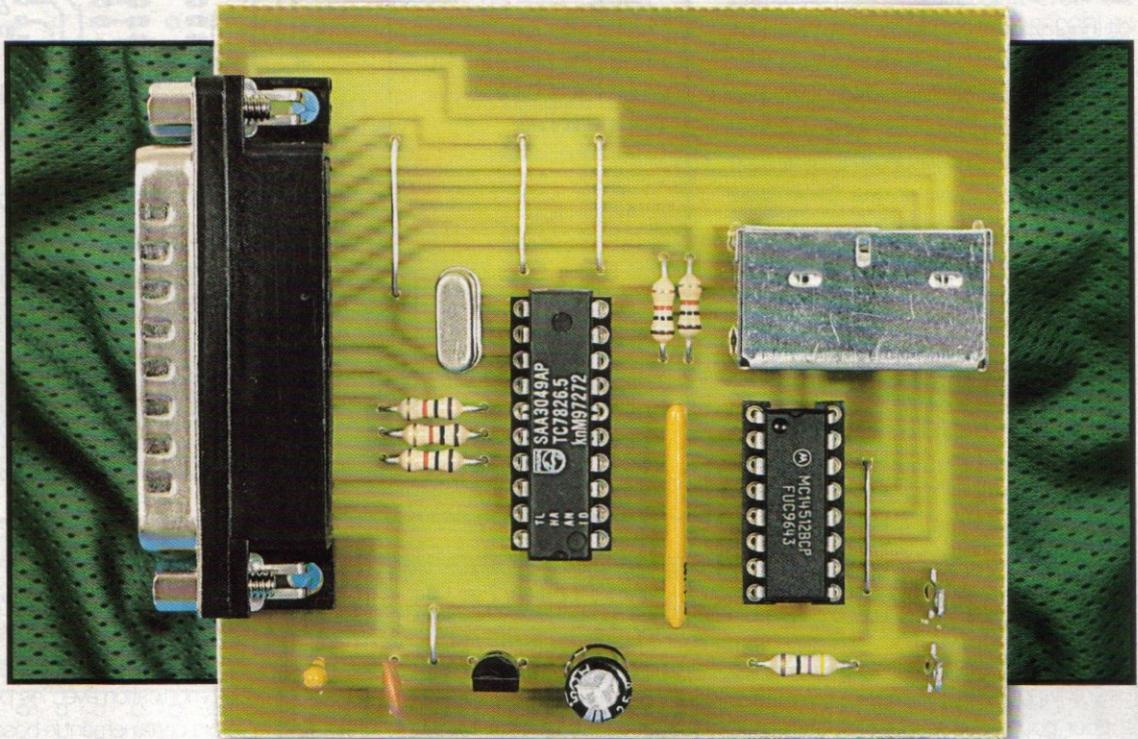
l'on souhaite utiliser est simplement déclarée au lieu d'être écrite de toutes pièces. Ceux de nos lecteurs qui souhaiteraient utiliser cette DLL pour d'autres applications de communication avec les ports série éprouveront certainement le besoin d'ouvrir ceux-ci avec des paramètres différents de ceux nous ayant servi d'exemple. Ils n'auront qu'à recompiler la DLL en modifiant à leur gré la valeur 154 dans la fonction "teletel", éventuellement renommée. La **figure 5** détaille, à cet effet, la signification de chaque bit de l'octet devant être chargé dans le registre AL du processeur et dont la valeur décimale est classiquement obtenue par simple addition des "poids" de tous les bits à 1.

P. GUEULLE



L'embase et le connecteur RJ11

Testeur de télécommandes IR



Les boîtiers de télécommande à infrarouges envahissent de plus en plus nos domiciles. Le confort auquel ils nous ont habitués fait que la panne de l'un d'eux provoque un agacement certain. Le petit montage décrit ci-après, connecté sur le port parallèle du PC, permet de s'assurer de la bonne santé de nos chères télécommandes.

Par exemple, en cas de difficulté de programmation d'une télécommande dite universelle, cette interface permettra de comparer les codes émis par le boîtier de commande original et celui de substitution. De plus, il est tentant de détourner l'utilisation de ces télécommandes à infrarouges afin de piloter divers montages personnels. Encore faut-il alors connaître les différents codes et adresses générés par ces émetteurs à infrarouges. C'est ici que ce montage prend tout son intérêt, en permettant d'afficher ces informations.

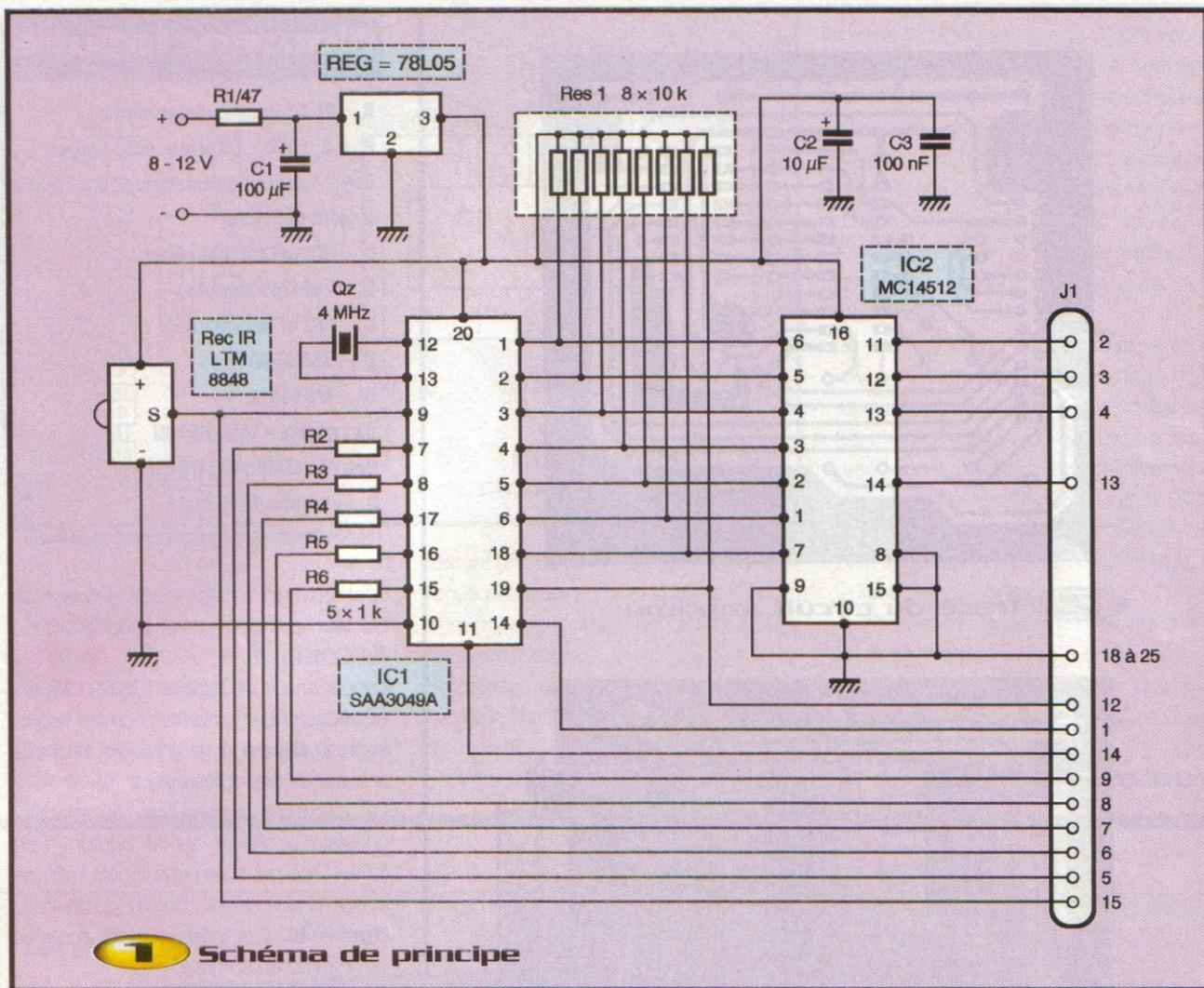
Le fonctionnement de ce montage est basé sur l'utilisation du circuit PHILIPS SAA3049A. Il s'agit d'un microcontrôleur programmé en décodeur des protocoles de codage RC5 et RECS80. On trouvera le premier de ces codes plus particulièrement sur les matériels de marque et sous marque PHILIPS (créateur de ce codage), tandis que le deuxième est plutôt utilisé sur les produits THOMSON et dérivés.

Les télécommandes utilisant un procédé de codage différent de ces deux normes ne seront, bien sûr, pas reconnues par notre testeur, pas plus que celles qui utilisent le code RC5 dit étendu (128 commandes), puisque le SAA3049A ignore ce type de commande. Le champ des modèles reconnus reste tout de même assez vaste.

Schéma électrique [figure 1]

L'alimentation, fournie par un bloc secteur, pourra être de faible puissance, puisque le montage ne consomme que quelques milliampères. Un régulateur de type 78L05 stabilise à 5V la tension d'entrée, filtrée par R_1/C_1 , tandis que C_2 et C_3 garantissent une tension d'utilisation parfaitement propre. Un récepteur à infrarouges, LITHEON LTM8848, transmet les trames reçues au SAA3049A sur son entrée en broche 9. Les broches d'adressage de ce circuit sont connectées aux sorties de données 5 à 9 du port paral-

lèle du PC, tandis que ses sorties de commandes sont envoyées sur les entrées d'un commutateur MOS à huit canaux de type 4512. La sortie en broche 14 de celui-ci rejoint l'entrée "Select" du port parallèle en broche 13. Les sorties de commandes du SAA3049A sont chacune "tirée" au +5V par des résistances de 10 k Ω , qui prennent la forme d'un réseau, notées Res*1 sur le schéma. En plus des sorties de commandes, le 4512 reçoit, en broche 7, l'information "Toggle bit", qui a pour but d'informer l'appareil télécommandé s'il s'agit ou non d'une nouvelle commande. Ce bit change donc d'état chaque fois qu'une nouvelle touche du boîtier de télécommande est sollicitée. Un quartz à 4 MHz anime l'oscillateur du SAA3049A. Par ailleurs, la sortie du récepteur à infrarouges est transmise sur l'entrée "Error" en broche 15 du port parallèle, ainsi que le signal de validité du code reçu, issu de la broche 19 du SAA3049A, sur l'entrée "Paper error", en broche 12 du port parallèle. Le fonctionnement du mon-



tage devient évident à la lecture du schéma : le programme surveille en permanence l'état de la ligne "Error" du port parallèle. Dès qu'une émission infrarouge est reçue, cette ligne passe au niveau bas. Le programme teste alors la ligne "Paper error". Si celle-ci est au niveau haut, c'est que le code reçu n'est pas valide et l'adresse suivante est sélectionnée. Les adresses sont ainsi sélectionnées successivement par les sorties 5 à 9 du port parallèle, jusqu'à ce que le signal de code valide soit actif. Dès que cette situation se présente, IC₂, commandé par les sorties 2 à 4 du port parallèle, sélectionne une à une chacune des sorties de IC₁ et transmet au PC les états correspondants par la ligne "Select" du port parallèle, afin de les afficher.

Réalisation et mise en œuvre [figure 2 & 3]

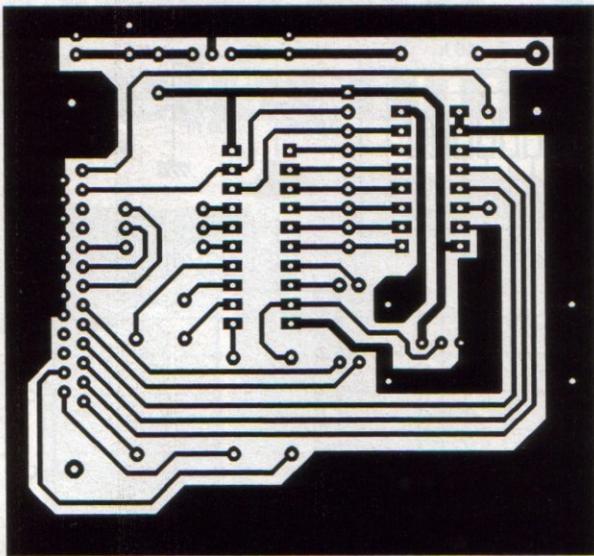
Le circuit imprimé simple face, de petite taille et dont le tracé est relativement aéré, ne devrait pas poser de problème particu-

lier. Les composants seront montés par ordre de taille, des plus petits aux plus grands, en commençant par les incontournables straps. Il faudra veiller au sens des composants polarisés ainsi qu'au réseau de résistances, dont le point de repère, qui correspond au point commun, doit être câblé du côté +5V.

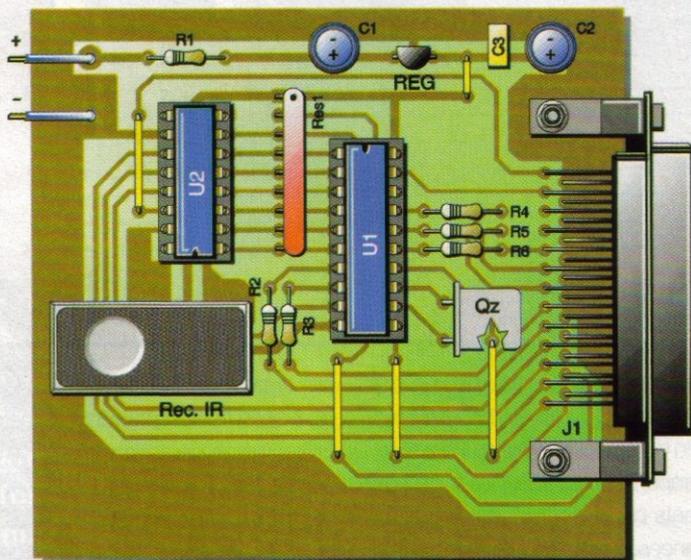
Il ne faudra pas oublier de souder les pattes de fixation du récepteur infrarouge car son boîtier métallique assure une partie des liaisons de masse du circuit imprimé.

Ce montage doit fonctionner dès la première mise sous tension. Toutefois, avant de le relier au PC à l'aide d'un câble 25 points mâle/femelle standard, il sera pu-





2 Tracé du circuit imprimé



3 Implantation des éléments

Nomenclature

- R₁ : 47 Ω (jaune, violet, noir)
- R₂ à R₆ : 1 kΩ (marron, noir, rouge)
- Res₁ : réseau de résistances 8 x 10 kΩ à point commun
- C₁ : 100 µF/25V chimique
- C₂ : 1 µF/35V tantale
- C₃ : 100 nF céramique
- IC₁ : SAA3049A
- IC₂ : MC14512
- Qz : quartz 4 MHz HC18U
- Rec IR : LTM8848 LITHEON
- J₁ : embase DB25 mâle

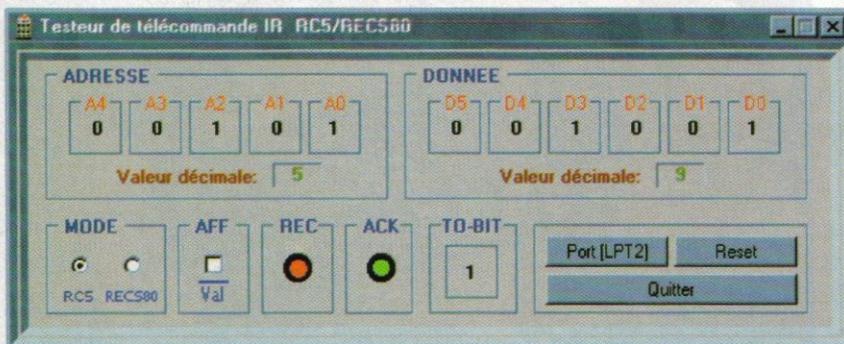
dent de vérifier la présence de la tension de 5V, par exemple sur la broche 20 du SAA3049A.

Lancer ensuite le logiciel TESTIR.EXE qui doit aussitôt se manifester, comme le présente la **figure 4**, aux valeurs affichées près qui seront toutes à zéro.

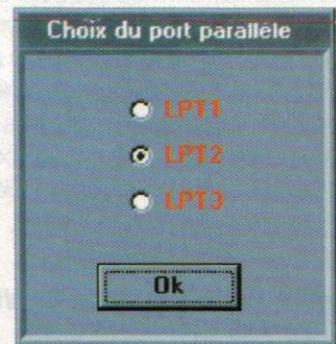
La première manœuvre à effectuer consiste à cliquer sur le bouton "Port [LPTx]", ce qui ouvre une fenêtre afin de choisir le port désiré comme le montre la **figure 5**. Les ports non disponibles seront affichés en grisé.

Le port choisi sera mémorisé dans le fichier TESTIR.CFG lorsque le programme sera quitté, ce qui évite d'avoir à refaire ce choix à chaque utilisation. Si ce fichier a été supprimé ou déplacé, c'est le port LPT1 qui sera choisi par défaut.

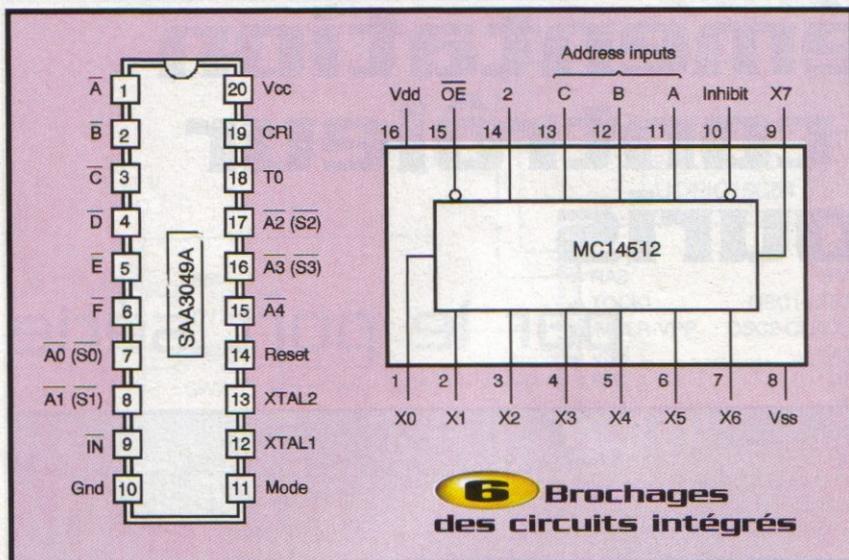
On peut alors se munir d'une télécommande à infrarouges et la diriger vers le montage. Appuyer sur une touche et la maintenir enfoncée. Le voyant "REC"



4 Lancement du logiciel



5 Choix du port parallèle



doit s'allumer en rouge tant que la touche est pressée et les adresses doivent défiler une à une. Dès que l'adresse valide est détectée, le comptage s'arrête, le voyant vert "ACK" s'allume et la valeur de la donnée s'affiche. Il convient alors de relâcher la touche : les voyants s'éteignent, mais l'adresse et la donnée

restent affichées. Si la touche est maintenue pressée, au bout d'une seconde, le comptage reprend son cycle à partir de l'adresse zéro.

Si l'adresse valide est zéro, l'affichage est, bien sûr, instantané. Si le compteur arrive à l'adresse 31 sans que le voyant "ACK" ne se soit allumé, c'est que la télécom-

mande n'utilise pas le code RC5. Cliquer alors sur le mode RECS80 et recommencer l'opération. Si toutes les adresses sont balayées à nouveau sans que le voyant "ACK" ne s'allume, alors le codage utilisé n'est pas reconnu par le SAA3049A et le seul diagnostic possible est que la télécommande émet quelque chose, ce qui met seulement hors de doute les diodes d'émission.

La case à cocher "AFF" permet simplement de basculer entre l'affichage correspondant aux niveaux électriques sur les broches du SAA3049A et celui correspondant à leurs états complémentaires. Le bouton "RESET" remet l'affichage à zéro et envoie une impulsion de RAZ sur le SAA3049A, en broche 14, par la ligne Strobe du port parallèle. Terminons en signalant qu'il faudra éviter d'utiliser l'interface en pleine lumière solaire, car celle-ci peut engendrer des perturbations qui risquent de produire des résultats erronés.

B. LEBRUN

744 pages, tout en couleurs

ENVOI CONTRE

Selectronic
L'UNIVERS ELECTRONIQUE

- MESURE - MATERIEL DE LABORATOIRE
- CONNEXIQUE - FILS ET CÂBLES
- ELECTRICITE - ECLAIRAGE
- OPTOELECTRONIQUE
- LIBRAIRIE TECHNIQUE
- ACTIF - PASSIF
- ROBOTIQUE
- OUTILLAGE
- ALARMES - SECURITE
- COFFRETS - BOUTONS
- COMMUTATEURS - RELAIS
- MATERIEL AUDIO ET VIDEO
- EQUIPEMENTS ET LOISIRS
- CIRCUITS IMPRIMES - CHIMIE
- DISSIPATEURS ET ACCESSOIRES

L'UNIVERS ELECTRONIQUE
Catalogue général
2001
Valeur 30 F

B.P. 513 - 59022 LILLE CEDEX - TEL. : 0 328 550 328 - FAX : 0 328 550 329 - www.selectronic.fr

30F (chèque ou timbres-poste)

Découvrez le **Nouveau**
Catalogue Général

Selectronic
L'UNIVERS ELECTRONIQUE

Toujours **PLUS** de Produits
et de Nouveautés !

Plus de 12.000 références !

Coupon à retourner à : **Selectronic BP 513 59022 LILLE Cedex - FAX : 0 328 550 329**

OUI, je désire recevoir le **"Catalogue Général 2001" Selectronic** à l'adresse suivante (ci-joint la somme de 30 F) :

Mr. / Mme : Tél :

N° : Rue :

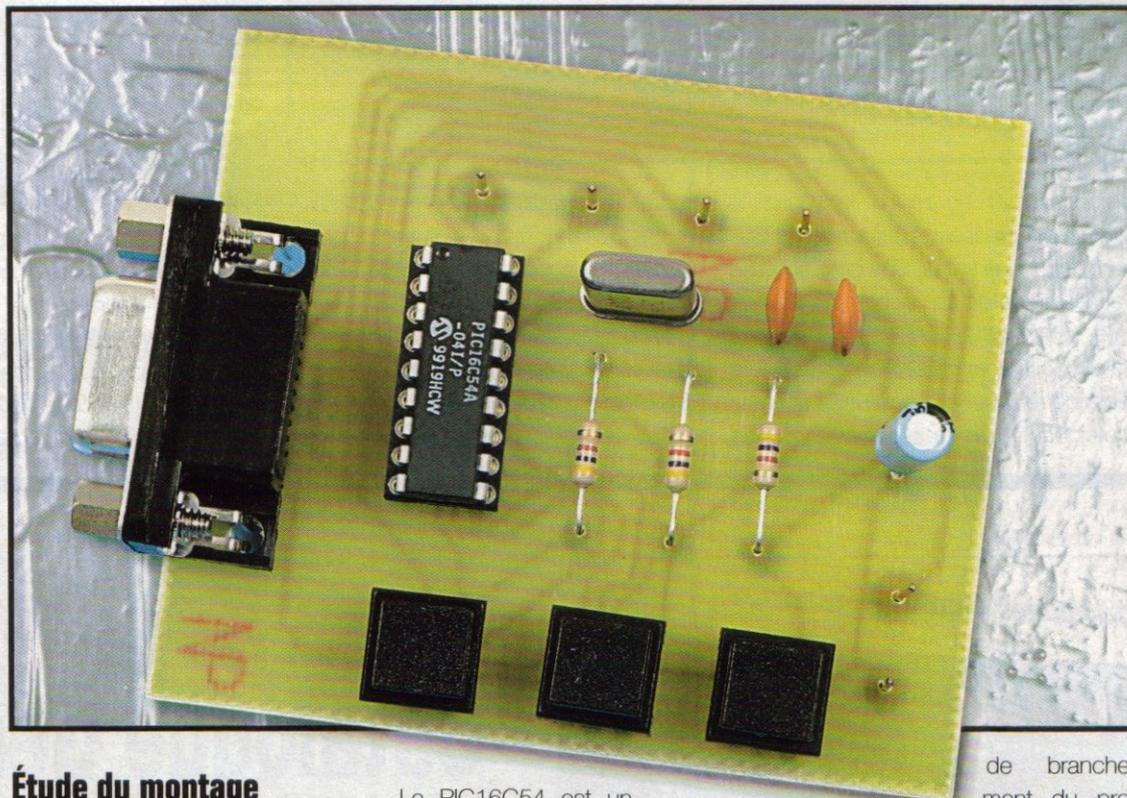
Ville : Code postal :

"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"

IPC

Implémentation d'un contrôleur de souris

par le port série



Étude du montage

Plusieurs sortes de souris peuvent être trouvées sur le marché, incluant les souris optiques, opto-mécaniques, ainsi qu'un accessoire qui s'en rapproche, à savoir les "trackballs".

Toutes ces souris sont interfacées en général à l'ordinateur par une carte d'interface dédiée ou encore par le port série RS232. Dans les deux cas, le mécanisme reste similaire.

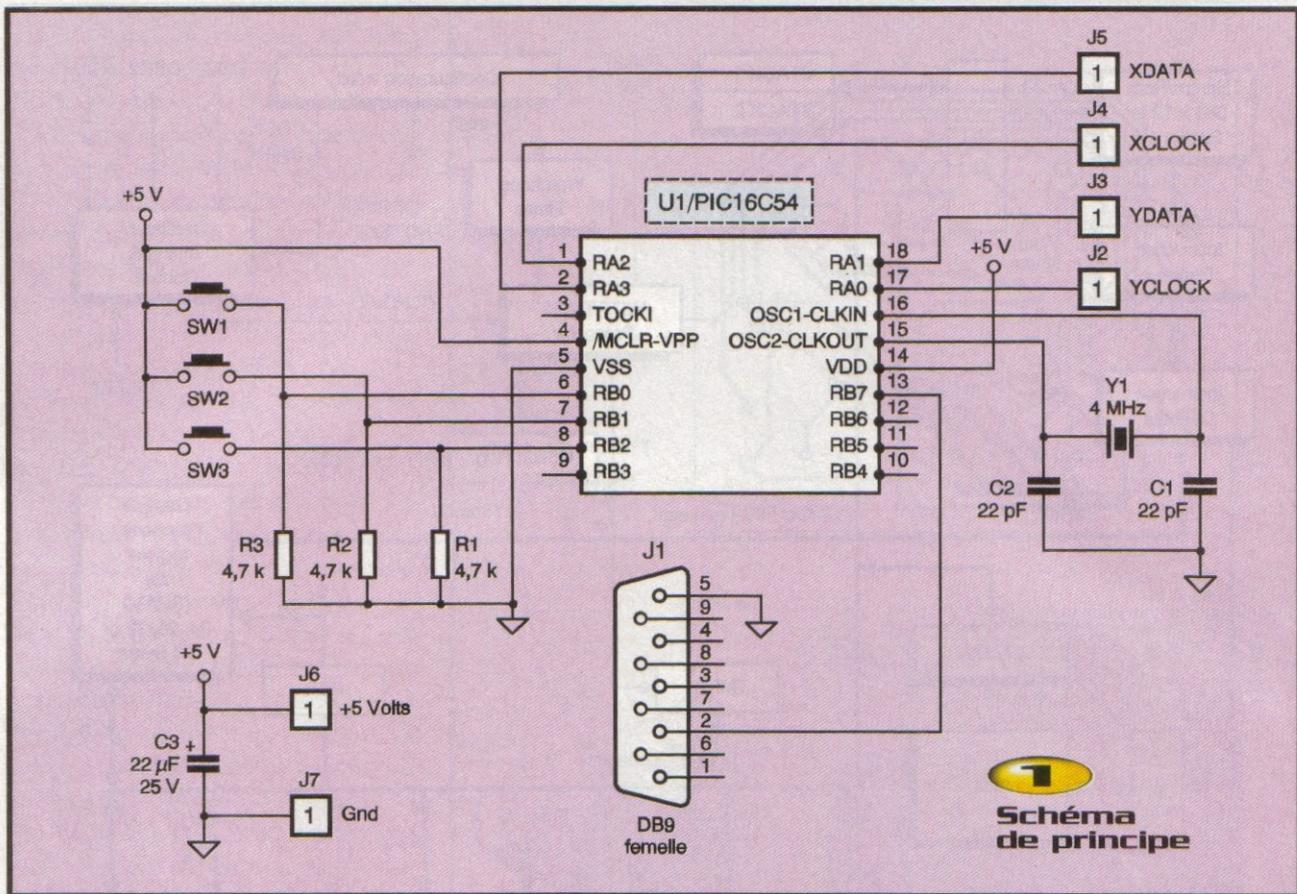
Notre carte d'application s'articule autour du microcontrôleur PIC16C54 de chez MICROCHIP, qui est la partie intelligente du montage pour dialoguer avec l'ordinateur. Par conséquent, les caractéristiques et les performances de notre souris dépendent avant tout du composant utilisé.

Le PIC16C54 est un microcontrôleur 8 bits très rapide en technologie CMOS. Nous allons décrire dans une première partie les caractéristiques principales de ce circuit intégré pour, ensuite, pouvoir expliquer dans une seconde partie le fonctionnement de notre montage ainsi que le déroulement du programme interne au composant.

Le PIC16C54 est un microcontrôleur 8 bits à très haute performance, peu cher, totalement statique et qui inclut sur sa puce une mémoire vive RAM de 25 octets pour les données et une mémoire effaçable PROM de 512 mots de 12 bits pour le programme. Ce composant utilise une architecture de type RISC avec seulement 33 instructions ; toutes ces instructions sont exécutables en un seul cycle sauf pour les instructions

de branchement du programme qui prennent 2 cycles. Le PIC16C54 délivre des performances d'un ordre de grandeur très supérieur à ses concurrents dans la même catégorie de prix. La largeur des instructions est sur 12 bits et sont fortement symétriques, ce qui en résulte un taux de compression dans un rapport de 2/1 par rapport aux autres microcontrôleurs 8 bits de sa classe. La facilité d'utilisation et de mémorisation des instructions réduit aussi le temps de développement de façon significative. Le PIC16C54 est équipé avec des caractéristiques spéciales qui réduisent le coût du système et les exigences de consommation de puissance. La remise à zéro du composant à la mise sous tension (POR : Power-On Reset) et la réinitialisation du comp-

La souris est devenue de plus en plus populaire dans le monde informatique en tant que point d'entrée dans un ordinateur. Il ne fait aucun doute que la demande pour accéder au maximum de ressources avec cet accessoire va en augmentant. Cet article décrit une manière originale de contrôler une souris par le port série.

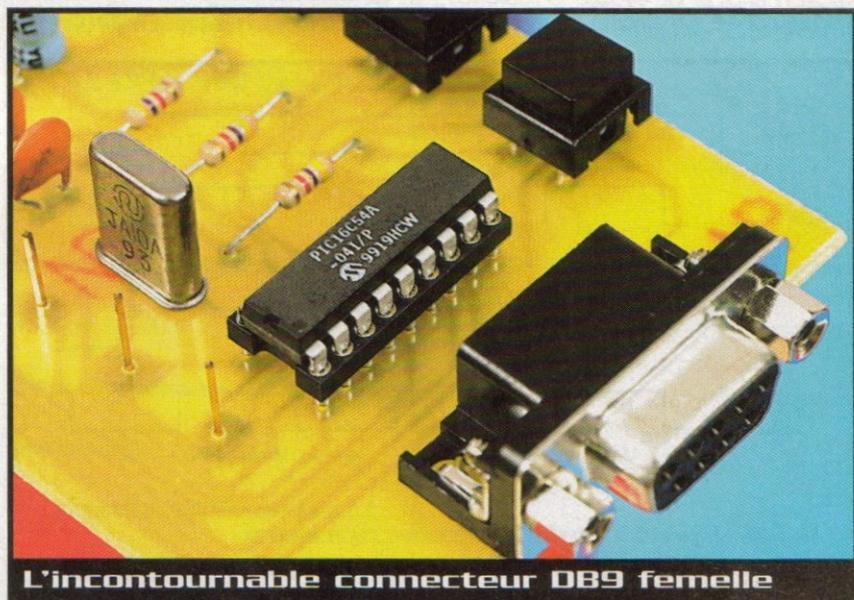


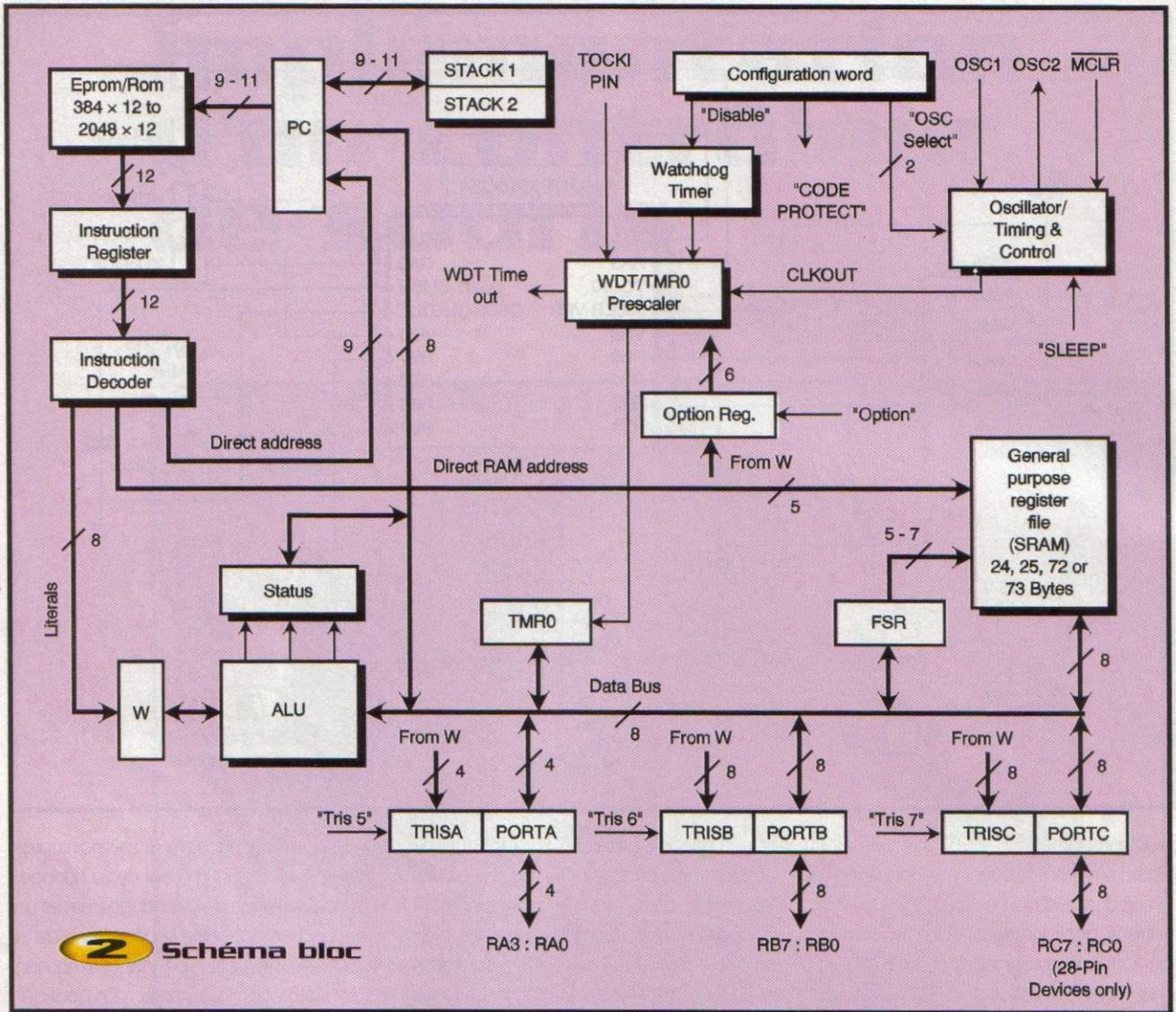
teur/chronométrateur (DRT : Device Rest Timer) éliminent le besoin de concevoir un circuit d'initialisation externe. Il y a le choix entre 4 configurations pour l'oscillateur, incluant l'oscillateur faible puissance (LP : Low Power) et l'oscillateur RC à bon prix. Le mode de mise en veille (SLEEP), le compteur de surveillance du bon déroulement du logiciel (Watchdog) et des caractéristiques de protection du code améliorent le coût du système global, la puissance et la fiabilité. Le PIC16C54 se présente sous la forme d'un boîtier DIP ; la version effaçable par des ultraviolets convient parfaitement pour le développement du code alors que la version programmable une seule fois est moins chère et est utilisée surtout pour la production en volume ou lorsque le code est au point. La haute performance du PIC16C54 peut être attribuée à un grand nombre de caractéristiques d'architecture que l'on trouve fréquemment dans les microprocesseurs à technologie RISC.

