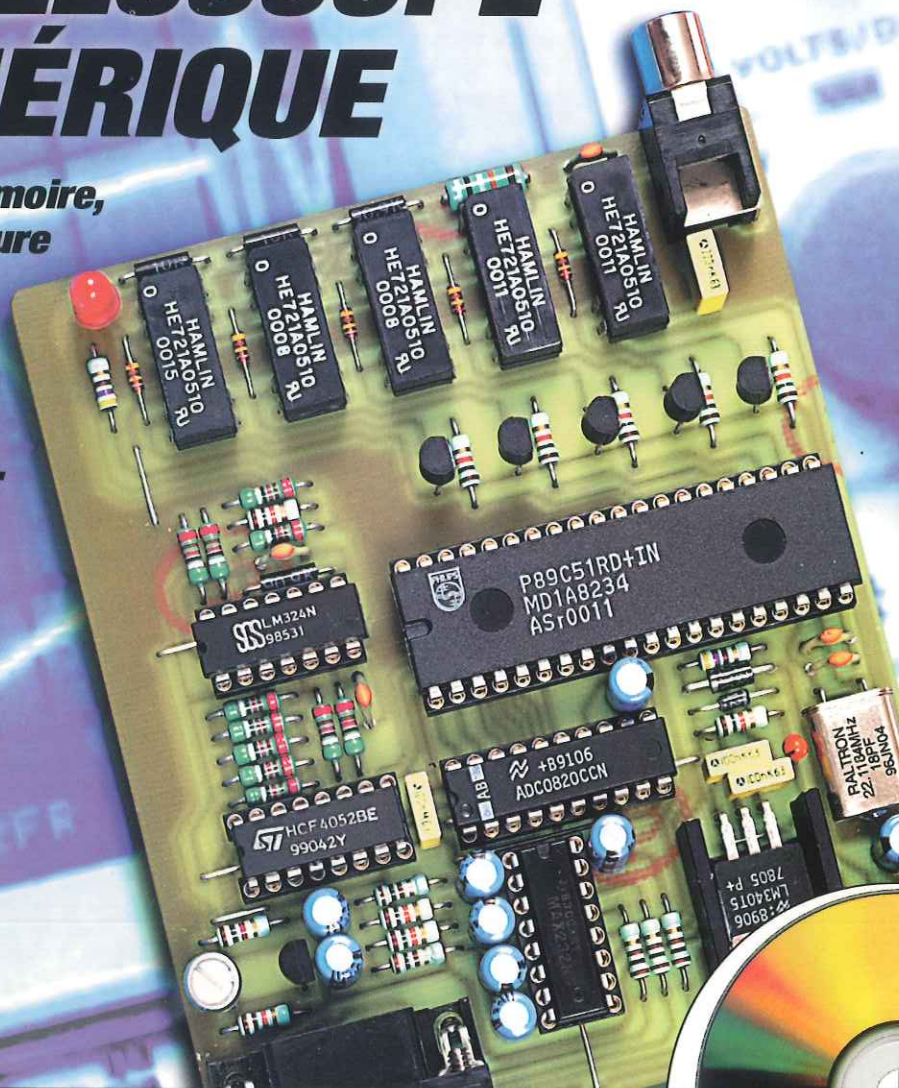


OSCILLOSCOPE NUMÉRIQUE

• **Fonction mémoire,
outil de mesure
performant,
impression
des courbes
avec choix
des couleurs.**



► **Nouveautés
sur les
cartes
à puces**

► **Espion
carte SIM**

► **Programmer
des PIC
en BASIC**

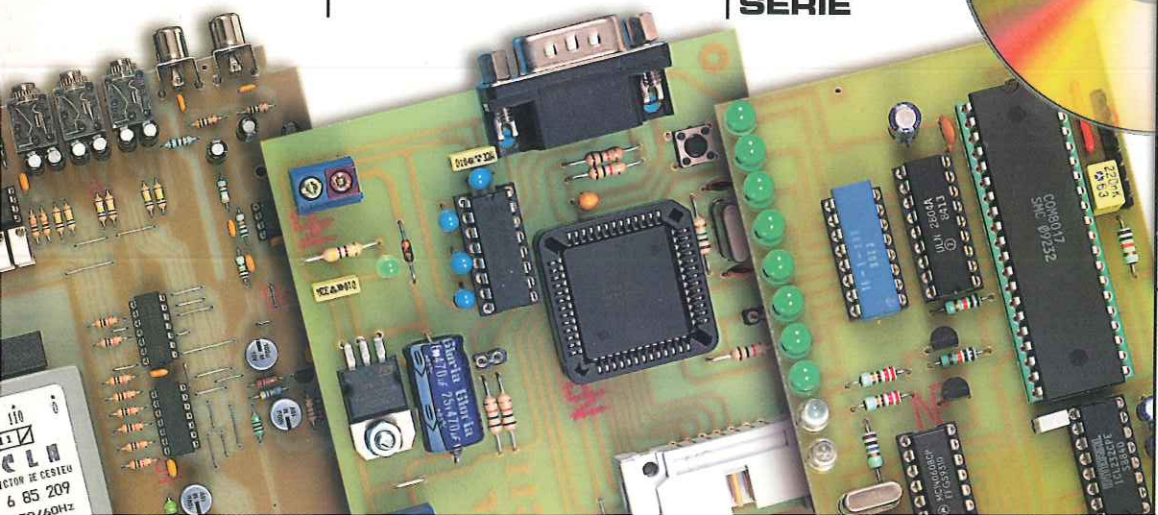
► **Port série :
3 sorties
sur triacs**

Etc.

**COMMUTATEUR
AUDIO**

**AFFICHEUR DE
MESSAGES**

**MONITEUR
LIAISON
SERIE**



CD Rom :

**TOUS LES
PROGRAMMES,
TOUS LES PCB,
DE NOMBREUSES
DEMOS... (VOIR P.15)**

T 3271-9H-32,80 F-RD



BELGIQUE : 180 FB - LUXEMBOURG : 180 FL
SUISSE : 7,00 FS - ESPAGNE : 490 Ptas
CANADA : \$ Can 7,50 - MAROC : 50 DH
ANTILLES - GU. 30 F

SOMMAIRE

ELECTRONIQUE PRATIQUE

HS N°09 - JUIN 2001
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**
Attaché de Direction : **Georges-Antoine VENTILLARD**
Directeur de la rédaction : **Bernard FIGHIERA** (84.65)
Directeur graphique : **Jacques MATON**
Maquette : **Dominique DUMAS, Jean-Pierre Rafini**

Avec la participation de : **Ph. André, U. Bouteville, H. Cadinot, A. Garrigou, F. Giamarchi, P. Gueulle, M. Laury, E. Migot, P. Morin, P. Oguic, J.P. Pitollet, D. Rey, Ch. Tavernier.**

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

Directeur de la diffusion et promotion :

Bertrand DESROCHE

Responsable ventes :

Bénédicte 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 dernière 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, communiquez 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 \$cd 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.

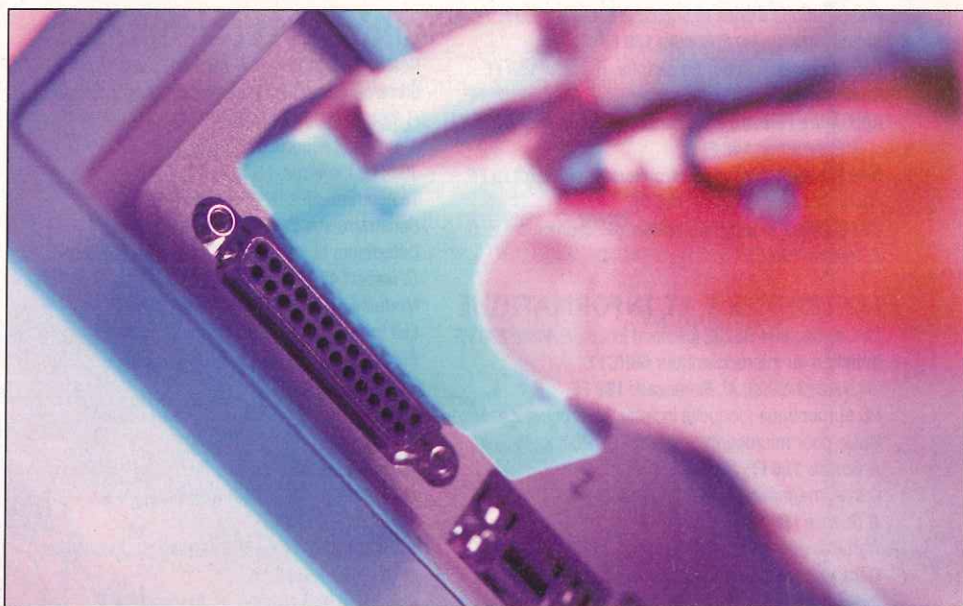
POSTMASTER : Send address changes to **Electronique Pratique**, c/o Express Mag, P.O. Box 2769, Plattsburgh, N.Y., 12901-0239.