La **figure 1** représente la structure interne simplifiée de ce circuit intégré. Pour commencer, le PIC16C54 utilise une architecture de type Harvard dans laquelle le programme et les données sont accédés par des bus séparés, ce qui amé-

liore la bande-passante par rapport à une architecture traditionnelle de type Von Neumann dans laquelle programme et données sont accédés par le même bus. La séparation de la mémoire programme et données permet aux instructions d'avoir une taille différente que les mots de 8 bits utilisés pour les données. Le code opérationnel des instructions est d'une largeur de 12 bits ce qui rend ainsi possible d'avoir toutes les instructions codées sur

un seul mot. L'accès à la mémoire programme se fait en un seul cycle horloge. Un double étage de décodage permet un chevauchement entre le décodage de la prochaine instruction pendant l'exécution de l'instruction courante. En conséquence, toutes les 33 instructions sont exécutées en un seul cycle excepté pour les instructions de branchement. Le PIC16C54 peut directement ou indirectement adresser sa liste de registres ou la

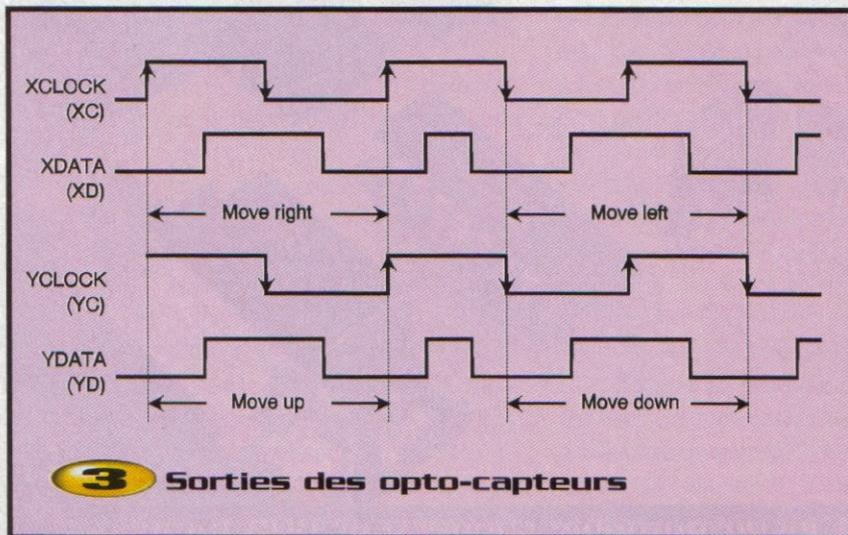




mémoire de donnée. Tous les registres à fonctions spéciales, incluant le compteur programme, sont implémentés dans la mémoire de donnée. Ce circuit intégré possède un jeu d'instruction orthogonal (encore appelé symétrique) qui rend ainsi

possible d'effectuer n'importe quelle opération sur n'importe quel registre en utilisant n'importe quel mode d'adressage. Cette nature symétrique et le manque de "situations spéciales optimales" rend la programmation du PIC16C54 simple mais

cependant efficace. De plus, l'apprentissage est réduit de façon significative. Ce microcontrôleur contient une unité arithmétique et logique (ALU) d'usage général sur 8 bits et des registres de travail ; il effectue des fonctions arithmétiques et booléennes entre les données qui se trouvent dans les registres de travail (W) et ses autres registres. L'ALU est aussi capable d'additionner, de soustraire et de réaliser des opérations de décalage et logique. Sans aucune autre mention particulière, les opérations arithmétiques sont en complément à deux par nature. Pour les instructions sur deux opérandes, un opérande est typiquement dans le registre de travail ; l'autre opérande est dans un autre de ses registres ou est une valeur immédiate constante. Pour les opérations avec un seul opérande, l'opérande est soit dans le registre de travail ou dans un autre registre. Le registre de travail W est sur 8 bits et est utilisé pour les opérations de

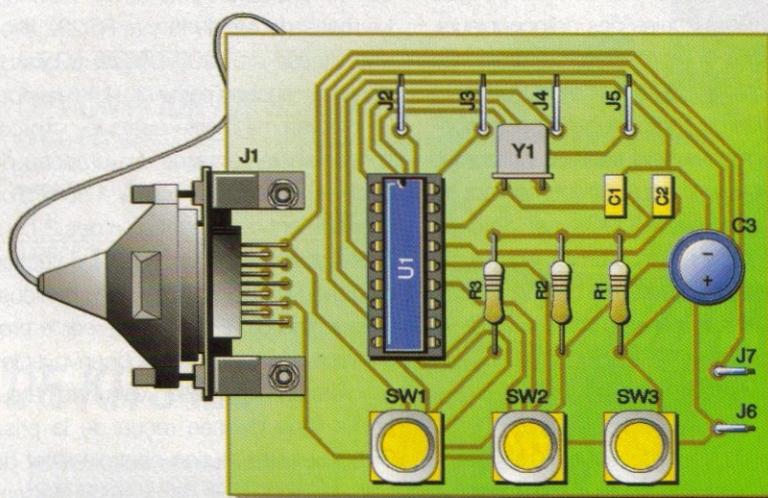
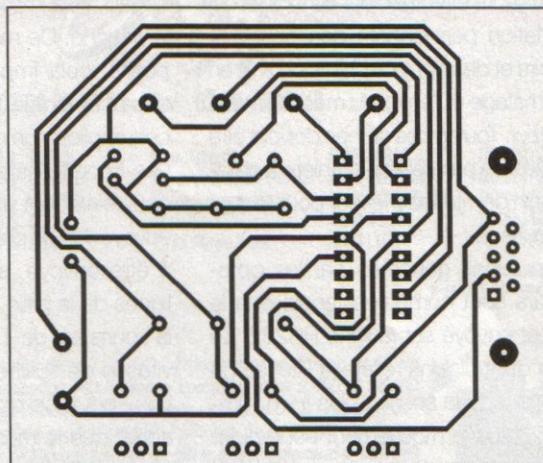


l'ALU ; ce n'est pas un registre adressable. Suivant l'instruction à exécuter, l'ALU peut affecter les valeurs des bits de la retenue C, la retenue numérique DC et du zéro Z dans le registre d'état (STATUS). Le PIC16C54 convient parfaitement dans la gamme d'applications pour l'automobile à haute vitesse de réponse, les dispositifs de contrôle des moteurs, les émetteurs/récepteurs à télécommande avec une faible puissance dissipée, les systèmes de pointage et le traitement des communications. La technologie en mémoire effaçable électriquement (EPROM) rend la personnalisation et les programmes d'application (codes transmis, vitesses des moteurs, fréquences reçues,...) extrêmement rapide et pratique. Le petit boîtier de 18 broches rend ce microcontrôleur parfait pour toutes les applications limitées en espace. Faible prix, faible consommation, haute performance, facilité d'utilisation et flexibilité de ses broches d'entrée-sortie rendent le PIC16C54 très polyvalent, même dans des champs d'applications où l'emploi d'un microcontrôleur n'avait pas été envisagé auparavant (comme par exemple les fonctions de chronométrage, d'impulsions codées en largeur PWM, de capture de comparaison, les communications série ou, encore, associé à un coprocesseur). Le lecteur désireux de connaître les différents périphériques internes ou les instructions ainsi que tous les modes d'adressage peuvent se référer au manuel du constructeur très bien documenté à ce sujet.

Étudions à présent le fonctionnement de notre montage ainsi que son programme associé. Le schéma de notre application est représenté à la **figure 2** et peut se diviser en plusieurs blocs fonctionnels : le microcontrôleur, la détection des 3 boutons de la souris, la détection du déplacement de la souris et le signal de génération pour le port RS-232.

Les 3 boutons-poussoirs sont connectés aux ports RBO à RB2 du PIC16C54. Lorsque le programme détecte une ouverture ou une fermeture de l'un de ces boutons-poussoirs, un message est établi et envoyé sur le port série. Les mouvements en X et Y sont mesurés en comptant les impulsions générées par les 2 optocoupleurs connectés sur XDATA et XCLOCK, d'une part, et sur YDATA et YCLOCK

4 Tracé du circuit imprimé



5 Implantation des éléments

Nomenclature

U₁ : PIC16C54 + support DIP18

R₁ à R₃ : 4,7 kΩ 1/4W (jaune, violet, rouge)

C₁, C₂ : 22 pF

C₃ : 22 µF/25V radial

Y₁ : quartz 4 MHz

SW₁ à SW₃ : boutons-poussoirs pour circuit imprimé

J₁ : connecteur DB9 femelle pour C.I.

J₂ à J₇ : connecteurs 1 point

d'autre part. De même, lorsqu'un mouvement en X ou Y est détecté, un message est établi et envoyé sur la ligne RS232. Les souris que l'on trouve dans le commerce utilisent toutes des techniques d'entrée série. Le protocole utilisé par Microsoft (avec 2 boutons) contient 3 octets de données qui décrivent l'état des 2 boutons-poussoirs et les mouvements relatifs des déplacements en X et en Y. Les souris qui comportent 3 boutons contiennent 5 octets de données : un octet décrit l'état des 3 boutons-poussoirs, 2 octets pour le mouvement relatif du déplacement en X et 2 octets pour le mouvement relatif du déplacement en Y. Deux lignes sont connectées à l'ordinateur par la prise série : le signal de masse et le signal donnée reçue (Received Data) ; ce dernier porte le message envoyé par la souris. Le travail le plus important est réa-

lisé par le programme du PIC16C54. La tâche majeure effectuée par ce logiciel est la scrutation permanente des boutons-poussoirs et des déplacements en X et en Y, le formatage et l'envoi du message vers l'ordinateur. Toutes ces tâches doivent être effectuées en parallèle de manière à obtenir un gain de vitesse meilleur pour le suivi de la souris.

Les impulsions générées par les optocoupleurs sont comptées tandis que le signal est envoyé sur la ligne RS232. Le nombre d'impulsions reflète la vitesse du mouvement de la souris : plus il y a d'impulsions, plus le mouvement est rapide. Les directions du mouvement sont déterminées par les derniers états et les états présents des sorties des optocoupleurs. La **figure 3** représente les sorties de ces derniers : XCLOCK et XDATA sont les sorties de l'optocoupleur correspondant au mouvement sur l'axe horizontal. XDATA est lu lorsqu'un front montant de XCLOCK est détecté. Pour un mouvement à droite, XDATA est soit au niveau bas sur le front montant de XCLOCK, soit au niveau haut sur le front descendant de YCLOCK. La détection du mouvement sur l'axe vertical suit la même logique avec YCLOCK et YDATA. Dans le tableau ci-dessous, X0 à X7 sont les données relatives au mouvement : Si X est positif, cela implique que la sou-

ris se déplace vers la droite ; si X est négatif, cela implique un mouvement vers la gauche. De manière similaire, si Y est positif, cela implique que la souris se déplace vers le bas et si Y est négatif, cela implique un mouvement vers le haut. Les impulsions générées par les optocoupleurs sont vérifiées avant chaque bit envoyé. Un bit prend 1/1200 de seconde à être envoyé, si la distance entre les lignes de la grille sur laquelle se déplace la souris est de 1 mm, ce qui donne une vitesse de suivi de 1200 mm/seconde. Les boutons-poussoirs sont scrutés après que le message soit envoyé et le temps utilisé pour envoyer le message sert comme temps pour les anti-rebonds. Le message est au format RS232 avec une vitesse de 1200 Bauds, 8 bits de données, aucune parité et 2 bits de stop. Le programme mérite quelques explications complémentaires. Le drapeau de déclenchement est mis à 1 lors d'un changement dans l'état d'un des 3 boutons-poussoirs ou lors de la détection d'un mouvement en X ou en Y. Le sous-programme BYTE est appelé par le programme principal 5 fois pour envoyer les 5 octets de l'information vers l'ordinateur par la ligne donnée reçue de la prise RS232 ; cette routine contrôle l'état de cette broche : si le drapeau de déclenchement est à 0, le signal de la broche

est toujours au niveau logique haut, si bien qu'aucun message n'est envoyé même si le sous-programme BYTE est appelé.

La routine BIT compte le nombre d'impulsions en provenance des sorties des optocoupleurs, détermine alors les directions des déplacements et génère un retard de 1/1200 seconde pour respecter la vitesse de transmission qui est de 1200 Bauds.

Réalisation pratique

Le câblage de notre circuit ne pose aucune difficulté particulière. Il est bien sûr recommandé de mettre le microcontrôleur sur support au cas où l'utilisateur désirerait modifier son programme interne. La **figure 4** représente le circuit imprimé côté soudures et la **figure 5** représente le circuit imprimé côté composants.

Le programme en code assembleur du microcontrôleur "souris.asm" est fourni sur le CD Rom joint à la revue.

Conclusion

Ce montage original peut répondre à de nombreux utilisateurs d'ordinateur qui désirent élargir l'utilisation de la souris classique à d'autres applications que le seul pointage de l'écran. Il convient d'utiliser une surface avec une grille qui possède des carrés de 1 mm pour le déplacement de la souris.

En ce qui concerne les optocoupleurs, le choix est laissé à l'utilisateur suivant son application souhaitée : des phototransistors standards peuvent très bien convenir.

M. LAURY

Position	7	6	5	4	3	2	1	0
Octet 1	1	0	0	0	0	L	M	R
Octet 2	X7	X6	X5	X4	X3	X2	X1	X0
Octet 3	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Octet 4	X7	X6	X5	X4	X3	X2	X1	X0
Octet 5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

SITE WEB: www.mdmagic.com (QR en téléchargement) CD ROM Démon des 3 logiciels TINA - QUICKROUTE et VINCENT: 65F.

QUICKROUTE, TINA & VINCENT Studio Draw!

Promo TINA: 710F
Sup manuel PAPIER: 50F

AMPLI DISCRET COMPLEMENTAIRE

AMPLIECO SMD

Edition de schémas.
Saisie automatique.
Routage automatique.
TOUT est compris!

Promo QR4 800Br: 1500 F

MDM électronique Simulateur TINA, DAO QUICKROUTE... Une DEMO vaut mieux qu'un long discours...

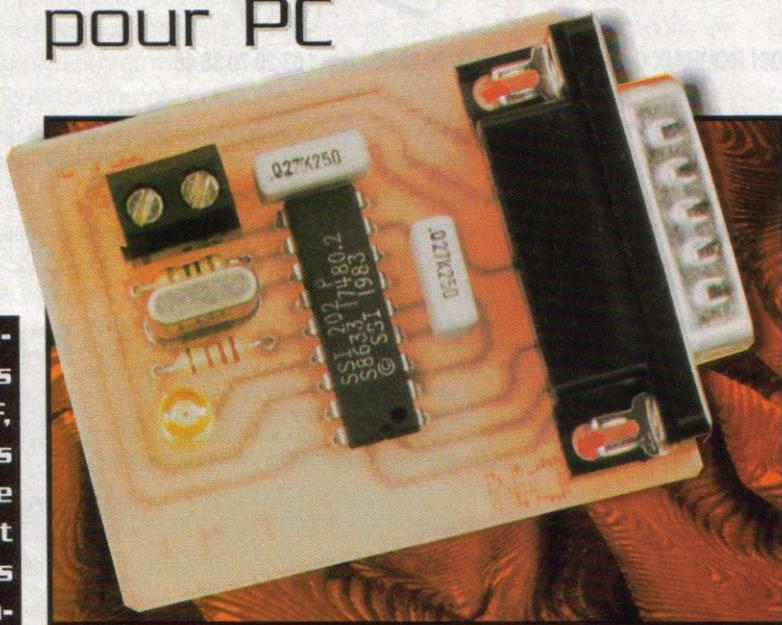
ZI de Carbon-blanc 33560 (près de BORDEAUX) TEL: (33) 0 556 06 37 89+ FAX: 0 556 38 08 05 WEB: http://www.mdmagic.com

Annance entièrement Réalisée avec VINCENT Studio Draw!

VALEUR SURE ! TOUT Y EST !

Emb+Port REC/AR: 35F

Un décodeur DTMF pour PC



Faut-il encore présenter les paires de tonalités DTMF, ces "fréquences vocales" presque universellement employées de nos jours pour la composition des numéros de téléphone ? Mais ce que l'on ne soupçonne pas toujours, c'est qu'un décodeur très simple peut aisément afficher les numéros (et codes de toutes sortes) ainsi composés ! Il suffit pour cela de capter, voire d'enregistrer sur cassette, ces sonorités bien particulières. Le présent montage permet l'utilisation, en direct ou en différé, d'un PC comme terminal de visualisation, apportant un confort certain grâce à un logiciel fonctionnant sous Windows.

Des tonalités normalisées

Successeur largement plus performant des cadrans rotatifs et des claviers "décimaux" fonctionnant par impulsions, le clavier téléphonique DTMF (Dual Tone Multi Frequency), "FV" (Fréquences Vocales) ou "Touch Tone" associe une "paire" de fréquences audibles à chaque chiffre ou signe qu'il est capable de transmettre. La normalisation internationale, très précise, est basée sur quatre fréquences "basses" (697, 770, 852, et 941 Hz) et quatre fréquences "hautes" (1209, 1336, 1477, et 1633 Hz). Il est possible de mélanger une fréquence basse et une fréquence haute de seize façons distinctes, ce qui permet de coder les dix chiffres de 0 à 9 plus six signes complémentaires, dont les indispensables étoile et dièse.

Absents des claviers les plus simples, qui ne gèrent pas la fréquence de 1633 Hz, les quatre codes "spéciaux" (baptisés A, B, C, et D) ne sont pratiquement jamais mis à contribution. En effet, la numérotation DTMF sert surtout à composer des numéros de téléphone, bien sûr, mais aussi des codes de cartes bancaires ou de cartes de télécommunications fixes ou mobiles. On les utilise pour passer des commandes,

recharger des comptes de pré-paiement, interroger des répondants à distance, paramétrer les services de son opérateur, naviguer dans des serveurs vocaux et, même, pour télécommander toutes sortes de dispositifs domotiques.

Naturellement, de très nombreux circuits intégrés ont été conçus tant pour générer que pour décoder les signaux DTMF. Parmi les fabricants les plus connus de décodeurs DTMF, on peut citer MITEL, TELTONE et TDK dont le SSI 75T202 est extrêmement populaire en vente au détail. C'est tout bonnement la nouvelle appellation du célèbre SSI 202, dont il existe d'ailleurs une "seconde source" parfaitement interchangeable chez RCA (le CD 22202).

Muni d'un bus de données à quatre lignes, le SSI 75T202 utilise un code hexadécimal modifié pour signaler les paires de tonalités qu'il reconnaît, à condition toutefois qu'on le configure en conséquence (broche 2 au niveau haut). Une sortie "data valid" (DV) passe au niveau haut dès qu'un code valide est détecté et y reste pendant toute la durée de celui-ci (en principe 100 ms ou davantage). Il est commode de lui faire piloter une LED de "diagnostic", confirmant le bon fonctionnement du décodeur en cas de doute.

Un décodeur "spécial PC"

Il serait difficile de faire plus simple que le schéma de la **figure 1** ! Un quartz de 3,58 ou 3,579 MHz (valeur très courante) et sa résistance associée sont les deux seuls composants externes qu'exige le SSI 75T202. Nous y avons ajouté un condensateur de découplage sur l'alimentation 5V et un condensateur de couplage sur l'entrée audio.

L'originalité du montage est de mettre à contribution la prise "joystick" du PC. Normalement destiné au branchement de manettes de jeu, ce connecteur DB15 est plus souvent libre que les ports série ou parallèle, du moins sur les PC affectés à des applications dites "sérieuses".

La communication avec le PC se fait donc par les quatre entrées tout ou rien dont est muni ce port (qui offre également quatre entrées "proportionnelles"), ce qui est tout juste suffisant pour les quatre lignes de données du SSI 75T202. Il faudra donc trouver un artifice logiciel pour se passer de la ligne de validation (DV). Un autre avantage du choix de cette prise est qu'un +5V de bonne qualité y est disponible pour alimenter le SSI 75T202 sans régulateur intermédiaire ni pile.

La réalisation pratique du montage suppose la gravure d'un circuit imprimé conforme au tracé de la **figure 2**, la mise en place des composants devant se faire selon le plan de la **figure 3**. On obtiendra ainsi un module compact, dont la largeur n'excède pas celle d'une fiche DB15 équipée d'un capot. Cela permet un enfilage direct à l'arrière du PC, le raccordement audio se faisant alors par un simple fil blindé dont la longueur n'a rien de critique (plusieurs mètres s'il le faut). Tel qu'il est monté (c'est à dire sans préamplificateur ni atténuateur en entrée), le décodeur fonctionne correctement dans une plage de niveaux d'entrée de 30 dB (gros-

sièrement entre 20 et 600mV RMS). Une sortie "casque" de magnétophone fera parfaitement l'affaire, la commande de volume de l'appareil étant réglée sur un niveau normal d'écoute, cela à défaut d'une prise "ligne", toujours préférable.

Il importe toutefois que le magnétophone soit en bon état général, car la précision des tonalités DTMF s'accommoderait mal de fluctuations de vitesse excédant 2% (ce qui est déjà sévère...).

Il faut également que le niveau des fréquences "hautes" et "basses" ne diffère pas de plus de 10 dB (neutraliser toute correction de tonalité et autres "Dolby" en enregistrant comme en lecture !).

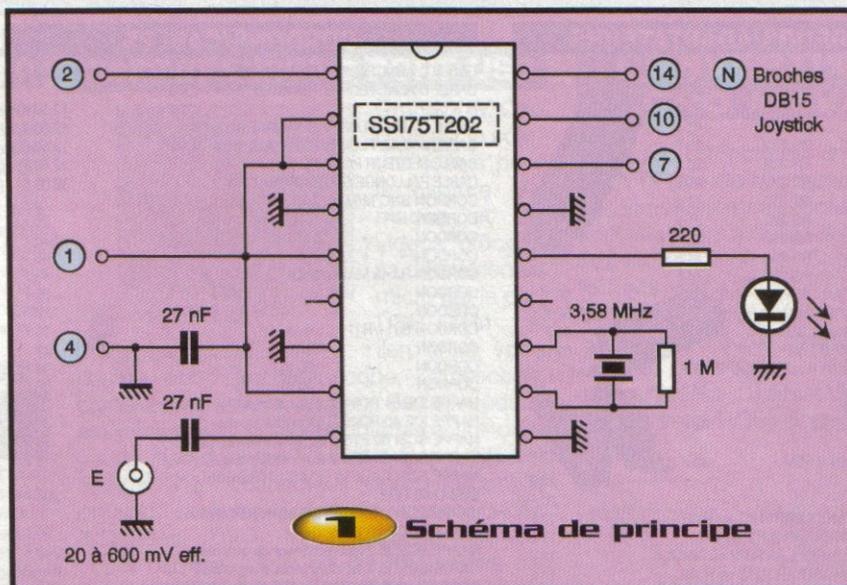
On préférera, par conséquent, l'enregistrement par "ventouse" magnétique ou par couplage galvanique, bien que la captation par un micro (par exemple à partir du son émis par le haut-parleur d'un poste "mains libres" ou d'un modem) puisse donner des résultats assez étonnants.

Le SSI 75T202 supporte d'ailleurs assez bien que de la parole ou du bruit ambiant soit superposé aux signaux utiles : le codage DTMF a été spécialement conçu pour cela !

Le logiciel de décodage

Le développement sous Delphi (version 3) d'un logiciel permettant l'exploitation confortable du montage en environnement Windows a posé un double problème :

- la lecture du port joystick (adresse 513) que Delphi 32 bits ne sait pas gérer directement ;
- la détection fiable des pauses interchiffres, même en cas de composition



ultrarapide par un numéroteur automatique (téléphone à mémoire, modem, etc.)

La solution adoptée est le recours à une courte routine en assembleur (asm) et aussi à un composant "timer" de Delphi. Ce dernier permet de mettre en œuvre une technique dite de "polling", c'est à dire d'interrogation du port à intervalles réguliers.

Compte tenu de la vitesse des processeurs modernes et de la simplicité de la tâche à exécuter, il n'est guère pénalisant de régler le temporisateur sur une valeur n'excédant pas 2 à 5 ms, sachant tout de même qu'une tonalité DTMF valide doit durer au minimum 100 ms.

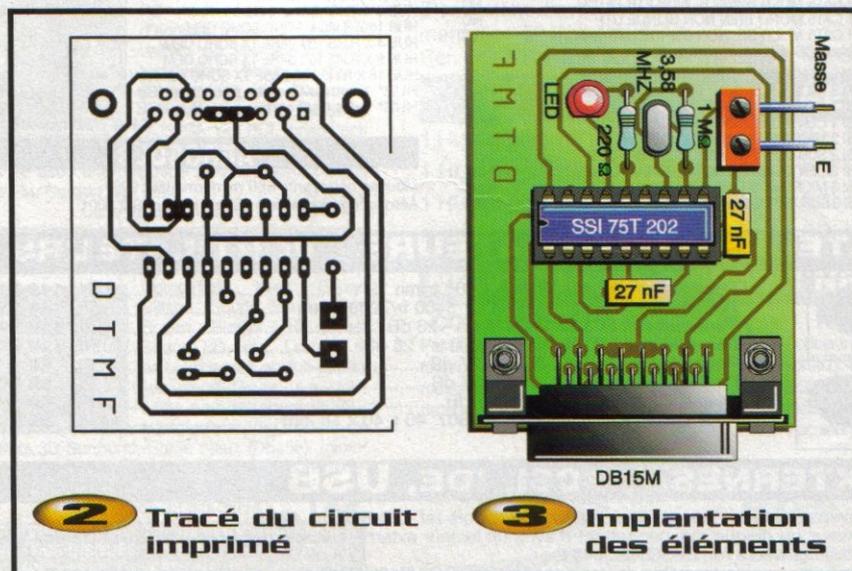
A condition de renoncer au décodage des quatre combinaisons "A, B, C, D", on peut considérer que le mot 0000 en sortie du SSI 75T202 correspond à une pause interchiffres. Ramené au port joystick, cela équivaut à 00001111 (0Fh ou 15 en décimal)

puisque les quatre bits de poids faible du port 513 restent à 1 lorsqu'on n'active pas les entrées proportionnelles. Il suffit dès lors de construire une table de correspondance entre états du port et chiffres composés, pour pouvoir afficher dans une ComboBox tout ce qui est détecté.

Pour plus de confort, nous avons prévu un bouton permettant d'effacer l'affichage et un autre permettant d'insérer manuellement un espace (en fait un caractère "underline") pour séparer à volonté un groupe de chiffres du suivant.

Notons, toutefois, que ces opérations pourraient tout aussi bien se faire par sélection à l'aide de la souris, selon la procédure commune à toutes les applications Windows ou bien être automatisées de diverses façons au moyen de "timers" supplémentaires.

P. GUEULLE



Nomenclature

IC₁ : SSI 75T202 (TDK)

D₁ : LED

R*1 : 1 MΩ (marron, noir, vert)

R*2 : 220 Ω (rouge, rouge, marron)

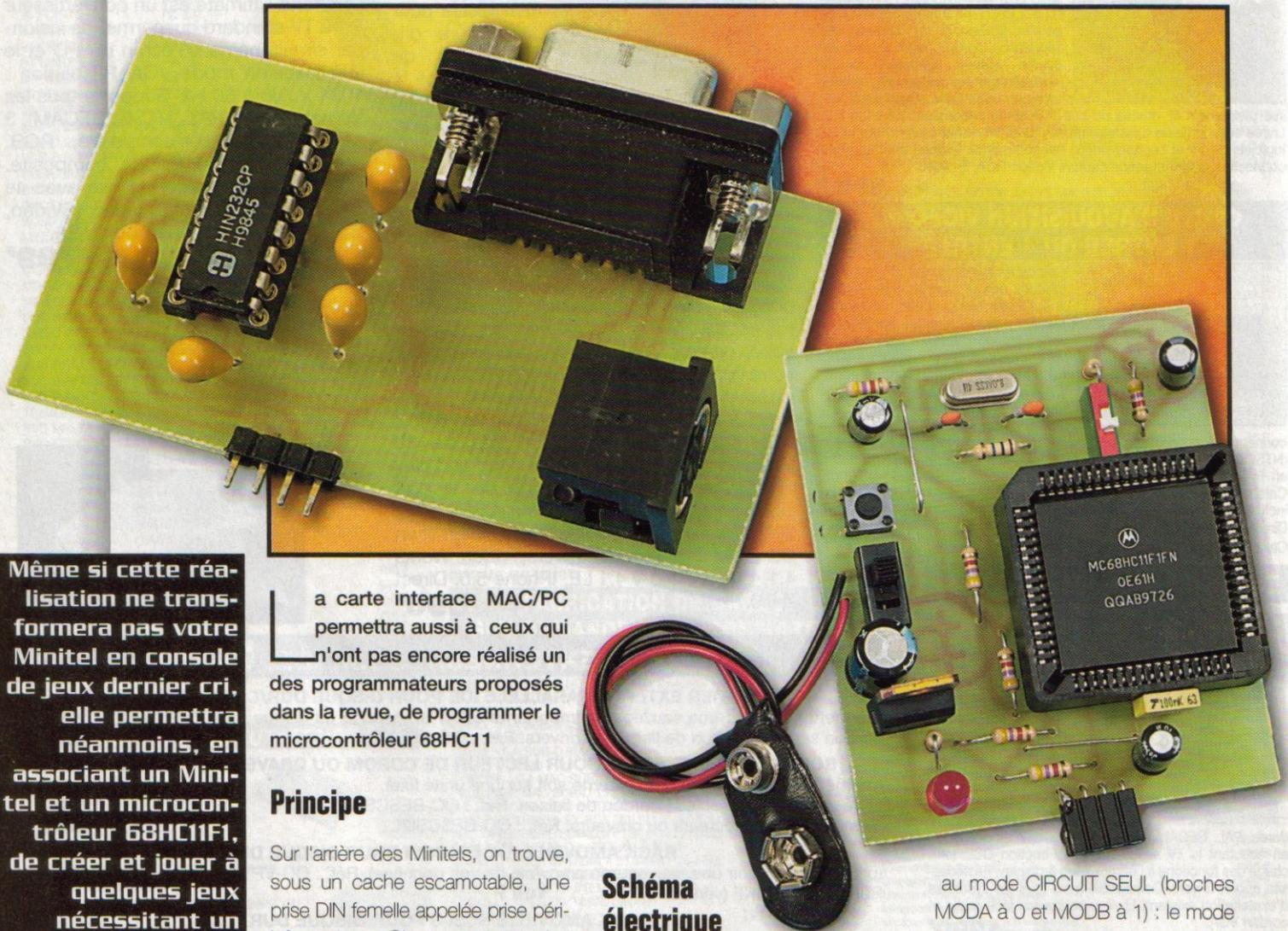
C*1, C*2 : 27 nF

1 embase DB15 mâle soudée pour C.I.

1 bornier 5,08mm à 2 circuits

1 quartz 3,579 MHz

Jeux sur Minitel®



Même si cette réalisation ne transformera pas votre Minitel en console de jeux dernier cri, elle permettra néanmoins, en associant un Minitel et un microcontrôleur 68HC11F1, de créer et jouer à quelques jeux nécessitant un écran et un clavier (c'est le rôle du Minitel) et un calculateur (c'est le rôle du microcontrôleur). Il est même possible, en ajoutant une petite carte interface de jouer sans Minitel en le remplaçant par un émulateur de Minitel comme HYPER-TERMINAL, petit logiciel de communications livré avec WINDOWS sur PC ou CLARIS WORKS sur MAC.

La carte interface MAC/PC permettra aussi à ceux qui n'ont pas encore réalisé un des programmeurs proposés dans la revue, de programmer le microcontrôleur 68HC11

Principe

Sur l'arrière des Minitels, on trouve, sous un cache escamotable, une prise DIN femelle appelée prise péri-informatique. C'est par cette prise que transiteront, en série, les données échangées avec notre carte : de même que tout appui sur une touche du clavier envoie le code de la touche appuyée sur la broche 3 (TX) de cette prise, toute réception de données sur la broche 1 (RX) est exploitée par le Minitel pour gérer la visualisation sur l'écran comme décrit plus loin dans le chapitre de présentation des Minitels.

Si on préfère utiliser un émulateur, le principe reste le même mais le montage aura besoin d'un circuit additionnel : il faudra en effet relier notre carte à un des ports série du PC (COM1 ou COM2) au travers d'une interface qui adaptera les signaux électriques pour les rendre compatibles avec ceux des PC.

Schéma électrique

Le schéma électrique de la **figure 1** fait apparaître la relative simplicité du montage. C'est bien souvent le cas lors de l'utilisation d'un microcontrôleur : l'essentiel est contenu dans le logiciel implanté dans sa mémoire programme. Le schéma de la **figure 2** décrit la connexion de la carte avec un Minitel, le schéma de la **figure 3**, la connexion au travers d'un circuit d'interface, en cas d'utilisation d'un émulateur.

Le 68HC11F1 est, bien sûr, le cœur de notre console. Il est cadencé par le quartz Q de 8 MHz accompagné des 2 condensateurs C_5 et C_6 et de la résistance R_9 . L'interrupteur à glissière K_2 , relié à la broche MODB, permet le passage du mode BOOTSTRAP (broches MODA et MODB à 0)

au mode CIRCUIT SEUL (broches MODA à 0 et MODB à 1) : le mode BOOTSTRAP sera utilisé pour la programmation et le mode CIRCUIT SEUL sera le mode de fonctionnement courant. La résistance R_8 , le condensateur C_4 et le bouton poussoir K_3 constituent le circuit de RESET ou d'initialisation. Quelques résistances de rappels stabilisent des niveaux hauts sur les broches PDO, PD1, XIRQ et IRQ, évitant des niveaux aléatoires et éventuellement perturbateurs sur ces broches.

Le circuit est alimenté par une tension continue de 5V fournie par une pile de 9V et un régulateur 7805. Une LED permet de vérifier la présence de cette tension sur le montage.

Le schéma de la figure 2 décrit la connexion directe entre la prise péri-informatique et notre carte : la broche réception PD1 de la carte est reliée à

la broche émission 3 du Minitel, la broche émission PDO de la carte est reliée à la broche réception 1 du Minitel.

Le schéma de la figure 3 décrit l'interface nécessaire entre la carte et un ordinateur MAC ou PC dans le cas de l'utilisation d'un émulateur et pour la programmation. L'adaptation des niveaux électriques est confiée au circuit MAX232 accompagné des condensateurs C_8 à C_{12} .

L'essentiel sur les Minitels

Ce montage et les programmes correspondants ne se comprennent qu'avec un minimum de connaissance sur les Minitels.

Codage des caractères dans un Minitel

Les caractères alphabétiques majuscules et minuscules, sans accentuation, les chiffres et certains symboles de ponctuations sont codés en ASCII (exemple \$65 correspond à e). Pour des raisons de simplicité, seuls ces caractères seront utilisés dans les 2 programmes MASTER68 et PUISSANCE68.

Les caractères accentués sont désignés au moyen d'une combinaison de 3 codes :

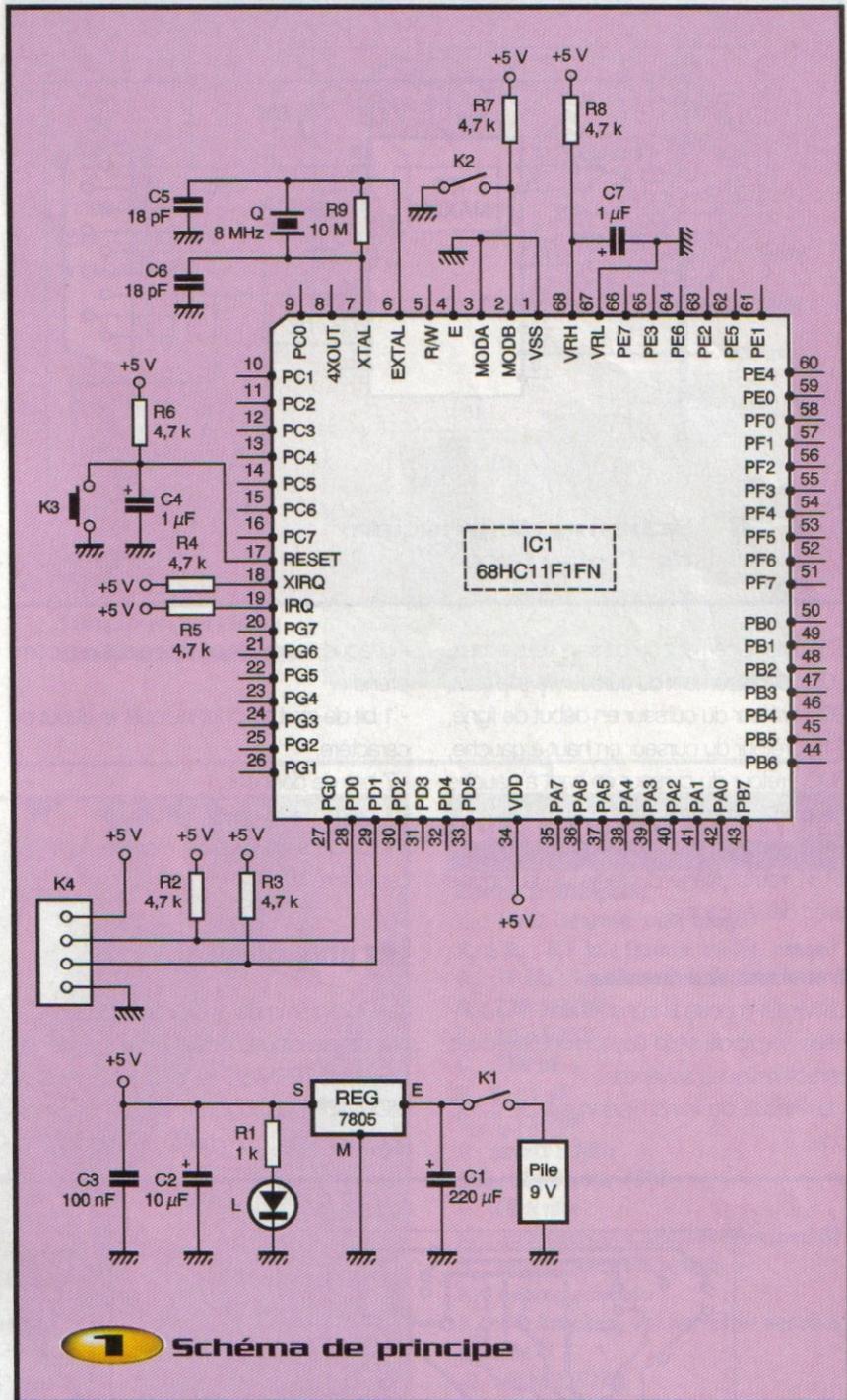
- un code de commande : \$19 (25)
 - un code représentant l'accent : \$42 accent aigu, \$48 tréma, \$43 accent circonflexe, \$41 accent grave
 - un code représentant la lettre sans accent
- Exemple \$19 \$42 \$65 correspond à la lettre é.

Il est aussi possible de créer des lettres spéciales et des caractères semi-graphiques mais ce thème sort du sujet de cet article.