« Ce numéro
a été tiré
à 50 700
exemplaires »

INTERFACES PC

- 05 Edito
- 06 Introduction : Les imprimantes
- 15 Utilisation du CDRom
- 20 Nouveautés "cartes à puce"
- 86 Concours Robotique 2001 - 2^{ème} édition



Les cartes à réaliser

- 22 Switch audio pour PC ou chaîne Hi-Fi
- 30 Moniteur de liaison série
- 34 "Espion" de cartes SIM
- 38 Programmer des PIC en Basic
- 42 Programmation du microcontrôleur AT89C51 par le port parallèle
- 48 Oscilloscope numérique pour PC
- 56 Programmateur Flash 8051
- 62 Détection automatique des systèmes connectés sur le port série
- 66 Port série : 3 sorties sur triacs
- 70 Afficheur de message à LED
- 80 Interface d'automatisation polyvalente
- 88 Interface 4 entrées/4 sorties triacs pour bus CAN
- 94 Programmateur de mémoire EEPROM

Infos

OPPORTUNITÉS

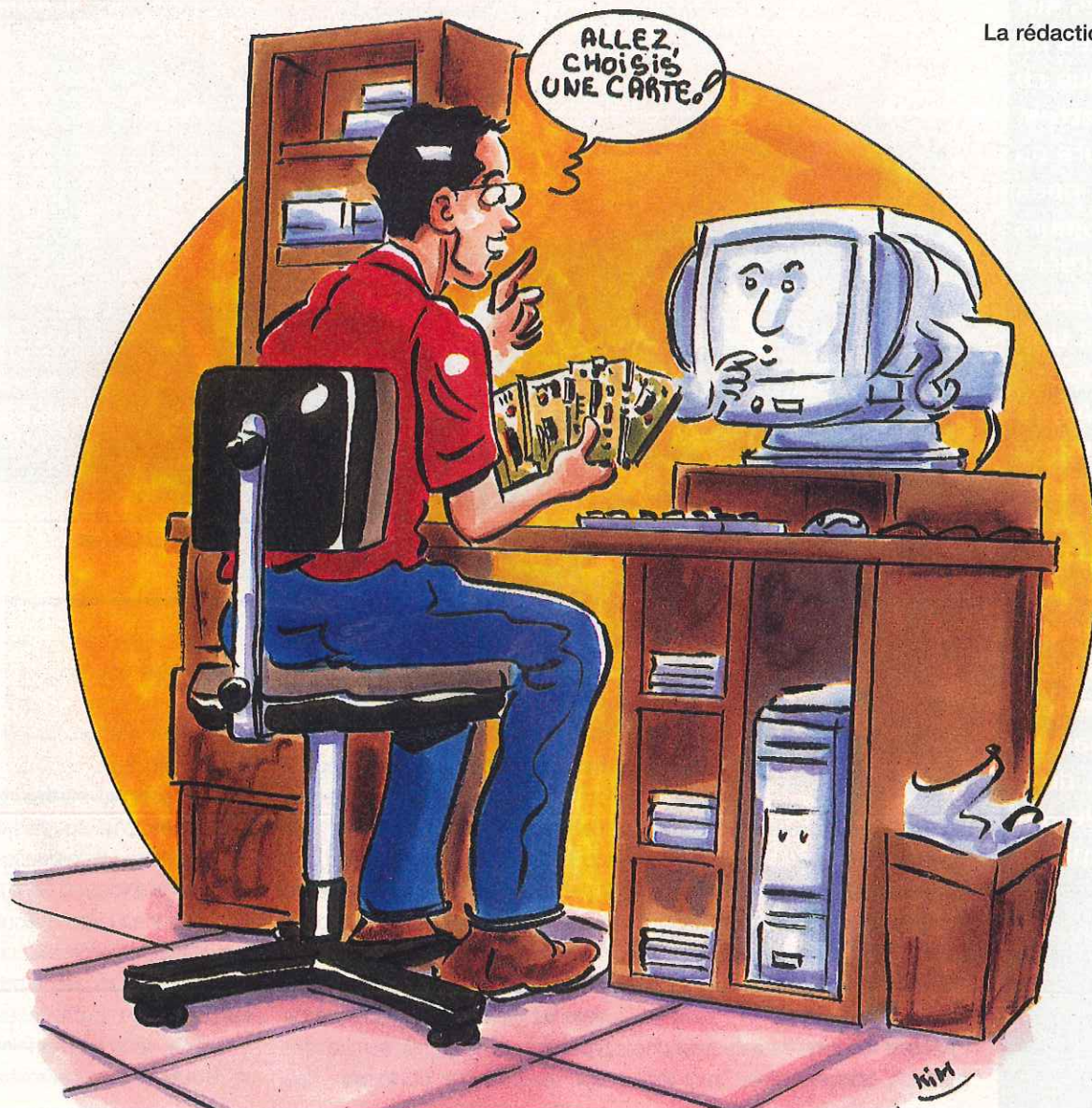
- 14
- 79

Les beaux jours reviennent enfin et nous donnent un peu de baume au cœur après un hiver particulièrement pluvieux. Souhaitons également que ce nouveau numéro d'Interfaces PC apporte à son tour une petite note de gaieté à nos lecteurs passionnés par l'électronique et le monde des PC. Comme à son habitude, ce numéro d'Interfaces PC fourmille de nouveautés et de montages attrayants, pour le plus grand plaisir de tous. Parmi les montages que vous pouvez découvrir dans le numéro 9 d'Interfaces PC, vous trouverez des programmeurs pour microcontrôleurs en tout genre. Étant donné l'impérieuse nécessité de faire appel à des microcontrôleurs pour réaliser, à peu de frais, des petits montages attrayants à raccorder à votre PC préféré, il était nécessaire de vous proposer des programmeurs mis au goût du jour. Citons, par exemple, le programmeur pour l'AT89C51 via le port parallèle, le programmeur de mémoire EEPROM et bien d'autres encore.

Si la programmation des microcontrôleurs ne vous intéresse pas spécialement, ou si vous êtes déjà bien équipés dans ce domaine, vous serez sans doute ravis de découvrir l'oscilloscope numérique pour PC publié dans ce numéro ou bien l'afficheur de messages à LED. Notez également que le bus CAN fait son apparition dans ces pages avec une interface de commande pour triac. Vous aurez ainsi l'occasion de mettre à profit les qualités de ce bus de terrain dans vos futures applications domotiques. Précisons, tout de suite, que vous trouverez dans un prochain numéro une petite interface CAN pour port série, très simple à mettre en œuvre, ainsi que bien d'autres montages articulés autour de ce bus très performant.

Nous vous invitons donc à découvrir sans tarder les montages passionnants publiés dans ce numéro. Et dès à présent, nous vous donnons rendez-vous au début de l'automne pour découvrir les montages, captivants eux aussi, qui vous seront proposés dans le prochain numéro d'Interfaces PC.

La rédaction



Les imprimantes



Malgré le fait que n'importe quelles données informatiques puissent être stockées sur toutes sortes de supports, que ce soit disquettes, disques durs ou CD-ROM, nous aurons toujours besoin de traces sur papier (courriers, listings, graphiques, photos, etc.). Les imprimantes, quelles qu'elles soient, assurent cette nécessité. L'article qui suit permettra de mieux comprendre le fonctionnement de chacune d'entre elles, de la plus simple à la plus complexe.

Historique

On peut distinguer deux sortes d'imprimantes :

- les machines utilisant l'impression à impact ou emboutissement du papier,
- les machines déposant l'encre sur le papier.

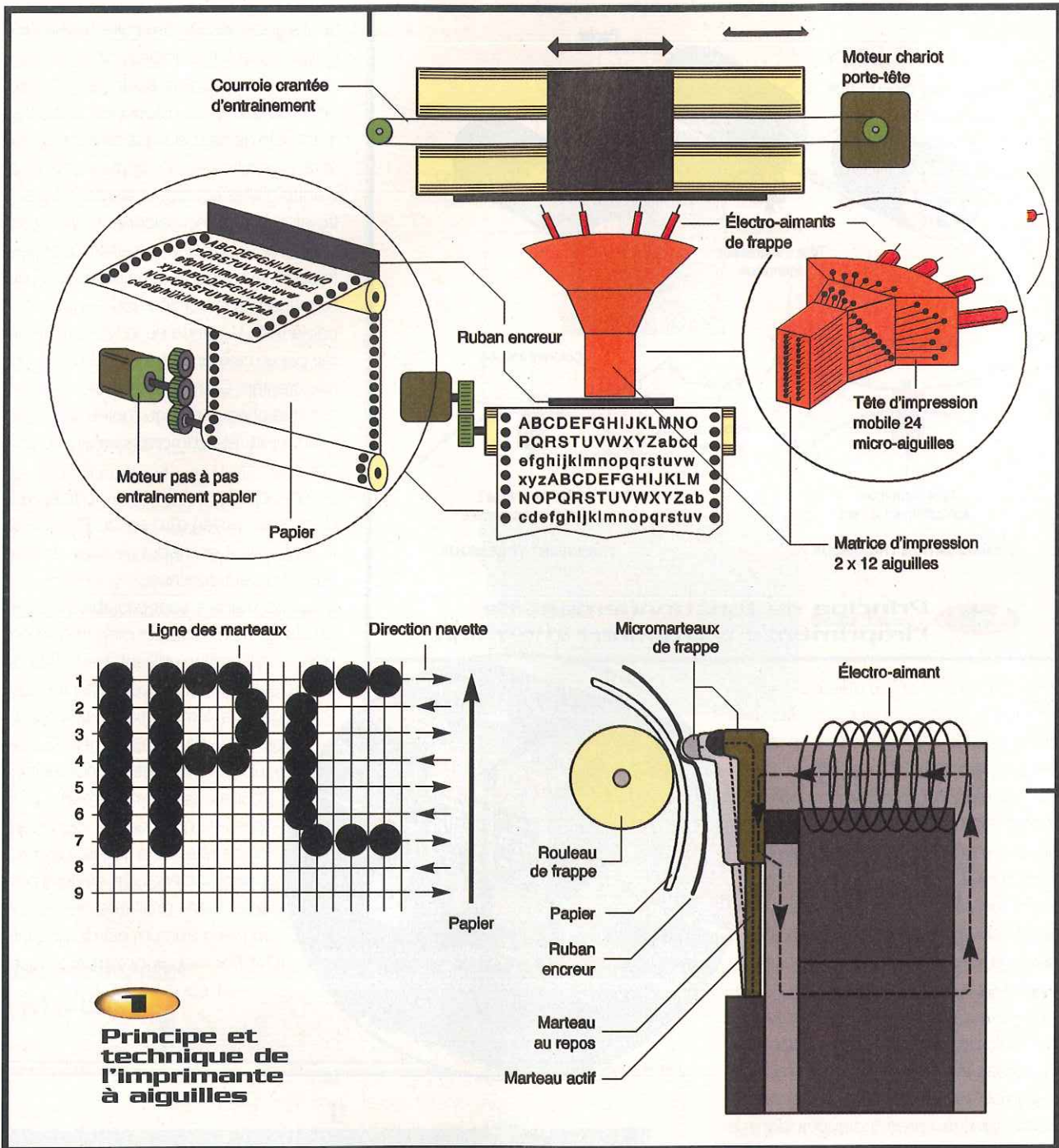
L'ancêtre des imprimantes, si on peut les nommer ainsi, sont les

machines à écrire existant depuis fort longtemps. Ces dernières, mécaniques, sur lesquelles les touches devaient être frappées très fortement afin que les caractères soit imprimés correctement sur le papier, ne permettaient que de réaliser des courriers. On voit immédiatement les inconvénients de ce type de machines qui sont la lenteur, le bruit, la correction des

erreurs, etc.