Codages des attributs de visualisation

La couleur des caractères, du fond et d'autres possibilités comme le clignotement ou la taille peuvent être définis par l'utilisateur. Comme dans le cas des lettres accentuées, il faut superposer plusieurs codes, ici 2 :

- un code de commande : \$1B (27)
- le code de la fonction :
- couleur caractère : noir \$40, rouge \$41, vert \$42, jaune \$43, bleu \$44, magenta \$45, cyan \$46, blanc \$47
- couleur fond : noir \$50, rouge \$51, vert \$52, jaune \$53, bleu \$54, magenta \$55,



1 Schéma de principe



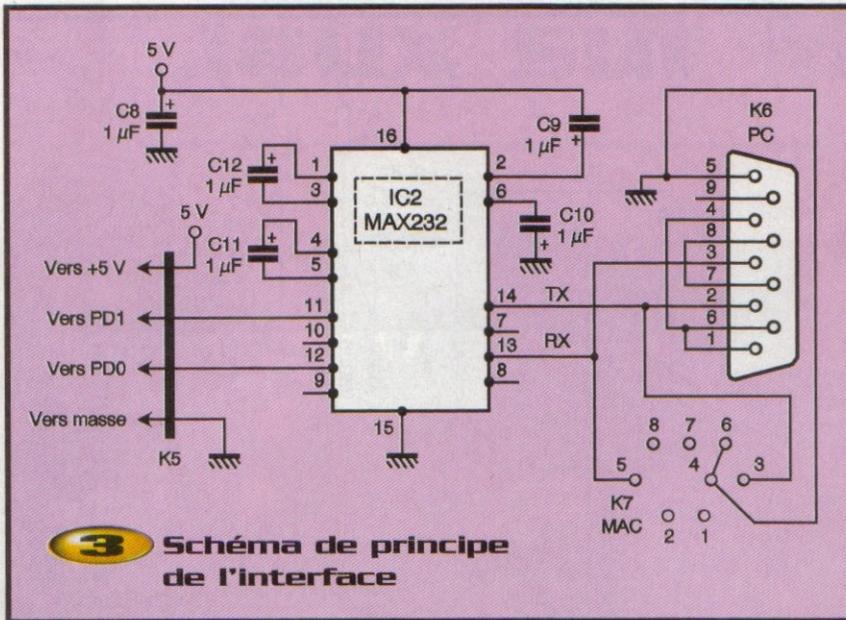
2 Liaison de la carte avec le minitel

- cyan \$56, blanc \$57
- clignotement \$48, fixe \$49
- grandeur normale \$4C, double hauteur \$4D, double largeur \$4E, double grandeur \$4F
- fond normal \$5C, inversion de fond \$5D

Fonctions de mise en page

Le curseur peut être positionné à un endroit déterminé de l'écran.

- \$08 : déplacement du curseur vers la gauche,
- \$09 : déplacement du curseur vers la droite,



3 Schéma de principe de l'interface

\$0A : déplacement du curseur vers le bas,
 \$0B : déplacement du curseur vers le haut,
 \$0D : retour du curseur en début de ligne,
 \$1E : retour du curseur en haut à gauche,
 \$0C : retour du curseur en haut à gauche avec effacement de l'écran,
 \$1F : séparateur de sous-article,
 \$18 : remplissage du reste de la rangée avec des espaces.

- chaque transmission de caractères comprend :
 - 1 bit de start (0) pour indiquer le début du caractère,
 - 7 bits de données,
 - 1 bit de parité paire,
 - 1 bit de stop (1) pour marquer la fin du caractère.

Transferts des données

Le Minitel impose la transmission des données en mode série asynchrone avec les caractéristiques suivantes :
 - la vitesse de transmission est de 1200 bds,

Les programmes

Les programmes proposés ici sont 2 grands classiques des jeux : le premier du type MASTERMIND et le second du type MORPION. Le fonctionnement des programmes étant commenté dans leur listing,

seuls quelques sous-programmes importants sont décrits ci-dessous.

Sous programmes généraux d'utilisation de la liaison série

Pour le 68HC11, chaque transmission de caractères comprend :

- 1 bit de start (0) pour indiquer le début du caractère,
- 8 bits de données,
- 1 bit de stop (1) pour marquer la fin du caractère.

On devine que, pour rendre le 68HC11 et le Minitel compatibles, il faudra que nos programmes transforment le 8ème bit des données envoyé par le HC11 en bit de parité paire et qu'à la réception des données par le HC11, on ne tienne compte que des 7 premiers bits.

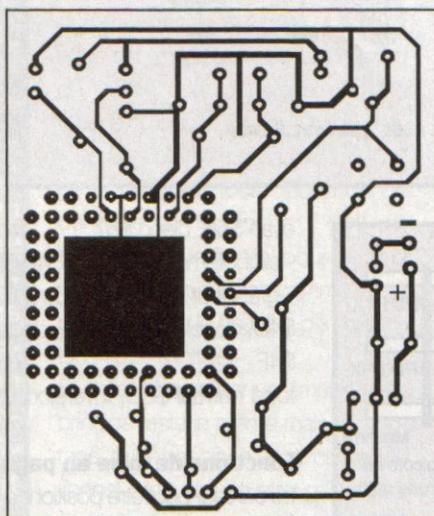
Initialisation de la liaison

SCSR	EQU \$2E
SCCR2	EQU \$2D
SCDR	EQU \$2F
BAUD	EQU \$2B
BASEREG	EQU \$1000

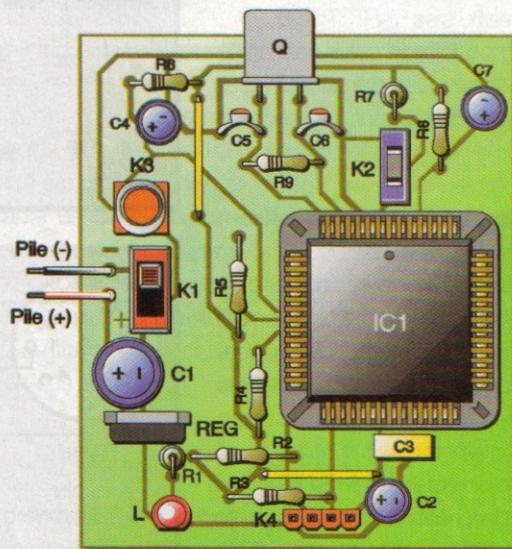
*initialisation générale

*vitesse transmission 1200 bds

```
ORG $FE00
LDS #$00C0
LDX #BASEREG
LDAA #00110011
```



4 Tracé du circuit imprimé



5 Implantation des éléments

STAA BAUD,X
BSET SCCR2,X %00001100

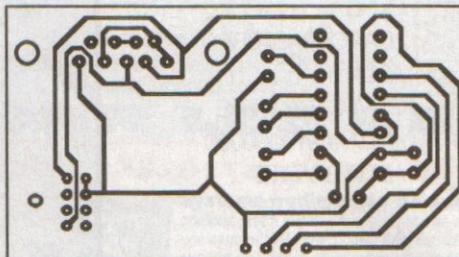
Transmission vers le HC11 de l'appui sur une touche du Minitel

En mode local (non connecté), la touche appuyée est automatiquement affichée sur le Minitel. Par contre, en mode connecté, mode sélectionné automatiquement par certains émulateurs comme HYPERTERMINAL, la touche appuyée n'est plus affichée : c'est donc au programme de renvoyer le code de la touche appuyée au Minitel. C'est le but des 2 lignes avec l'astérisque.

TOUCHE LDAA SCSR,X
ANDA #S20
BEQ TOUCHE
LDAA SCDR,X
ANDA #%01111111
JSR ENVOI
* à enlever avec un vrai

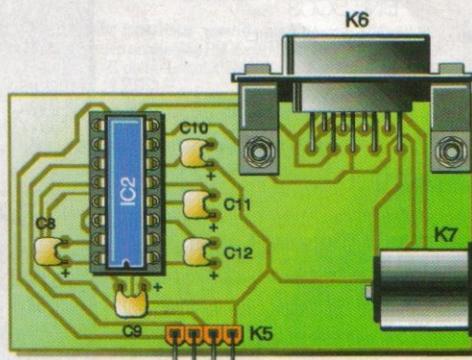
6

Tracé du circuit imprimé



7

Implantation des éléments



Nomenclature

Carte principale

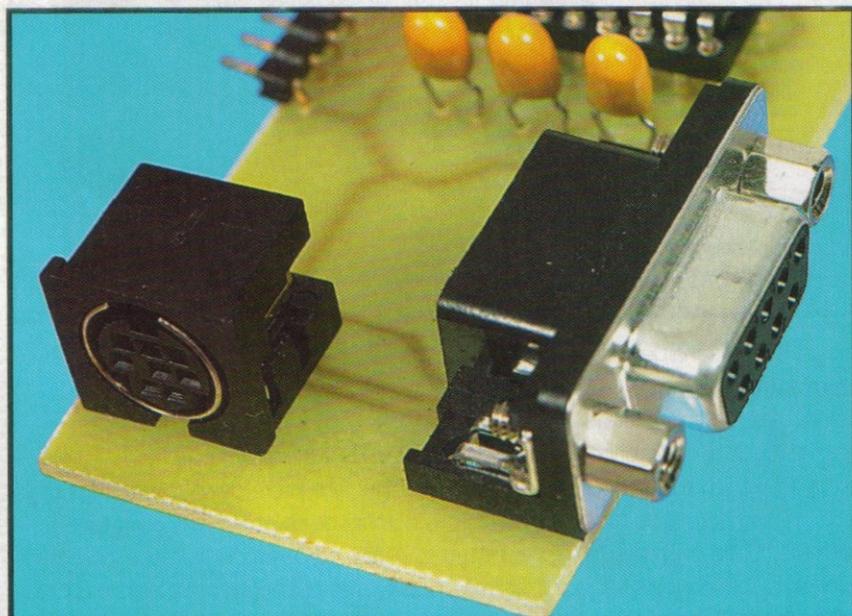
- R₁ : 1 kΩ (marron, noir, rouge)
- R₂ à R₆ : 4,7 kΩ (jaune, violet, rouge)
- R₈ : 10 MΩ (marron, noir, bleu)
- C₁ : 220 µF/25V
- C₂ : 10 µF/25V
- C₃ : 100 nF
- C₄, C₇ : 1 µF
- C₅, C₆ : 18 pF
- Q : quartz 8 MHz
- REG : régulateur 7805
- L : LED 5 mm
- K₁ : interrupteur d'alimentation pour CI
- K₂ : interrupteur DIL simple
- K₃ : bouton poussoir
- K₄ : 4 broches de barrette sécable femelle
- IC₁ : 68HC11F1FN
- 1 support PLCC 68 broches
- 1 coupleur de pile 9V
- 2 straps

Carte interface MAC/PC

- K₅ : 4 broches de barrette sécable mâle soudée
- K₆ : connecteur DB9 pour PC
- K₇ : connecteur mini-DIN 8 pour MAC
- IC₂ : MAX232
- C₈ à C₁₂ : 1 µF
- 1 support 16 broches

Câble Minitel/carte

- 1 prise DIN 5 mâle
- 3 broches de barrette sécable mâle



Les connecteurs mini-DIN 8 pour MAC et DB9 pour PC

ANDA #%01111111
* minitel en mode local
ADDA #S20
RTS

Envoi vers le Minitel de données en provenance du HC11

ENVOI ASL SCSR,X
BCC ENVOI
BSR PARITE
STAA SCDR,X
RTS

Transformation du 8^{ème} bit en bit de parité paire

Le caractère à envoyer est au départ dans l'accumulateur A.
On le stocke alors dans CHIFFRE et CARACTERE.
Les bits de CARACTERE sont ensuite additionnés un à un, le résultat apparaissant sur le bit 7 de l'accumulateur A.
PARITE PSHB
STAA CHIFFRE
STAA CARACTERE

```

CLR BPARITE
CLRB
LDAACARACTERE
ASLA
STAA CARACTERE
ANDA #%10000000
ADDA BPARITE
STAA BPARITE
INCB
CMPB #7
BNE BOU3
ADDA CHIFFRE
PULB
RTS
    
```

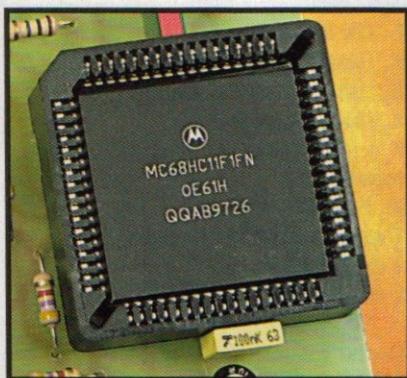
La réalisation

Carte principale (figures 4 et 5)

Elle ne pose pas de problèmes particuliers, seule la proximité de quelques pistes et pastilles autour du support PLCC est à surveiller attentivement. Après perçage des trous, on passera à l'implantation des 2 straps et des autres composants. On fera, là aussi, particulièrement attention à l'orientation du support PLCC : la petite flèche de repère qui indique la patte 1 pointera vers l'interrupteur à glissière de MODB.

Carte interface MAC/PC (figures 6 et 7)

Là non plus, il ne devrait pas y avoir de problèmes. On choisira de souder le connecteur DB9 pour PC ou mini DIN 8 pour MAC suivant son équipement informatique.



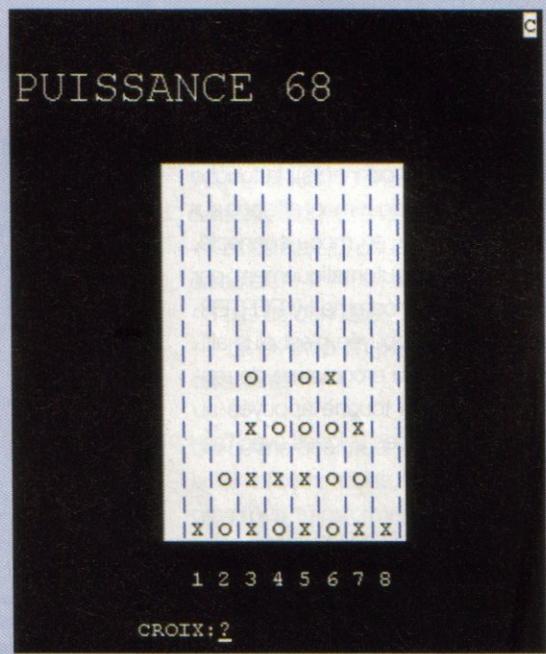
Mise en œuvre de notre console de jeux

Programmation

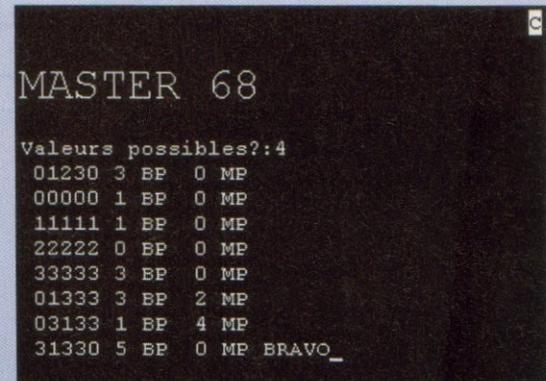
Les 2 programmes MASTER68 et PUISSANCE68 dont vous pouvez apprécier l'aspect figures 8 et 9 sont disponibles sur le CD Rom joint à la revue. Deux autres programmes (calendrier perpétuel et codeur



Le prog puissance 68



Le prog master 68



morse) sont téléchargeables à partir du site www.kitmicro.com.

Connectez la carte interface entre votre ordinateur et la carte principale et positionnez l'interrupteur K₂ pour avoir MODB à 0. Assemblez puis chargez le programme. Cette opération terminée, positionnez K₂ pour avoir MODB à 1 et passer en mode CIRCUIT SEUL. Le montage est opérationnel.

Utilisation

Avec un Minitel, branchez le câble 3 fils entre la prise péri-informatique et les broches masse, PD0 et PD1 de la carte principale, allumez le Minitel et restez en mode local.

Avec un émulateur, branchez l'interface sur un des ports COM1 ou COM2, démarrez l'émulateur, vérifiez les paramètres de la transmission (minitel, 1200Bds, 7 bits de données, parité paire, 1 bit de stop).

Alimentez le montage puis appuyez sur une

touche de votre clavier ou du Minitel : la présentation du logiciel chargé s'affiche.

MASTER68

Entrez le nombre de valeurs maximum (le nombre de couleurs dans le vrai MASTER-MIND), par exemple 4. Dans ce cas, le 68HC11 va créer un code avec les chiffres 0 à 3. Tapez votre première combinaison (5 chiffres à entrer), c'est parti. Si vous vous êtes trompé de chiffres en entrant une combinaison, vous pouvez effacer la ligne en tapant C (c majuscule)

PUISSANCE68

2 joueurs s'affrontent, le joueur aux croix joue en premier. Chacun entre à tour de rôle le numéro de la colonne où mettre son symbole.

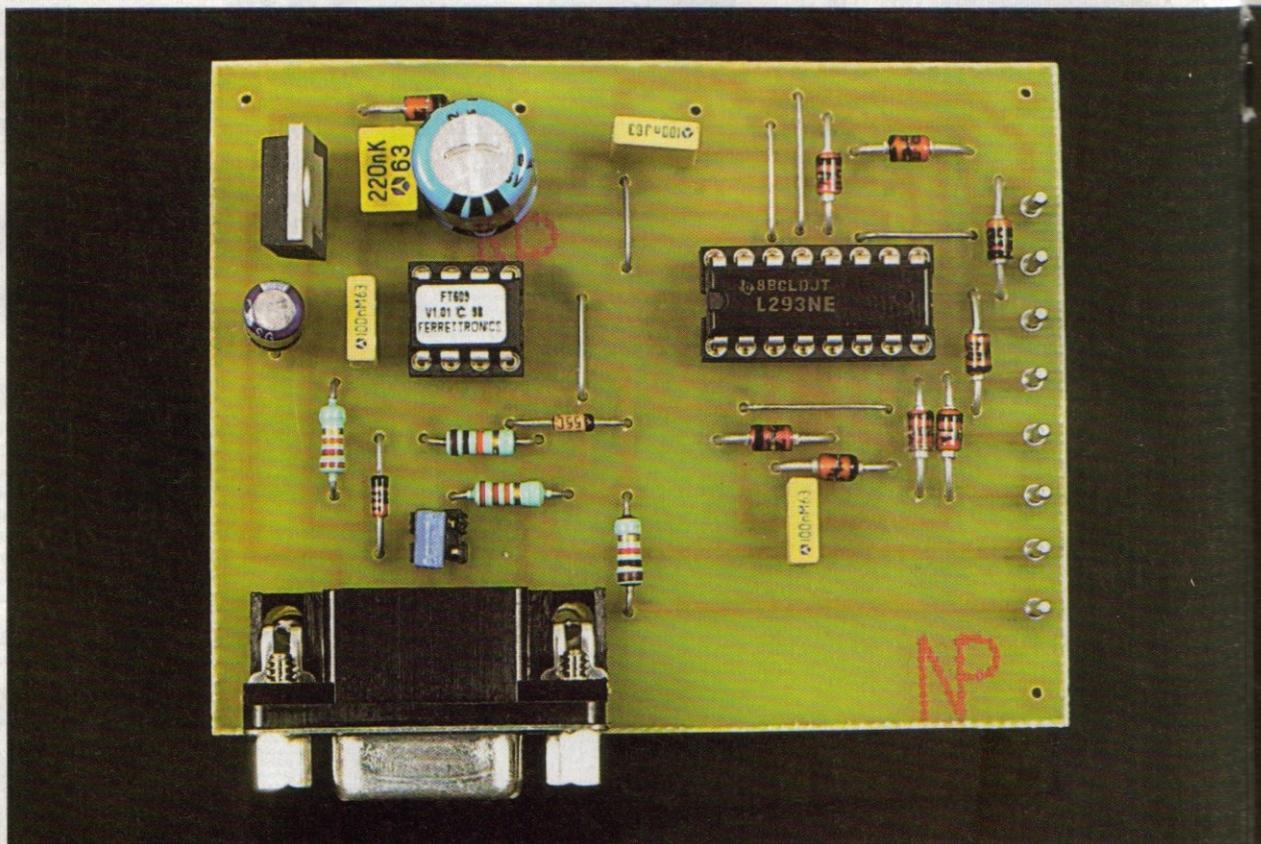
Pour une nouvelle partie, appuyez sur le bouton de RESET.

A. REBOUX

Contrôleur de moteurs pas à pas

sur le port série

Les moteurs pas à pas présentent de nombreux avantages par rapport aux moteurs à courant continu classiques. Malheureusement, et c'est un peu le revers de la médaille, ils sont plus difficiles à mettre en œuvre car, au lieu de la banale tension continue nécessaire pour faire tourner les moteurs classiques, il faut ici générer des signaux rectangulaires aux chronogrammes bien précis. Si cette génération est encore assez facile pour les moteurs unipolaires, elle est nettement plus délicate en ce qui concerne les moteurs bipolaires.



Bien sûr, nous direz-vous, quelques circuits spécialisés existent mais leur souplesse de programmation laisse souvent à désirer et il faut réaliser une interface spécifique pour les commander à partir d'un PC. C'est pour tenter de résoudre tous ces problèmes que nous vous proposons, non pas un, mais deux montages vous permettant de commander des moteurs pas à pas unipolaires et bipolaires à partir du port série de n'importe quel compatible PC. Et, pour que votre bonheur soit complet, sachez que cette commande peut être faite dans n'importe quel langage de programmation car elle se résume à l'envoi de quelques octets sur le port série. Contrairement à ce que l'on pourrait croire, il ne faut, pour parvenir à cela, que deux circuits intégrés !

Le FT609 de Ferretronics

Une telle simplicité de réalisation est rendue possible aujourd'hui grâce à la commercialisation, en France, par Selectronic, des circuits intégrés de la firme américaine Ferretronics. Cette société dispose, en effet à son catalogue, d'un circuit baptisé FT609 qui réalise l'interface intelligente entre n'importe quelle liaison série classique et des circuits de puissance pour moteurs pas à pas.

En fait, ce circuit permet de commander aussi bien des moteurs unipolaires que bipolaires ; il suffit juste pour cela de le faire suivre par le circuit de puissance "qui va bien". Nos deux réalisations du jour vont donc avoir comme base commune un FT609 suivi, selon le cas, du circuit de puissance pour

moteurs unipolaires ou pour moteurs bipolaires.

L'interface pour moteurs unipolaires

Puisqu'il faut bien commencer par l'une ou l'autre, voici tout d'abord en **figure 1** le schéma de l'interface pour moteurs unipolaires. Même si vous n'avez d'yeux que pour les moteurs bipolaires, lisez néanmoins sa description car nous ne répéterons pas tout ce qui est commun avec celle pour moteurs bipolaires que nous verrons dans un instant. Le port série du PC aboutit sur une prise SubD 9 points femelle classique et les straps S_1 et S_2 permettent d'adapter le montage à votre câble de liaison, selon qu'il est droit ou croisé. Une réduction des niveaux

RS232 en TTL est ensuite assurée par D_1 , R_1 et R_2 . C'est assez peu élégant mais cela suffit, dans le cas présent, pour faire en sorte que le FT609, repéré IC_1 , ne reçoive que des niveaux TTL.

Ce circuit délivre alors, sur ses sorties repérées A à D, les chronogrammes propres à être appliqués à un circuit de puissance pour

tion suivante soit respectée : $VZ \leq 49 - VM$

Dans la majorité des applications classiques, vous pouvez cependant vous passer de cette diode et la remplacer par un court-circuit.

La tension d'alimentation du moteur est également utilisée pour alimenter IC_1 au

tance R_3 et la zéner DZ_1 protègent l'entrée du FT609 d'une éventuelle tension excessive appliquée sur "inter".

L'interface pour moteurs bipolaires

Comme nous vous l'avions annoncé, le schéma visible **figure 2** ressemble comme un frère au précédent. Toute la partie entourant le FT609 est en effet identique et seul change le circuit d'interface avec le moteur, qui est ici un L293 de ST Microelectronics ou d'une de ses nombreuses secondes sources.

Ce circuit est prévu pour commander directement des moteurs bipolaires et admet comme tension maximum d'alimentation $+VM$ jusqu'à 36V. Il peut fournir un courant permanent de 1A et un courant de crête de 2A. Même si une version avec diodes de protections intégrées existe au catalogue de ST, nous n'avons pu la dénicher chez aucun revendeur et avons donc prévu notre montage pour la version la plus répandue, c'est à dire sans diodes. De ce fait, ces dernières sont à câbler en externe et sont repérées ici D_2 à D_5 .

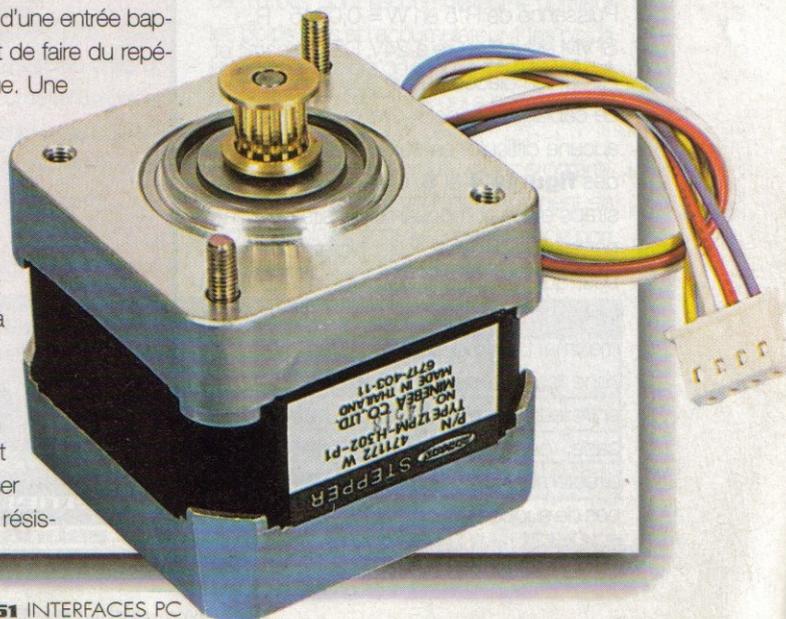
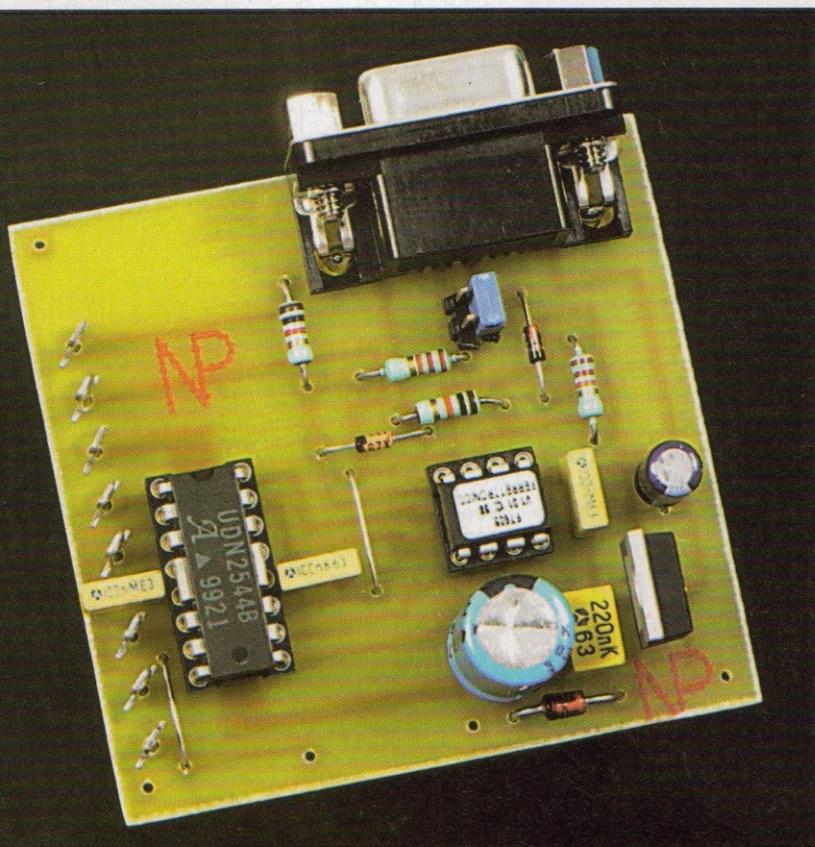
Réalisation des deux versions

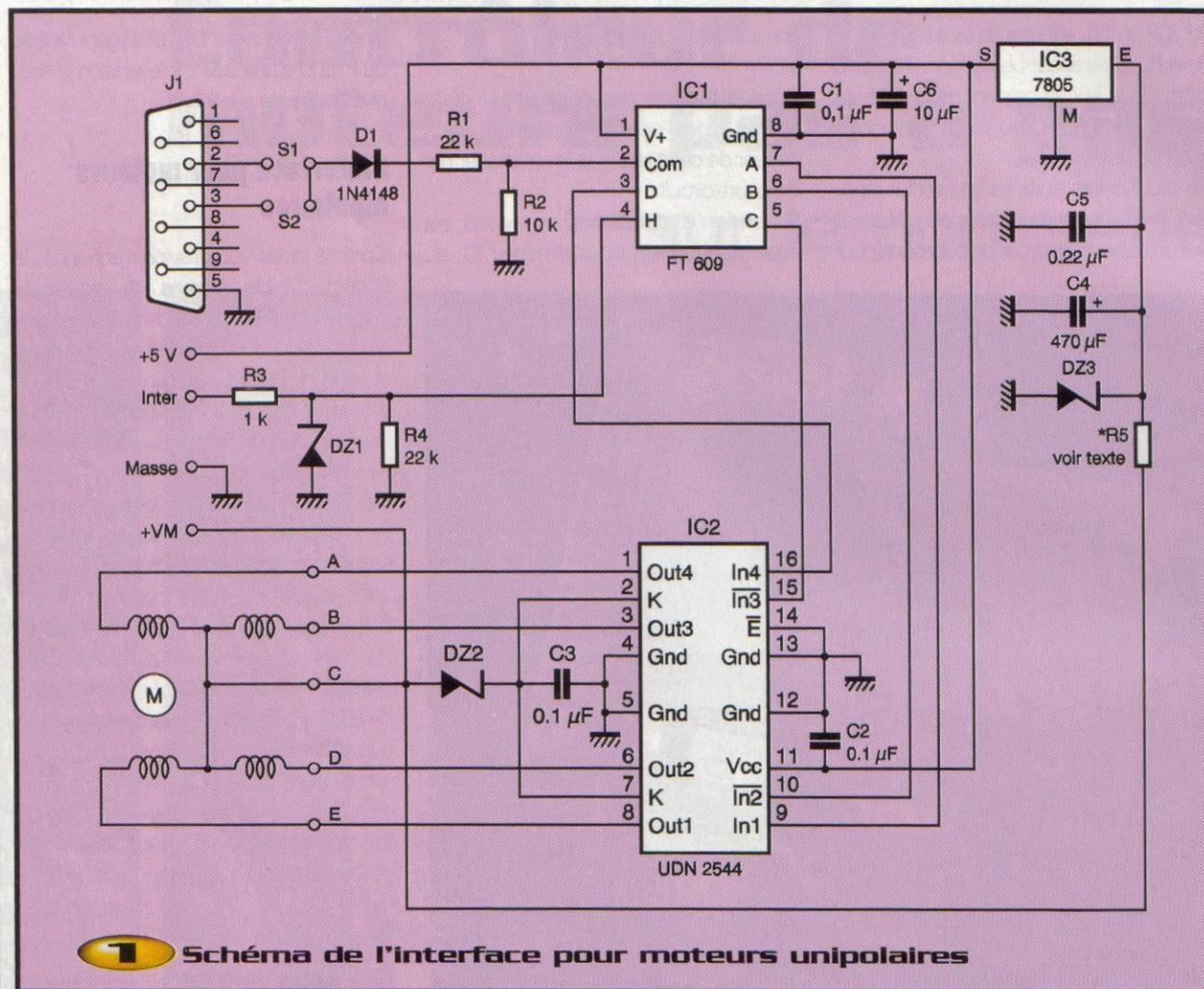
Le circuit imprimé de la version unipolaire vous est présenté **figure 3** alors que celui de la version bipolaire est disponible **figure 5**.

Dans les deux cas, l'approvisionnement des composants ne devrait pas vous poser de problème particulier, mais faites attention au L293 dont il existe une version en boîtier DIL 20 pattes qui ne convient pas ici,

moteurs pas à pas. Dans cette application unipolaire, un UDN2544 d'Allegro est utilisé. Ce circuit est en fait un assemblage de darlington "de puissance" et de portes logiques dans un seul et même boîtier 16 pattes. Il peut être alimenté sous une tension allant jusqu'à 50V et fournir, sur ses sorties à destination du moteur, un courant maximum de 1,8A. En outre, il contient en interne les diodes d'écrêtage des transitoires qui se produisent lorsque le courant est brutalement coupé dans les bobinages du moteur. L'alimentation de ce circuit, et donc du moteur, est assurée par la tension $+VM$ qui doit donc rester inférieure à 50V et qui doit être adaptée au moteur utilisé. La zéner DZ_2 est facultative. Elle permet d'améliorer la commande des phases du moteur en augmentant la vitesse de décroissance de la tension aux bornes des bobines, ce qui confère au moteur de meilleures performances. Sa valeur doit être telle que la rela-

travers d'un régulateur intégré classique IC_3 . Afin que la tension d'entrée de ce régulateur ne dépasse pas ses possibilités en présence d'une tension VM très importante, un circuit de limitation faisant appel à R_5 et DZ_3 est prévu. Enfin, le FT609 dispose d'une entrée baptisée "home" qui permet de faire du repérage de position absolue. Une commande permet en effet de faire tourner le moteur jusqu'à ce que cette entrée passe au niveau logique haut, c'est à dire au +5V. Il est donc possible de relier à l'entrée "inter" de notre montage un interrupteur, ou tout autre capteur, établissant un contact avec le +5V pour disposer de cette fonction. La résis-





bien sûr. Si vous trouvez du L293 avec diodes (suffixe D chez ST Microelectronics mais pas chez Texas !), les diodes D_2 à D_9 sont évidemment inutiles.

Pour les deux montages, la diode zener DZ_3 et la résistance R_5 ne sont nécessaires que si VM dépasse 24V. Dans ce cas, R_5 est à calculer avec la relation suivante :

Valeur de R_5 en $\Omega = (VM - 15) / 0,05$

Puissance de R_5 en W = $0,0045 \cdot R_5$

Si VM est inférieure à 24V, DZ_3 disparaît et R_5 est remplacée par un court-circuit.

Le câblage des deux circuits ne présente aucune difficulté en suivant les indications des **figures 4 et 6**. Commencez par les straps et les composants passifs pour finir par les composants actifs. Notez, à ce propos, que les circuits de puissance chauffent s'ils travaillent près de leurs possibilités maxima. Leur radiateur est constitué par la zone cuivrée du circuit imprimé placée entre leurs pattes mais, pour qu'elle soit efficace, ces circuits doivent être soudés directement sur ce dernier. N'utilisez donc pas de support pour IC_2 contrairement à ce

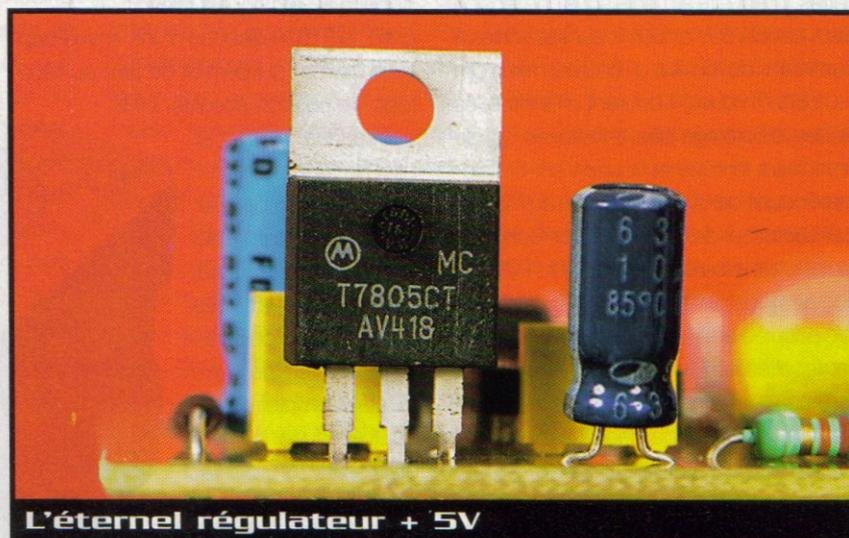
qui est visible sur notre maquette, pour laquelle c'était indispensable pour nous permettre diverses expérimentations.