Ensuite, sont arrivées les machines électriques, plus souples d'emploi mais du même type que les précédentes, employant non plus les tiges sur lesquelles étaient placés les caractères en relief comme en imprimerie, mais des modules sur lesquels étaient gravés les caractères. Ce sont les modèles :

A boule : les caractères sont



1
Principe et technique de l'imprimante à aiguilles

répartis sur une sphère. Le caractère qui doit être imprimé est sélectionné par le mouvement de la sphère (axiale et angulaire). Une fois le caractère sélectionné, la sphère frappe le ruban encreur ;

à marguerite : comme son nom l'indique, les caractères sont répartis sur un disque constitué de lames souples ressemblant à une marguerite. Ce disque est actionné par un moteur, qui positionne le caractère à imprimer, et un électro-aimant associé à un marteau frappe le caractère et l'imprime sur le papier ;

à bande : les caractères sont gravés sur une bande mobile en rotation ;

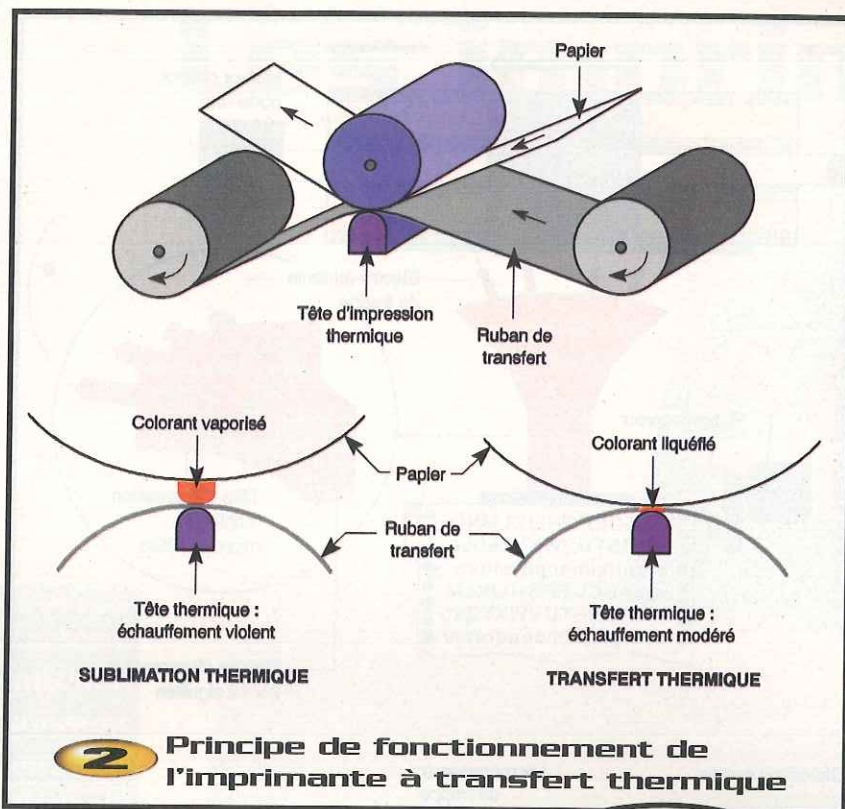
à cylindre : c'est un tambour sur lequel sont placés les caractères qui imprime sur le papier et à l'aide du ruban encreur les caractères. Ces derniers sont sélectionnés par la rotation et la position angulaire et verticale du cylindre. Ces modèles ne permettent cependant pas la création de graphisme, de caractères gras, de soulignés, etc., ce qui limite leur utilisation à la simple rédaction de textes, sans illustrations. Les plus modernes des machines à écrire disposent maintenant d'un afficheur à cristaux liquides sur lequel apparaît ce qui a été frappé sur le clavier avant impression, ce

qui correspond au traitement de texte que nous connaissons bien.

Les imprimantes à impact

Les imprimantes à caractères préformés Il existe deux types d'imprimantes à impact. Celles utilisant, comme nous l'avons dit plus haut, la technique des machines à écrire adaptée aux imprimantes :

- les boules,
- les marguerites,
- les bandes,



- les cylindres, et celles utilisant l'impression par impact à aiguilles. Il ne faut pas croire que les imprimantes à impact ont été abandonnées. Ce serait un tort. Elles permettent l'impression de documents simples ou de listings en double, voire triple, ce que les autres imprimantes ne permettent pas (encre ou laser). Les imprimantes utilisant les caractères préformés ne sont ni plus, ni moins, que des machines à écrire équipées d'un microprocesseur et de ses composants permettant de transformer les données envoyées par l'ordinateur en caractères imprimables. Leurs performances sont semblables à celles indiquées dans le premier paragraphe. Elles ne sont pas rapides étant donné le mode de fonctionnement utilisé.

Les imprimantes à aiguilles

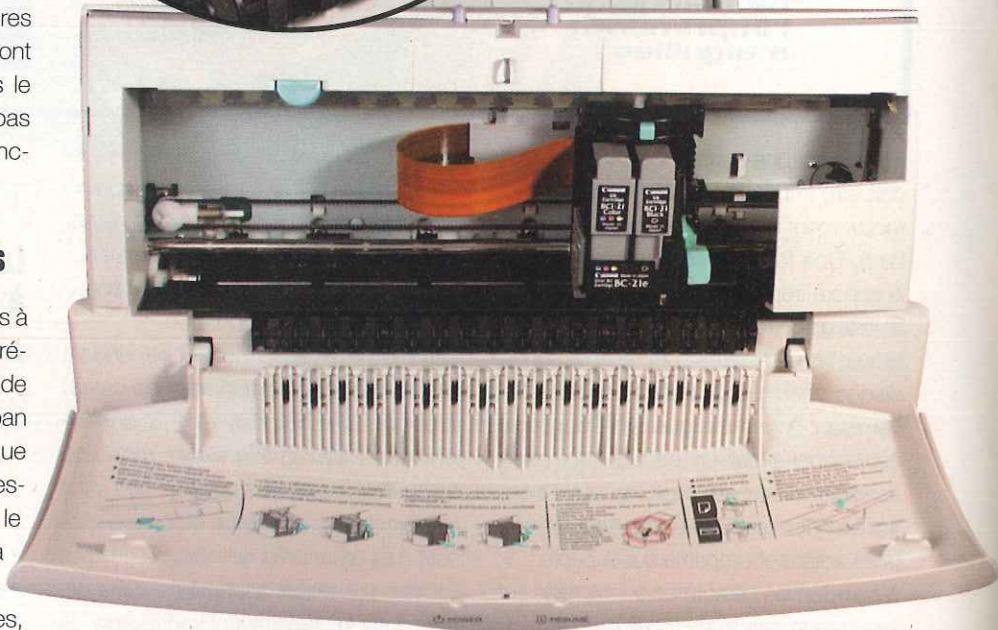
Il en va tout autrement des imprimantes à aiguilles. Les caractères ne sont pas préformés mais dessinés par une matrice de 9 ou 24 aiguilles qui frappent un ruban encreur et impriment le papier à chaque fois qu'une aiguille est sollicitée. Le dessin représenté en **figure 1** montre le principe de ce type d'impression et la technique employée. Le dessin de gauche montre les lignes et les colonnes,

tandis que le dessin de droite détaille l'intérieur d'une tête d'impression.

Les aiguilles, qu'elles soient au nombre de 9 ou 24, sont disposées verticalement. Afin de frapper le ruban encreur qui sera appliqué sur le papier, chacune d'entre elles est propulsée par un électro-aimant sur une colonne. Puis la tête se déplace latéralement et les aiguilles frappent la seconde colonne, et ainsi de suite jusqu'à la dernière (80 en principe, papier A4). L'avance de la tête est assurée par un moteur pas-à-pas et une courroie crantée. Cette avance a été calculée afin que chaque pas du moteur ne fasse avancer la tête d'impression que d'une colonne.

Arrivée en fin de ligne, le papier est déroulé au moyen d'un second moteur et la tête revient en début de ligne et l'impression peut continuer.

Ces imprimantes sont pilotables à partir du clavier de l'ordinateur mais également par une application utilisant des codes de contrôle. La plupart dispose de plusieurs jeux de caractères et sont capables de reproduire des graphismes. Celles équipées d'une tête comportant 24 aiguilles sont nettement plus rapides que celles disposant d'une tête à 9 aiguilles. Les premières offrent bien évidemment une bien meilleure résolution dont le maximum est de 360 DPI (Dot Per Inch = point par pouce, 1 pouce = 2,54 cm).



modernes des imprimantes matricielles acceptent le papier listing dont l'avance est assurée par des roues à picots ainsi qu'un chargeur de feuille automatique. Un commutateur mécanique permet de sélectionner le papier qui est employé. Pour information, nous devons également parler des imprimantes matricielles sans impact qui utilisent des procédés et des papiers spéciaux :

- la plus simple est l'imprimante thermique dont la tête d'impression est équipée d'électrodes chauffantes. Le papier thermique change de couleur lorsqu'une source de chaleur lui est appliquée,
- l'imprimante sur papier photosensible qui utilise deux sortes de papier. L'un est recouvert de capsules microscopiques photosensibles contenant les trois couleurs de base : le cyan, le magenta et le jaune. Lorsque celles-ci sont exposées à une lumière comportant trois longueurs d'onde différentes, les capsules sont activées et déposent leur contenu sur le second papier servant de révélateur
- l'imprimante à transfert thermique qui utilise des rubans et des films de transfert spéciaux. Ceux-ci sont recouverts d'une encre spéciale à base de cire. Des électrodes chauffantes fondent cette cire sur le papier.

Le dessin représenté en **figure 2** montre le principe de cette technologie.

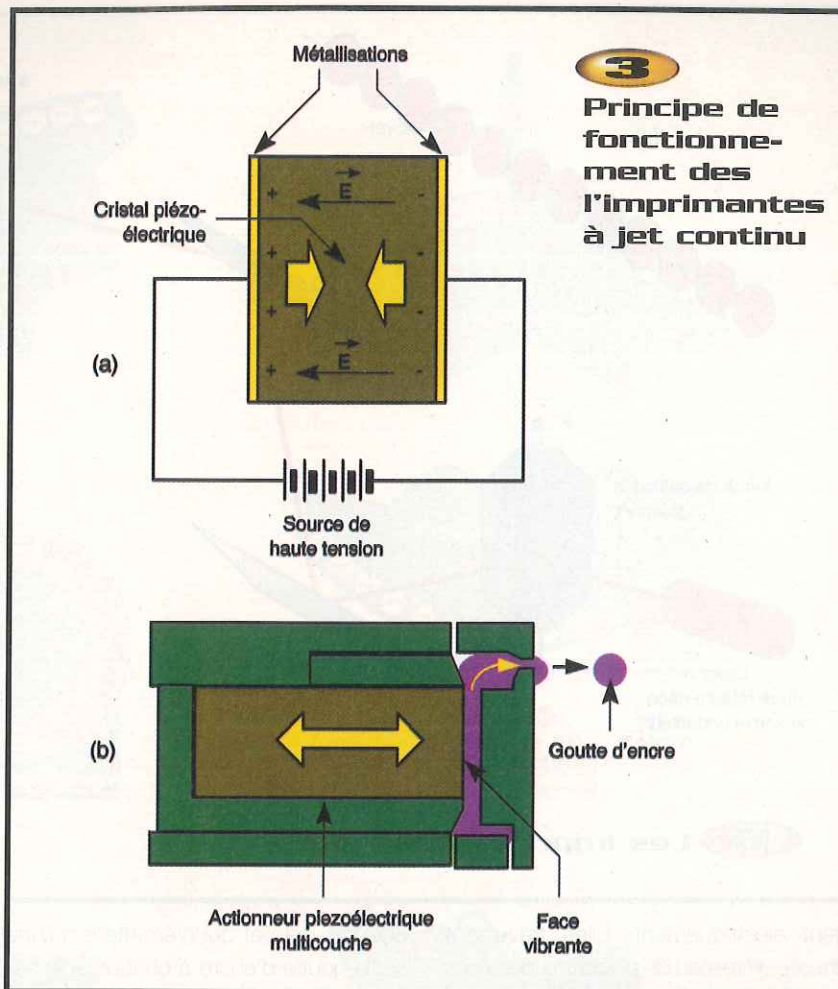
Les imprimantes à jet d'encre

Dans cette technologie, l'impression se fait sans impact, ce qui ne permet pas d'obtenir des doubles pendant l'impression d'une page. Cela ne nous semble pas un handicap étant donné que l'on peut demander, vu la vitesse d'impression, plusieurs exemplaires d'un même document. Mais alors, revers de la médaille, une plus grande quantité d'encre est utilisée et le prix de revient est augmenté d'autant. Cependant, la rapidité et la qualité de l'impression compensent très largement le coût de revient.

Comme pour l'impression matricielle à aiguilles, l'impression par procédé du jet d'encre comporte plusieurs technologies :

- le jet continu,
- le jet discontinu à basse pression,
- le jet discontinu à bulles.

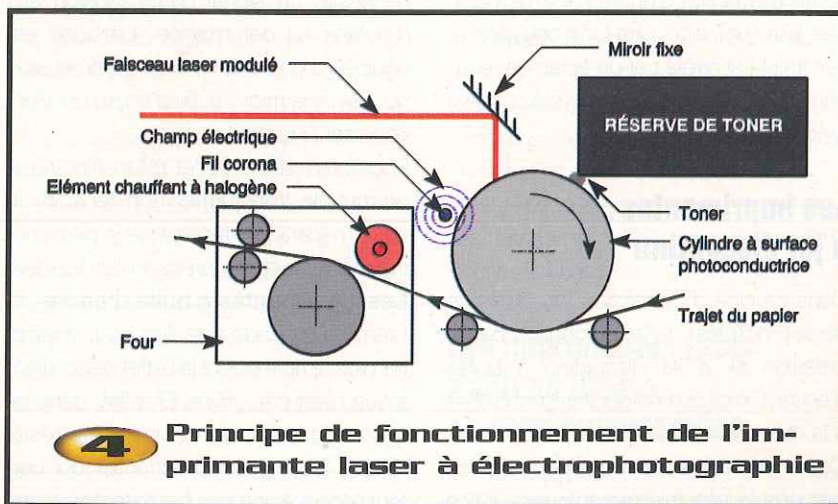
Ces imprimantes n'utilisent pas de ruban,

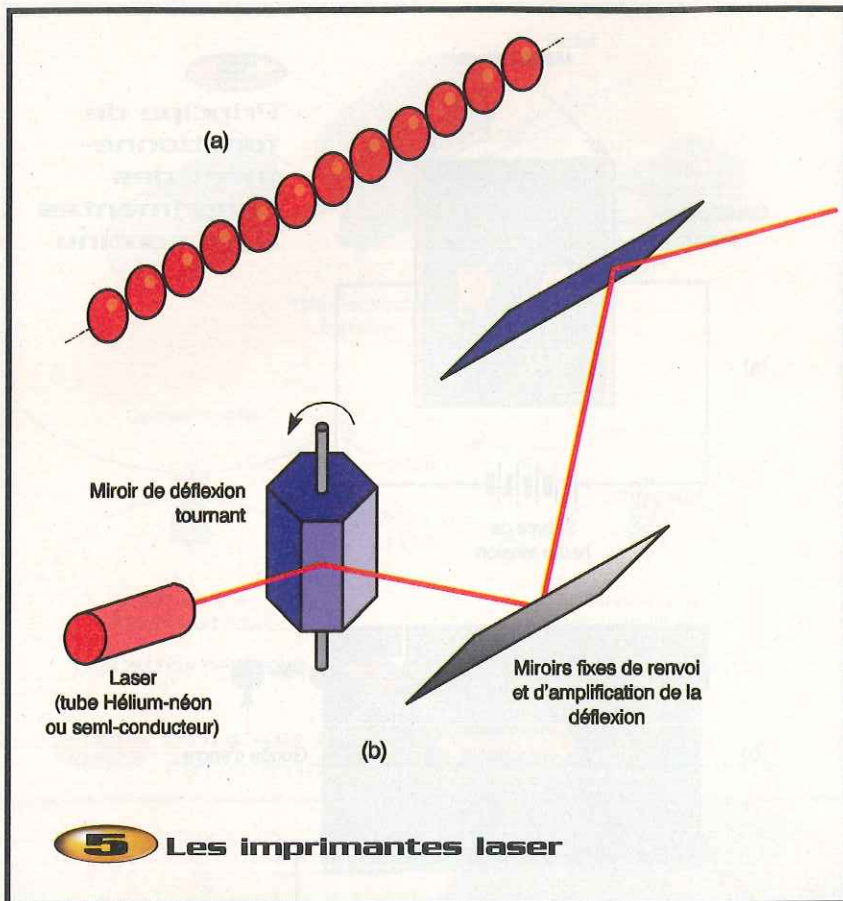


mais des cartouches d'encre noire et couleurs (les trois couleurs de base). Leur technologie est bien plus complexe que les imprimantes à impact car dans ce procédé de fines gouttelettes d'encre doivent impressionner le papier. La qualité de ce système d'impression dépasse largement celle des imprimantes matricielles pour peu que l'on emploie un papier spécial, relativement cher mais qui donne une résolution de type photographique.

Les imprimantes à jet continu

Ces imprimantes utilisent, comme leur nom l'indique, un jet continu d'encre qui est projeté sur le papier au moyen d'une buse capillaire. Mais avant de sortir de la buse, un résonateur piézo-électrique transforme ce jet en fines gouttelettes calibrées (quelques dizaines de μm) passant entre deux électrodes qui les char-





gent électriquement. Elles traversent ensuite un réseau de plaques défectrices plus ou moins polarisées qui dévient leur trajectoire horizontalement ou verticalement afin de dessiner le caractère ou le graphique. Ce système peut être comparé au système employé dans les téléviseurs ou les oscilloscopes lorsque le faisceau d'électrons est plus ou moins dévié avant d'atteindre la surface interne du tube cathodique afin d'obtenir une image. La représentation schématisée de ce type d'impression est représentée en **figure 3**. Puisque le jet est continu, une certaine quantité d'encre n'est pas utilisée. Elle subit alors une forte déviation et son trajet est arrêté par un écran de récupération. Cette encre est recyclée et peut à nouveau traverser la buse.

Les imprimantes à jet discontinu

Dans ce type d'impression, on distingue deux procédés : le jet discontinu basse pression et le jet discontinu à bulles d'encre. On dit que les gouttes sont créées à la demande (DOD, Drop On Demand). Ces imprimantes sont équipées de nombreuses buses microscopiques (300 à

600 par pouce) qui n'émettent qu'une seule goutte d'encre à chaque sollicitation, ce qui rappelle le procédé d'impression matricielle à impact mais qui procure une résolution non comparable en qualité.

Les imprimantes à jet discontinu basse pression utilisent, comme pour l'imprimante à jet continu, l'effet piézo-électrique. Mais dans ce cas, le cristal piézo-électrique comprime un diaphragme qui, par effet de pression, projette l'encre à travers les buses (ou cracheurs). Se reporter à nouveau en figure 3. Le cristal nécessite un signal haute tension afin d'obtenir sa déformation. La buse est équipée d'une chambre de compression qui éjecte la micro-bulle d'encre qui vient imprimer le papier.

Pour information, l'effet piézo-électrique permet de transformer une tension en force mécanique, tandis que la piézométrie concerne la compression des liquides.

Les imprimantes à bulle d'encre utilisent un procédé que l'on peut qualifier de plus simple puisque l'effet piézo-électrique n'est pas utilisé. En effet, dans ce cas les buses sont équipées de résistances électriques chauffantes qui portent l'encre à une très haute température.

Celle-ci se gazéifie et pousse vers l'extérieur une bulle d'encre. Lorsque la résistance n'est plus alimentée, la bulle se rétracte et projette vers le papier une goutte d'encre.

Les imprimantes laser

Dernières-nées des imprimantes, elles offrent une résolution et un rendu graphique très nettement supérieurs à celles que nous avons décrites auparavant. Il faut cependant signaler que les consommables sont d'un prix de revient élevé. Le prix d'une cartouche d'encre (toner) vaut effectivement plusieurs centaines de francs.