Une fois le montage terminé et vérifié, il ne vous reste plus qu'à le relier au moteur de votre choix et à la tension d'alimentation VM de ce dernier, sans dépasser les maxima de chaque carte qui sont, rappelons-le, de

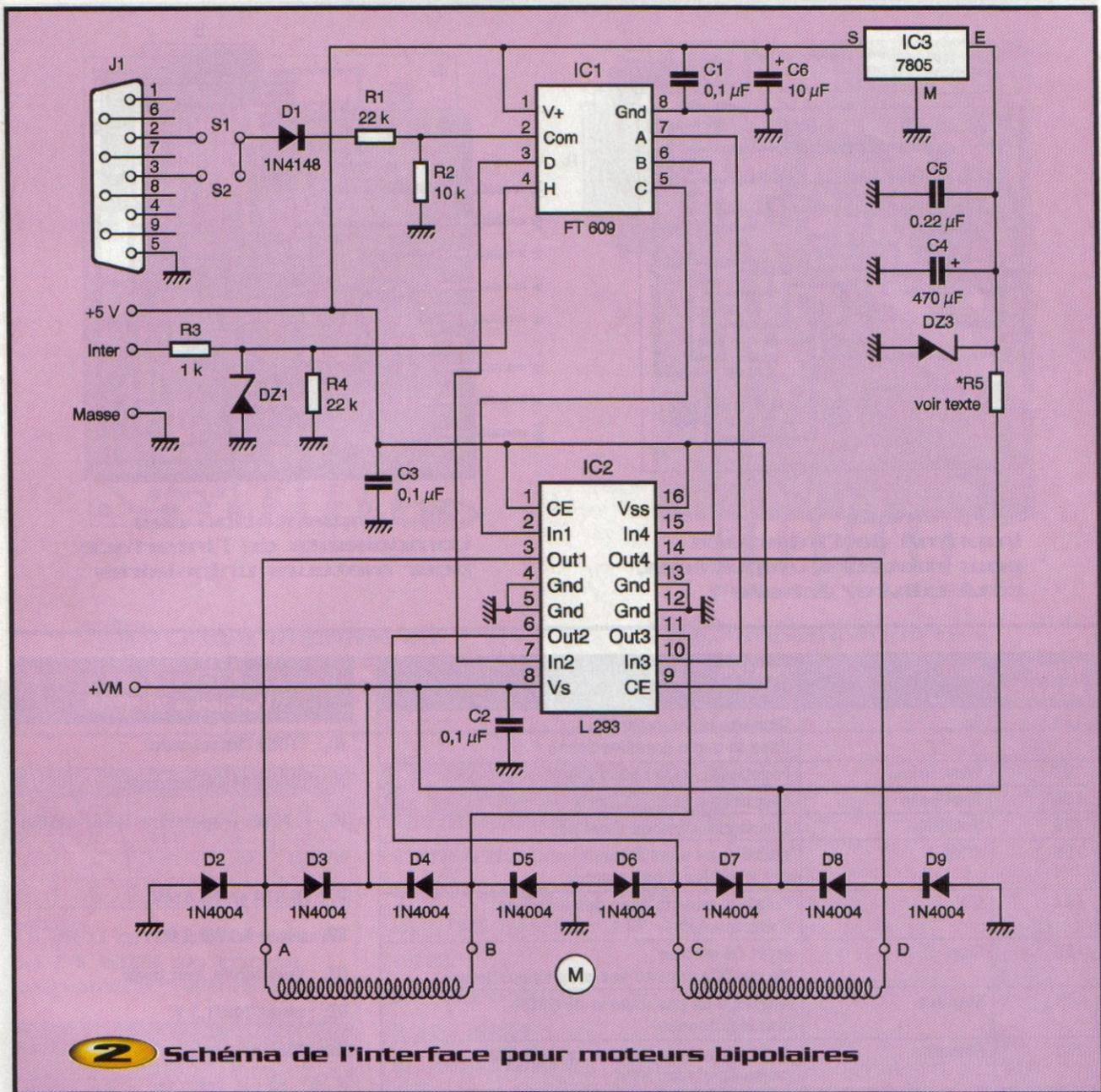
50V pour la version unipolaire et de 36V pour la version bipolaire.

Logiciel et utilisation

Comme nous vous l'avons annoncé en introduction, aucun logiciel spécifique n'est nécessaire car le FT609 n'a besoin que de



L'éternel régulateur + 5V



2 Schéma de l'interface pour moteurs bipolaires

quelques octets de commande pour générer automatiquement tous les signaux correspondants. N'importe quel logiciel capable d'émettre ceux-ci sur le port série convient donc pour piloter nos interfaces. Ceci étant, voici donc le principe d'envoi de ces octets de commande et leurs fonctions.

Sachez tout d'abord que le FT609 contient quatre accumulateurs dans lesquels on ne peut écrire que les 4 bits de poids faibles car les 4 bits de poids fort servent en fait d'adresses et de codes de commande de ces derniers comme le montre le **tableau 1**.

Pour envoyer un ordre au FT609, il suffit de lui envoyer l'octet correspondant, via la liaison série, qui doit fonctionner à 2400 bits

par seconde, sans parité, avec 8 bits de données et 1 bit de stop.

Si la commande nécessite une valeur numérique, comme par exemple celle qui ordonne au moteur de tourner d'un certain nombre de pas, cette valeur numérique doit avoir été écrite dans les quatre accumulateurs avant l'envoi de la commande.

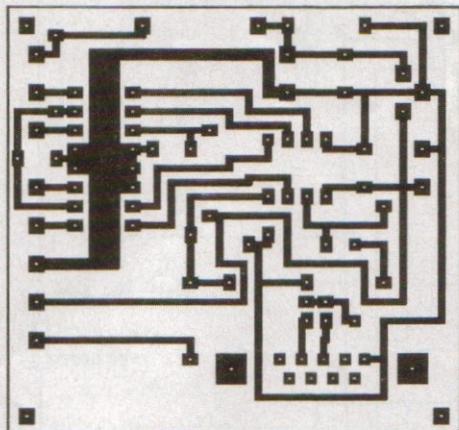
Les 4 bits de données "utiles" contenus dans chacun de ces quatre accumulateurs

sont mis bout à bout pour former un mot de 16 bits dont l'accumulateur 4 contient les poids forts et l'accumulateur 1 les poids faibles. C'est alors ce mot de 16 bits qui est utilisé par la commande qui a besoin d'une donnée numérique.

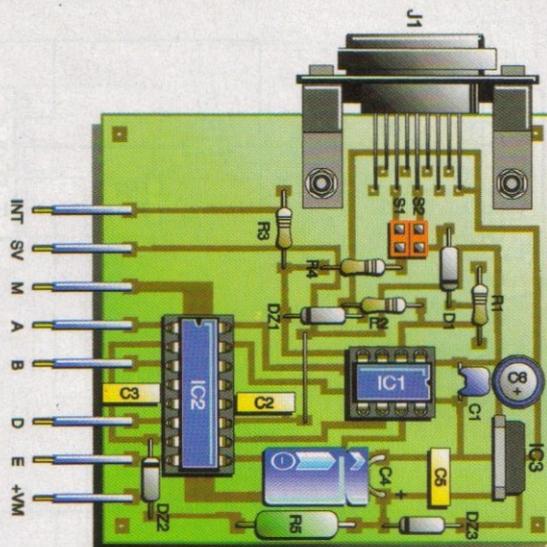
Ainsi par exemple, si vous voulez programmer le FT609 pour un fonctionnement en mode bipolaire, vous lui enverrez par le port série le seul octet 194 (en décimal) puisque

Valeur binaire	Nom	Fonction
1000xxxx	LoadAccumNibble1	Charge l'accu 1 avec xxxx et remet à zéro les accus 2 à 4
1001xxxx	LoadAccumNibble2	Charge l'accu 2 avec xxxx
1010xxxx	LoadAccumNibble3	Charge l'accu 3 avec xxxx
1011xxxx	LoadAccumNibble4	Charge l'accu 4 avec xxxx

T1 Commandes relatives aux accumulateurs



3 Circuit imprimé de l'interface pour moteurs unipolaires, côté cuivre, échelle 1



4 Implantation des composants de l'interface pour moteurs unipolaires

Valeur décimale	Nom	Fonction
192	Go	Démarre le mouvement dans le mode présélectionné
193	WaveDrive	Fonctionnement unipolaire
194	TwoPhase	Fonctionnement bipolaire
195	HalfStep	Fonctionnement en demi pas
196	CCW	Rotation en sens inverse des aiguilles d'une montre
197	CW	Rotation dans le sens des aiguilles d'une montre
198	Stop	Arrêt du moteur. Ne modifie aucune des programmations
199	StepOne	Avance d'un pas dans la direction présélectionnée
200	HomeOn	Arrêt du moteur lorsque l'entrée Home passe au niveau haut
203	HomeCountOn	Tourne un certain nombre de pas après que Home soit passée au niveau haut
204	CountOn	Valide le comptage du nombre de pas prédéfini
205	LoadCount	Charge le nombre de pas
206	Use2544	Utilisation avec un UDN 2544
207	StoreStepTime	Voir texte
208	StoreRampInterval	Voir texte
209	StoreRampStepTime	Voir texte
210	StoreRampStartTime	Voir texte
213	RampOn	Valide le mode Ramp
214	HomeOff	Arrête la commande Home
215	CountOff	Arrête la commande Count
217	RampOff	Arrête le mode Ramp
218	KeepEnergized	Laisse le moteur alimenté, même à l'arrêt
219	DeEnergized	N'alimente pas le moteur à l'arrêt
222	StoreRampEndTime	Voir texte
223	Continue	Reprise de la rotation après une commande Stop

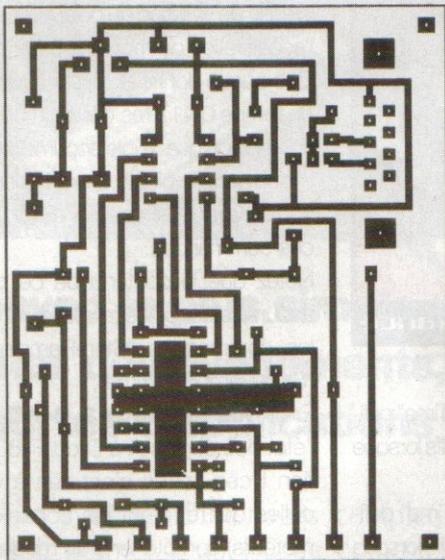
Liste des commandes reconnues par le FT 609

Nomenclature

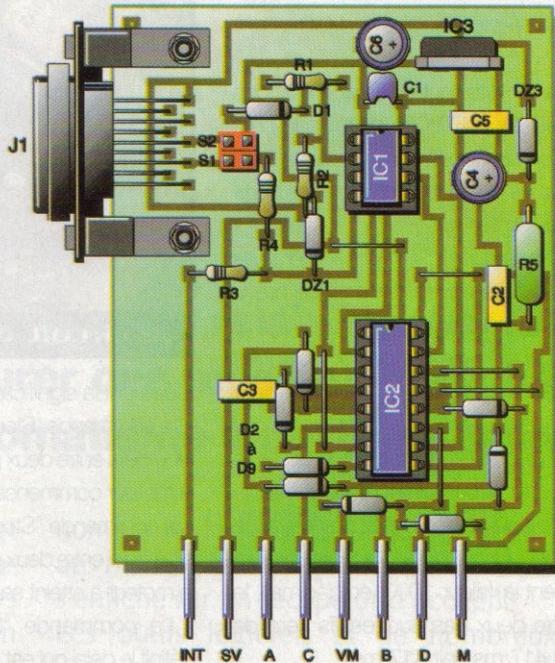
Version à UDN254

- IC₁ : FT609 (Sélectronic)
- IC₂ : UDN2544 (Sélectronic)
- IC₃ : 7805 (régulateur +5V, boîtier T0220)
- D₁ : 1N 914 ou 1N 4148
- DZ₁ : zéner 5,1V/0,4 W
- DZ₂ : facultative, voir texte
- DZ₃ : zéner 24V/1,3 W (facultative, voir texte)
- R₁, R₄ : 22 kΩ 1/4W 5% (rouge, rouge, orange)
- R₂ : 10 kΩ 1/4W 5% (marron, noir, orange)
- R₃ : 1 kΩ 1/4W 5% (marron, noir, rouge)
- R₅ : facultative, voir texte
- C₁ à C₃ : 0,1 µF mylar
- C₄ : 470 µF/25V chimique radial
- C₅ : 0,22 µF mylar
- C₆ : 10 µF/63V chimique radial
- 1 connecteur SubD 9 points femelle coudé pour CI
- 1 support de CI 8 pattes





5 Circuit imprimé de l'interface pour moteurs bipolaires, côté cuivre, échelle 1



6 Implantation des composants de l'interface pour moteurs bipolaires

	A	B	C	D
1	+5V	0	0	0
2	0	+5V	0	0
3	0	0	+5V	0
4	0	0	0	+5V

T3 Fonctionnement du FT 609 en mode unipolaire "Wavedrive"

	A	B	C	D
1	+5V	+5V	0	0
2	0	+5V	+5V	0
3	0	0	+5V	+5V
4	+5V	0	0	+5V

T4 Fonctionnement du FT 609 en mode deux phases "TwoPhases"

	A	B	C	D
1	+5V	0	0	0
2	+5V	+5V	0	0
3	0	+5V	0	0
4	0	+5V	+5V	0
5	0	0	+5V	0
6	0	0	+5V	+5V
7	0	0	0	+5V
8	+5V	0	0	+5V

T5 Fonctionnement du FT 609 en mode demi pas "HalfStep"

Nomenclature

Version à L293

IC₁ : FT609 (Sélectronic)

IC₂ : L293 en boîtier DIL 16 pattes

IC₃ : 7805 (régulateur +5V, boîtier T0220)

D₁ : 1N914 ou 1N4148

D₂ à D₉ : 1N4004

DZ₁ : zéner 5,1V/0,4 W

DZ₃ : zéner 24V/1,3 W (facultative, voir texte)

R₁, R₄ : 22 kΩ 1/4W 5% (rouge, rouge, orange)

R₂ : 10 kΩ 1/4W 5% (marron, noir, orange)

R₃ : 1 kΩ 1/4W 5% (marron, noir, rouge)

R₅ : facultative, voir texte

C₁ à C₃ : 0,1 μF mylar

C₄ : 470 μF/25V chimique radial

C₅ : 0,22 μF mylar

C₆ : 10 μF/63V chimique radial

1 connecteur SubD 9 points femelle soudé pour CI

1 support de CI 8 pattes

cette commande ne nécessite aucune donnée numérique.

Par contre, si vous voulez le faire tourner de 4900 pas (en décimal, soit 1324 en hexadécimal), vous réaliserez les opérations successives suivantes :

- envoi de 84 (en hexadécimal, soit 10000100 en binaire) ce qui charge le quartet de poids faible de 1324 dans l'accumulateur 1,
- envoi de 92 (en hexadécimal soit 10010010 en binaire) ce qui charge le quartet suivant de 1324 dans l'accumulateur 2,
- envoi de A3 (en hexadécimal soit 10100011 en binaire) ce qui charge le quartet suivant de 1324 dans l'accumulateur 3,
- envoi de B1 (en hexadécimal soit 10110001 en binaire) ce qui charge le quartet de poids fort de 1324 dans l'accumulateur 4,
- envoi de l'octet 205 (en décimal, voir **tableau 2**) qui charge le compteur du FT 609 avec le nombre de pas désiré, c'est à dire en fait avec ce que l'on vient d'envoyer dans les accumulateurs,
- envoi de l'octet 204 (en décimal, voir tableau 2) ce qui ordonne au FT 609 de valider le comptage,
- envoi de l'octet 192 (en décimal, voir tableau 2) ce qui ordonne au FT 609 de commencer à faire tourner le moteur.

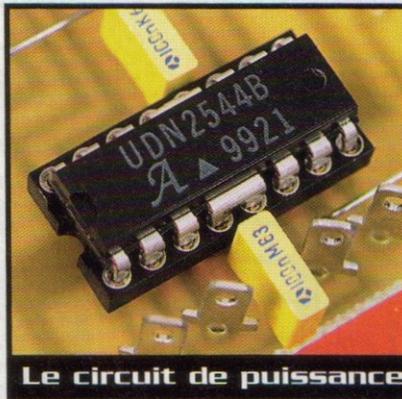
Rassurez-vous cette succession d'opérations a été plus longue à décrire qu'à réaliser.

Le tableau 2 liste toutes les commandes reconnues par le FT609. Ce tableau peut presque se passer de commentaires si ce n'est pour les quelques commandes particulières suivantes :

la commande "Use2544" doit impérativement être utilisée avec le montage prévu pour les moteurs unipolaires, afin que le FT609 génère des signaux logiques conformes aux polarités attendues par l'UDN2544.

La commande "StoreStepTime" utilise le contenu des accumulateurs pour fixer le délai entre deux pas. L'unité utilisée est de 0,417 ms. Ainsi, si les accumulateurs contiennent la valeur 1000 (en décimal), le délai entre deux pas successifs sera de 1000 . 0,417 ms soit 417 ms.

Enfin, les commandes "Ramp" permettent de définir l'accélération et la décélération du moteur en faisant varier les délais entre les pas successifs. Les variables qui y sont uti-



Le circuit de puissance

lisées ont la signification suivante :

- La commande "StoreRampStartTime" définit le délai entre deux pas successifs lorsque le moteur commence à tourner.

La commande "StoreRampEndTime" définit le délai entre deux pas successifs lorsque le moteur a atteint sa vitesse maximum.

- La commande "StoreRampStepTime" définit le délai qui est ajouté au délai initial en phase d'accélération. En d'autres termes c'est le pas de la variation de vitesse.

- La commande "StoreRampInterval" définit le temps pendant lequel le moteur tourne

avec le délai spécifié, c'est à dire encore la vitesse de variation du délai, ou bien encore l'accélération du moteur.

Ces quatre données utilisent comme unité de temps 0,417 ms qui est multiplié par le contenu du quadruple accumulateur. Celui-ci doit donc avoir été chargé avec la valeur voulue avant l'exécution de chacune de ces commandes.

Notez que le contenu de ce quadruple accumulateur doit être inférieur à 255 pour les commandes "StoreRampInterval" et "StoreRampStepTime".

Enfin, **les tableaux 3, 4 et 5** précisent l'état des sorties A à D du FT609 (attention ! ces lettres n'ont rien à voir avec celles qui repèrent les connexions aux moteurs) compte tenu du mode de fonctionnement choisi.

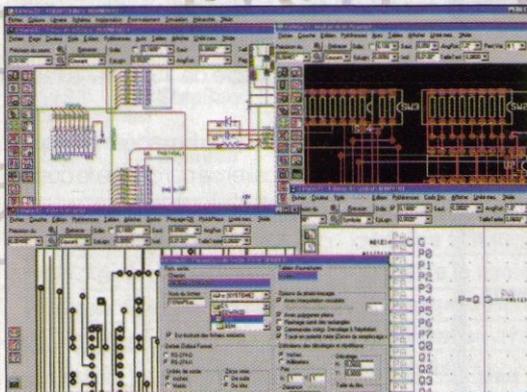
Vous disposez ainsi de toutes les informations nécessaires à la mise en œuvre de ces deux interfaces et vous pouvez constater qu'elle est en fait d'une grande simplicité grâce au FT609.

C. TAVERNIER

NOUVEAU*
en français

EDWin 32

VERSION WINDOWS 95-98 & NT
(compatible an 2000)



Version 1.7 toujours disponible avec des menus d'aide en français.

OPTIONS :

- Librairie complète* (15000 composants) dont CMS
- Base de données étendue*
- Simulation en mode mixte*
- EDSpice - Moteur Spice
- Autorouteur Arizona*
- Analyse thermique*
- Analyse CEM
- Intégrité du signal (plug-in à la CEM)
- CEM + intégrité du signal
- EDCoMX - Générateur de modèles - code Spice
- Aides en français

Conditions particulières pour Education nationale.
Professionnels nous consulter.

Passage de Edwin 32 (ang.) à Edwin 32 (français) 850 F TTC
*Pack Edwin 32 (français) amateur complet 3300 F TTC

MERCURE TELECOM ZA de l'Habitat Bat N°6
BP 58 - Route d'Ozoir - 77680 Roissy-en-Brie
Appel gratuit : 0805 00 80 88 - Fax : 01 64 40 49 18
e-mail : edwin@mercuretelecom.com Internet : mercuretelecom.com

NOUVEAU*

Plus besoin de connaître
la programmation !

Ce que vous dessinez
c'est ce que vous programmez
avec

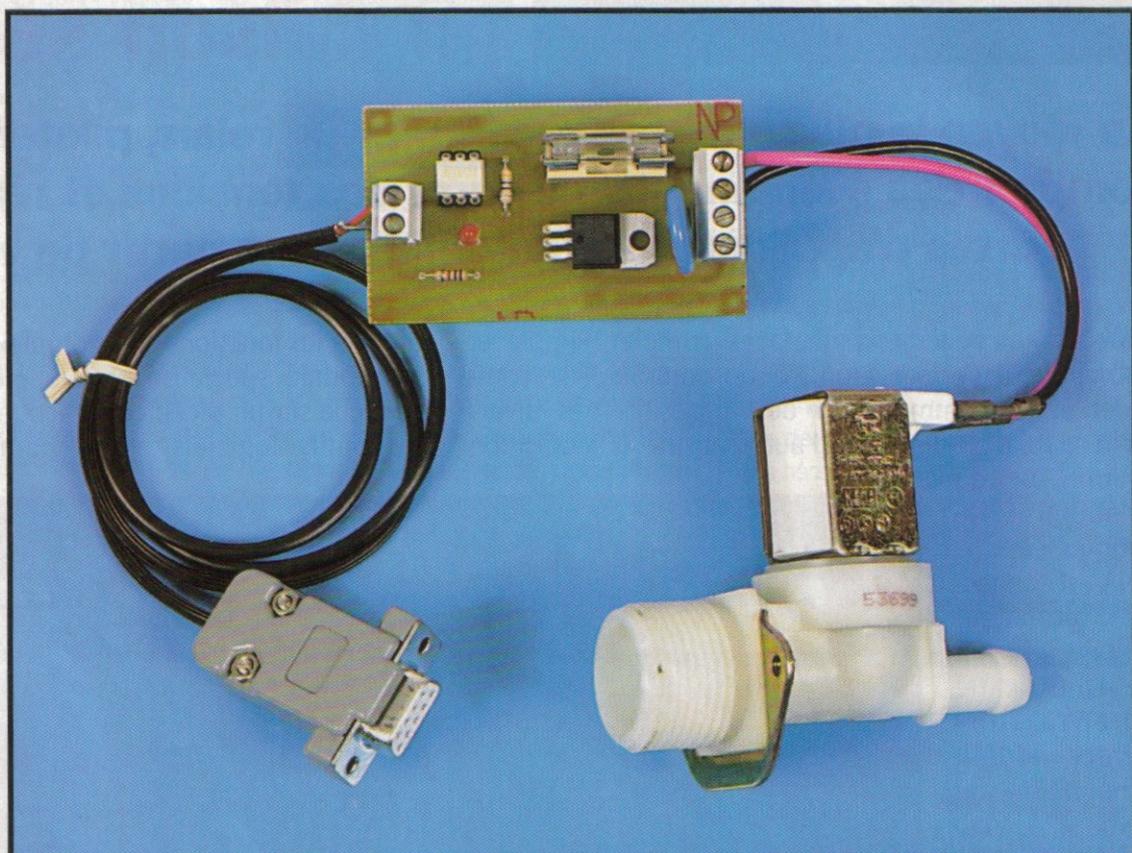


Pour les familles de microcontrôleurs suivantes :
ST62XX - ST72XX - PIC12BXX - PIC14BXX, etc.

Démo gratuite téléchargeable via internet :
www.mercuretelecom.com

MERCURE TELECOM ZA de l'Habitat Bat N°6
BP 58 - Route d'Ozoir - 77680 Roissy-en-Brie
Appel gratuit : 0805 00 80 88 - Fax : 01 64 40 49 18
e-mail : realizer@mercuretelecom.com Internet : mercuretelecom.com

Programmateur d'arrosage



Qui n'a jamais rêvé de pouvoir déclencher l'arrosage de son jardin ou de sa pelouse d'un simple clic de souris ? Ce sera désormais chose possible à condition de réaliser le petit montage décrit dans ces quelques lignes.

Connectez-le au port série de n'importe quel PC et vous serez en mesure de contrôler l'ouverture ou la fermeture d'une électrovanne qui a la même fonction qu'un robinet et, ceci, en restant tranquillement derrière votre écran d'ordinateur. Le programme utilisable avec ce montage fonctionne sous Windows 95/98 et NT, vous disposez donc d'un environnement graphique très confortable et ergonomique. Vous pourrez programmer la date et l'heure du début et de la fin de la période d'arrosage.

Électrovanne

Le principe du montage repose sur l'utilisation d'un organe particulier : l'électrovanne.

Une électrovanne permet de stopper ou non l'écoulement d'un fluide, dans notre cadre d'utili-

sation il s'agira d'eau. Le clapet chargé d'obstruer le conduit où circule l'eau est actionné par un électroaimant. Lorsque la bobine est soumise à une tension, elle crée un champ magnétique qui vient attirer un noyau qui est lui-même solidaire du clapet.

Certains appareils domestiques disposent d'électrovannes, c'est notamment le cas de la machine à laver, servant pour remplir automatiquement le bac en eau. Ces types d'électrovannes sont faciles à se procurer, vous en trouverez notamment dans le catalogue CONRAD Electronic. Deux modèles sont disponibles : simple ou double. Avec le double vous avez deux électrovannes qui ont une entrée commune, mais deux sorties indépendantes. Notre montage, disposant que d'une sortie, pourra piloter que le modèle simple.

Caractéristiques :

Pression admissible : 0,2 à 10 bars
Alimentation : 230V / AC
Puissance : 8W
Entrée filetage : 20 / 27 mm nylon équipée d'un filtre
Sortie sur mâle : diamètre 12mm, longueur 18 mm
Connexion électrique : par cosses de 6,3 mm
Fixation : par plaque entre axe 45 mm

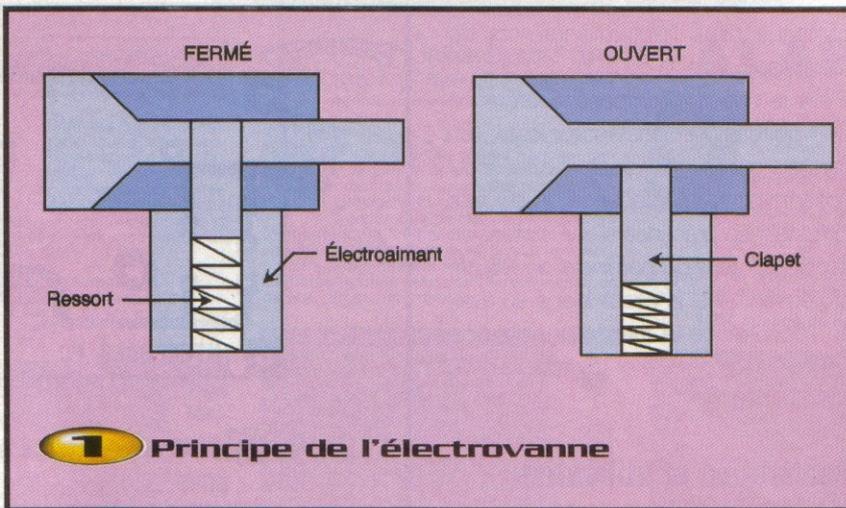
Partie électronique (fig. 1)

La commande de notre carte s'effectue par une sortie du port RS232. L'utilisation de la tension du secteur pour alimenter l'électrovanne nous conduit à réaliser une isolation galvanique afin de protéger correctement l'ordinateur. Cette isolation est réalisée à l'aide d'un optocoupleur (du type MOC3063), un tel circuit se compose

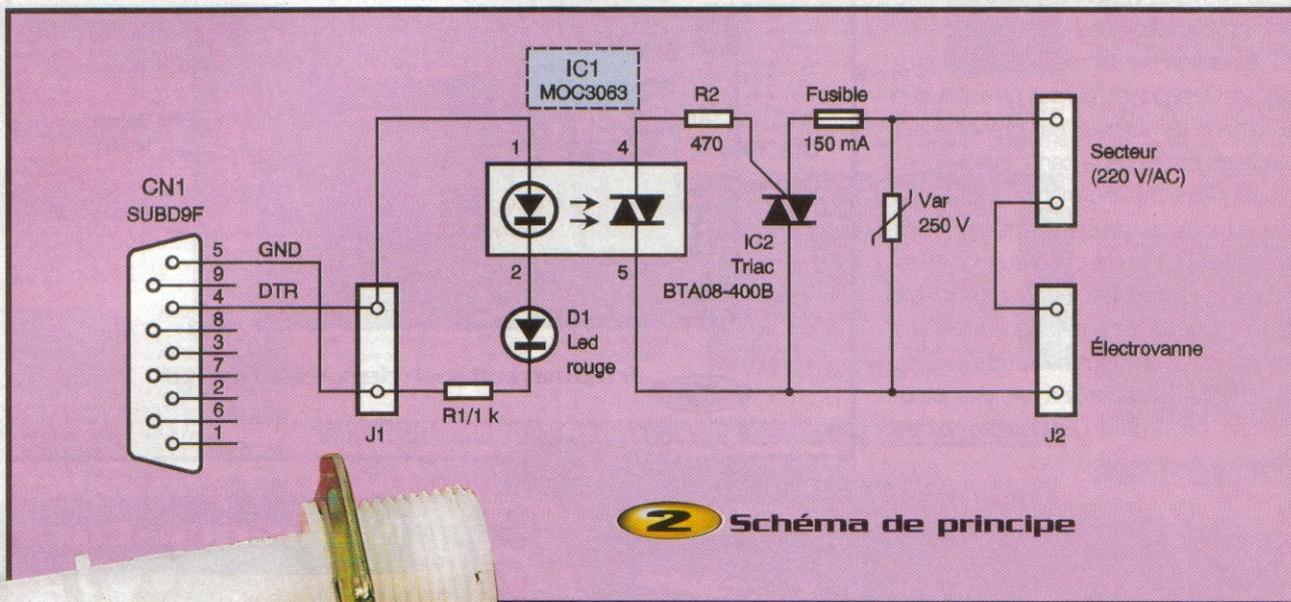
de deux parties distinctes (isolation galvanique de 600V) : la première est constituée d'une diode infrarouge qui va venir mettre en conduction le triac contenu dans la deuxième partie.

Le courant de l'ordre de 5mA, nécessaire à l'activation de la diode infrarouge, est généré par la sortie DTR (broche n° 4) du port série associée à la résistance R₁ d'une valeur de 1 kΩ. La diode d₁, mise en série, permet de valider visuellement l'état de la diode interne au circuit.

La faible puissance du triac interne à l'optocoupleur (I_{MAX} = 100mA) ne permet pas une alimentation directe de l'électrovanne.

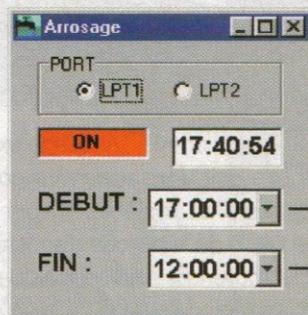


1 Principe de l'électrovanne



2 Schéma de principe

3 Présentation



Copie d'écran :

Début arrosage : le 22/04/00 à 17:00:00
Fin arrosage : le 22/04/00 à 12:00:00

avril 2000							
lun.	mar.	mer.	jeu.	ven.	sam.	dim.	
27	28	29	30	31	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	29	30	
1	2	3	4	5	6	7	

Aujourd'hui : 22/04/00

avril 2000							
lun.	mar.	mer.	jeu.	ven.	sam.	dim.	
27	28	29	30	31	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	29	30	
1	2	3	4	5	6	7	

Aujourd'hui : 22/04/00

Un deuxième triac, mis en cascade, permet de disposer d'une puissance beaucoup plus importante. Celui utilisé ici est un BTA 08-400B en boîtier TO220. Il peut fonctionner sous une tension maxi de 400V et débiter un courant de 8A, ce qui est largement suffisant compte tenu de la faible consommation de l'électroaimant. Il est donc inutile de prévoir un dissipateur thermique. Cependant, il est nécessaire de protéger le triac des phénomènes d'auto-induction générés par la bobine

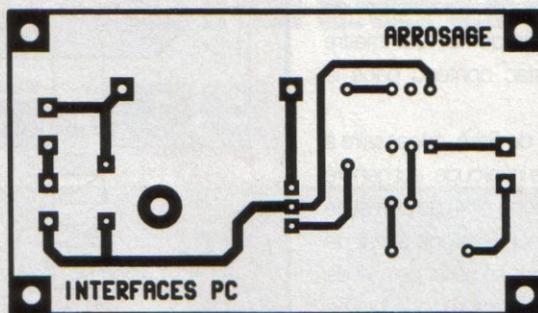
contenue dans l'électrovanne lors de l'établissement et la coupure du courant. C'est le rôle du composant référencé VAR qui est une varistance montée en parallèle sur l'électrovanne. Un tel composant voit son impédance chuter très fortement en présence d'une surtension (tension > tension nominale de 250V), protégeant ainsi le circuit placé en aval, en l'occurrence le triac. Un fusible de 150mA vient compléter la protection de notre montage.

Réalisation et utilisation

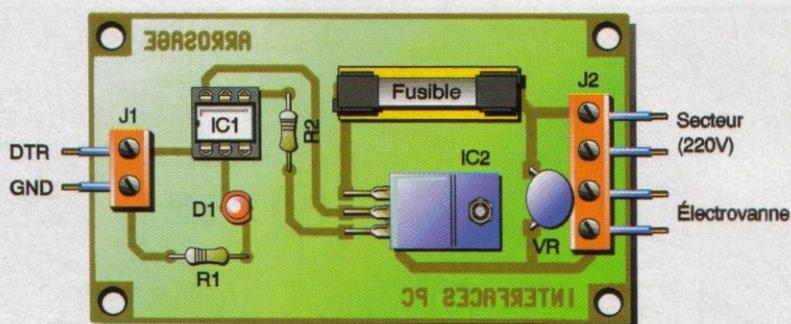
Vous trouverez figures 4 et 5 le cuivre et le l'implantation des composants.

La réalisation ne doit pas poser de problème. Cependant, gardez à l'esprit que la tension du secteur est présente sur plusieurs pistes de la carte, aussi soyez prudent lors de vos manipulations une fois le montage mis sous tension.

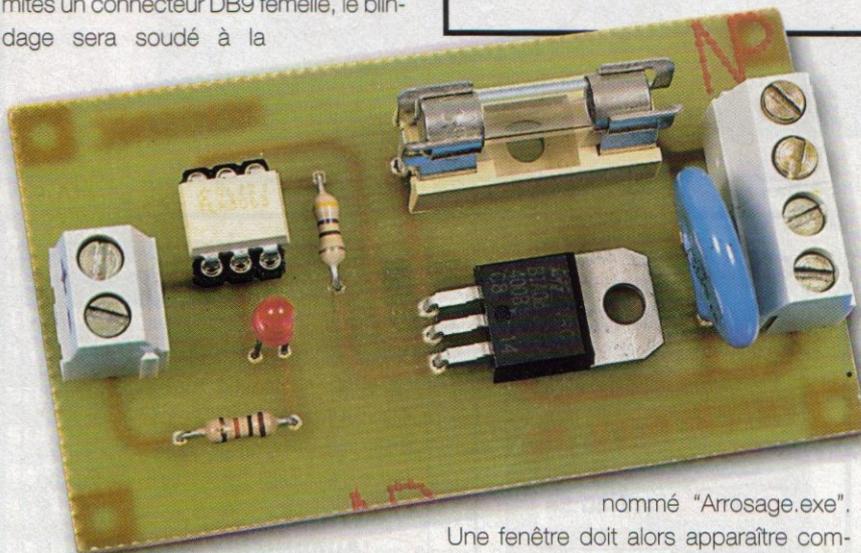
La connexion de la carte au port série pourra s'effectuer à l'aide d'un câble blindé (type audio), la longueur utilisée pourra atteindre plusieurs dizaines de mètres. Vous soudez à une des extrémités un connecteur DB9 femelle, le blindage sera soudé à la



4 Tracé du circuit imprimé



5 Implantation des éléments



broche 5 correspondant à la masse, l'autre conducteur à la broche 4 correspondant à la sortie DTR. L'autre extrémité prendra place dans le bornier J₂ de la carte.

Le programme d'utilisation a été développé avec le logiciel DELPHI 4. Il est donc prévu pour fonctionner sous l'environnement Windows. Pour lancer l'exécution, il suffit de double-cliquer sur l'icône de l'unique fichier du programme

nommé "Arrosage.exe".

Une fenêtre doit alors apparaître comportant l'heure du système, l'heure du début et de fin de la période d'arrosage, qui sont à l'ouverture initialisées à 00 :00 :00. En tabulant, ou à l'aide de la souris, vous allez pouvoir modifier ces horaires. En cliquant sur les flèches vous pouvez également modifier la date qui, par défaut, correspond à la date du jour (cerclée en rouge sur le calendrier). Lorsque la date et l'heure de début correspondent à la date et à l'heure actuelle, le voyant, à l'origine de couleur verte,

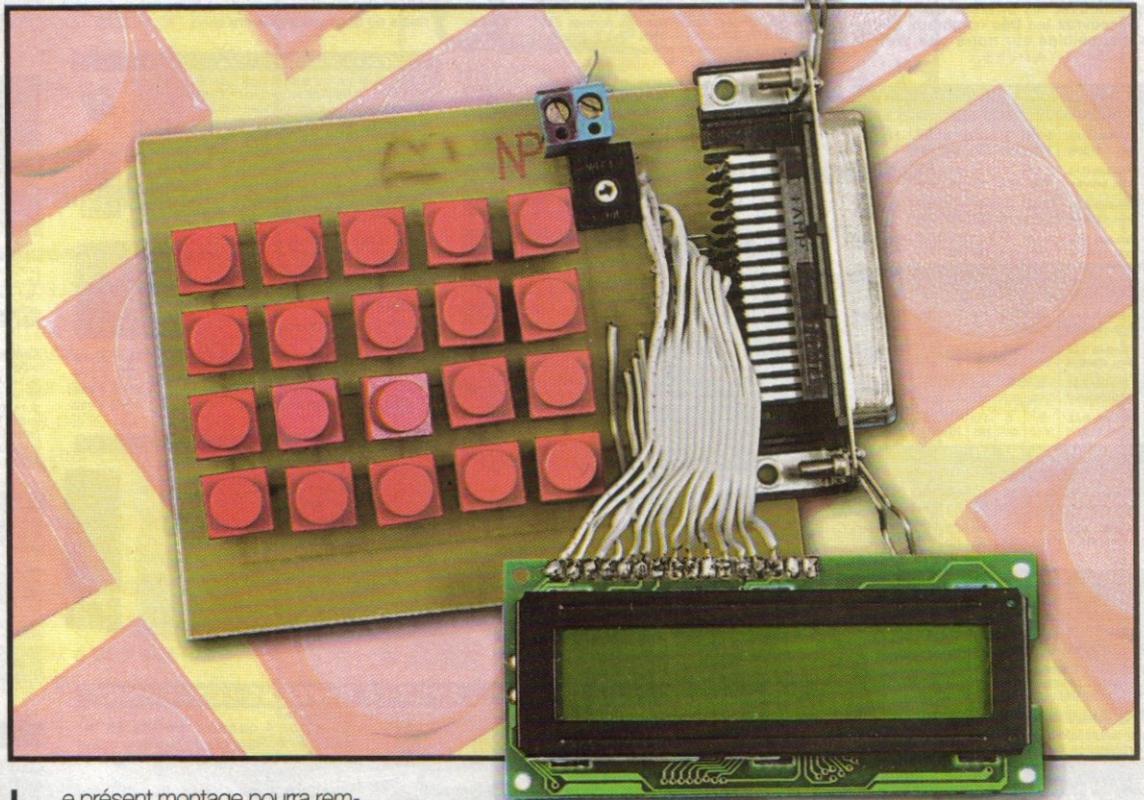
Nomenclature

- R₁ : 1 kΩ
- R₂ : 470 Ω
- VAR : varistance 250V
- Fusible : 150mA + support pour CI
- D₁ : LED rouge
- CN₁ : connecteur DB9 femelle (à câbler)
- J₁ : 1 bornier 2 plots
- J₂ : 2 borniers 2 plots
- IC₁ : MOC3063 (ou TLP3063) + support DIL 6 broches
- IC₂ : triac BTA 08-400B
- Électrovanne (modèle simple) : disponible chez CONRAD Electronic
- 2 cosses femelles de 6,3mm

devient rouge avec à l'intérieur le mot "ON" signalant que l'électrovanne est activée (sortie DTR = 1). Lorsque l'heure et la date de fin correspondent à l'heure et la date actuelle, le voyant redevient vert avec le mot "OFF" affiché signalant que l'électrovanne est hors tension (sortie DTR = 0).