On distingue parmi ces imprimantes, deux types principaux qui utilisent deux procédés différents :

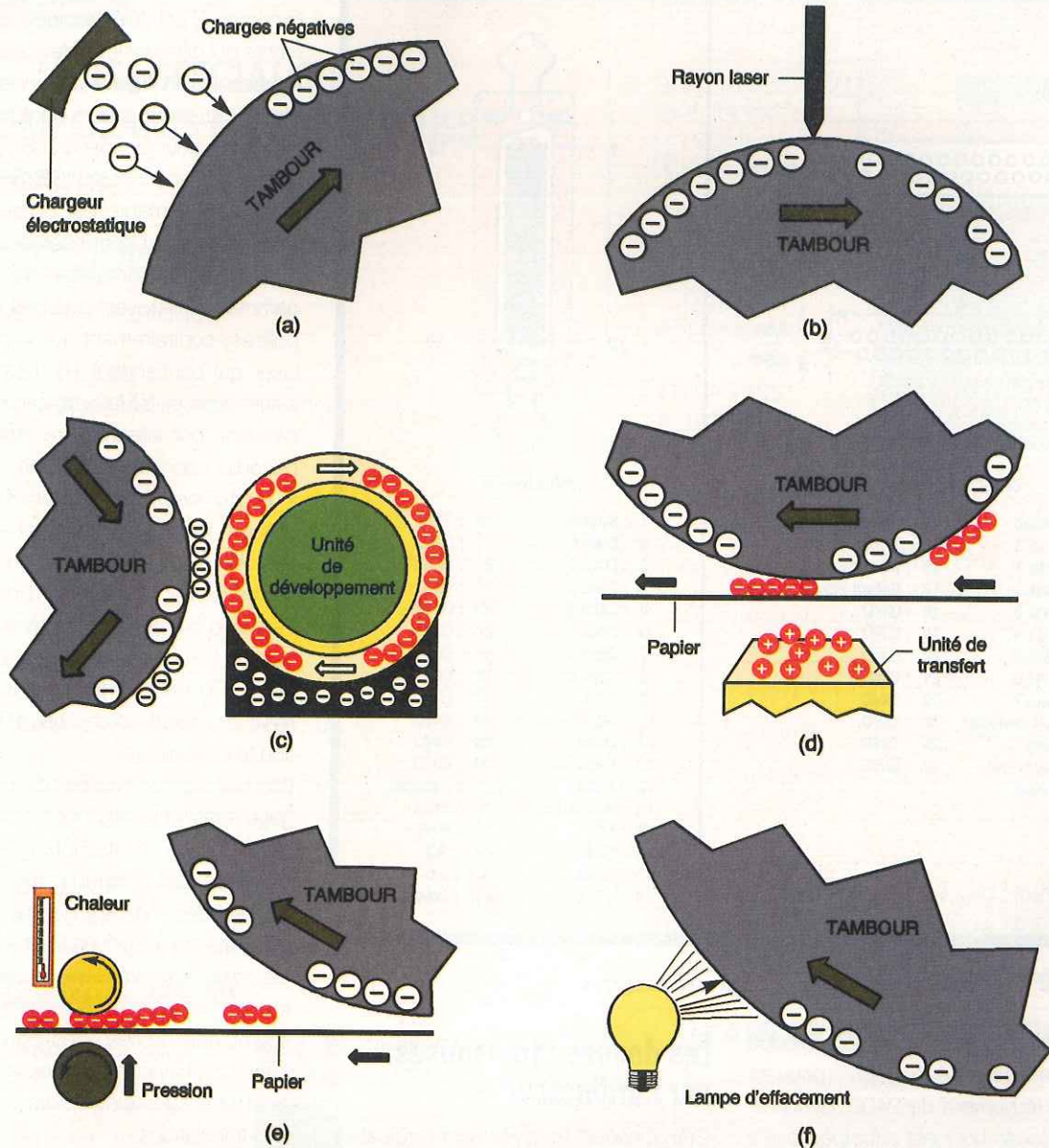
- l'électrophotographie,
- l'électroluminescence.

L'encre est constituée d'une poudre et le système fonctionne pratiquement comme une photocopieuse. Sur certaines d'entre elles, le tambour impressionnant le papier est inamovible, alors que sur d'autres, le tambour est inséré dans le toner, ce qui est nettement plus fiable. En aucun cas le tambour ne doit être exposé à une lumière directe, sous peine de dégâts irréversibles.

L'imprimante laser à électrophotographie

Le principe de fonctionnement de cette imprimante est représenté en **figure 4**. Il est plus complexe que celui des imprimantes que nous avons décrites plus haut. Le principal composant de ces machines est un tambour (ou cylindre) recouvert d'une matière photoconductrice composée de sélénium.

Lors de l'envoi d'un document par l'ordinateur, le tambour entre en rotation devant un chargeur électrostatique (effet Corona) et est électrisé en recevant des charges négatives. La surface du cylindre est ensuite soumise au balayage du faisceau laser, faisceau qui est modulé par les instructions reçues par l'imprimante. Lorsque le rayon laser illumine fortement l'une des parties de la surface du cylindre, les charges négatives sont éliminées. Lorsqu'une partie du tambour n'est pas soumise à la lumière du rayon laser les charges négatives restent en,



6 Fonctionnement du chargement du tambour

place. Puisque l'intensité du rayon est modulée, les charges électriques disponibles sur la totalité de la surface du tambour le sont également. On obtient ainsi une image semblable à un négatif photographique. Le tambour continuant sa rotation, il passe devant le toner dont les particules, sous l'effet des charges électriques, se collent à la surface du cylindre. Les particules noires ou colorées sont ensuite déposées sur le papier, lequel est ensuite chauffé à une température élevée qui provoque la fusion des particules d'encre qui resteront indélébiles.

On peut, à l'aide de ces imprimantes,

imprimer toutes sortes de supports, que ce soit du papier ou des transparents. Ces derniers sont très utilisés par les électroniciens car cela leur permet d'obtenir des typons de circuits imprimés dont le tracé est totalement fidèle au dessin que l'on a conçu sur l'ordinateur.

Le dessin donné en **figure 5**, en "b", représente le balayage du tambour de l'imprimante par un rayon laser dont le faisceau est dévié par un miroir de déflexion tournant et par deux autres miroirs fixes. La figure dessinée en "a" montre le second système employé par les imprimantes laser à LED.

La **figure 6** explique le fonctionnement du chargement du tambour de l'imprimante laser par les charges négatives puis le tracé de l'image, le développement de l'image puis son impression sur le papier et la cuisson de ce dernier, et en dernier lieu l'effacement des charges sur le tambour (respectivement a, b, c, d, e, f).

Les imprimantes laser à LED ou diodes électroluminescentes

Dans ce type d'imprimantes, le rayon laser n'existe plus et est remplacé par

CONNECTEUR DB25 PORT PARALLELE	CONNECTEUR CENTRONICS																																																														
<p>DB 25</p> <p>47 2.84 2.54* 2.77</p> <p>* Suivant modèle</p> <p>BROCHAGE</p> <table> <tr><td>1 : Strobe</td><td>14 : Auto feed</td></tr> <tr><td>2 : Data 0</td><td>15 : Error</td></tr> <tr><td>3 : Data 1</td><td>16 : Initialize</td></tr> <tr><td>4 : Data 2</td><td>17 : Select Input</td></tr> <tr><td>5 : Data 3</td><td>18 : GND</td></tr> <tr><td>6 : Data 4</td><td>19 : GND</td></tr> <tr><td>7 : Data 5</td><td>20 : GND</td></tr> <tr><td>8 : Data 6</td><td>21 : GND</td></tr> <tr><td>9 : Data 7</td><td>22 : GND</td></tr> <tr><td>10 : Acknowledge</td><td>23 : GND</td></tr> <tr><td>11 : Busy</td><td>24 : GND</td></tr> <tr><td>12 : Paper out</td><td>25 : GND</td></tr> <tr><td>13 : Select</td><td></td></tr> </table>	1 : Strobe	14 : Auto feed	2 : Data 0	15 : Error	3 : Data 1	16 : Initialize	4 : Data 2	17 : Select Input	5 : Data 3	18 : GND	6 : Data 4	19 : GND	7 : Data 5	20 : GND	8 : Data 6	21 : GND	9 : Data 7	22 : GND	10 : Acknowledge	23 : GND	11 : Busy	24 : GND	12 : Paper out	25 : GND	13 : Select		<p>18 19 36</p> <p>BROCHAGE</p> <table> <tr><td>1 : Strobe</td><td>19 : GND</td></tr> <tr><td>2 : Data 0</td><td>20 : GND</td></tr> <tr><td>3 : Data 1</td><td>21 : GND</td></tr> <tr><td>4 : Data 2</td><td>22 : GND</td></tr> <tr><td>5 : Data 3</td><td>23 : GND</td></tr> <tr><td>6 : Data 4</td><td>24 : GND</td></tr> <tr><td>7 : Data 5</td><td>25 : GND</td></tr> <tr><td>8 : Data 6</td><td>26 : GND</td></tr> <tr><td>9 : Data 7</td><td>27 : GND</td></tr> <tr><td>10 : Acknowledge</td><td>28 : GND</td></tr> <tr><td>11 : Busy</td><td>29 : GND</td></tr> <tr><td>12 : Paper out</td><td>30 : GND</td></tr> <tr><td>13 : Select</td><td>31 : Initialize</td></tr> <tr><td>14 : Auto feed</td><td>32 : Error</td></tr> <tr><td>15 : NC</td><td>33 : GND</td></tr> <tr><td>16 : GND</td><td>34 : NC</td></tr> <tr><td>17 : GND</td><td>35 : + 5 V</td></tr> <tr><td>18 : + 5 V</td><td>36 : Select Input</td></tr> </table>	1 : Strobe	19 : GND	2 : Data 0	20 : GND	3 : Data 1	21 : GND	4 : Data 2	22 : GND	5 : Data 3	23 : GND	6 : Data 4	24 : GND	7 : Data 5	25 : GND	8 : Data 6	26 : GND	9 : Data 7	27 : GND	10 : Acknowledge	28 : GND	11 : Busy	29 : GND	12 : Paper out	30 : GND	13 : Select	31 : Initialize	14 : Auto feed	32 : Error	15 : NC	33 : GND	16 : GND	34 : NC	17 : GND	35 : + 5 V	18 : + 5 V	36 : Select Input
1 : Strobe	14 : Auto feed																																																														
2 : Data 0	15 : Error																																																														
3 : Data 1	16 : Initialize																																																														
4 : Data 2	17 : Select Input																																																														
5 : Data 3	18 : GND																																																														
6 : Data 4	19 : GND																																																														
7 : Data 5	20 : GND																																																														
8 : Data 6	21 : GND																																																														
9 : Data 7	22 : GND																																																														
10 : Acknowledge	23 : GND																																																														
11 : Busy	24 : GND																																																														
12 : Paper out	25 : GND																																																														
13 : Select																																																															
1 : Strobe	19 : GND																																																														
2 : Data 0	20 : GND																																																														
3 : Data 1	21 : GND																																																														
4 : Data 2	22 : GND																																																														
5 : Data 3	23 : GND																																																														
6 : Data 4	24 : GND																																																														
7 : Data 5	25 : GND																																																														
8 : Data 6	26 : GND																																																														
9 : Data 7	27 : GND																																																														
10 : Acknowledge	28 : GND																																																														
11 : Busy	29 : GND																																																														
12 : Paper out	30 : GND																																																														
13 : Select	31 : Initialize																																																														
14 : Auto feed	32 : Error																																																														
15 : NC	33 : GND																																																														
16 : GND	34 : NC																																																														
17 : GND	35 : + 5 V																																																														
18 : + 5 V	36 : Select Input																																																														

7 Brochage des connecteurs

des barrettes de diodes LED (diodes électroluminescentes) qui peuvent atteindre le nombre de 2400. L'impression sur le tambour est effectuée par la modulation de ces diodes lors de sa rotation. Lorsque le négatif est créé, l'impression s'effectue de la même manière que pour l'imprimante à rayon laser. Dans ce procédé, c'est la taille et le nombre des LED qui déterminent la résolution de l'imprimante.