D. REY

Interface écran et clavier sur port parallèle



Nombre d'entre nos lecteurs utilisent de vieux PC "démodés" pour des applications diverses. Programmation de composants, gestion d'alarme, automates divers sont des applications courantes pour nos 8086, mais, très souvent aussi, nous ne disposons pas d'un écran pour chaque machine car, si un vieux PC se trouve facilement à 200/300F, un écran est toujours plus cher

Le présent montage pourra remplacer avantageusement l'écran et le clavier d'une machine consacrée à une application simple. Ce montage permet de connecter simplement un écran LCD et un petit clavier de 20 touches sur le port parallèle de votre PC. De plus, le seul composant de ce montage est le potentiomètre de réglage du contraste de l'écran.

Partie affichage

Le montage est très simple. L'écran est utilisé en mode 4 bits. Cela signifie que seuls les bits de 4 à 7 sont connectés et, ce, aux bits 0 à 3 du port parallèle. Les bits 4 à 6 de ce même port sont utilisés pour contrôler l'afficheur LCD. Le bit 4 est connecté à la broche RS. Cette broche sert à sélectionner le registre de l'afficheur parmi celui contenant les caractères et celui des instructions. Le bit 5 est connecté à la broche RW

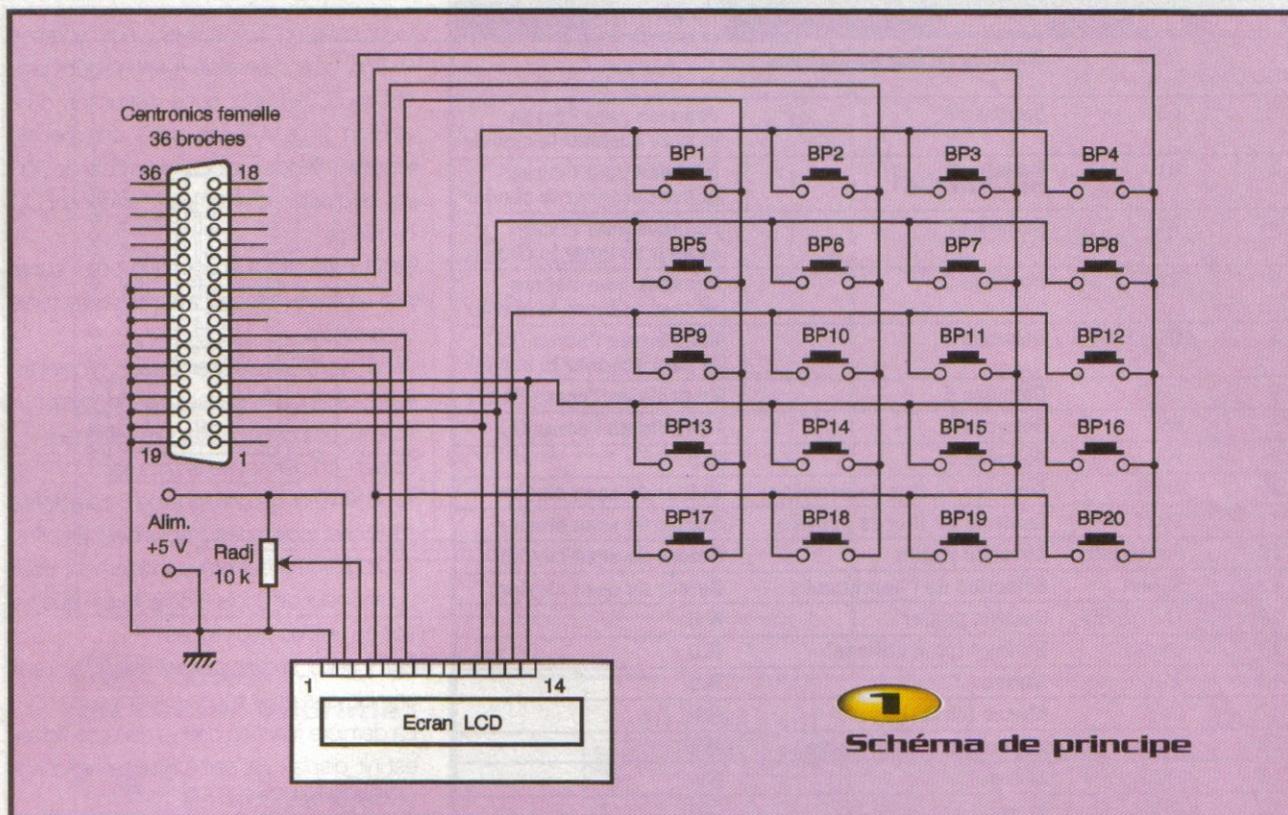
qui permet de sélectionner le mode de lecture ou d'écriture. En écriture, on peut écrire des instructions et des données dans la RAM de l'afficheur. En lecture, on peut relire ces mêmes informations, ainsi que des bits de

statuts (cette fonctionnalité de lecture n'est pas utilisée ici). Enfin, le dernier bit connecté du port parallèle est le 6, il est connecté à la broche E de l'afficheur LCD. Cette broche est la plus importante de la présente applica-

Broche	NOM	Description
1	GND	Masse de l'alimentation
2	Vcc	Alimentation 5V
3	Cstr	Réglage du contraste
4	RS	Sélection du registre : 0=instructions 1=données
5	R/W	0=écriture 1=lecture
6	E	Validation des données présentes sur le bus
7	D0	Données 0
8	D1	Données 1
9	D2	Données 2
10	D3	Données 3
11	D4	Données 4
12	D5	Données 5
13	D6	Données 6
14	D7	Données 7

Note : En mode 4 bits, comme ici, seules les broches D4 à D7 sont utilisées, les broches D0 à D3 doivent rester non connectées.

T1 Brochage de l'écran LCD



tion : Elle permet de valider les données présentes sur le bus 4 bits de l'afficheur. C'est aussi une programmation un peu spéciale de cette broche qui a permis de connecter à la fois l'afficheur et le clavier sur le même port parallèle.

Partie clavier

Le clavier est géré d'une manière assez répandue. Il forme une matrice entre 5 sorties et 4 entrées, ce qui donne 20 combinaisons possibles. Pendant que la broche 6 du port parallèle, et par suite la broche E de l'afficheur, est à l'état bas, on peut modifier comme on le souhaite les autres bits de ce port et cela n'aura aucune influence sur l'affichage tant que cette broche ne repassera pas à l'état haut.

On scanne donc le clavier de la manière suivante : On active successivement les 5 sorties (bits 0, 1, 2, 3 et 4 du port) et on regarde sur 4 entrées celle qui est active. On effectue ainsi en 5 fois un scan total du clavier.

Pour un récapitulatif de toutes les connexions, reportez-vous à la **figure 1** ainsi qu'aux tables 1 et 2. La **table 1** rappelle le brochage d'un afficheur LCD et la **table 2** celui du port parallèle côté imprimante, ainsi que l'utilisation de ces broches dans la présente application.

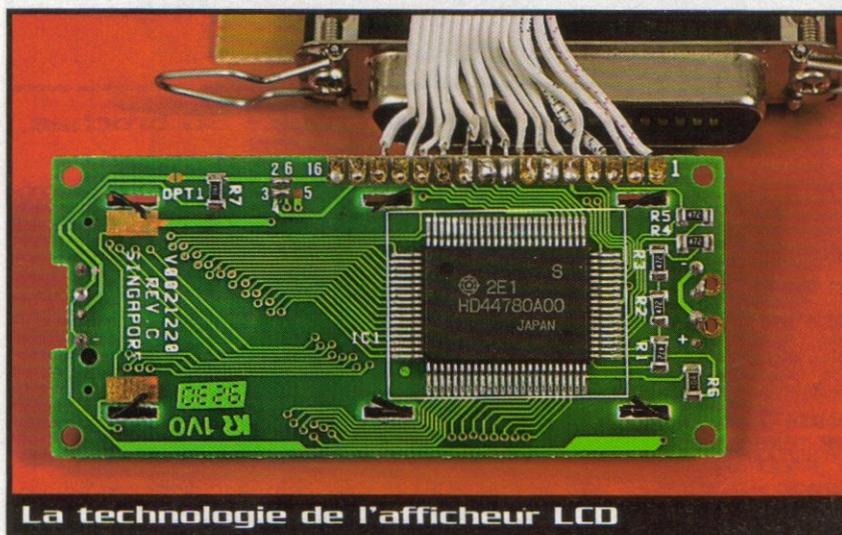
Logiciel

Comme souvent, la réduction du nombre de composants se traduit par une complication, toute relative d'ailleurs, du logiciel accompagnant le montage. Celui-ci est à adapter à votre application particulière et, pour vous simplifier la tâche, un petit programme de démonstration vous est fourni pour deux systèmes d'exploitation et dans deux langages différents. Ainsi, vous trouverez sur le CD-ROM accompagnant ce hors-série une version QBasic pour DOS et une version C pour Unix. Cette dernière est facilement adaptable pour tout système d'exploitation possédant le compilateur libre

GCC. Pour plus d'informations sur le logiciel libre en général, vous pouvez vous référer au site de l'APRIL (Association pour la Promotion et la Recherche en Informatique Libre) dont fait partie l'auteur de ce montage.

Analysons version en C, car c'est un langage assez peu utilisé dans nos réalisations d'habitude.

Le programme est constitué principalement de 3 fichiers : screen.h, keyboard.h et screen_keyb.c, plus un fichier contenant les instructions pour la compilation : Makefile. Le respect des majuscules est très important dans le monde Unix, car ce sys-



Broche	NOM	Description	Utilisation ici
1	/STR N.U.	Contrôle du flux de données	
2	D0	Données 0	Données pour l'écran et pour scanner le clavier
3	D1	Données 1	Données pour l'écran et pour scanner le clavier
4	D2	Données 2	Données pour l'écran et pour scanner le clavier
5	D3	Données 3	Données pour l'écran et pour scanner le clavier
6	D4	Données 4	Contrôle de l'écran (RS) et pour scanner le clavier
7	D5	Données 5	Contrôle de l'écran (R/W)
8	D6	Données 6	Contrôle de l'écran (E)
9	D7	Données 7	N.U.
10	/ACK	Contrôle du flux de données	Retour du scan clavier
11	BUZY	Contrôle du flux de données	Retour du scan clavier
12	Paper Out	Absence papier	Retour du scan clavier
13	Select	Sélection de l'imprimante	Retour du scan clavier
14	/AUTOFEED	Gestion papier	N.U.
15	OSCXT	Horloge (plus utilisée)	N.U.
16	GND	Masse	N.U.
17	GND	Masse (châssis)	N.U.
18	5V	Généré par l'imprimante	N.U.
19	GND	Masse	GND
20	GND	Masse	GND
21	GND	Masse	GND
22	GND	Masse	GND
23	GND	Masse	GND
24	GND	Masse	GND
25	GND	Masse	GND
26	GND	Masse	GND
27	GND	Masse	GND
28	GND	Masse	GND
29	GND	Masse	GND
30	GND	Masse	GND
31	/INIT	Reset	N.U.
32	FAULT	Erreur	N.U.
33	N.C.	N.C.	N.U.
34	N.C.	N.C.	N.U.
35	N.C. ou 5V ou Return	Fonction très ambiguë	N.U.
36	/SELECT	Sélection imprimante	N.U.

Note 1 : N.C. = non connecté
Note 2 : N.U. = non utilisé dans ce montage

T2 Brochage du port Centronics 36 broches, port parallèle

tème, par opposition au DOS, reconnaît les majuscules des minuscules. Il en est de même pour le langage C dont la naissance est liée à celle d'Unix. Ainsi, la variable "table" et la variable "Table" sont deux variables bien séparées, alors qu'en BASIC ou en PASCAL nous ne ferions pas de distinction.

Intéressons-nous tout d'abord au fichier screen.h. Ce fichier commence par "#include

<asm/io.h>". Ce fichier contient les fonctions outb et inb qui servent respectivement à lire et à écrire un octet à une adresse mémoire précise de la machine. Ici, cette adresse est 0x378. C'est l'adresse des 8 broches de données du port parallèle n°1. On stocke cette adresse dans la constante PORT pour utilisation ultérieure dans les fonctions. La première fonction définie est delay. Elle permet de définir une temporisation indépendante du type de machine.

Pour cela, on lit une valeur sur l'adresse 0x378. Le port parallèle ayant un débit précis de 500ko/sec, cette instruction dure environ 2 µs. Utilisée dans une boucle répétée 100 fois, on obtient un délai de 200 µs, ce qui donne le temps à l'afficheur LCD de réagir.

send_byte sert à envoyer un octet sur le port 4 bits de l'afficheur avec ses caractères de contrôle.

send_word sert à décomposer un octet en ses 8 bits pour le passer à send_byte.

screen_off permet d'éteindre l'afficheur.

screen_on permet de le rallumer.

send_line envoie une ligne de 8 caractères à l'afficheur, parmi les 2 possibles. En effet, le programme est prévu initialement pour un afficheur de 2 fois 8 caractères, que ce soit en une ou deux lignes.

Enfin, send_screen permet d'envoyer l'ensemble des deux lignes à l'afficheur.

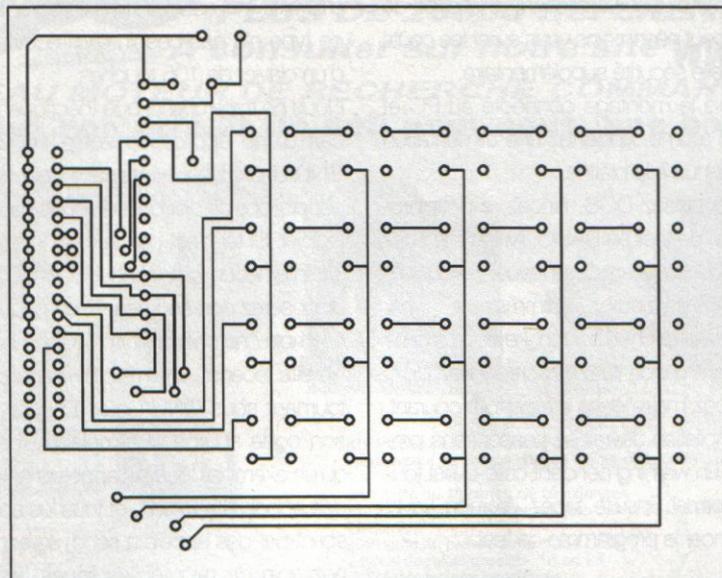
La dernière fonction définie dans ce fichier est init_display. Elle sert à initialiser l'afficheur en début de programme.

Passons maintenant au fichier keyboard.h.

Ce fichier commence comme le précédent. Il définit une constante supplémentaire, POIN, pour Port Input. Ce port, 0x379, est le port d'état du port parallèle. La seule fonction définie dans ce fichier, keyscan, renvoie la touche pressée s'il y en a une, sinon un espace.

Intéressons-nous de plus près à cette fonction. La fonction usleep permet de définir un temps d'attente en microsecondes. Ce délai laissé sert d'anti-rebondissement au clavier. Ensuite démarre le scan du clavier. On active la dernière ligne, on lit le registre d'état du port et on déduit de la valeur présente sur ce port la touche pressée. Ensuite, on passe à la 4ème ligne et ainsi de suite jusqu'à la première ligne.

Le dernier fichier, et le plus important aussi, est screen_keyb.c. Au début de ce fichier, un certain nombre de fichiers sont inclus dans le programme, dont les deux définis précédemment. Cette séparation du programme en plusieurs fichiers rend plus facile la réutilisation des fonctions de contrôle de l'écran et du clavier dans une de vos réalisations personnelles, car c'est le but premier de ce montage. Ensuite sont déclarées quelques variables utilisées par le programme. table[16] sert à stocker la totalité de l'écran, tandis que tab1[8] et tab2[8] sont destinées à accueillir les deux lignes de 8 caractères. key contiendra la



2 Tracé
du circuit imprimé

3 Implantation
des éléments

actuellement pressée. Les touches ont reçu comme noms les 20 premières lettres de l'alphabet.

Réalisation

La réalisation est relativement simple. Une simple photocopie du dessin du circuit sur transparent suffit, car les pistes sont assez larges (**figure 2**). Le perçage s'effectuera directement en 1 mm car, aussi bien le connecteur Centronics que les boutons-poussoirs nécessitent au moins ce diamètre. Le bornier d'alimentation, quant à lui ainsi que le trimmer de réglage du contraste, nécessitent un perçage de diamètre 1,3 mm.

Comme vous l'avez sûrement remarqué, le nombre de straps est assez réduit. Cela a

Nomenclature

ELCD : écran LCD à bus parallèle

1 prise Centronics 36 broches femelle

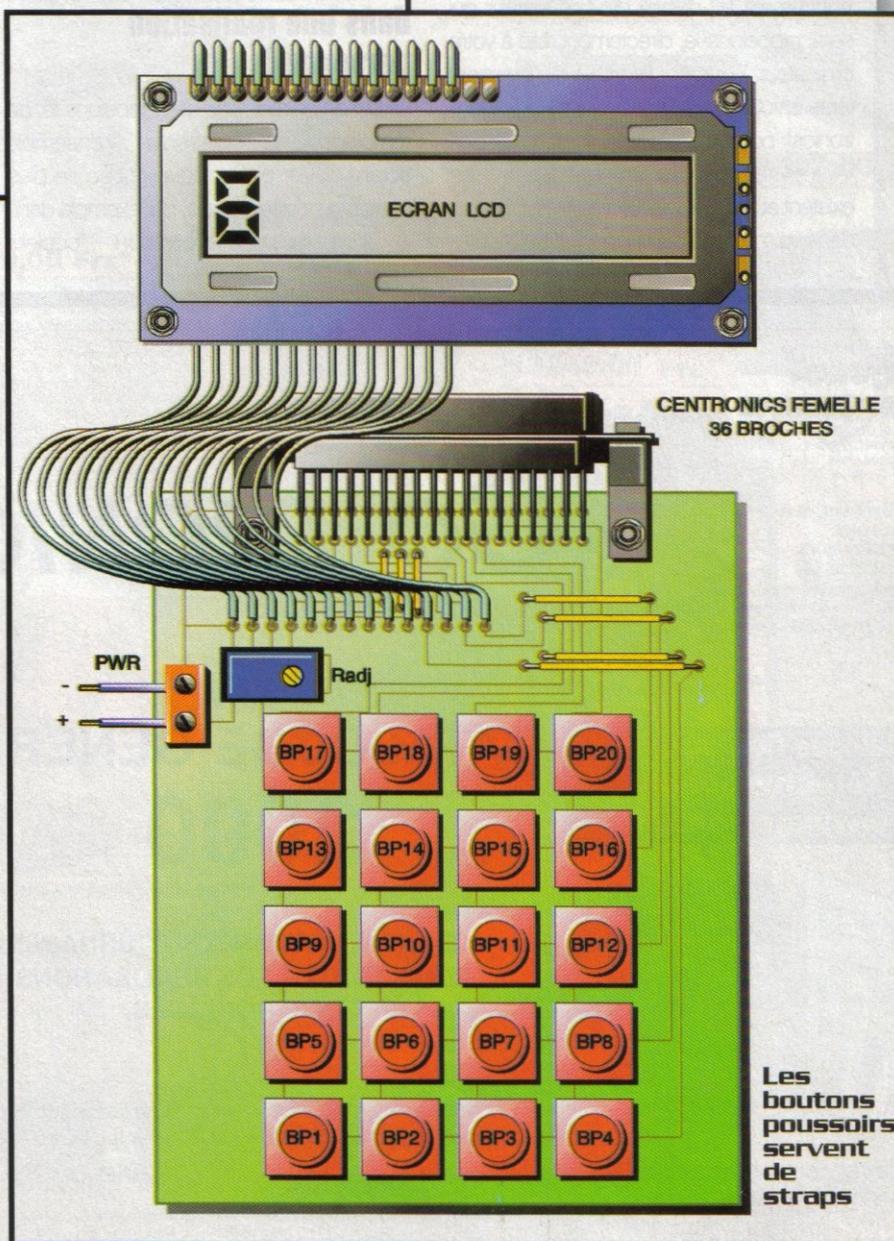
BP1 à BP20 : boutons poussoirs

WR : bornier 2 points

Radj : 10 k Ω linéaire horizontal

7 straps

touche actuellement pressée; i et j sont des variables d'importance mineure, ne servant que pour des boucles dans le programme. La fonction main est la première fonction exécutée lors du lancement d'un programme en C. Elle commence par autoriser l'accès aux entrées/sorties au programme via la fonction ioperm, comme Input Output Permission. Souvenons-nous que nous sommes dans le monde Unix, qui exerce un contrôle très strict sur le matériel, aussi bien que sur le logiciel. La suite du programme est très courte et claire grâce à toutes les fonctions déclarées dans screen.h et keyboard.h. Le programme commence par afficher une petite animation, puis le texte " Touche : " et la touche



Les
boutons
poussoirs
servent
de
straps

été possible grâce à l'utilisation de poussoirs particuliers (rassurez-vous, c'est le cas de la majorité de ceux que vous trouverez dans le commerce) : Les broches des poussoirs sont reliées deux à deux, ainsi, le poussoir sert de strap.

L'implantation (**figure 3**) commencera par les 7 straps. Attention, l'un d'entre eux, au moins, entre partiellement sous le connecteur Centronics (cela dépend du type de connecteur). Soudez ensuite un morceau de nappe à 14 conducteurs pour la connexion de l'afficheur. Terminez par les poussoirs, le réglage du contraste, le bornier et enfin le connecteur Centronics 36 broches femelle. Ce type de connecteur permet d'utiliser les câbles d'imprimante qui sont très bon marché.

Vérifiez ensuite qu'il n'y a aucun court-circuit, notamment au niveau du connecteur qui sera, rappelons-le, directement relié à votre ordinateur. Vous pouvez faire les premiers tests en utilisant le "Protecteur de port Centronics", publié dans le Hors-Série Interfaces PC n°4. Si minimes soient-ils, les risques existent et ne sauraient être négligés, surtout dans un montage qui apporte une alimenta-

tion externe. Une vérification minutieuse du circuit peut néanmoins vous éviter les coûts d'une telle sécurité supplémentaire.

Une fois le montage connecté au PC et celui-ci allumé, apportez une alimentation 5V continu à la platine.

Si vous utilisez DOS, lancez le petit programme de test en BASIC, livré sur le support accompagnant cette revue. Si vous utilisez GNU/Linux, commencez par désarchiver l'archive screen_keyb.tgz à l'aide de la commande 'tar xzvf screen_keyb.tgz', puis tapez 'make' dans le répertoire courant. La compilation devrait se passer sans problème; un warning pendant celle-ci est toutefois normal. Ensuite, tapez './screen_keyb' pour lancer le programme de test.

Mise en œuvre dans une réalisation personnelle

Le but de ce montage est, rappelons-le, de remplacer le classique ensemble écran+clavier par quelque chose de plus compact et plus adapté, par exemple dans le cas d'une application "copieur

d'EPROM". Il est clair qu'un programme de ce type n'a pas besoin d'un écran 14" et d'un clavier de 105 touches.

Nous ne traiterons ici que théoriquement le cas d'une application écrite en C pour GNU/Linux. Il est nécessaire pour faire les adaptations de disposer du code source du logiciel. Si tel n'est pas le cas, téléchargez sur Internet un logiciel équivalent dont vous disposerez des sources. Vous pouvez par exemple chercher sur <http://freshmeat.net>. Ce site recense une majorité d'applications tournant sous GNU/Linux. Ensuite éditez son code source et remplacez (ce n'est qu'un exemple) tous les appels à getch par des appels à keyscan, et tous les appels à sprint par des appels à send_screen. Prenez soin de rajouter les lignes '#include "screen.h"' et '#include "keyboard.h"', ainsi que les commandes d'initialisation de l'afficheur. Si tout cela vous semble obscur, mais que vous êtes motivés pour modifier un programme, vous pouvez contacter l'auteur par e-mail et il vous aidera dans la mesure du temps disponible.

M. LUCZAK
mluczak@april.org



devient

GO TRONIC

4, route Nationale - B.P. 13
08110 BLAGNY
TEL.: 03.24.27.93.42
FAX: 03.24.27.93.50
WEB: www.gotronic.fr
Ouvert du lundi au vendredi (9h-12h/14h-18h)
et le samedi matin (9h-12h).

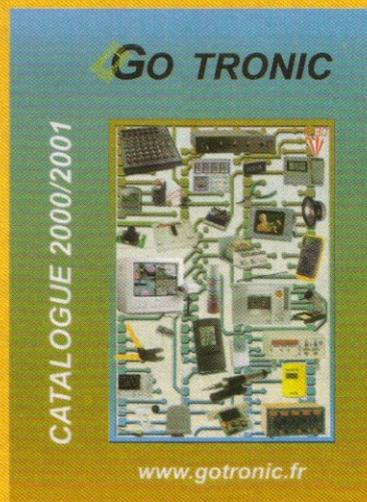
catalogue Go Tronic 2000
téléchargeable
www.gotronic.fr

CATALOGUE GENERAL 2000



Recevez le catalogue 2000 contre
29 FF (60 FF pour les DOM-TOM
et l'étranger).
Gratuit pour les Ecoles et les
Administrations.

LE CATALOGUE INCONTOURNABLE
POUR TOUTES VOS REALISATIONS
ELECTRONIQUES.



PLUS DE 300 PAGES de
composants, kits, livres,
logiciels, programmeurs,
outillage, appareils
de mesure, alarmes...

Veuillez me faire parvenir le nouveau catalogue général **GO TRONIC** (anc. Euro-composants). Je joins mon règlement de 29 FF (60 FF pour les DOM-TOM et l'étranger) en chèque, timbres ou mandat.

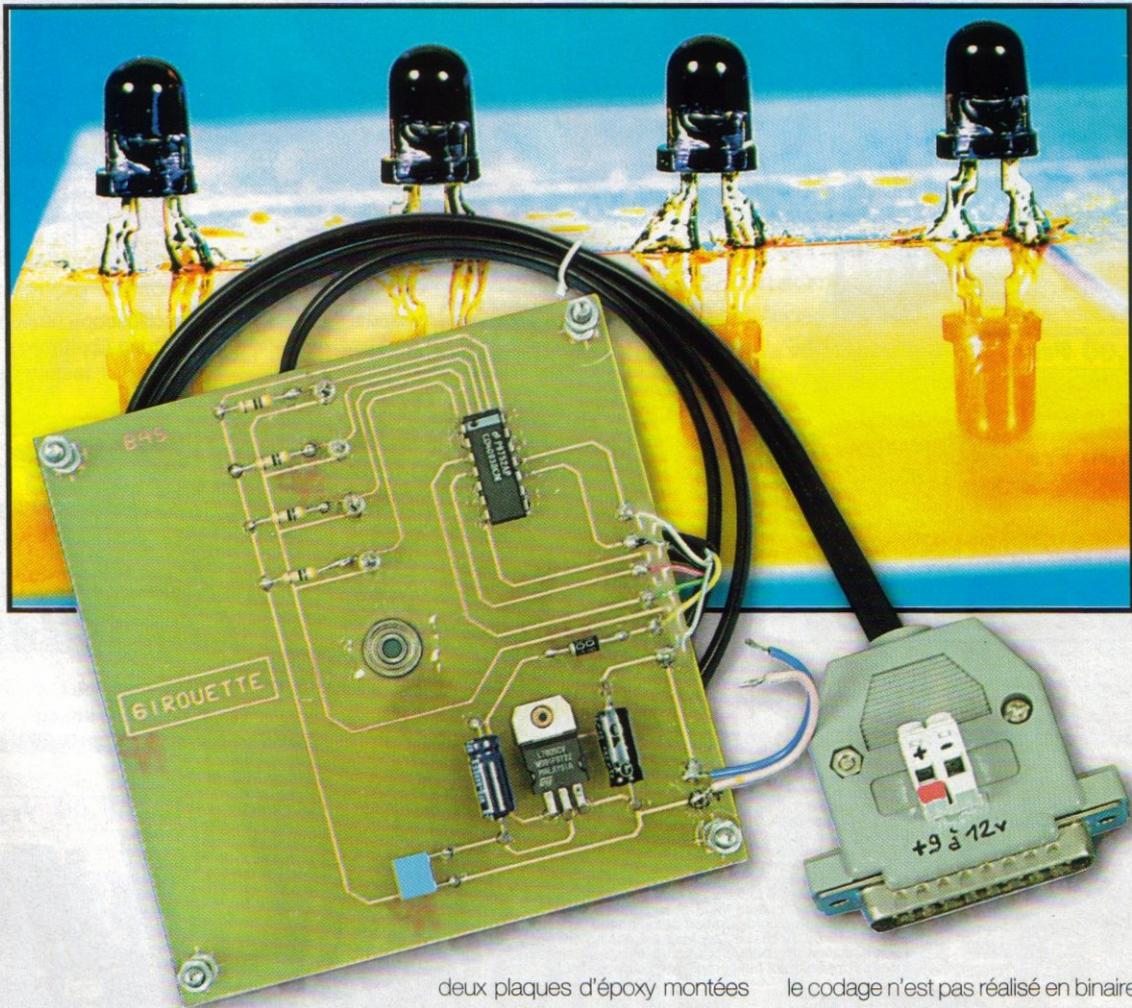
NOM : PRENOM :

ADRESSE :

CODE POSTAL :

VILLE :

Girouette électronique



Lorsque l'on s'intéresse à la météorologie, l'instrument le plus simple à réaliser est la girouette. Constituée d'une plaque de forme variable mobile autour d'un axe vertical fixé le plus souvent sur le toit, elle permet d'indiquer la direction du vent. Si cette solution est la plus simple elle est néanmoins peu pratique ; en effet il faut aller à l'extérieur pour lire l'orientation de la girouette.

Avec le montage présenté ici plus besoin de vous déplacer, la position occupée par la girouette s'affiche sur l'écran de votre ordinateur. Celle-ci étant déterminée par un ensemble de 4 barrières infrarouges qui balayent un disque de codage. Cette solution permet la lecture de 16 positions différentes et a surtout l'avantage de pratiquement supprimer toute force de frottement, ainsi la girouette est sensible au moindre souffle de vent.

Principe de fonctionnement

Le schéma mécanique de notre girouette vous est présenté **figure 1**. La partie fixe est constituée de

deux plaques d'époxy montées l'une au-dessus de l'autre. La plaque supérieure comporte les diodes IR émettrices qui sont au nombre de 4. Les phototransistors sont positionnés sur la plaque inférieure. Le disque de codage est solidaire de la partie tournante de la girouette. Le guidage de l'axe mobile est confié à deux roulements à bille, ce qui laisse une totale liberté de rotation.

Le disque de codage est représenté **figure 2**. Les parties blanches sont transparentes et laissent donc passer le faisceau IR, inversement les parties opaques (noires) coupent le faisceau. Le disque est découpé en 16 portions, chaque portion comporte 4 secteurs. Ainsi, on réalise un codage de chacune des 16 positions que peut occuper la girouette. Vous aurez sûrement remarqué que

le codage n'est pas réalisé en binaire pur mais en code de Gray (du nom de son inventeur). Ce système de numérotation permet d'éviter un comportement erratique qui survient lors du passage d'une position à la suivante, car un seul bit peut changer d'état d'une position à l'autre. Par exemple imaginons que l'on passe d'une position 5, codée en binaire par la valeur 0 1 0 1, à la position suivante 6 codée par la valeur 0 1 1 0. Du passage de 5 à 6 il y a deux bits qui ont changé d'état, ce qui peut générer des erreurs lorsque les faisceaux IR se trouvent sur la frontière entre la position 5 et la position 6, on peut obtenir la valeur 0 1 0 0 qui correspondrait à la position 4. En code de Gray, la position 5 est codée par la valeur 0 1 1 1 et la position 6 par la valeur 0 1 0 1, ainsi lors du pas-

sage de la position 5 à 6, nous avons uniquement le bit numéro 2 (secteur C) qui change d'état.

Chaque portion correspond à une position de la girouette, nous avons donc les orientations suivantes : N, 1/4 Nord/Ouest, Nord/Ouest, 3/4 Nord/Ouest, Ouest, 3/4 Sud/Ouest, Sud/Ouest, 1/4 Sud/Ouest, Sud, 1/4 Sud/Ouest, Sud/Ouest, 3/4 Sud/Ouest, Ouest, 3/4 Nord/Ouest, Nord/Ouest, 1/4 Nord/Ouest.

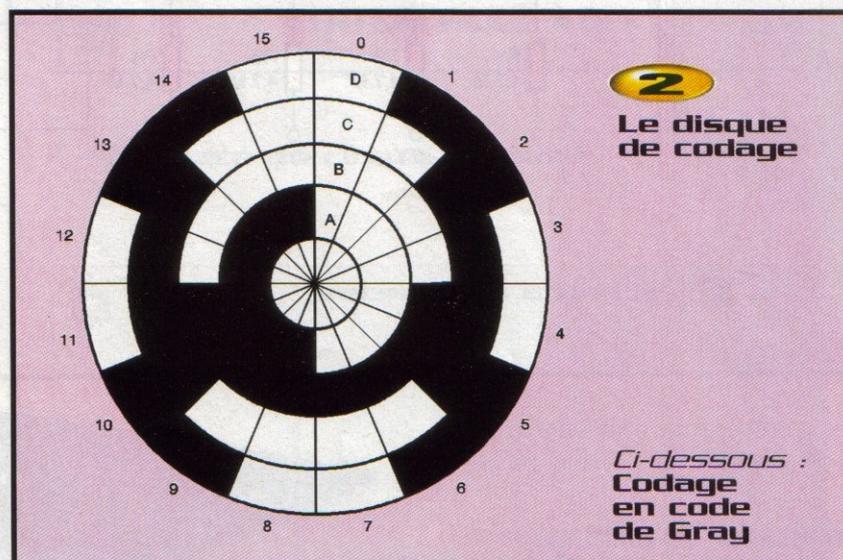
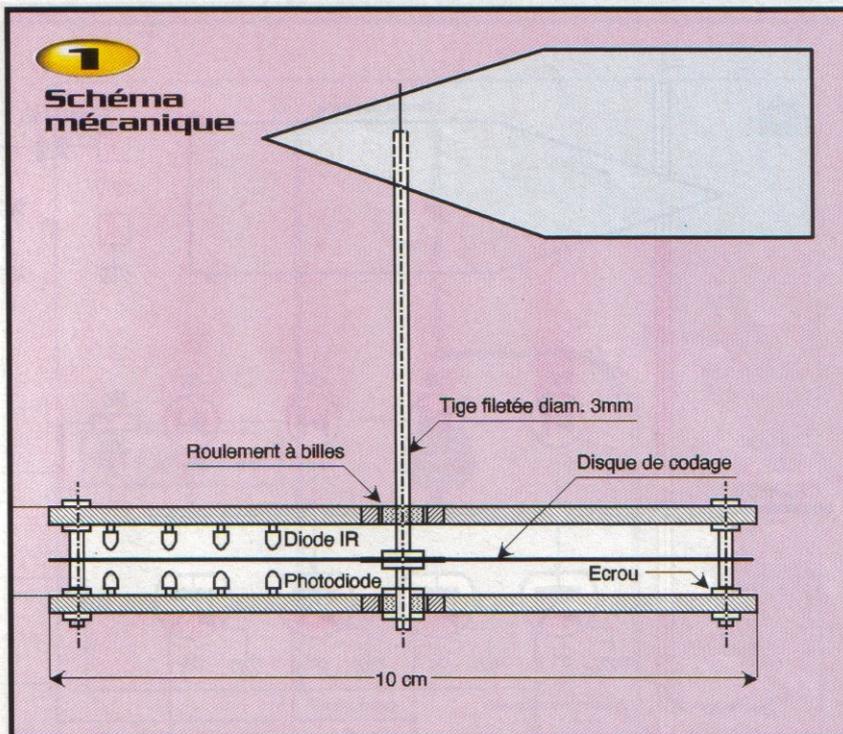
Exemple : si la valeur lue par les phototransistors est 1 0 1 0, cela signifie que le vent souffle suivant la direction Est (**voir tableau de la figure 2**).

Schéma électrique

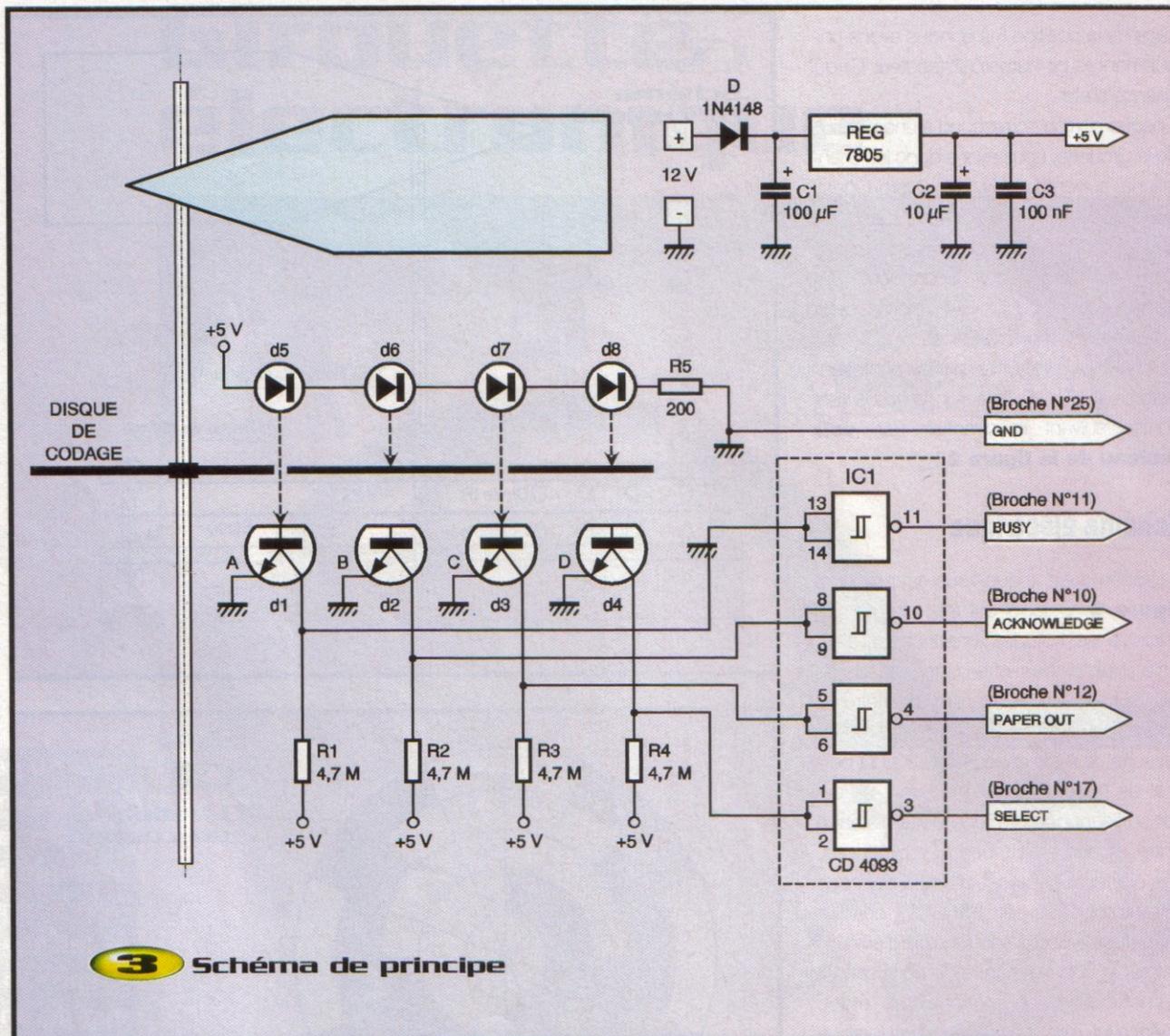
Le schéma électrique vous est présenté **figure 3**. Celui-ci est très simple. On retrouve les 4 diodes émettrices d₅ à d₈. Il s'agit de diodes infrarouges standards (réf. SFH 409) mais ayant un diamètre de 3mm. Elles sont toutes les quatre reliées en série, la résistance R₅ de 200 Ω permet de limiter l'intensité à 1 mA, ce qui est suffisant compte tenu de la distance entre diodes émettrices et réceptrices. Les diodes réceptrices d1 à d4 sont des phototransistors (réf. SFH309), chaque collecteur est relié à la tension d'alimentation par une résistance de polarisation. L'émetteur du phototransistor étant, quant à lui, relié à la masse. Ainsi lorsque le phototransistor reçoit le faisceau IR, celui-ci devient passant, on relève entre le collecteur et la masse une tension proche de 0V. Lorsque le faisceau est coupé, le phototransistor est bloqué, on relève sur sa sortie une valeur proche de 3,5V. Chacune des 4 sorties est reliée à une porte logique inverseuse trigger de Schmitt, tant que la tension appliquée à l'entrée de la porte est inférieure à 2,5V (tension de seuil), la sortie est à l'état haut, inversement si la tension d'entrée est supérieure à 2,5V la sortie passe à l'état bas. Ceci permet d'envoyer à l'ordinateur des signaux bien carrés.

La transmission du code de Gray s'effectue sur les lignes busy (broche 11), acknowledge (broche 10), paper out (broche 12) et select (broche 13) du port parallèle d'un ordinateur. Rappelons que l'entrée busy est active à l'état bas.

L'alimentation du montage est réalisée en +12V, tension régulée par le circuit 7805,



Portion	direction	A	B	C	D	valeur
0	N	0	0	0	0	0
1	1/4 NO	0	0	0	1	8
2	NO	0	0	1	1	12
3	3/4 NO	0	0	1	0	4
4	O	0	1	1	0	6
5	3/4 SO	0	1	1	1	14
6	SO	0	1	0	1	10
7	1/4 SO	0	1	0	0	2
8	S	1	1	0	0	3
9	1/4 SE	1	1	0	1	11
10	SE	1	1	1	1	15
11	3/4 SE	1	1	1	0	7
12	E	1	0	1	0	5
13	3/4 NE	1	0	1	1	13
14	NE	1	0	0	1	9
15	1/4 NE	1	0	0	0	1



3 Schéma de principe

deux condensateurs permettent de filtrer et de découpler cette tension d'entrée.

Réalisation du montage

Les figures 4 et 5 représentent respectivement la partie supérieure et inférieure de l'élément fixe de notre girouette, chacune ayant une dimension de 10 x 10cm. Après gravure des deux plaques, vous percerez les différentes pastilles à l'aide d'un foret de 0,8mm de diamètre, celles destinées à recevoir le régulateur seront percées avec un foret de 1mm. Avant de passer à la mise en place des composants, il vous faudra également percer les deux pastilles centrales à l'aide



d'un foret de 10mm. Une fois ce travail réalisé, vous y positionnerez les roulements à billes, qui seront alors collés à l'aide de colle forte. Point particulier pour ce montage, tous les composants seront soudés côté cuivre. Concernant les condensateurs C₁ et C₂, n'utilisez pas de modèles verticaux, ceux-ci viendraient entraver la rotation du disque de codage. Le condensateur C₃ sera soudé à 90° avant d'être soudé. Les diodes et autres phototransistors seront soudés de manière à ce qu'il y ait une distance d'environ 8mm entre la pointe de la diode et la surface de la plaque. Une fois la mise en place des composants effectuée, vous pouvez déjà véri-

fier le bon fonctionnement des diodes IR. Mettez le montage sous tension en lui appliquant une tension de l'ordre de 12V. N'oubliez pas de relier les deux plaques entre elles, sinon la plaque supérieure ne sera pas alimentée. En superposant les deux plaques telles que les diodes émettrices et les phototransistors soient alignés, vous vérifierez que la tension entre l'émetteur et la masse de chacun des transistors est proche de 0V.

En retirant la plaque supérieure, les phototransistors ne recevant plus le faisceau, vous devez relever entre la masse et l'émetteur une tension supérieure à 2,5V (tension de seuil des portes du 4093). Si ce n'est pas le cas, c'est que la résistance de polarisation ne convient pas, essayez avec une valeur plus faible de l'ordre de 470 k Ω .

Le disque de codage sera réalisé en imprimant le fichier "DISQUE.BMP", utilisez de préférence un papier assez épais. Les parties banches seront découpées à l'aide d'un cutter, la partie restante (noire) sera collée sur un disque en matière plastique transparente suffisamment rigide (épaisseur d'environ 2 mm), vous pouvez par exemple le découper dans un boîtier de CD ROM. Ensuite, vous percerez le centre du disque à l'aide d'un foret de 3mm.

Il ne vous reste plus qu'à assembler les différents éléments :

Préparez tout d'abord la plaque inférieure en la munissant de 4 vis, Solidarisez le disque de codage et la tige filetée à l'aide de deux écrous, utilisez deux rondelles afin d'éviter de déformer le disque lors du serrage,

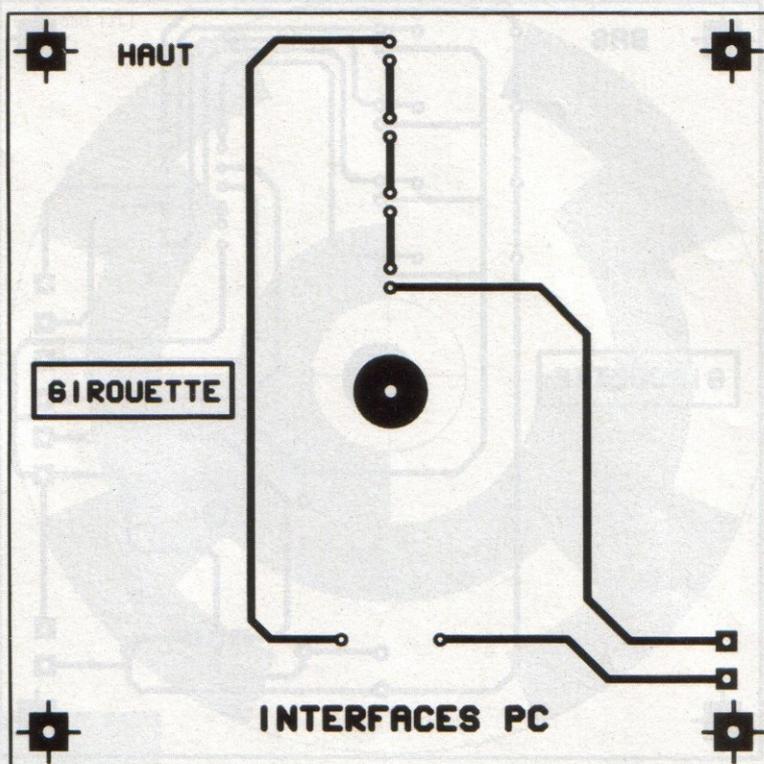
Engagez alors la tige filetée dans le roulement inférieur que vous fixerez à l'aide de deux écrous,

Enfilez la plaque supérieure sur la tige filetée, Réglez ensuite l'écartement entre les deux plaques à 2cm, utilisez à cet effet des entretoises ou des écrous,

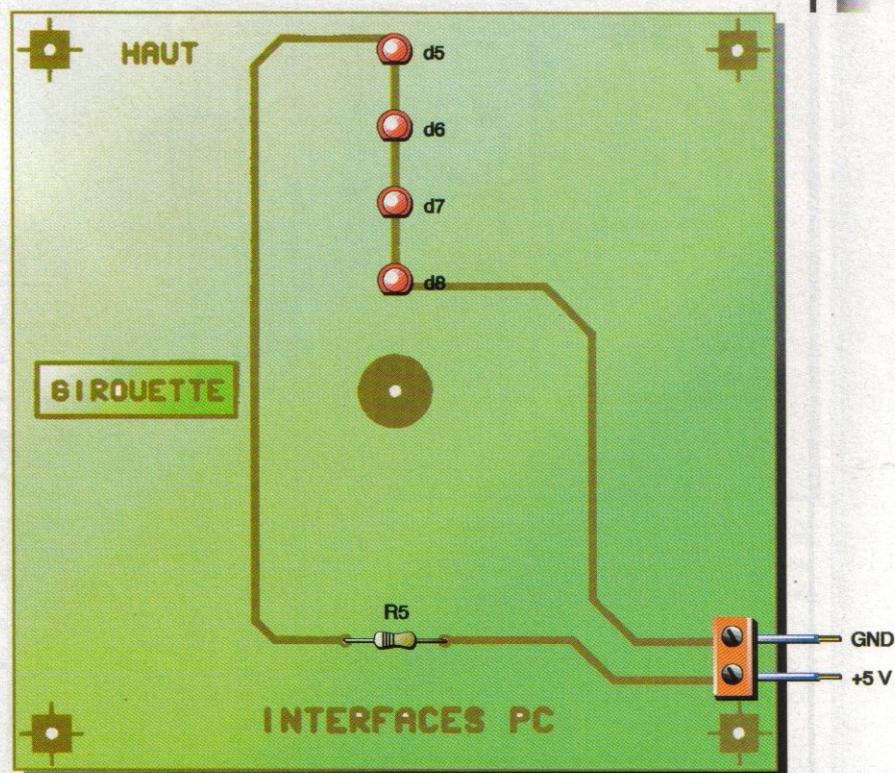
Vérifiez que le disque possède une totale liberté de rotation.

La plaque mobile sera taillée dans du bois tendre. Vous pourrez vous inspirer de la forme présentée sur les schémas, seul point important : la surface à gauche de l'axe de rotation sera plus petite que celle de droite afin d'offrir une résistance moindre au vent, ce qui garantira une bonne orientation.

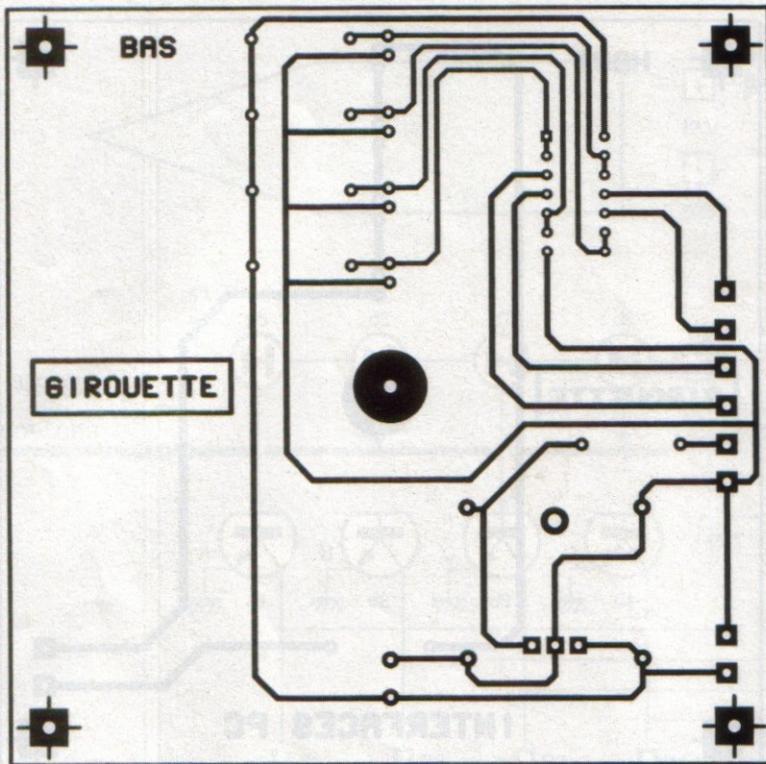
Fixez cette plaque à l'aide de colle sur la tige filetée en vous assurant que la portion 0 (celle utilisée pour codifier le Nord) est exac-



4 Tracé du circuit imprimé

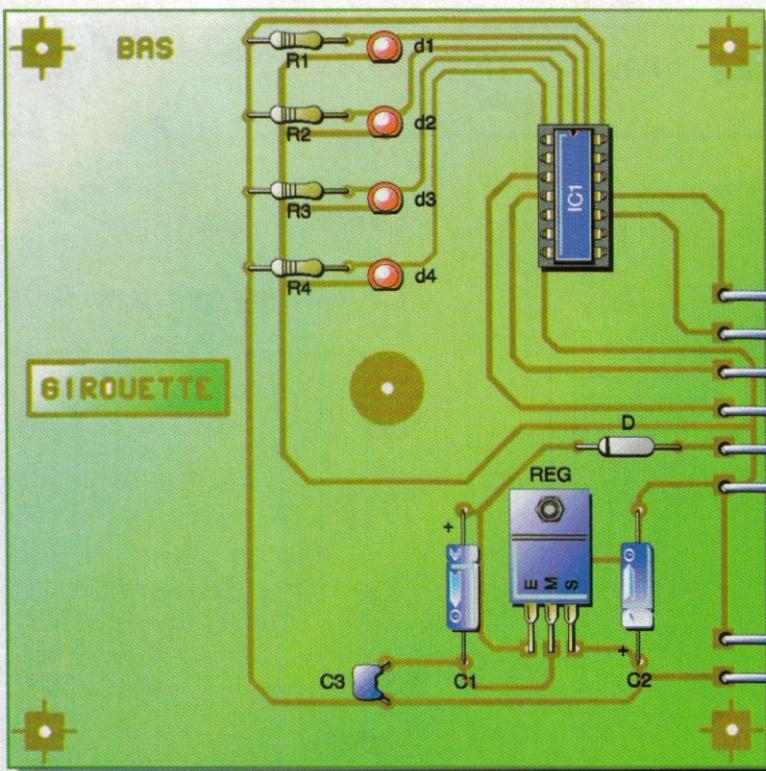


6 Implantation des éléments



6 IROUETTE

5 Tracé du circuit imprimé



6 IROUETTE

7 Implantation des éléments où les composants sont soudés côté "cuivre"

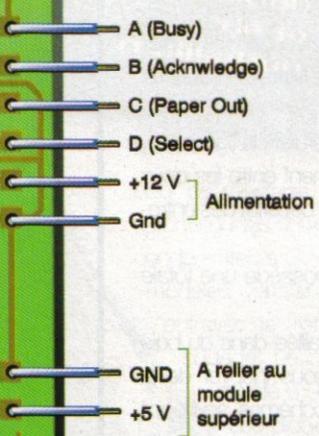
Nomenclature

Partie électronique

- R₁ à R₄ : 4,7 MΩ
- R₅ : 200 Ω
- D₁ à d₄ : phototransistor IR SFH 309 Δ 3mm (CONRAD Electronic)
- D₅ à d₆ : diodes IR SFH 409 Δ 3mm (CONRAD Electronic)
- IC₁ : CD4093
- C₁ : 100 μF horizontal chimique
- C₂ : 10 μF horizontal chimique
- C₃ : 100 nF LCC
- d : diode 1N4148
- REG : régulateur 7805
- CN : connecteur DB25 femelle, à câbler
Câble 6 conducteurs

Partie mécanique

- 1 plaque de bois tendre 14 x 6cm épaisseur 3mm
- 1 disque en plastique transparent Δ 10cm épaisseur 2mm
- 1 tige filetée Δ 3mm
- 2 roulements à billes Δ extérieur 10mm Δ intérieur 3mm
- 4 vis longueur 2,5mm
- 8 écrous + rondelles



tement alignée avec la pointe de la plaque. La liaison au PC sera réalisée avec un câble comportant 6 conducteurs munis d'un connecteur DB25, la longueur de celui-ci pourra atteindre plusieurs dizaines de mètres sans craindre d'éventuels parasites. En effet, les signaux envoyés à l'ordinateur auront une fréquence très basse (le vent ne changeant pas souvent de direction).

Le logiciel

Le programme permettant de visualiser graphiquement la position occupée par la girouette, a été développé avec DELPHI 4 et fonctionne sous Windows 95/98 et NT. L'installation du logiciel consiste simplement à lancer l'exécution du fichier "SETUP.EXE" et à suivre les instructions affichées. Par défaut, tous les fichiers du programme seront copiés sous le répertoire "C:\Program Files", mais vous pouvez aussi choi-



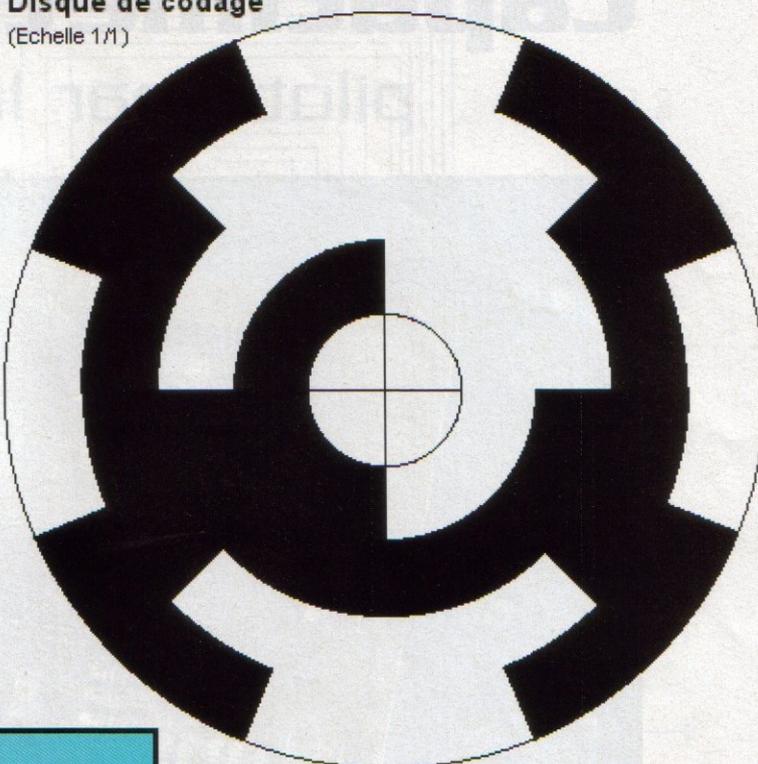
Aspect du disque

sir votre propre emplacement. Un raccourci nommé "Girouette" est automatiquement placé dans votre bureau. Cliquez 2 fois de suite sur cet icône et le programme s'exécute. Une copie d'écran vous est présentée figure 6. Une flèche rouge vient indiquer la direction du vent. Une petite fenêtre permet d'indiquer la position exacte de la girouette. Il est possible d'utiliser le montage sur LPT1 ou sur LPT2, le choix s'effectuant d'un simple click de souris.

Lors de la première exécution du programme, orientez manuellement la girouette vers le nord. A l'écran la flèche rouge doit indiquer également le nord. Si ce n'est pas le cas, c'est que vous avez mal positionné la plaque lors de son collage sur l'axe. Ce n'est pas grave, pour corriger il suffit de débloquer légèrement les deux écrous qui

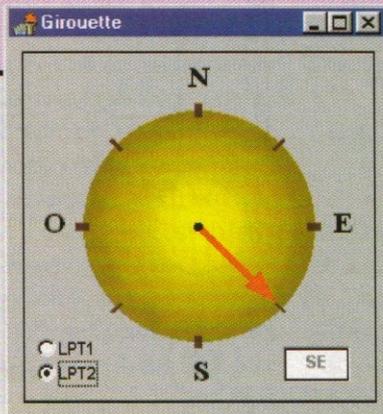
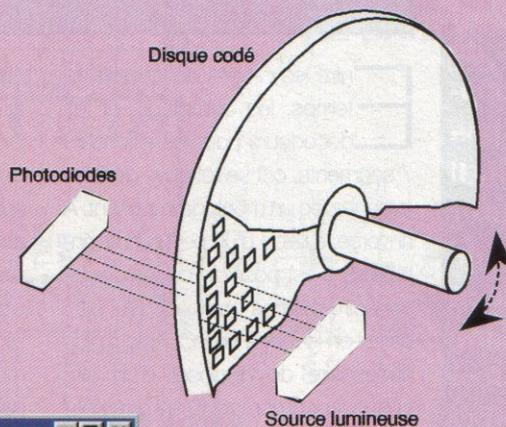
Disque de codage

(Echelle 1/1)



8 Réalisation du disque à l'échelle 1

9 Principe du codage



10 Copie d'écran

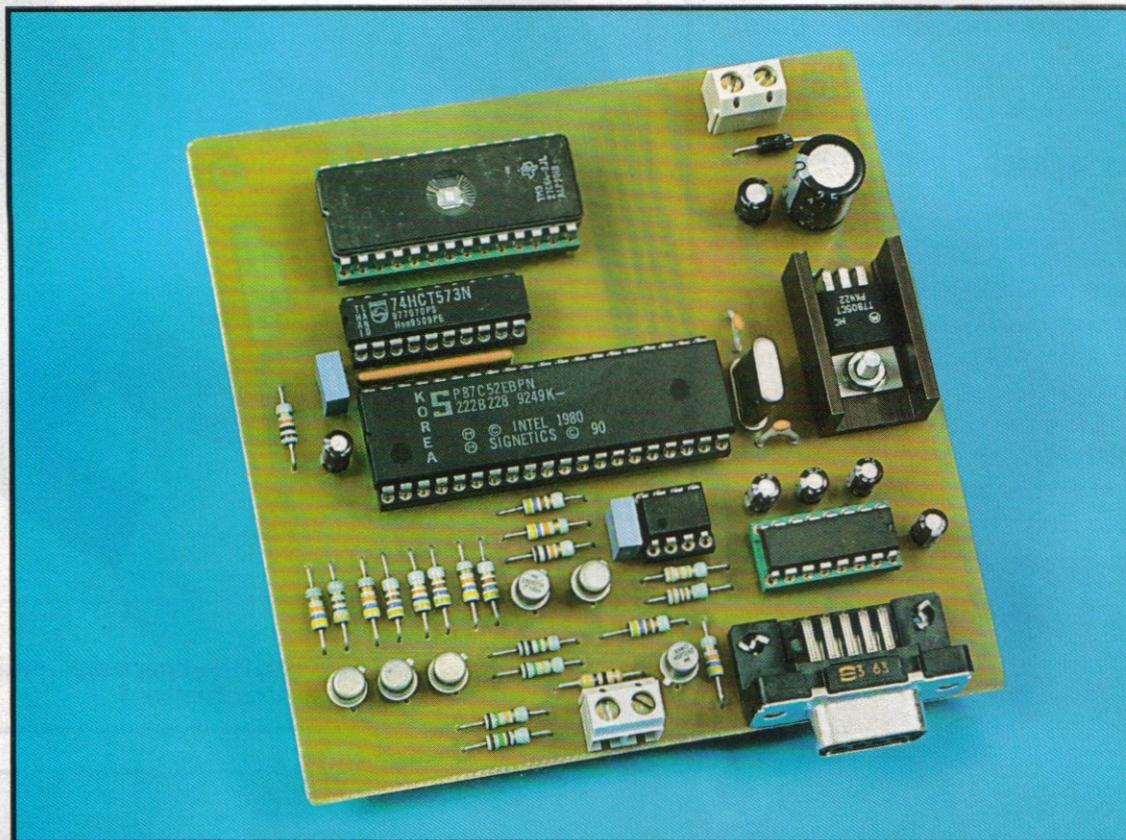
maintiennent le disque de codage puis de le faire tourner jusqu'à ce que la position sur l'écran de l'ordinateur soit identique à celle occupée par la plaque mobile de la girouette. Il ne vous reste alors plus qu'à bloquer les deux écrous afin de rendre solidaire le disque et l'axe de rotation.

Pour conclure, signalons que ceux qui voudraient profiter de ce montage sous MS DOS trouveront également un programme écrit en QBasic.

D. REY

Capacimètre

piloté par liaison RS232



Le principe de mesure d'un capacimètre n'est pas bien compliqué. Il suffit de mesurer le temps de charge du condensateur à l'aide d'une tension fixe au travers d'une résistance dont la valeur est parfaitement connue. En logique discrète ce type d'appareil prend vite des proportions imposantes, surtout si la gamme de mesure est étendue.

Entre les circuits de la base de temps, les compteurs et les décodeurs pour les afficheurs 7 segments, on se retrouve généralement avec un montage imposant. A l'inverse, à l'aide d'un petit microcontrôleur, il est possible de réaliser un capacimètre très compact. Qui plus est, si on le relie à un PC, il n'est plus nécessaire de disposer d'un affichage encombrant. C'est précisément le montage que nous vous proposons de réaliser avec nous dans ces pages.

Schéma

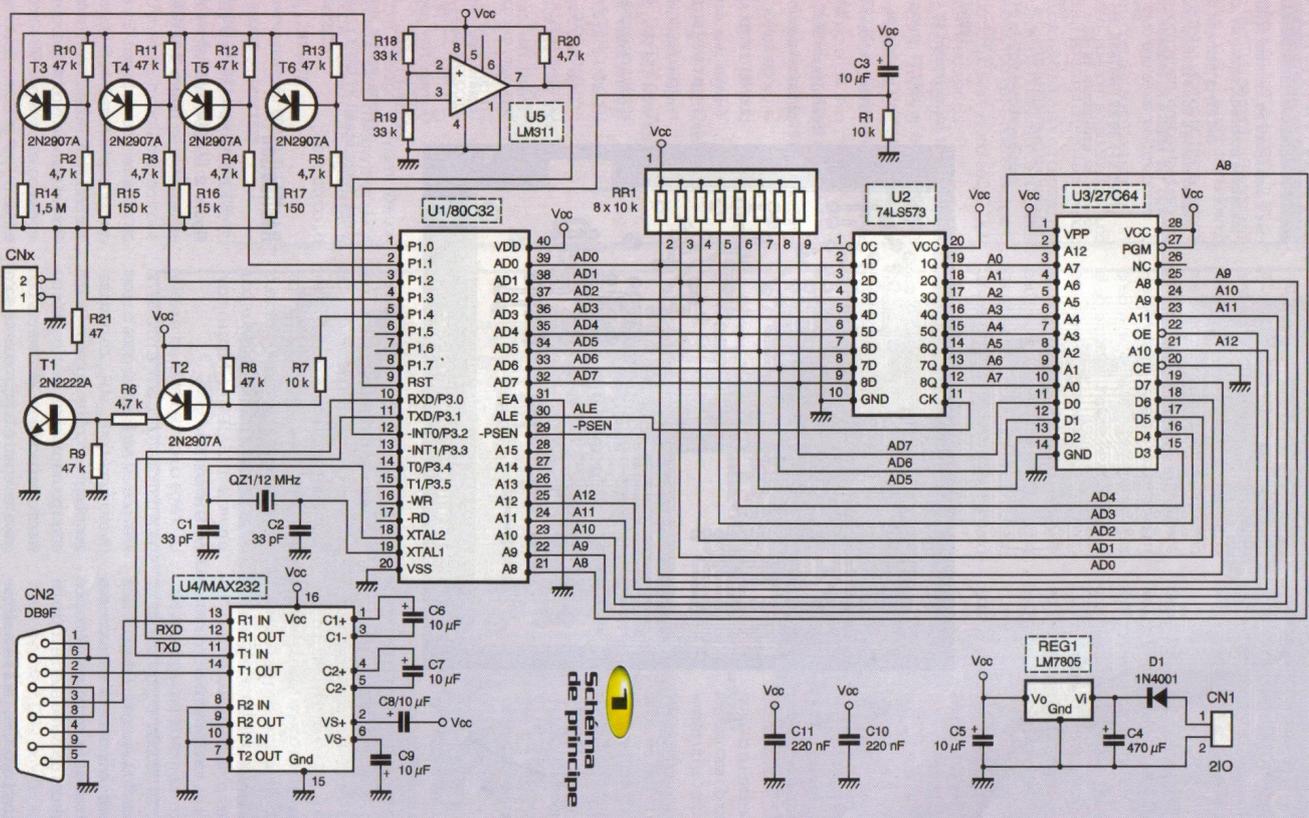
Le schéma de notre montage est reproduit en **figure 1**. Le cœur de ce montage est, bien entendu, le microcontrôleur U_1 , qui est un modèle 80C32. Ce microcontrôleur ne dispose pas de mémoire EPROM interne. Il faut donc lui adjoindre une EPROM externe qui contiendra le

programme à exécuter. Pour notre application, une EPROM de type 27C64 suffira largement. Le bus des données du microcontrôleur est multiplexé avec le poids faible du bus des adresses de sorte qu'il est nécessaire d'ajouter le circuit latch U_2 pour réaliser le démultiplexage. Le circuit U_2 est synchronisé le plus simplement du monde grâce au signal ALE généré par le microcontrôleur. Le raccordement du bus des données et du bus des adresses, dans un ordre apparemment incohérent, permet de simplifier le "routage" du circuit imprimé. Ceci est possible à condition de programmer l'EPROM U_3 avec un fichier traité de façon adéquate.

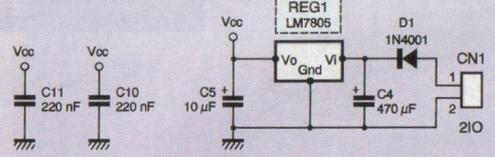
Si vous lisez régulièrement ces pages, vous devez commencer à y être habitué. Nous ne nous attardons donc pas plus longtemps sur ce point.

Le microcontrôleur U_1 dispose d'un oscillateur interne qui nécessite seu-

lement 1 quartz et 2 condensateurs pour être fonctionnel. Le circuit de remise à zéro du microcontrôleur est composé d'une simple cellule RC. Le port série du microcontrôleur est mis à profit pour ce montage, ce qui arrange bien nos affaires, car il n'est pas nécessaire d'ajouter un circuit spécialisé (UART). Par contre, les sorties RXD et TXD sont aux niveaux TTL ce qui n'est pas directement compatible avec une liaison RS232. Il faut transformer les niveaux 0 ou 5V en niveaux -12V et +12V (en réalité +/-9V à +/-15V). Etant donné que notre montage ne dispose que d'un régulateur 5VDC, il est plus simple d'utiliser un circuit MAX232 pour transformer les niveaux TTL en niveaux RS232. En effet, ce circuit contient des convertisseurs DC-DC qui permettent d'élever la tension VCC et de l'inverser pour obtenir les niveaux requis. Les condensateurs C_6 à C_9 sont justement nécessaires à la mise



Schema de principe



en œuvre des convertisseurs internes du circuit MAX232. La connectique retenue pour CN₂ permet de connecter notre montage au port série de votre PC par un câble fils à fils direct (pas besoin de croiser RX et TX, etc...).

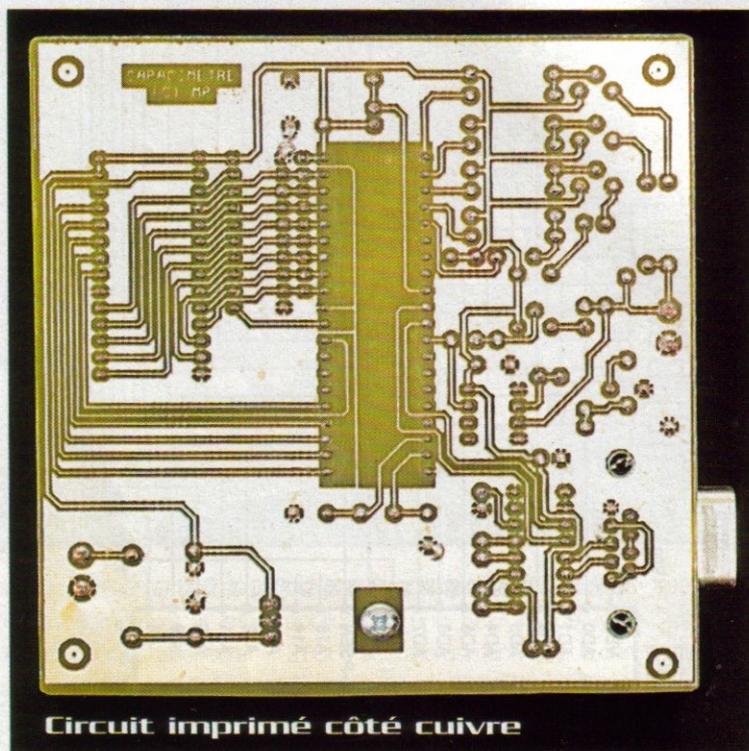
Le montage sera alimenté par une tension de 12VDC qui n'a pas besoin d'être stabilisée. Une tension correctement filtrée fera très bien l'affaire, à condition de pouvoir fournir au moins 250mA sous 12VDC. La diode D₁ permet de protéger le montage en cas d'inversion du connecteur d'alimentation.

La partie réellement spécifique de notre montage est constituée des transistors T₁ à T₆ et du comparateur U₅. Les transistors T₃ à T₆ permettent de charger le condensateur que l'on souhaite mesurer à l'aide de l'une des résistances R₁₄ à R₁₇, selon la gamme de la mesure. Le choix du transistor qui sera actif est sous contrôle du microcontrôleur. Ce dernier doit placer le port de sortie correspondant à l'état bas pour "tirer" un courant de base sur le transistor choisi (puisqu'il s'agit de transistor PNP). Vous noterez que tous les transistors du montage sont utilisés en commutation tout ou rien et qu'ils disposent tous d'une résis-

tance placée entre leur émetteur et leur collecteur. Ces résistances sont très importantes, car elles permettent d'absorber le courant de fuite des ports du microcontrôleur, ce qui garantit l'état parfaitement bloqué des transistors lorsqu'ils ne sont pas pilotés. Sans ces résistances, le courant de fuite des ports du microcontrôleur (1µA environ) se transformerait en courant de base, ce qui, multiplié par le gain des transistors, se traduirait par un courant de l'ordre de 100µA dans le collecteur. Cela fausserait complètement la mesure des condensateurs (on ne pourrait même plus mesurer des condensateurs de l'ordre de 10 nF).

Avant chaque mesure, il faut s'assurer que le condensateur sous test est correctement

déchargé (surtout si vous enchaînez une nouvelle mesure rapidement). C'est le transistor T₁ qui va s'occuper de cette tâche. La résistance R₂₁, en série avec le collecteur du transistor T₁, sert à limiter le courant de décharge du condensateur à une valeur acceptable. Cette précaution est très importante pour la mesure des condensateurs de forte valeur (10 µF et plus). Sans la résistance, le courant transitoire de la décharge du condensateur serait très élevé, au point que le transistor ne serait plus correctement saturé. Pour des condensateurs importants, cela se traduit par une énergie dissipée par T₁ très élevée,



Circuit imprimé côté cuivre

ce qui pourrait l'endommager. La résistance R₂₁ a donc été choisie pour limiter le courant de décharge à 100mA environ. Du coup, le temps de décharge des condensateurs de forte valeur (4700 µF par exemple) peut être très long. Tout ceci aura une incidence sur le temps de mesure pour la gamme 10 µF à 9999 µF. Nous en reparlerons un peu plus loin.

La charge du condensateur est surveillée par le comparateur U₅ dont le potentiel de référence est fixé à VCC/2 grâce au pont diviseur formé par R₁₈ et R₁₉. Le comparateur n'est pas monté en trigger de Schmitt ce qui pourra provoquer des instabilités sur la sortie de ce dernier. Ceci n'est pas bien grave, eu égard à la façon dont ce signal sera traité par le logiciel du microcontrôleur.

Tant que le condensateur n'est pas suffisamment chargé, la sortie sur comparateur U₅ est maintenue à l'état haut grâce à la résistance R₂₀. L'entrée -INT0 du microcontrôleur est donc à 1, ce qui autorise le fonctionnement du compteur interne T0. Le compteur est cadencé par l'horloge interne du microcontrôleur après une division par 12. Compte tenu de la fréquence retenue pour le quartz de ce montage, cela nous procure une base de temps de 1µs qui est très pratique pour la suite des calculs.