Pour conclure la description des différentes imprimantes, il convient de signaler la différence de transcription des données issues de l'ordinateur. Tandis que les imprimantes matricielles ou à jet d'encre prennent en compte ces données ligne par ligne et l'impriment lorsqu'elle est pleine, les imprimantes laser reçoivent une page entière et l'impriment ensuite toujours en mode graphique, que ce soit du texte ou des graphismes. Elles sont si rapides que leur impression se compte en page par minute (10 à 12 PPM est maintenant un minimum).

Les données transmises par l'ordinateur

Ces données, envoyées par l'ordinateur, "tiennent" toujours sur un octet. Ce sont

soit des instructions, soit des données à imprimer. C'est l'électronique de l'imprimante qui différencie l'octet. Les dessins représentés en **figure 7** donnent le brochage des connecteurs dont sont équipés l'ordinateur (connecteur SUBD 25) et l'imprimante (connecteur CENTRONICS). Chaque imprimante est dotée de plusieurs polices qui sont insérées dans des ROM. Les imprimantes de bas de gamme ne disposent que de quelques polices, contrairement aux imprimantes laser qui contiennent en mémoire plusieurs dizaines de jeux de caractères qui peuvent, par ailleurs, être augmentées par adjonction de cartouches.

Il existe deux principales sortes de polices : les polices bitmap et les polices vectorielles. Elles sont totalement différentes de par leur qualité d'impression. La police bitmap est imprimée point par point et l'on ne peut en changer la résolution. On peut effectivement l'agrandir mais le résultat obtenu est d'une définition très médiocre.

Les polices vectorielles utilisent un langage spécifique (vectorisé) dont les principaux sont le Postscript., le PCL5 et le True Type. Le Vectoriel permet de décrire par des dessins géométriques les données qui sont envoyées par l'ordinateur. Ce sont des formules qui décrivent la forme des données à imprimer, que ce soient des droites, des courbes, des contours. Chacune de ces impressions peut être agrandie, déformée ou subir des translations.

P. OGUIC

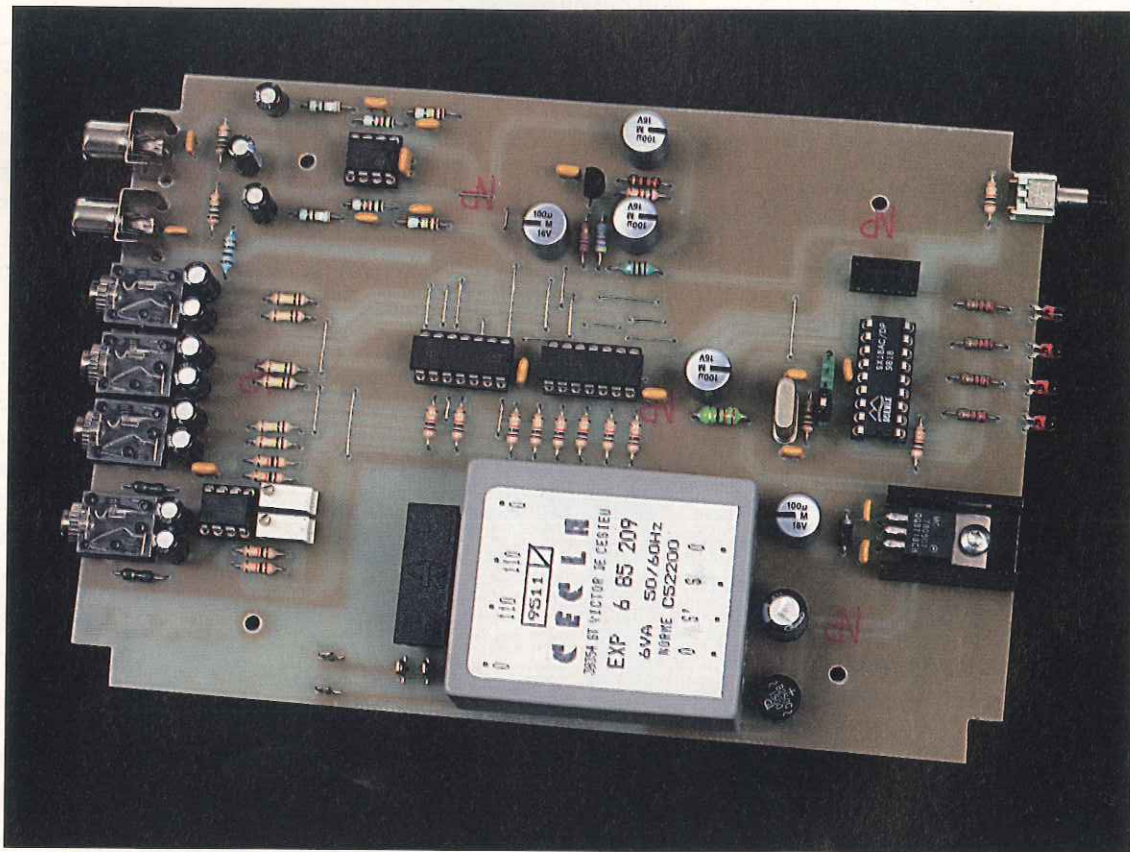


Quelques cartouches d'encre Canon

Switch audio

pour PC ou chaîne Hi-Fi

Dans une installation audio, il est fréquent de manquer d'entrée pour connecter un nouvel appareil. Commence alors une valse des fiches : déconnexion, reconnexion, ... Ce travail est d'autant plus délicat que, généralement, les connecteurs sont placés à l'arrière des appareils. Actuellement, la plupart des PC possèdent une carte son (SB ou autres). Si on désire connecter plusieurs appareils sur l'entrée de cette carte, on se rend vite compte que l'on dispose d'une seule entrée "LINE". De la même manière, sur cette chaîne Hi-Fi, il est fréquent de ne pas disposer d'assez d'entrées sur l'amplificateur (problème de plus en plus courant avec le Home Cinéma).



La réalisation que nous vous proposons permettra de palier élégamment à ces lacunes. Vous disposerez alors de 3 entrées supplémentaires, dont 1 entrée RIAA. Vous pourrez passer de l'une à l'autre des entrées par une simple pression sur un bouton-poussoir.

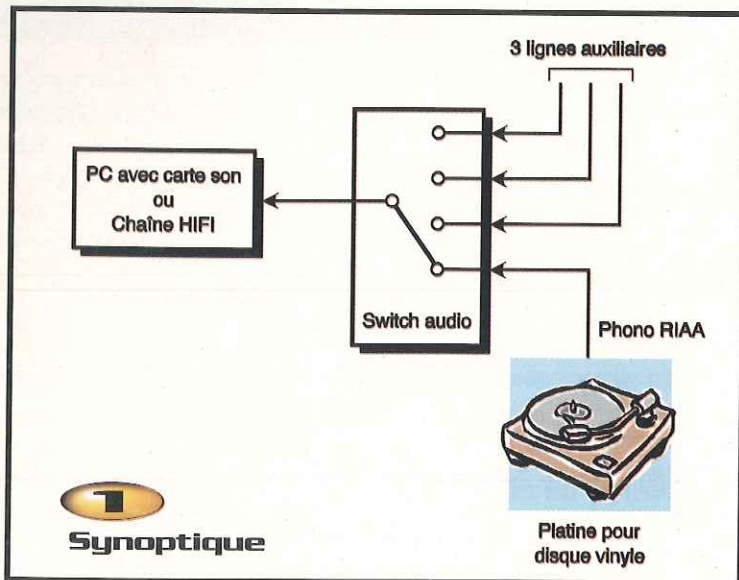
Revenons sur l'entrée RIAA. Il s'agit d'une entrée dédiée à la connexion d'une platine pour disque vinyle. Elle possède une préamplification spécifique. Grâce à cette entrée, vous pourrez, par exemple, connecter votre platine disque à votre PC. Si vous disposez d'un graveur, vous pourrez ainsi créer une copie de sauvegarde de vos disques vinyles. De même, si votre amplificateur ne possède pas d'entrée RIAA, ce qui est couramment le cas des nouveaux amplificateurs, vous pourrez à nouveau profiter de votre collection de vinyles.

Connectique

Venons-en maintenant à l'aspect technique en décrivant les différents étages de la réalisation. Tout d'abord, pour un gain de place, les entrées/sorties ligne se connectent à

des fiches Jack 3,5mm. A ce titre, sur la **figure 3**, vous pourrez voir le repérage normalisé des connecteurs Jack 3,5mm.