Lorsque le condensateur est suffisamment chargé, la tension à ses bornes dépasse VCC/2. Dès lors, la sortie du comparateur U₅ bascule à l'état bas, ce qui inhibe automatiquement le compteur T0. Le programme du microcontrôleur n'a plus qu'à lire le contenu du compteur T0 pour connaître le temps que le condensateur a mis pour se charger jusqu'à VCC/2.

Ce temps est donné par la formule théorique $T = 0,7 \times R \times C$, R étant la valeur de la résistance pour la gamme de la mesure en cours.

Connaissant le temps T, il est donc très facile pour notre microcontrôleur de calculer la valeur du condensateur :

$C = T / (0,7 \times R)$. Bien entendu, pour chaque

nouvelle mesure, le logiciel du microcontrôleur se chargera d'inhiber et de remettre à zéro le compteur T0 pendant la décharge du condensateur.

Réalisation

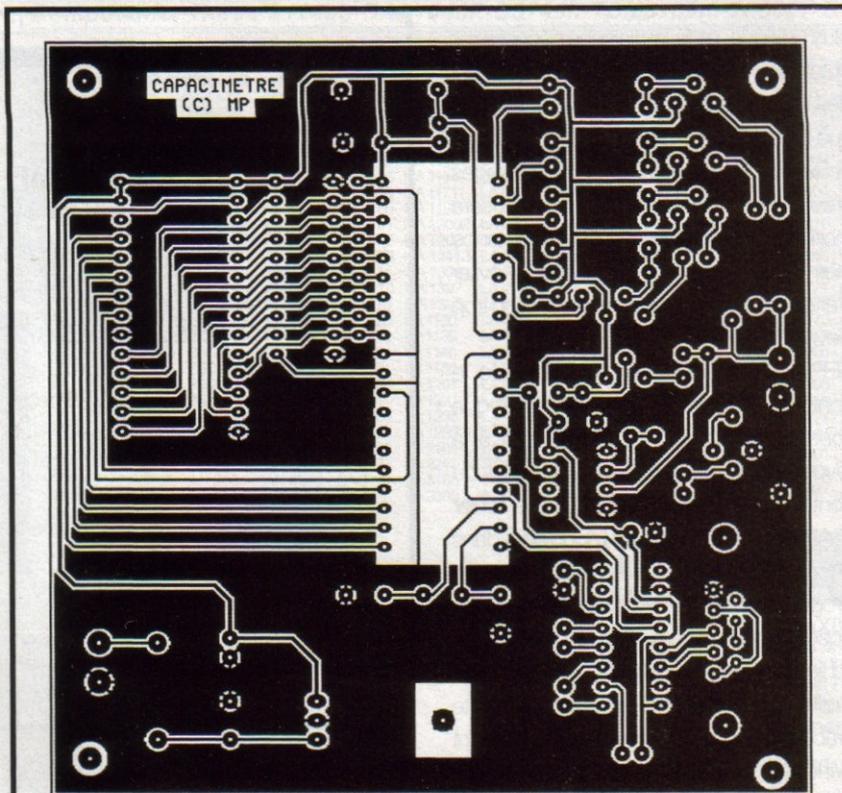
Le dessin du circuit imprimé est visible en **figure 2**. La vue d'implantation associée est reproduite en **figure 3**. Les pastilles seront percées à l'aide d'un foret de 0,8mm de diamètre, pour la plupart. En ce qui concerne les borniers à vis, D₁ et le régulateur REG₁, il faudra percer avec un foret de 1mm de diamètre. N'oubliez pas de percer les passages des vis pour REG₁ et CN₂ à l'aide d'un foret de 3,5mm de diamètre.

Comme d'habitude, procurez-vous les

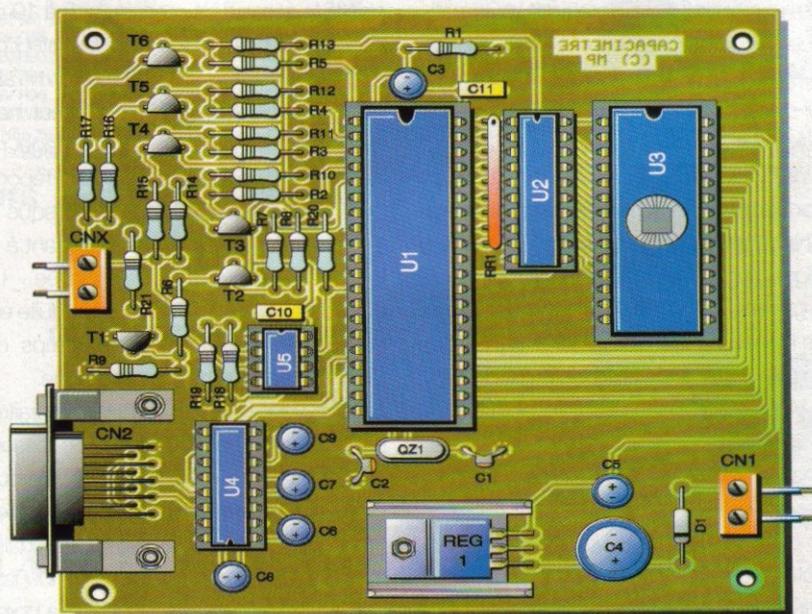
Nomenclature

- CNx, CN₁ : borniers de connexion à vis
2 plots, au pas de 5,08mm, à souder sur
circuit imprimé, profil bas
- CN₂ : connecteur SubD 9 points femelle,
sorties coudées, à souder sur circuit
imprimé
(ex. : réf. HARTING 09 66 112 7601)
- C₁, C₂ : 33 pF céramique au pas de
5,08mm
- C₃, C₅ à C₉ : 10 µF/25V sorties radiales
- C₄ : 470 µF/25V sorties radiales
- C₁₀, C₁₁ : 220 nF
- D₁ : 1N4001
(diode de redressement 1A/100V)
- QZ₁ : quartz 12 MHz en boîtier HC49/U
- REG₁ : régulateur LM7805 (5V) en boî-
tier TO220 + dissipateur thermique
18°C/W
(ex. : Shaffner réf. RAWA 400 9P)
- RR₁ : réseau résistif 8x10 kΩ en boîtier
SIL
- R₁, R₇ : 10 kΩ 1/4W 5%
(marron, noir, orange)
- R₂ à R₆, R₂₀ : 4,7 kΩ 1/4W 5%
(jaune, violet, rouge)
- R₈ à R₁₃ : 47 kΩ 1/4W 5%
(jaune, violet, orange)
- R₁₄ : 1,5 MΩ 1/4W 5%
(marron, vert, vert)
- R₁₅ : 150 kΩ 1/4W 5%
(marron, vert, jaune)
- R₁₆ : 15 kΩ 1/4W 5%
(marron, vert, orange)
- R₁₇ : 150 Ω 1/4W 5%
(marron, vert, marron)
- R₁₈, R₁₉ : 33 kΩ 1/4W 5%
(orange, orange, orange)
- R₂₁ : 47 Ω 1/4W 5% (jaune, violet, noir)
- T₁ : 2N2222A
- T₂ à T₆ : 2N2907A
- U₁ : microcontrôleur 80C32 (12 MHz)
- U₂ : 74LS573 ou 74HCT573
- U₃ : EPROM 27C64 temps d'accès 200 ns
- U₄ : driver de lignes MAX232
- U₅ : LM311P

composants avant de dessiner le circuit, au cas où il vous faudrait adapter un peu l'implantation. Il n'y a pas de difficulté particulière pour l'implantation. Soyez tout de même attentifs au sens des condensateurs et des circuits intégrés et respectez scrupuleusement le découplage des



2 Tracé du circuit imprimé



3 Implantation des éléments

lignes d'alimentations si vous voulez éviter les surprises. Veillez bien à choisir un connecteur femelle pour CN₂. Car un modèle mâle s'implante parfaitement sur le circuit imprimé, mais les points de connexions se retrouvent inversés symétriquement par rapport à l'axe ver-

tical. En ce qui concerne le câble nécessaire pour relier notre montage à un PC, il vous suffira de fabriquer un câble équipé d'un connecteur DB9 mâle d'un côté et d'un connecteur DB9 femelle de l'autre côté (liaison fil à fil de la broche 1 à la broche 9). L'utilisation de connecteurs à sertir est

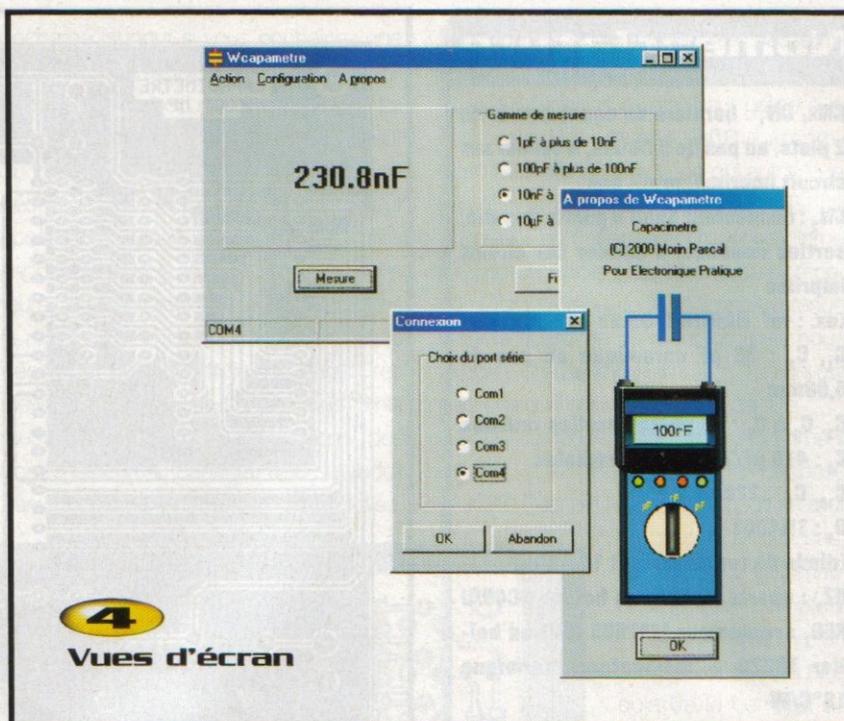
plus pratique, mais les liaisons nécessaires étant peu nombreuses, vous pourrez utiliser des connecteurs à souder. Enfin, ajoutons que le connecteur CN₂ sera immobilisé par deux boulons montés dans les passages prévus à cet effet. Le régulateur REG₁ sera monté sur un radiateur ayant une résistance thermique inférieure à 18°C/W pour éviter d'atteindre une température de jonction trop élevée.

L'EPROM U₃ sera programmée avec le contenu du fichier "CAPAMTR.ROM" que vous trouverez sur le CDROM joint à la revue. Ce fichier est le reflet binaire du contenu de l'EPROM tandis que le fichier "CAPAMTR.HEX" contient l'équivalent dans le format "HEXA INTEL". Selon le modèle de programmeur d'EPROM dont vous disposez, vous utiliserez l'un ou l'autre des fichiers.

L'utilisation de ce montage est immédiate grâce au programme pour l'environnement Windows (95/98/NT) qui l'accompagne. Lors de la première mise en service, il vous faudra définir quel est le port série auquel est raccordé le montage. Pour cela, utilisez l'option "Configuration" du menu principal. Ce paramètre sera enregistré automatiquement par le programme dans la base des registres de Windows, pour vous éviter de le saisir à nouveau lors des utilisations suivantes.

Il ne vous reste plus qu'à brancher le condensateur à mesurer au connecteur CN_x, à choisir la gamme de mesure et à cliquer sur le bouton "Mesure". Le choix de la gamme de mesure influence la précision de la mesure, bien évidemment. Mais, dans un premier temps, vous pouvez approcher la valeur du condensateur en demandant la mesure sur n'importe quelle gamme. Ensuite, selon le résultat affiché, vous pouvez choisir la gamme de mesure appropriée et demander une nouvelle mesure pour obtenir une valeur plus précise. Pour la gamme allant de 10 µF à 9999 µF, le temps de la mesure atteint plus de 5 s (3 s pour la décharge complète du condensateur et un peu plus de 1 s pour la charge). Soyez donc patient sur cette gamme de mesure. Pour les autres gammes, la mesure demande moins de 1/2 seconde.

Pour ne pas pénaliser le temps de la mesure des condensateurs de valeur inférieure à 10 µF, le programme n'a pas été surchargé pour vérifier tous les cas de figures qui peuvent se présenter. Pour les



4 Vues d'écran

gammes de mesure inférieure à 10 µF, le temps de décharge est beaucoup plus court. Cela provoque quelques petits effets de bord qui ne sont pas bien gênants si l'on sait à quoi cela correspond : Si vous choisissez la gamme de mesure 1 pF à 10 nF et que vous branchez un condensateur de l'ordre de 10 µF sur CN_x, le programme affichera un message d'avertissement indiquant que la réponse du montage ne répond pas. En réalité, le montage répond bien, mais il répond trop tard puisque la résistance de charge correspondant à la gamme de mesure est élevée. La constante de temps R/C ainsi produite est trop longue par rapport aux temps de réponse attendus par le programme.

Si vous voyez ces messages apparaître, pensez à vérifier que la gamme de mesure est bien adaptée. Cette remarque concerne uniquement la mesure des condensateurs chimiques. Notez également que, pour la mesure des condensateurs chimiques, il faut connecter la borne (-) du condensateur au point n°1 du connecteur CN_x, c'est à dire à la masse du montage. Sinon la mesure risque d'être complètement faussée.

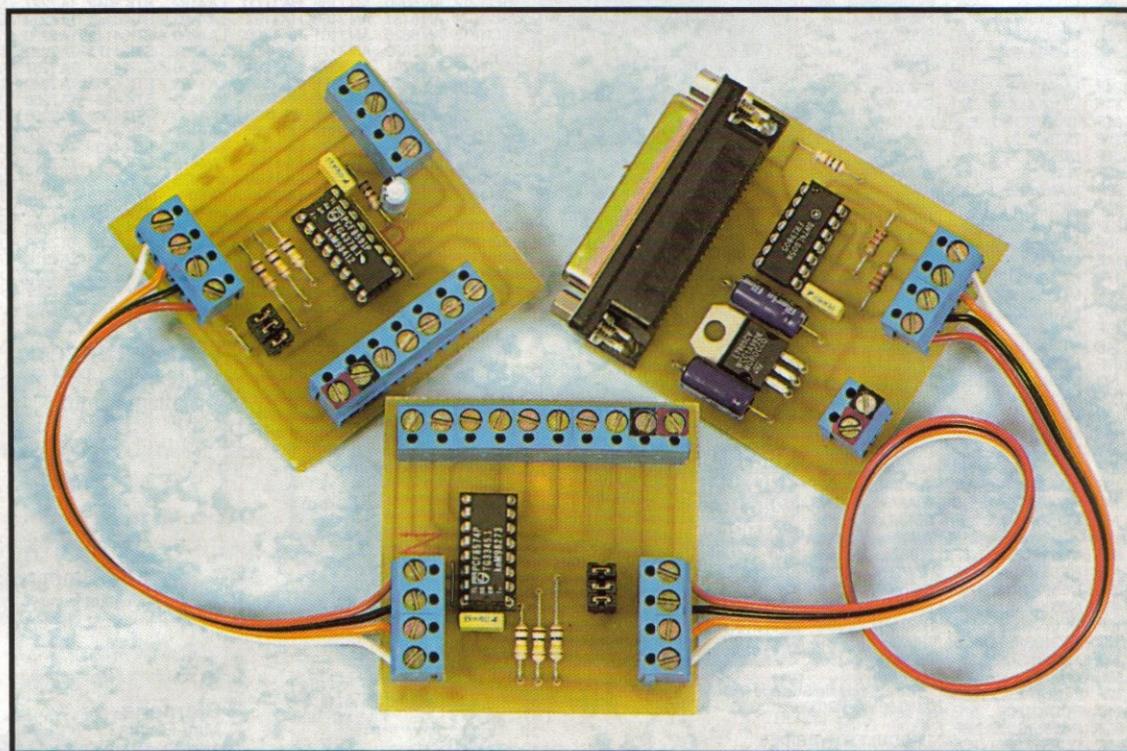
Si vous regardez de près les fichiers sources des programmes qui vous seront remis avec le fichier nécessaire à la programmation de l'EPROM, vous constaterez que le microcontrôleur ne calcule vraiment pas la valeur du condensateur à mesurer. En réalité, il mesure simplement le temps

que met le condensateur à se charger jusqu'à VCC/2. Comme nous l'avons déjà dit, le temps que met le condensateur à se charger vaut typiquement $0,7 \times R \times C$. Mais en raison des chutes de tensions introduites par les transistors T₃ à T₆, le facteur 0,7 est légèrement affecté. Le programme Windows en tient compte puisque c'est lui qui, finalement, réalise le calcul de la valeur du condensateur. Si vous souhaitez adapter le programme à vos besoins ou en écrire un autre (pour imprimer un ticket avec la valeur du condensateur, par exemple), sachez que les commandes à passer au montage via la liaison RS232 sont très simples. Il suffit d'envoyer le caractère "G" au montage suivi du caractère "1" à "4" pour sélectionner la gamme de mesure (1 correspond à la gamme 10 µF à 9999 µF, 2 correspond à la gamme 10 nF à 10 µF, etc...). Le montage doit répondre "G" suivi d'un retour à la ligne et du caractère ">". Pour lancer la mesure, il suffit d'envoyer le caractère "M" au montage. Ce dernier doit répondre "M." dans un premier temps suivi de "[xxxx]" à la fin de la mesure (donc au bout de 5 s pour la gamme de mesure 10 µF à 9999 µF). La valeur xxxx qui est transmise correspond au temps de charge du condensateur pour atteindre VCC/2 exprimé en µs (xxxx est une chaîne ASCII). Vous savez maintenant tout, pour ce qui concerne ce petit montage. Il nous reste à vous souhaiter une bonne réalisation.

P. MORIN

Bus I2C

sur le port parallèle



Le bus I2C a été développé par la société PHILIPS. Il s'agit d'une liaison synchrone qui nécessite seulement deux lignes pour communiquer. Les données sont transmises sous forme série au rythme de l'horloge qui peut atteindre une vitesse de 100 Kbits/seconde.

Dans le cadre de cet article, nous allons vous montrer comment transformer très simplement le port parallèle de votre ordinateur en un véritable bus I2C. Votre PC sera alors en mesure de piloter tous les circuits maîtrisant ce mode de communication. Deux circuits utilisant le protocole I2C vous seront présentés : le premier (PCF8574) permet de disposer de 8 entrées/sorties logiques, le deuxième (PCF8591) permet de générer et de lire un signal analogique.

Principe de fonctionnement du bus I2C (figure 1)

La ligne où transite les données est appelée SDA (Serial Data), SCL (Serial CLock) étant la ligne d'horloge. Chacune de ces lignes est dotée d'une résistance de rappel à la tension d'alimentation, chaque circuit associé au bus peut la mettre à un niveau bas grâce à des sorties à collecteur ouvert.

Le principe fondamental d'une liaison synchrone est le suivant :

L'état de la ligne de donnée (SDA) est valide uniquement lorsque la ligne d'horloge (SCL) est au niveau haut, le changement d'état de la ligne SDA s'effectue lorsque la ligne SCL est à l'état bas.

Les circuits qui envoient les ordres sont appelés maîtres, ceux recevant les données sont des esclaves. Chaque circuit a sa propre adresse qui est codée sur 7 bits, cela signifie que l'on peut théoriquement connecter jusqu'à 128 circuits différents sur un même bus. L'adresse se compose d'une partie fixe propre au type de circuit et d'une partie variable codée par des lignes d'adresses. Par exemple, si le circuit comporte trois lignes d'adresses cela signifie que l'on peut avoir jusqu'à $2^3 = 8$ circuits du même type sur le bus.

Le dialogue entre le maître et les esclaves suit un protocole bien défini : Lorsqu'un maître veut envoyer un ordre à un esclave il émet, premièrement, une condition de départ

(START) afin de prévenir l'ensemble des abonnés, SDA passe du niveau haut au niveau bas alors que SCL est au niveau haut. Ensuite il envoie l'adresse de l'esclave qu'il désire joindre, en commençant par le bit de poids fort, le maître indique qui veut "lire" sur l'esclave en positionnant le bit 0 de l'adresse à 1 (qui est le 8ème bit transmis) et écrire en le positionnant à 0. L'esclave répond en envoyant un signal d'acquiescement (ACKNOWLEDGE) qui a lieu lors de l'envoi par le maître de la 9ème impulsion d'horloge et qui correspond à la mise à l'état haut de la ligne SDA. Ensuite le maître peut envoyer l'ordre codé sur 8 bits, dans le cas d'une écriture, ou recevoir la donnée dans le cas d'une lecture. L'esclave signalera la prise en compte de cette transmission par un deuxième signal d'acquiescement. La fin d'une transmission est spécifiée par le maître par une condition d'arrêt (STOP), SDA passe du niveau bas au niveau haut alors que SCL est au niveau haut.

Dans notre cas de figure, notre bus

ne comportera qu'un seul maître : le PC, lui seul, délivrera le signal d'horloge. Cependant la ligne SDA restera bidirectionnelle car, comme nous l'avons vu précédemment, un esclave reçoit mais aussi envoie des données au maître.

Interface port parallèle / bus I2C (figure 2)

Seulement trois lignes du port parallèle sont utilisées pour simuler un bus I2C. La sortie STROBE (broche 1) est chargée de générer le signal d'horloge (SCL). Deux lignes sont utilisées pour générer le signal SDA, en effet celui-ci est bidirectionnel. L'entrée Error (broche 15) permettra de lire les données reçues par SDA, alors que la sortie D0 (broche 2) permettra d'en transmettre.

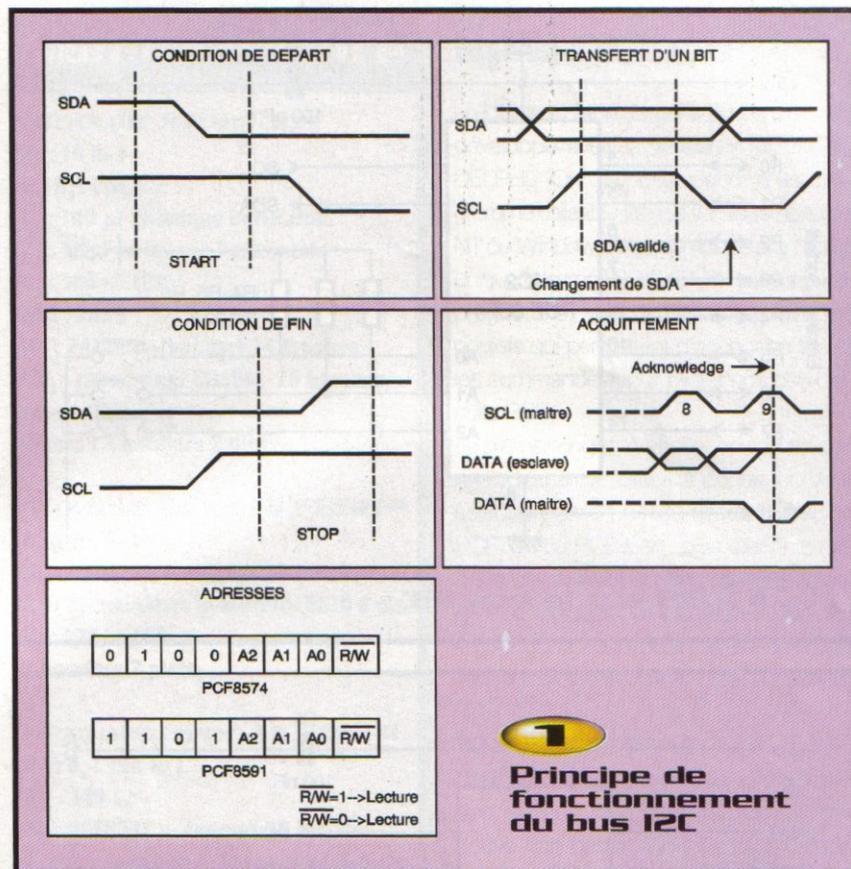
Chacune de ces lignes est tamponnée par le circuit IC₁ : un 74LS05. En effet, celui-ci dispose de 6 portes logiques inverseuses mais qui fonctionnent à collecteur ouvert, ce qui est nécessaire pour être compatible avec un bus I2C. On retrouve, bien entendu, trois résistances de rappels R₁, R₂ et R₃.

Rappelons que la sortie Strobe et que l'entrée Error sont actives à l'état bas, mais avec le passage par les portes IC_{1a} et IC_{1b}, on retrouve un fonctionnement normal. Toutefois ce n'est pas le cas de la sortie D0 qui sera active au niveau bas après le passage par la porte IC_{1c}, il faudra en tenir compte lors de la conception du logiciel.

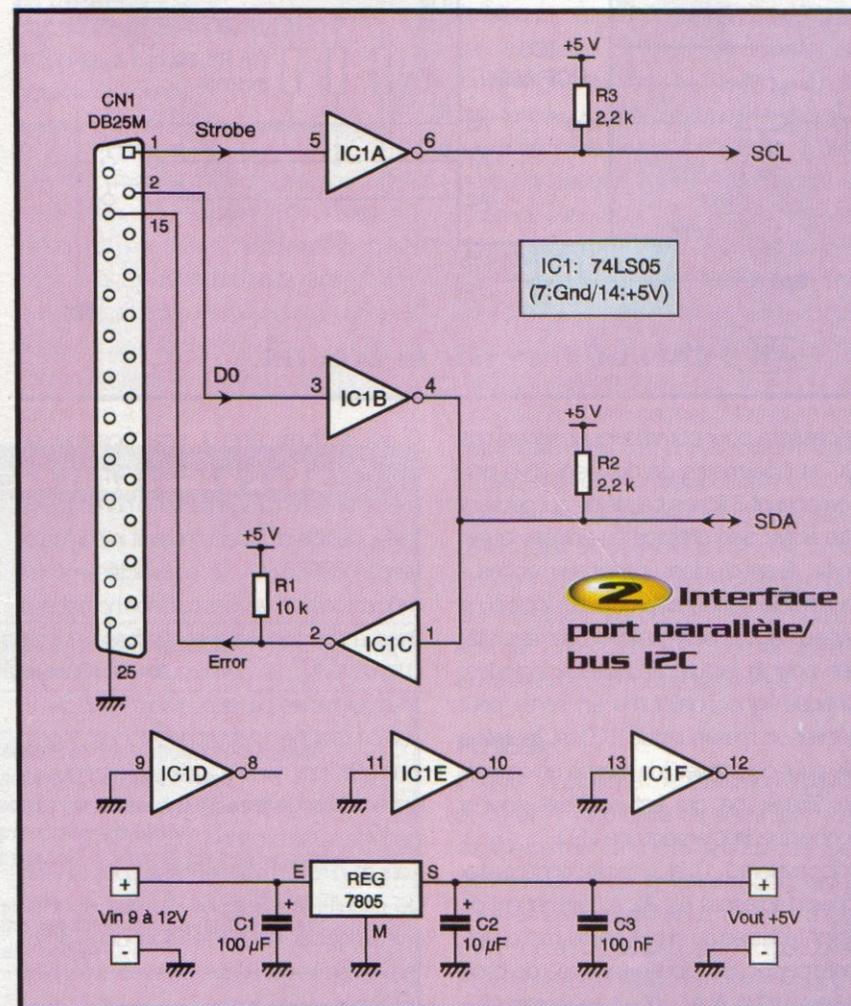
Les trois lignes utilisées sont disponibles sur un connecteur standard mâle au format DB25. Vu la taille particulièrement réduite de notre interface, il sera possible de se passer de câble et ainsi connecter directement celle-ci à la sortie parallèle de votre PC. L'alimentation est architecturée autour d'un classique régulateur 7805 qui peut délivrer un maximum de 1A, ce qui n'est pas excessif compte tenu des nombreux circuits qui pourront, par la suite, être connectés au bus, ceux-ci pourront directement tirer leur tension d'alimentation à partir de l'interface. De ce fait, la tension d'alimentation régulée à +5V ainsi que la masse sont disponibles, avec les lignes SDA et SCL, sur des borniers à vis ce qui facilitera la connexion des circuits à piloter.

E/S sur le bus I2C (figure 3)

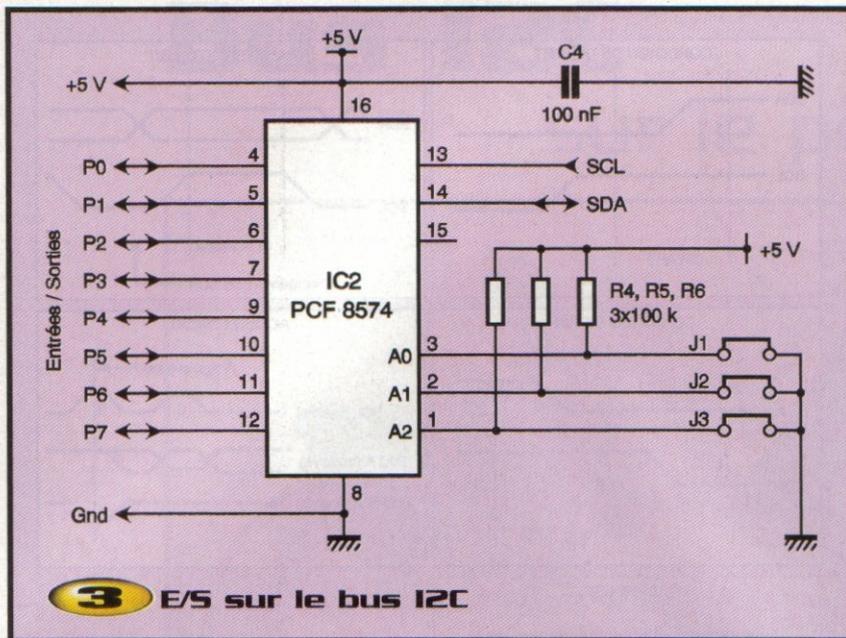
Le premier circuit que nous allons piloter, un PCF8574, dispose de toute la logique



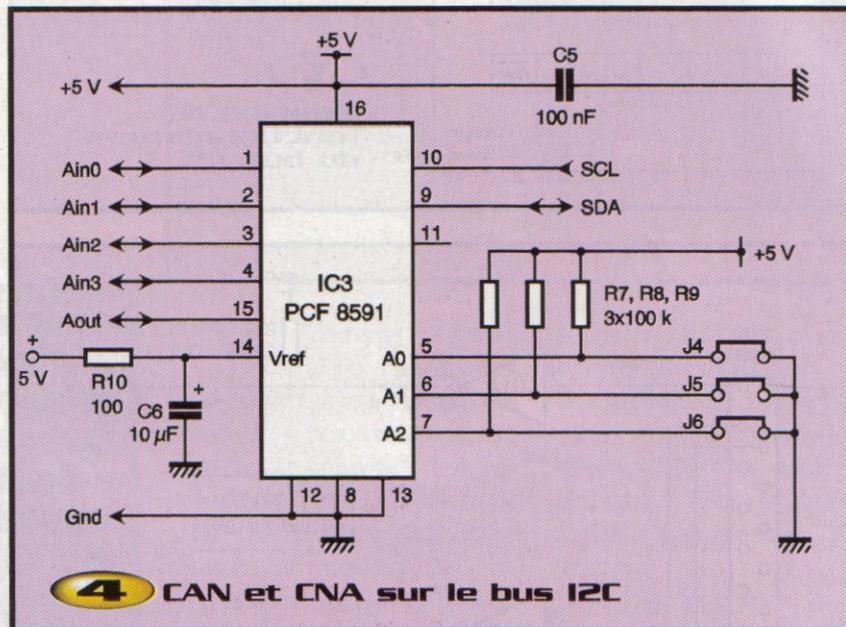
1 Principe de fonctionnement du bus I2C



2 Interface port parallèle / bus I2C



3 E/S sur le bus I2C



4 CAN et CNA sur le bus I2C

C.A.N. et C.N.A. sur le bus I2C (figure 4)

Le deuxième circuit que nous vous proposons de piloter est le PCF8591. Un tel composant dispose de deux fonctions : Il est capable de convertir une tension analogique en une donnée numérique (C.A.N.) et de réaliser l'opération inverse, soit convertir une donnée numérique en un signal analogique (C.N.A.).

Quatre entrées sont associées au CAN (Ain0, Ain1, Ain2, Ain3) ainsi il est possible de convertir quatre tensions analogiques distinctes comprises entre 0 et +5V en une donnée numérique codée sur 8 bits, ce qui nous amène à une résolution de 20mV.

Une seule sortie (Aout) est utilisée pour le CNA, qui délivre une tension comprise entre 0 et +5V par pas de 20mV.

La référence de tension haute de +5V est prélevée sur l'entrée VREF (broche 14).

L'adresse fixe de base du PCF8591 est 72 avec trois lignes d'adresses (1001A₂A₁A_{0x}) dont l'état est défini par les cavaliers J₄, J₅ et J₆, ce qui permet d'adresser jusqu'à 8 circuits de ce type sur un même bus. Le mode d'utilisation de ce circuit dépend d'un octet de commande. Les bits 0 et 1 définissent la voie d'entrée active du CAN. Le bit 2, lorsqu'il est à 1, autorise la commutation automatique de l'entrée utilisée par le CAN, à chaque lecture le numéro de la voie active s'incrémente automatiquement. Les bits 4 et 5 définissent le mode de mesure. On peut, par exemple, référencer toutes les entrées par rapports à la masse, ce qui est le cas le plus classique. On peut aussi effectuer des mesures différentielles (voir tableau). La partie CNA n'utilise que le bit 6 qui, lorsqu'il est à 1, autorise son fonctionnement, dans le cas contraire la sortie est mise à haute impédance.

L'utilisation de la sortie analogique s'effectue en envoyant l'adresse du circuit puis en envoyant l'octet de commande en s'assurant que le bit 6 est bien à 1. Ensuite on fait suivre l'octet de donnée qui spécifie la valeur de la tension désirée. Signalons que l'on peut faire suivre plusieurs octets de donnée sans pour cela renouveler l'envoi de l'octet d'adresse et de commande. On peut ainsi générer un signal analogique qui pourra varier à une fréquence maximale d'environ 11 kHz puisque la fréquence maximale admise par le bus I2C est de 100 kHz et qu'il faut 9 coups d'horloges

nécessaire pour communiquer sur un bus I2C et il permettra de disposer d'un port comportant 8 lignes parallèles qui peuvent être aussi bien utilisées en entrées qu'en sortie. Avant d'utiliser une ligne en entrée, il faut simplement lui appliquer par logiciel un niveau haut, ensuite celle-ci peut être utilisée pour la lecture de niveaux logiques. Chaque ligne configurée en sortie peut débiter un maximum de 100µA, toutefois elle peut commuter à la masse un courant de 25mA, ce qui est suffisant pour y connecter directement une LED.

Trois cavaliers J₁, J₂ et J₃ correspondants aux lignes d'adresses A₂, A₁, A₀ permettent de définir l'adresse du circuit, en ouvrant ou fermant ceux-ci on modifie l'adresse de base qui est 0100A₂A₁A_{0x}, (X = 1 => lecture / X =

0 => écriture) ainsi il sera possible de connecter 8 circuits du même type sur un même bus. Afin d'augmenter le nombre E/S, il est possible d'utiliser un circuit intégré référencé PCF8574A, il a exactement les mêmes caractéristiques que son homologue mise à par son adresse de base qui est 0111A₂A₁A_{0x}. Le nombre de circuits maximum pilotables par le bus sera donc de 16 soit un total de 128 entrées/sorties. Signalons que, lors que les lignes configurées en entrées changent d'état, une interruption est générée sur la broche 13 de IC₁ (INT).

Lors d'un reset matériel (mise sous tension du circuit), les 8 lignes du port sont au niveau logique haut. Elles sont donc directement prêtes pour être utilisées aussi bien en entrée qu'en sortie.

pour transmettre un octet (8 bits de donnée + 1 bit d'acquiescement).

Pour récupérer l'octet correspondant à la conversion d'une des entrées Ainx, il suffit d'envoyer l'adresse du circuit en prenant soin de mettre le bit 0 à 1 signalant au PCF8991 que l'on effectue une opération de lecture. Celui-ci envoie alors l'octet représentant le résultat de la conversion réalisée sur l'entrée analogique définie par l'octet de commande envoyé lors de l'utilisation de la sortie analogique.

On peut réaliser l'acquisition d'un signal ayant une fréquence max. de 5,5 kHz, en respectant la condition de Shannon qui dit que la fréquence d'échantillonnage (ici 11 kHz) doit être au moins le double de la fréquence du signal échantillonné.

Réalisation pratique

Les schémas nécessaires à la réalisation des trois montages vous sont fournis figures 5, 7 et 9. Il faudra vous munir de trois carrés d'époxy présensibilisés de dimension 55 x 55 mm. Vous pourrez utiliser la méthode photographique pour insoler le circuit en réalisant un transparent à partir des fichiers disponibles sur le CD ROM accompagnant la revue. Après gravure dans un bain de perchlore et un rinçage abondant à l'eau tiède, vous pourrez passer à la mise en place des composants.

L'approvisionnement des composants électroniques ne doit pas poser de problème : les circuits du type PCF sont assez courants chez les détaillants ou même par correspondance. Si vous comptez utiliser un nombre considérable de circuits sur le bus,

Nomenclature

Module interface

R₁ : 10 kΩ
R₂, R₃ : 2,2 kΩ
C₁ : 100 µF chimique horizontal
C₂ : 10 µF chimique horizontal
C₃ : 100 nF LCC
REG : 7805
IC₁ : 74LS05 + support 14 broches
CN₁ : connecteur DB25M, 25 broches, coudé 90°
Divers : 3 borniers 2 plots

Module avec PCF8584

R₄ à R₆ : 100 kΩ
IC₂ : PCF8574 + support 16 broches
J₁ à J₃ : cavaliers + barrette HE10 6 cts
C₄ : 100 nF LCC
9 borniers 2 plots

Module avec PCF8591

R₇ à R₉ : 100 kΩ
R₁₀ : 100 Ω
IC₃ : PCF8591 + support 16 broches
J₄ à J₆ : cavaliers + barrette HE10 6 cts
C₅ : 100 nF LCC
C₆ : 10 µF chimique vertical
6 borniers 2 plots
1 bornier 3 plots

prévoyez d'ajouter un dissipateur thermique au régulateur de tension 7805. Gardez à l'esprit qu'il peut délivrer un maximum de 1A. Concernant le câble de liaison entre l'interface et les différents modules, ne pas dépasser une dizaine de mètres, en effet la charge capacitive du bus ne doit pas être supérieure à 250 pF.