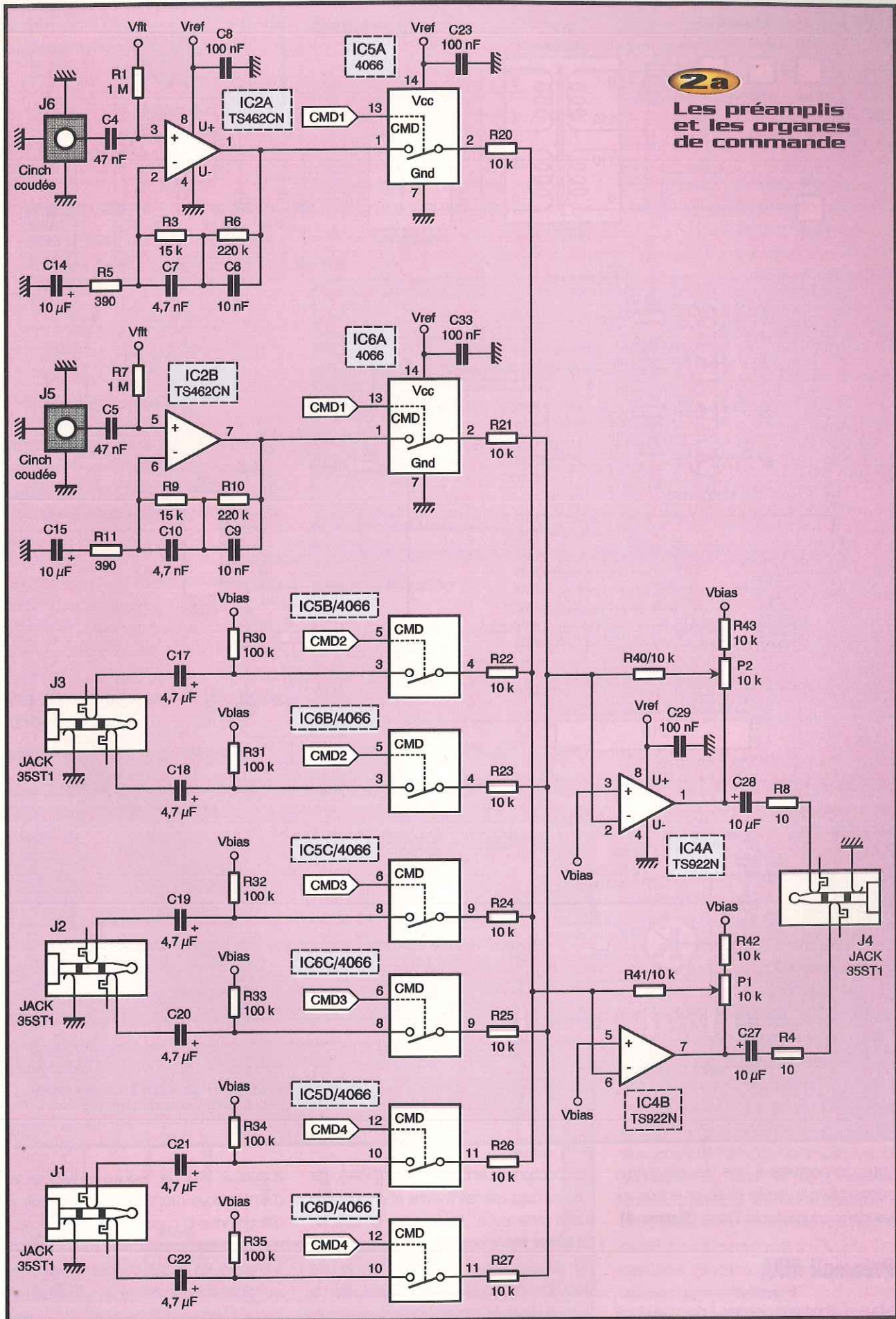
L'entrée RIAA fait, quant à elle, appel à des fiches Cinch. Sur la plupart des platines, il y a un fil de masse que

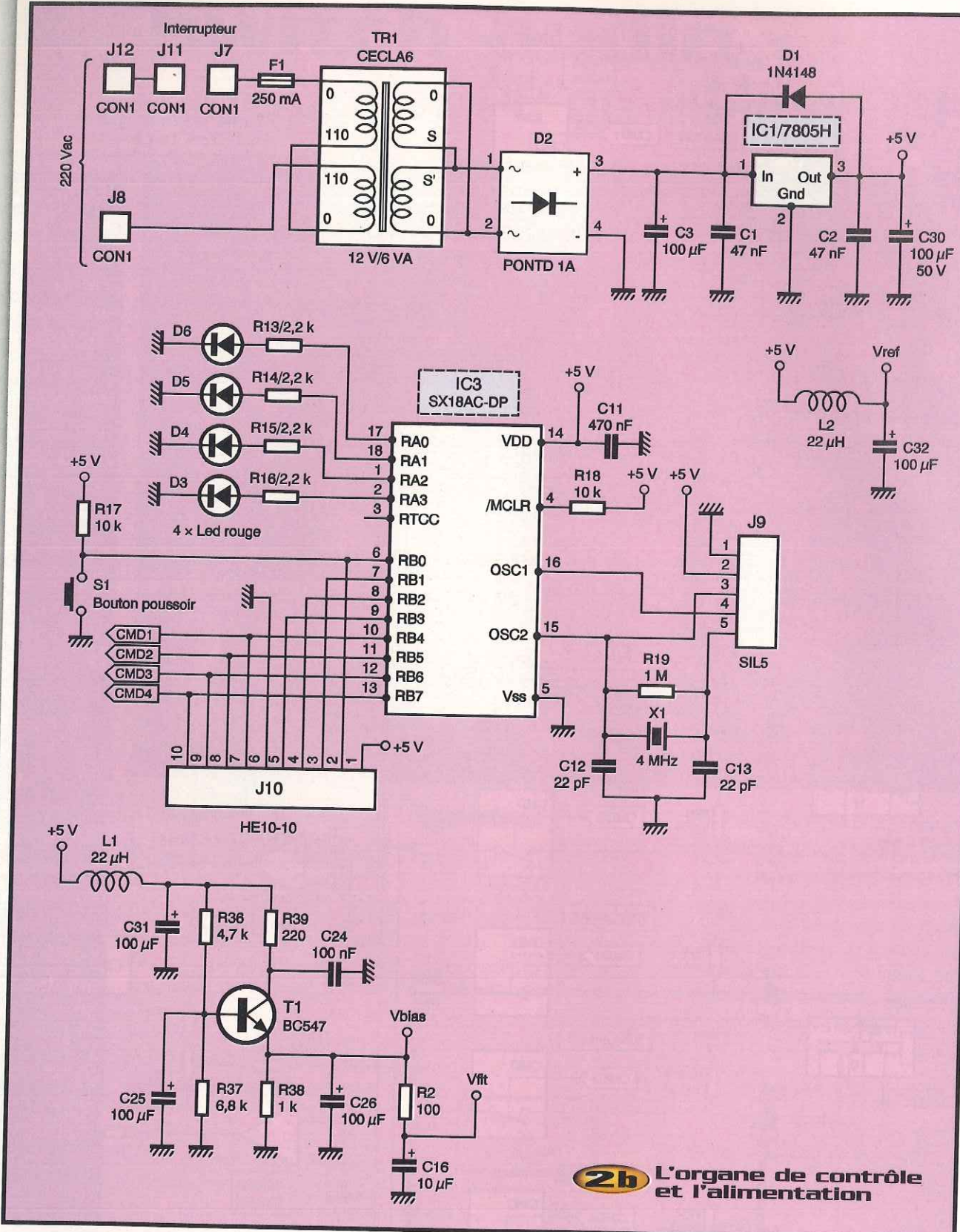


1
Synoptique

2a

Les préamplis et les organes de commande





2b L'organe de contrôle et l'alimentation

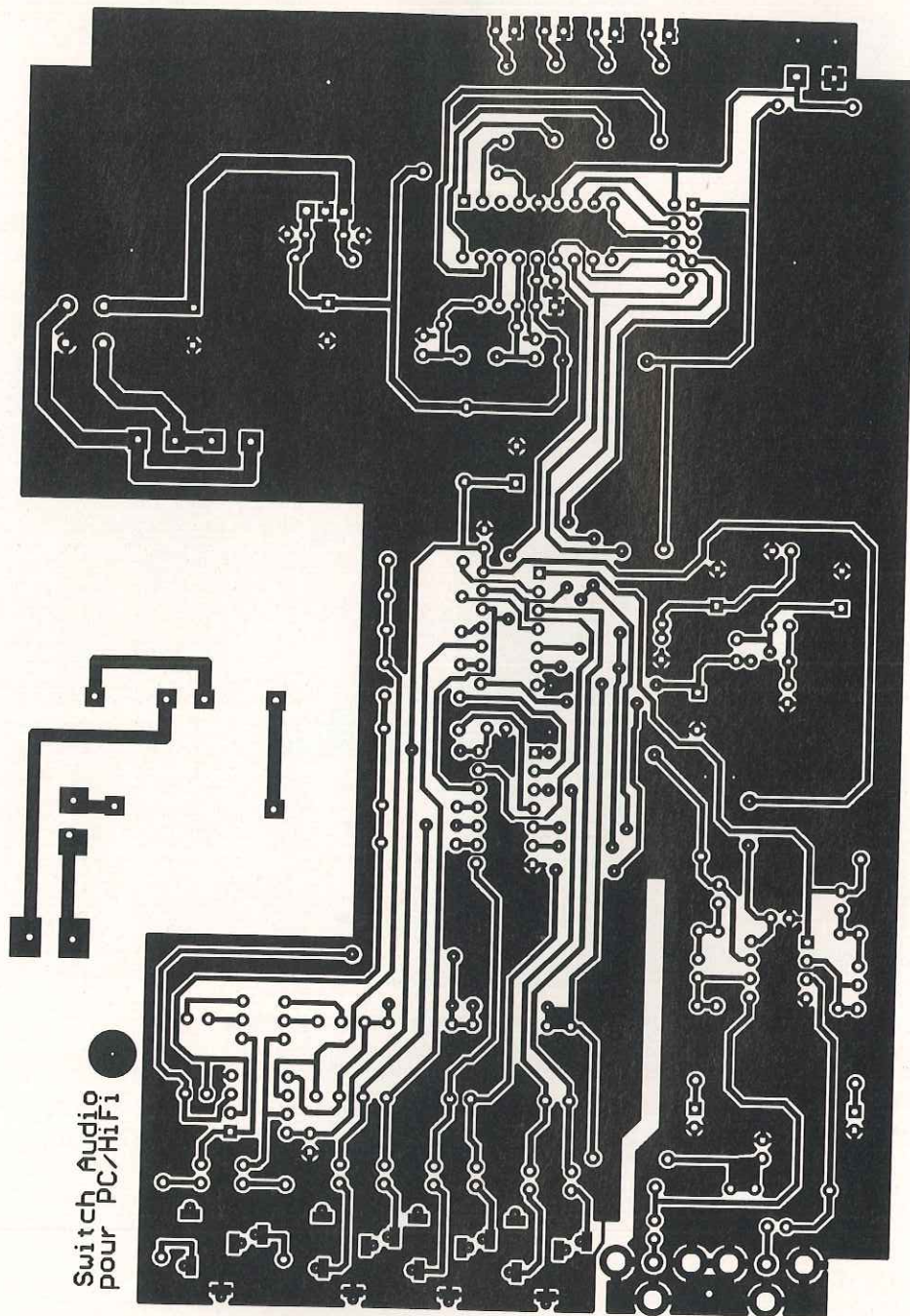
vous connecterez à une vis, elle-même connectée à la pastille située en entrée sur les deux connecteurs Cinch (**figure 4**).