Logiciel d'utilisation

Le logiciel chargé de gérer le bus I2C a été développé avec la version 4 du logiciel DELPHI. On dispose donc d'un programme utilisable par les versions 95/98 et NT de Windows. Il se compose d'un fichier unique nommé "I2C.EXE". A l'exécution du programme, vous voyez apparaître cinq onglets qui permettent d'accéder à toutes les commandes pour piloter nos deux circuits PCF.

Le premier onglet "Adresse" permet de définir l'adresse de circuit à contacter. Vous sélectionnez le type de circuit PCF8574, PCF75A ou PCF8591, puis, par l'intermédiaire de cases à cocher, vous définissez les états logiques des lignes d'adresses A0, A1, A2 qui correspondent à l'état des cavaliers (interrupteur ouvert = case cochée).

PCF8574 et PCF8574A

L'onglet "Lecture" permet de venir lire l'état des entrées du circuit sélectionné. On trouve huit cases à cocher qui reflètent l'état des entrées du circuit, la mise à jour se faisant chaque fois que l'on vient cliquer sur le bouton lecture. On trouve trois fenêtres où vient s'afficher l'adresse du circuit dans la première, la seconde affiche la valeur décimale lue sur le port 8 bits, la dernière signale que la lecture demandée a réussi ou échoué, en fait le programme test si le bit d'acquiescement a bien été envoyé par le circuit. Si vous définissez une adresse qui n'existe pas, la lecture sera impossible et le message "pas de réponse" sera affiché.

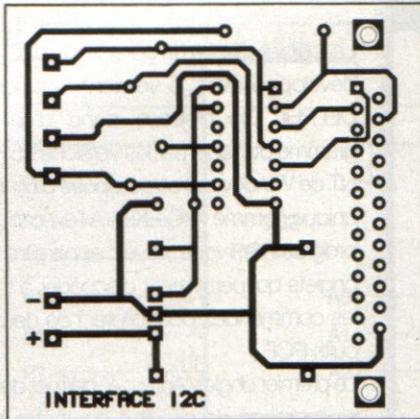
L'onglet "Ecriture" permet de définir l'état des sorties du circuit, on retrouve la même présentation que l'onglet lecture. Il suffit de venir cliquer sur les cases à cocher pour définir l'état de la sortie correspondante. Une fois vos choix définis, vous effectuez la mise à jour en cliquant sur le bouton envoyer. Un bouton "RAZ" permet de faire la mise à zéro de toutes les sorties.

PCF8591

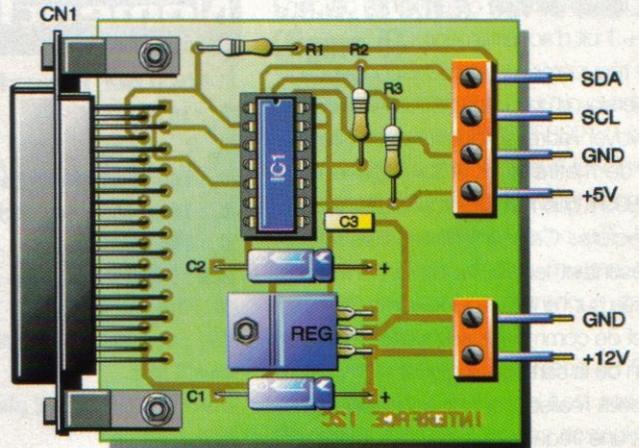
L'onglet "Analog" permet l'utilisation du CAN et du CNA. Une barre de défilement horizontale permet, à l'aide de la souris, de spécifier la valeur de la tension analogique générée sur la sortie Aout. La mise à jour s'effectue en cliquant sur le bouton envoyer. Pour la lecture d'une des entrées analogiques, sélectionnez tout d'abord la voie

Octet de commande

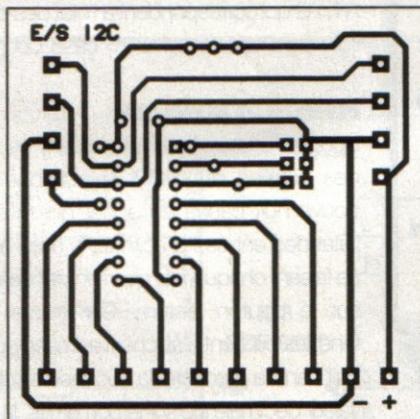
Bit 0 et 1	00 01 10 11	Voie Ain0 active Voie Ain1 active Voie Ain2 active Voie Ain3 active
Bit 2	0 1	Commutation automatique de voie inactive Commutation automatique de voie active
Bit 3	0	Toujours à 0
Bit 4 et 5	00 01 10 11	Toutes les voies référencées par rapport à la masse Ain0 à Ain2 référencées par rapport à Ain3 Ain0 et Ain1 référencées par rapport à la masse et Ain2 référencée par rapport à Ain3 Ain0 référencée par rapport à Ain1 et Ain2 référencée par rapport à Ain3
Bit 6	0 1	Sortie analogique Aout inactive (haute impédance) Sortie analogique Aout active
Bit 7	0	Toujours à 0



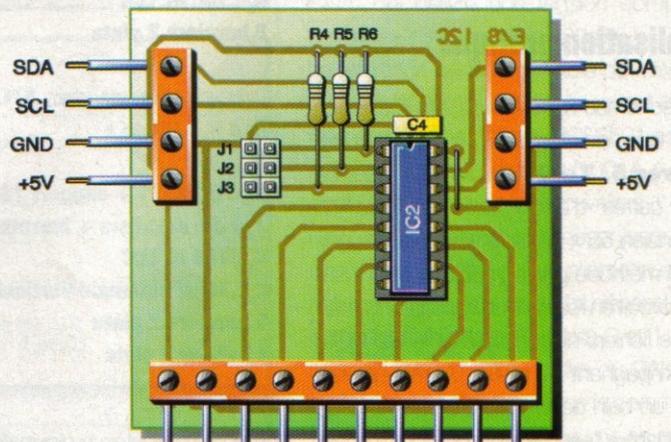
5 Tracé du circuit imprimé du module interface



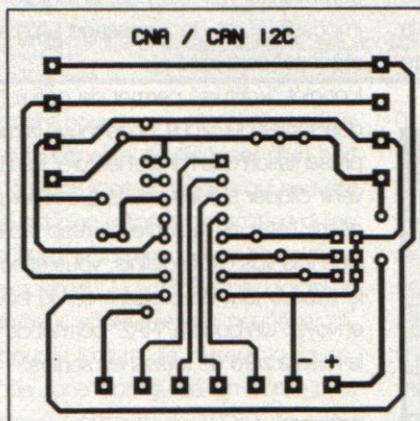
6 Implantation des éléments du module interface



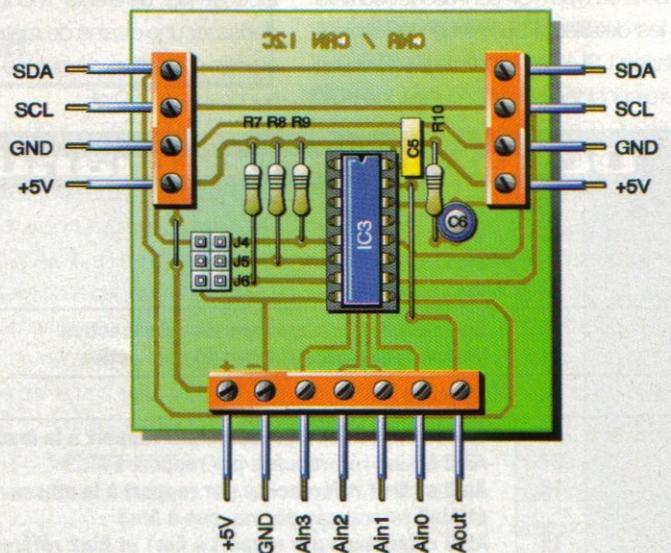
7 Tracé du circuit imprimé du module PCF 8584



8 Implantation des éléments du module PCF 8584



9 Tracé du circuit imprimé du module PCF 8591

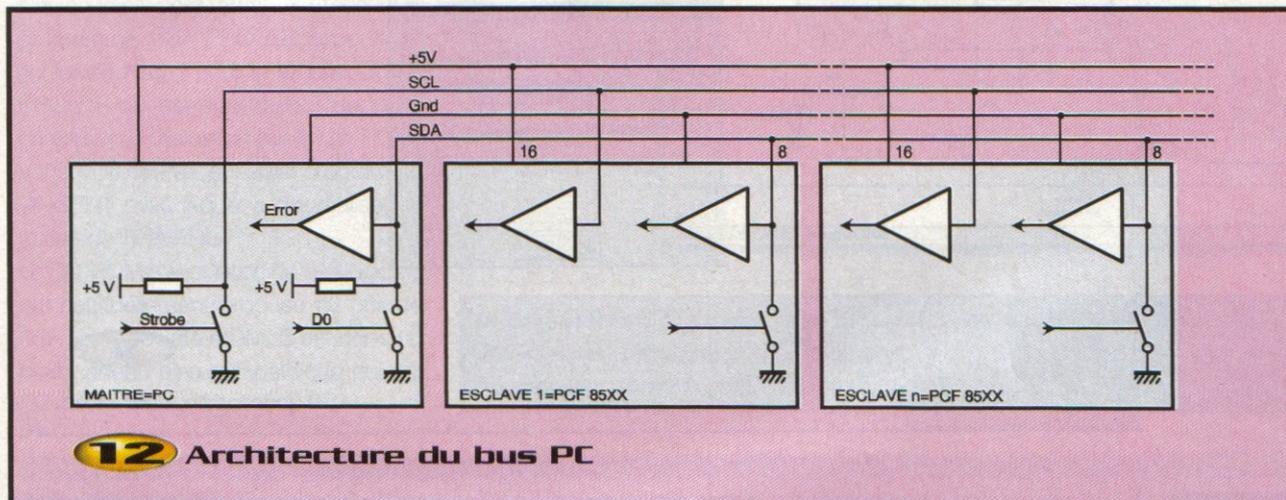
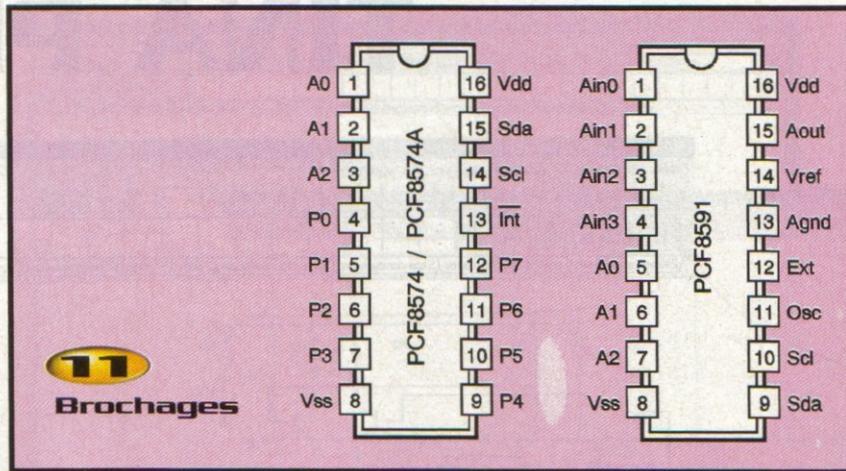


10 Implantation des éléments du module PCF 8584

concernée, puis en cliquant sur le bouton lecture le résultat de la conversion doit venir s'afficher directement en volts.

Le dernier onglet nommé "Divers" permet, d'une part de définir sur quel port votre interface I2C est connectée : LPT1 ou LPT2, d'autre part une donnée temporisation permet de définir une temporisation créée en nombre de boucle, en effet la ligne d'horloge ne doit pas dépasser la fréquence de 100 kHz, ce qui peut être le cas avec les ordinateurs les plus rapides.

Pour ceux qui désirent doter leur ancien ordinateur, ne disposant pas de Windows

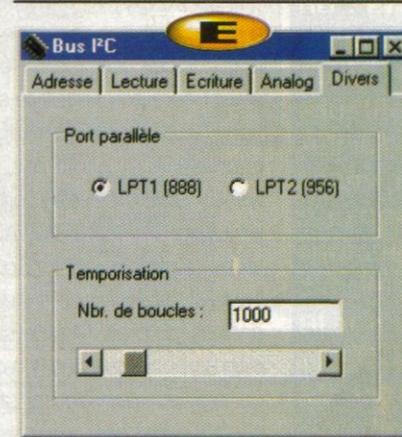
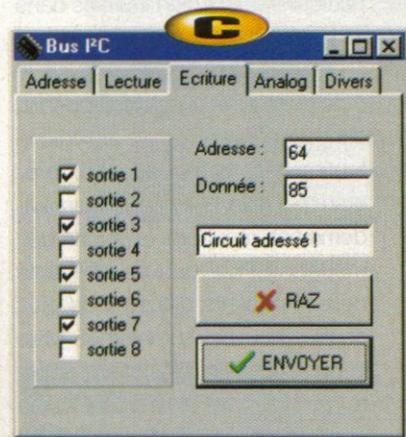
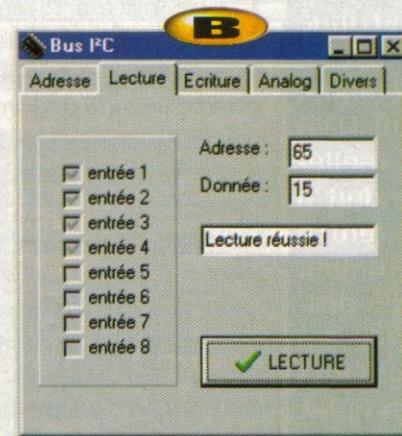
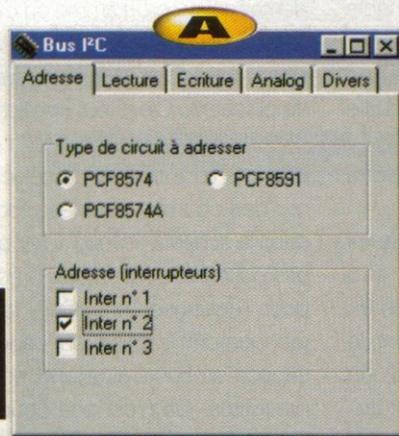


12 Architecture du bus PC

d'un bus I2C, trouveront également sur le CD ROM un programme écrit en QBasic. Il reprend exactement le même fonctionnement que celui écrit avec Delphi mise à part l'interface utilisateur qui est beaucoup moins conviviale !

D. REY

- A** : Analogie
- B** : Adresse
- C** : Ecriture
- D** : Lecture
- E** : Divers



Un parafoudre pour ligne de transmission



Les lignes de transmission en tous genres, et tout particulièrement les lignes téléphoniques, sont exposées à des surtensions accidentelles pouvant avoir pour origine des décharges atmosphériques (foudre) ou le contact fortuit avec un conducteur d'alimentation électrique

Loin d'être l'apanage des lignes aériennes de rase campagne, ces phénomènes peuvent fort bien concerner les citadins les plus invétérés : il suffit par exemple que les lignes téléphoniques d'un immeuble partagent la même gaine que les câbles d'antenne collective. Bien entendu, modems, PC, télécopieurs ou postes téléphoniques n'apprécient guère ce genre d'incident, dont il est pourtant si simple de les protéger...

Faire mieux pour moins cher

Une fois n'est pas coutume, le montage que nous vous proposons de réaliser revient plutôt moins cher, tout en étant au moins aussi efficace, que les dispositifs de protection disponibles dans le commerce. Difficile, en effet, de trouver un (bon) parasurtenseur "courants faibles" à moins de quelques centaines de francs, un peu comme si le prix de ces "assurances-vie" était calculé à partir de la valeur des équipements qu'ils sont chargés de protéger. Présentés la plupart du temps sous la forme d'une fiche téléphonique gigogne, les parafoudres "grand

public" font rarement appel à plus d'une famille de composants de protection.

Il en existe pourtant trois, dont les caractéristiques diffèrent très sensiblement, et qu'il est avantageux d'associer pour aboutir à une sécurité optimale : les éclateurs à gaz, les diodes zéner bidirectionnelles (Transils) et les varistances à l'oxyde de zinc (Gemov ou SIOV).



La version commerciale

Il faut également distinguer deux catégories de protections : en mode commun, et en mode différentiel.

La première concerne les surtensions qui affectent les deux fils de ligne à la fois, et cela par rapport à

la terre. Un parasurtenseur "mode commun" dépourvu de prise de terre ne sert donc strictement à rien !

La seconde se rapporte aux surtensions apparaissant entre les deux fils de ligne et, par conséquent, pas forcément référencées à la terre. Elles peuvent être assez facilement éliminées par une simple varistance placée en parallèle avec la ligne, type de protection dont sont de plus en plus souvent équipés d'origine les matériels péri-téléphoniques sensibles et coûteux.

Dans certains cas, la ligne téléphonique de l'abonné est munie d'une "boîte d'entrée de poste" située en amont de la première prise de l'installation. Équipée de deux fusibles, elle est également prévue pour recevoir un double éclateur à gaz, rarement présent en pratique. Son efficacité suppose en effet son raccordement à une bonne prise de terre, dont il n'est pas si courant de disposer là où il le faudrait.

Simple mais efficace

Le schéma de la **figure 1** ne fait appel qu'à un très petit nombre de composants, mais apporte un degré

élevé de protection. Il est en effet très proche de ce que l'on rencontre en amont des transmetteurs d'alarme professionnels ou des autocommutateurs privés.

L'association des trois familles de composants éliminateurs de surtensions garantit une efficacité optimale, tant en mode différentiel qu'en mode commun, à condition toutefois que la nomenclature soit respectée à la lettre et qu'une bonne prise de terre soit mise à contribution.

Destiné à être placé en amont des matériels à protéger, le montage pourrait bien sûr être incorporé à l'installation intérieure entre le câble de ligne et la première prise murale, bien qu'il ne soit nullement "agrée".

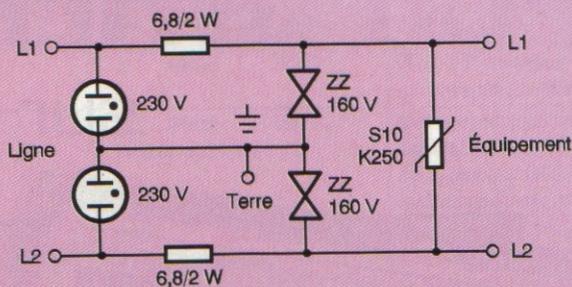
En pratique, on préférera souvent l'insérer en "volant" entre un conjoncteur mâle (bornes 1 et 3) et le câble à deux conducteurs de l'équipement à protéger. Muni de deux borniers à trois circuits (dont chacun possède une borne de terre centrale), ce module compact se prête de toute façon parfaitement à toutes les situations.

Sa réalisation suppose la gravure d'un petit circuit imprimé conforme au tracé de la **figure 2** dont l'épaisseur des pistes n'est en aucun cas le fruit du hasard. Il faut, en effet, prendre conscience du fait que le jour où le montage aura besoin d'entrer en action, il se trouvera sans doute parcouru par plusieurs centaines, voire quelques milliers, d'ampères. Malgré la brièveté probable de la décharge, il est certain que des pistes plus fines risqueraient de fondre, compromettant naturellement l'action des composants de protection.

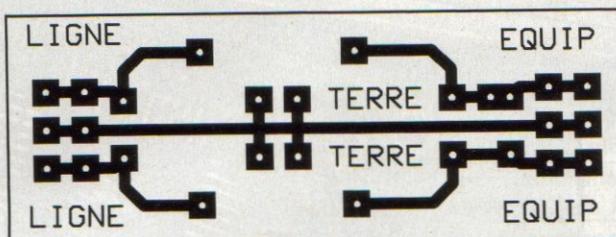
Le câblage sera exécuté selon le plan de la **figure 3** dont on nous permettra de réaffirmer que la nomenclature doit être scrupuleusement respectée. L'utilisation de composants présentant des tensions de déclenchement non conformes pourrait soit ruiner l'efficacité de la protection, soit perturber le fonctionnement de la ligne, notamment en réception d'appels (la tension de sonnerie est en effet de l'ordre de 80V efficaces).

Certains des composants prescrits pouvant ne pas être disponibles partout, nous indiquons, à toutes fins utiles, leurs codes de commande RADIOSPARES.

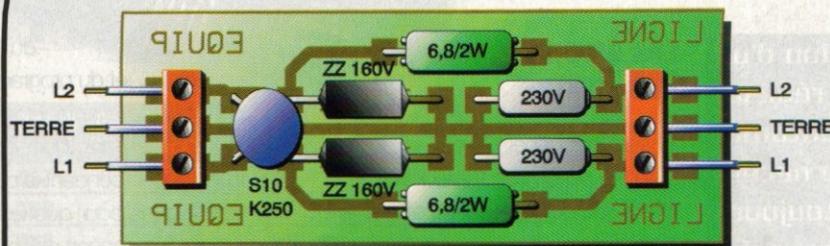
P. GUEULLE



1 Schéma de principe



2 Tracé du circuit imprimé



3 Implantation des éléments

Nomenclature

- 2 borniers à vis 3 circuits 5,08 mm (code 131-9147)
- 2 résistances vitrifiées 6,8 Ω / 2 ou 3W (code 165-4147)
- 2 éclateurs à gaz 230V (code 204-2371)
(ou 1 éclateur double, code 167-5510, avec adaptation de brochage)
- 2 Transils bidirectionnels 160 V (code 169-1168)
- 1 varistance S10K250 ou S14K250 (code 204-2157)

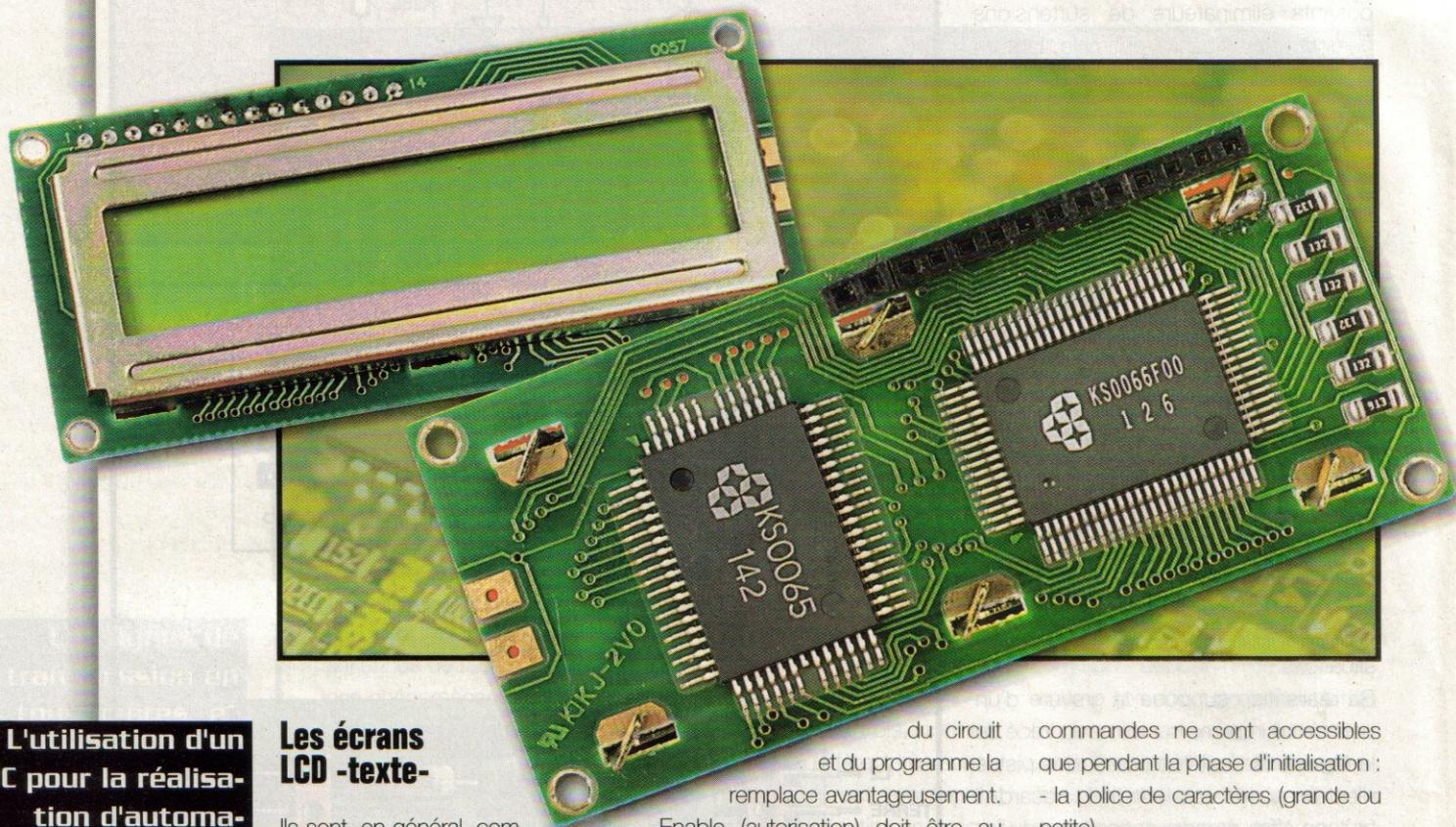
Destiné à remplacer une protection inexistante ou à seconder celle pouvant éventuellement être déjà en place, ce montage ne prétend en aucune façon se substituer à un quelconque élément de l'installation d'origine et, en particulier, à sa boîte d'entrée de poste. Si celle-ci est munie de fusibles, il faut savoir que l'efficacité accrue de la protection augmente leurs chances de fusion.

Il est bien évident que c'est là le prix à payer pour la mise en sécurité d'équipements importants et onéreux, mais que cela peut rendre nécessaires de plus fréquentes vérifications (par exemple d'un transmetteur d'alarme).

Il faut enfin savoir que ce montage est sans effet sur les surtensions pouvant arriver par le secteur électrique et qu'il peut également être prudent d'équiper le tableau général de parafoudres spécialisés.

Un afficheur LCD

sur le port parallèle



L'utilisation d'un PC pour la réalisation d'automatisme ne nécessite pas toujours un écran. En revanche si le contrôle du process est nécessaire un petit afficheur 2 ou 4 lignes est souvent suffisant. Les lecteurs d'Interface PC apprécieront sûrement ce système qui permet d'économiser un écran et surtout de la place.

Les écrans LCD -texte-

Ils sont, en général, commandés par un circuit intégré du type HD44780 ou équivalent, cet article y fait référence.

Exigez de votre revendeur une notice technique de votre LCD.

Sans rentrer dans les détails, voici ce qu'il faut savoir :

- Possibilité de travailler en mode 4 ou 8 bits, c'est à dire envoi des données sur 4 ou 8 lignes logiques. Dans notre cas, on utilise 8 bits.

- En plus des lignes de données, il faut ajouter des lignes de commandes :

RS (sélecteur de registre) qui permet de passer alternativement de l'envoi de commandes (niveau logique 0) à l'envoi de données (niveau logique 1). RW (lecture-écriture) Cette ligne permet d'accélérer le fonctionnement du LCD. Quand elle est mise au niveau logique 1, elle passe les lignes en mode lecture et signale au programme (si celui-ci l'interroge) la disponibilité du circuit. Elle n'est pas indispensable. Une gestion correcte

du circuit et du programme la remplace avantageusement.

Enable (autorisation) doit être au niveau logique 1 pendant la configuration des lignes pour que les commandes et les données soient prises en compte (**graphe 1**)

- Sur les afficheurs 4 lignes, il y a 2 circuits de contrôle donc 2 Enable, le premier pour les 2 lignes hautes, le second pour les 2 lignes basses.

Les commandes et les données

Les commandes sont envoyées quand le port RS est à 0, certaines

commandes ne sont accessibles que pendant la phase d'initialisation :

- la police de caractères (grande ou petite)

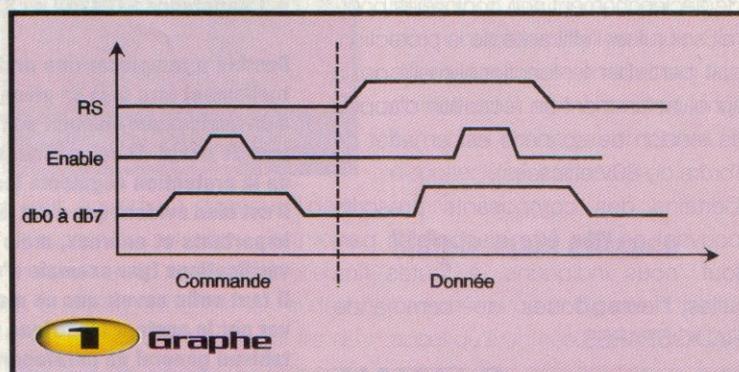
- le nombre de lignes (1 ou 2)

N'oubliez pas d'envoyer une impulsion Enable pour la prise en compte. Nous avons rajouté les valeurs que nous préconisons à initialisation pour l'interface par le port parallèle (**tableau 1**).

Quelques explications

- l'adresse DDRAM définit la position du caractère qui va être affiché, à l'initialisation cette adresse=00

- CGRAM, il est possible de définir des caractères particuliers (se repor-



ter à la notice de chaque afficheur)

- ASCII est une norme américaine de définition des codes de caractères : exemple 0=48 1=49 etc..

Liaison port parallèle / écran LCD

Tout ce qui suit concerne un écran LCD 4 lignes de 40 caractères, pour un écran de 1 ou 2 lignes, il suffit de supprimer la seconde ligne Enable (select in).

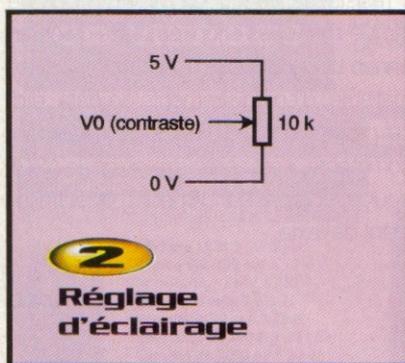
Pour cette application nous avons utilisé le port LPT1 de notre carte PC, à l'adresse 37A-378, donc n'oubliez pas de changer les adresses en cas d'utilisation du port LPT2 (27A-278)

Le **tableau 2** suivant montre le câblage utilisé :

Attention, ceci n'est qu'à titre indicatif, il faut respecter le brochage de votre LCD.

Vérifiez soigneusement le câblage avant la mise sous tension.

L'alimentation des LCD est en général du type :



En cas d'écran rétro-éclairé, il ne faut pas oublier que la consommation peut être importante (600 mA pour un grand LCD 4 lignes) et il ne faut pas omettre une résistance de protection pour limiter le courant et la tension.

Programmation

Pour comprendre le port parallèle, nous vous suggérons de vous reporter à l'article "Mémento des ports PC" du N°1 d'Interfaces PC, mais comme nous ne sommes pas méchants nous allons résumer ce dernier.

Le port parallèle est géré par trois registres : Données, Contrôles et État, nous allons utiliser les deux premiers. Données (&H378) et contrôle (&H37A)

Commandes	Code	Description
	RS poids fort /poids faible	
Efface + home	0 0000 0001	efface l'afficheur et remet le curseur en position 0
Home	0 0000 0010 valeur décimale=2	curseur en position 0 mais sans effacement, la valeur de l'adresse DDRAM reste inchangée (voir ci-dessous)
Réglage de la direction du curseur et de l'affichage	0000 01 i/d S à l'initialisation : 0 0000 0110 valeur décimale=6	i/d : curseur 1=incrément 0= décrément S=accompagnement de l'affichage
Contrôle de l'affichage	0 0000 1DCB à l'initialisation : 0 0000 1100 valeur décimale=12	D = afficheur on/off C = curseur on/off B = clignotement-curseur
Mouvement du curseur et de l'afficheur	0 0001 S/C R/L00 pas nécessaire à l'initialisation	S/C=1 afficheur 0=curseur R/L=1 droite 0=gauche
Réglage des fonctions	0 001DL NFO0 à l'initialisation : 0 0011 1100 valeur décimale=60	DL=1 interface 8 bits 0=4 bits N=1 : 2 lignes 0=1 ligne F=1 grande police 0= petite
Réglage de l'adresse DDRAM	0 1AAA AAAA	Positionne l'affichage à l'adresse AAA AAAA
Réglage de l'adresse CGRAM	0 01AA AAAA	Sélectionne un caractère défini par l'utilisateur
Données	Code	Description
Écriture d'une donnée	1 DDDD DDDD	écrit la valeur ASCII : DDDDDDD

T1 Valeurs préconisées

ECRAN LCD	Port du LCD	DB25	
DB7	1	D7	9
DB6	2	D6	8
DB5	3	D5	7
DB4	4	D4	6
DB3	5	D3	5
DB2	6	D2	4
DB1	7	D1	3
DB0	8	D0	2
Enable 1	9	strobe	1
R/W	10	autofeed	14
RS	11	init	16
VO(Contraste)	12		
VSS (0V - Masse)	13	18-19-20-21-22-23-24-25	
VDD (5V)	14		
Enable2	15	select in	17

T2 Câblage employé

Il est accessible sur une prise DB25 femelle, 18 broches sont utiles, nous en utiliserons 12

toutes en sorties. les autres sont en entrées.

Les données sont en D0 à D7 soit 8 bits, donc on peut envoyer 256 données (de 0 à 255).

Pour envoyer des données on utilisera l'instruction OUT ou OUTPORT (voir l'article précédent pour les équivalences de langage).

Le programme a été écrit en QuickBASIC soit : OUT &H378, 65 par exemple.

Pour piloter les lignes de contrôles, il faut utiliser le registre de contrôle (exemple OUT &H37A, 219) et le **tableau 3** :

Description du programme de test de l'afficheur - LCD.BAS -

Les procédures

Tout d'abord deux petites routines sont indispensables même si la première n'est pas utilisée dans le programme de test.

DelayS (secondes) permet de temporiser pendant un nombre de secondes défini, **Pause** (Millisecondes) permet de temporiser pendant un nombre de millisecondes défini, cette procédure est à ajuster suivant le micro-ordinateur utilisé.

Puis la procédure d'initialisation : InitLcd

Dans notre cas nous répétons 2 fois les mêmes étapes, une fois pour l'afficheur

	Strobe enable 1	autofeed R/W	init RS	select in enable 2	Valeur en décimale	fonction
départ	1	1	1	0		
	0	0	0	1	211	enable 2 cmd
	1	1	1	1	212	
	0	1	1	1	213	
	1	0	1	1	214	
	0	0	1	1	215	enable 2 data
	1	1	0	0	216	
	0	1	0	0	217	
	1	0	0	0	218	enable 1 cmd
	0	0	0	0	219	remise à 0
	1	1	1	0	220	
	0	1	1	0	221	
	1	0	1	0	222	enable 1 data
	0	0	1	0	223	fonction data

R/W est toujours à 0 dans notre cas

T3 Pilotage des lignes de contrôle

caractère	1	2	3	4	5	6	-----	35	36	37	38	39	40
DDRAM 1° Ligne	00	01	02	03	04	05	-----	22	23	24	25	26	27
DDRAM 2° ligne	40	41	41	43	44	45	-----	62	63	64	65	66	67

T4 Ecritures

haut, une fois pour l'afficheur bas.

Enable1, enable2 : procédures d'autorisation des commandes.

Ena1RS, Ena2RS : procédures d'acceptation de données.

WrLcd12, WrLcd34 : procédures d'écriture de données respectivement lignes 1 et 2, et lignes 3 et 4

AT12, AT34 : procédures de positionnement des caractères sur la ligne.

On peut écrire le caractère à un emplacement choisi, par exemple 10, en fait on positionne l'adresse DDRAM à 10 Attention : l'écriture dans la DDRAM est en hexadécimale et la seconde ligne ne commence pas après le dernier caractère de la première. Ainsi, dans le cas de notre afficheur 40 caractères, l'adresse &H31 (instruction **AT34** &H31) fait perdre un certain nombre de données. En revanche, &H27 imprimera toute la chaîne de caractères correctement ! (le premier caractère en fin de 1° ligne, les suivants au début de la seconde) c'est ainsi, il faut le savoir. Pour écrire en début de 2° ligne, il vaut mieux utiliser l'instruction **AT34** &H40.

(tableau 4)

Dans le cas d'un 4 lignes, la DDRAM est dupliquée, il y a deux contrôleurs, donc on retrouvera les mêmes adresses DDRAM, seules les lignes ENABLE 1 et 2 les différencient. C'est la raison de nos procédures nommées nnn12 et nnn34.

Comment envoyer une commande à l'afficheur ?

Exemple l'initialisation :

- Écrire la valeur décimale de la somme des

bits dans le port parallèle

- puis l'autoriser avec **enable**

exemple **port &H378,48**

Enable1

ou utiliser les procédures décrites ci-dessus.

Comment envoyer une donnée à l'afficheur ?

- Écrire la valeur ASCII dans le port parallèle (exemple pour écrire A)

- puis l'autoriser avec **ena1RS**

- exemple **port &H378,65**

Ena1RS

ou plus simplement **WrLcd12** asc("A")

Exemple : pour écrire "Afficheur LCD" à la 1° ligne en commençant par le 10° caractère : A\$= "Afficheur LCD"

at12 10 : rem commence au caractère 10 for j%=1 to len(a\$)

WrLcd12 asc(mid\$(a\$,j%,1))

next

Conclusion

Un afficheur, un potentiomètre de 10 k Ω , 1 résistance de 1,2 Ω 1W et un peu de remue-ménages et on remplace un écran par un LCD pour des applications d'automatisme, d'enregistrements de data, etc. Le programme LCD.BAS est largement commenté, il est enregistré en code ASCII pour être lisible sur n'importe quel traitement de texte.

J.P. DUVAL