Préampli RIAA

Nous n'allons pas détailler à nouveau tous

les fonctionnements du préampli RIAA, car cela a déjà été fait mainte et mainte fois dans cette revue. Juste pour mémoire, un préampli RIAA est nécessaire pour amplifier et déaccentuer les signaux lus par une tête sur un disque vinyle. En effet, sur un disque vinyle, le son est très fortement pré-

accentué afin de réduire la distorsion, d'améliorer le rapport signal sur bruit et d'augmenter la capacité d'enregistrement sur un disque. La préaccentuation consiste à amplifier les hautes fréquences au détriment des basses fréquences. Par conséquent, la déaccentuation consiste à réali-



7
Circuit
imprimé

Switch Audio
pour PC/HiFi

gramme est d'une extrême simplicité, toujours dans un but d'initiation au SX18. Si vous désirez en savoir plus sur le SX18 de SCENIX, vous pouvez vous référer au site Web de SCENIX :

<http://www.ubicom.com> et au livre "Les Microcontrôleurs SCENIX" aux éditions DUNOD.

Buffer de sortie

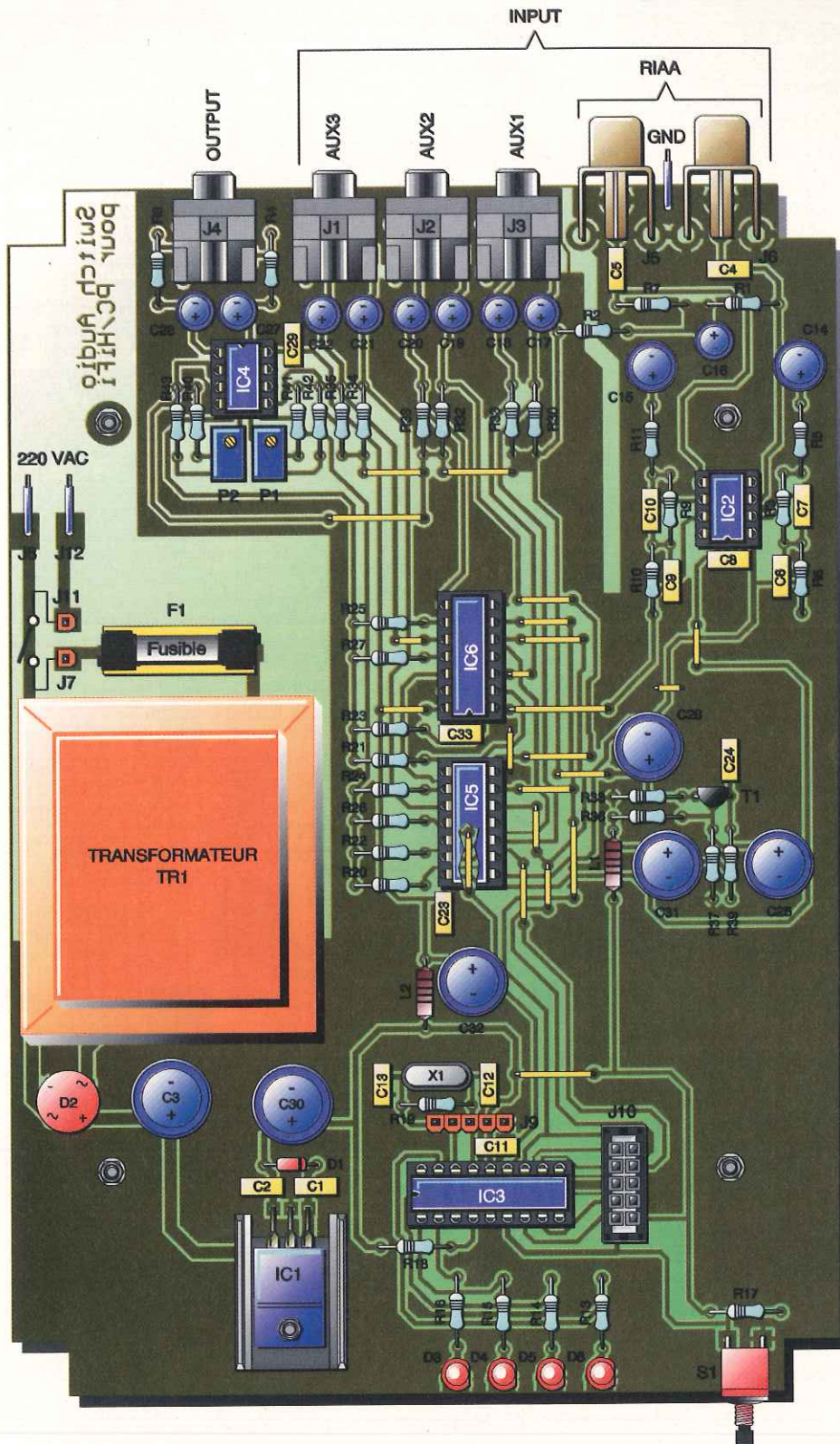
Dernière les commutateurs analogiques, si on désire améliorer la chaîne analogique, on peut bufferiser le signal. Cette bufferisa-

tion permet de régler le niveau de sortie et de driver correctement le câble d'interconnexion. Comme précédemment, l'ampli OP utilisé peut être changé sans grande difficulté. Pour notre part, nous avons utilisé un TS922, également très bien adapté à des applications audio. De plus, le TS922 permet de connecter directement un casque 32Ω sur la sortie. Les niveaux de sortie sont réglables par l'intermédiaire de P_1 et de P_2 . Sachez que les niveaux lignes varient entre 224 et 775mVrms suivant les normes. La meilleure solution consiste à réaliser des essais avec le PC ou l'ampli de

la chaîne hi-fi, et de choisir un niveau correspondant aux autres périphériques, afin qu'il n'y ait pas trop de différences de niveau entre l'entrée, où est connecté le Switch, et les autres entrées, ce qui est généralement plus plaisant.

Alimentation

L'alimentation générale est de type linéaire. Elle est embarquée sur la carte afin d'éviter l'usage d'un bloc secteur. On peut remarquer le soin particulier qui est apporté au filtrage des alimentations (VREF) et des ten-



8
Implantation
des
composants

sions de polarisation (VBIAS et VFLT). Il ne s'agit pas d'un luxe inutile. La qualité de la chaîne audio et, notamment, le bruit présent dans cette chaîne dépendent directement de la qualité des différents filtrages. Par conséquent, il est possible de supprimer les filtres au détriment du bruit. Ce qui présente

un risque important, sachant que la chaîne audio est au voisinage d'un microcontrôleur.

Coffret

La carte a été réalisée de manière à entrer dans un boîtier D30. Ce boîtier, très

célèbre, a l'avantage d'être répandu, peu cher et facile à utiliser. La fixation sera possible par quatre vis sur les mini-collettes présentes au fond du boîtier D30. En ce qui concerne les faces avant et arrière, nous vous proposons une version sur les **figures 5 et 6.**

Réalisation

Le circuit imprimé est en simple face de 176x120mm. Sa réalisation et son câblage ne représentent pas de difficulté particulière. L'approvisionnement des composants reste assez facile. Dans le cas où vous chargez les amplis OP, respectez bien les quelques consignes citées précédemment. Les connecteurs J₁₂ et J₉ vous permettront de souder les fils d'alimentation secteur et J₇ et J₁₁ de connecter un interrupteur afin de mettre hors ou sous tension l'ensemble de la carte. Attention au choix des connecteurs Jack 3,5mm car il existe 3 ou 4 sortes de brochage. Les connecteurs choisis sont facilement repérables grâce à leur capot transparent.

Ph. ANDRE

Nomenclature

C₁, C₂, C₄, C₅ : 47 nF
 C₃, C₂₅, C₂₈, C₃₀ à C₃₂ : 100 µF
 C₈, C₉ : 10 nF
 C₇, C₁₀ : 4,7 nF
 C₆, C₂₃, C₂₄, C₂₉, C₃₃ : 100 nF
 C₁₁ : 470 nF
 C₁₂, C₁₃ : 22 pF
 C₁₄ à C₁₆, C₂₇, C₂₈ : 10 µF
 C₁₇ à C₂₂ : 4,7 µF
 L₁, L₂ : 22 µH
 R₁, R₇, R₁₉ : 1 MΩ
 R₂ : 100 Ω
 R₃, R₉ : 15 kΩ
 R₄, R₈ : 10 Ω
 R₅, R₁₁ : 390 Ω
 R₆, R₁₀ : 220 kΩ
 R₁₃ à R₁₆ : 2,2 kΩ
 R₁₇, R₁₈, R₂₀ à R₂₇, R₄₀ à R₄₃ : 10 kΩ
 R₃₀ à R₃₅ : 100 kΩ
 R₃₆ : 4,7 kΩ
 R₃₇ : 6,8 kΩ
 R₃₈ : 1 kΩ
 R₃₉ : 220 Ω
 P₁, P₂ : potentiomètres 10 kΩ
 D₁ : 1N4148
 D₂ : pont redresseur
 D₃ à D₆ : LED rouges 3mm
 T₁ : BC547
 F₁ : porte fusible + fusible 250mA
 IC₁ : 7805H
 IC₂ : TS462CN
 IC₃ : SX18AC-DP
 IC₄ : TS922N
 IC₅, IC₆ : 74HC4066
 J₁ à J₄ : jacks 3,5mm stéréo
 J₅, J₆ : Cinch coudées
 J₇, J₈, J11, J12 : cosse poignard
 J₉ : barrette sécable 5 broches
 J₁₀ : HE10 (10 broches femelles)
 S₁ : bouton-poussoir
 TR₁ : transfo CECLA 6VA/12V
 X₁ : quartz HC49 4 MHz

LISTING 1 : Programme Scenix pour switch audio

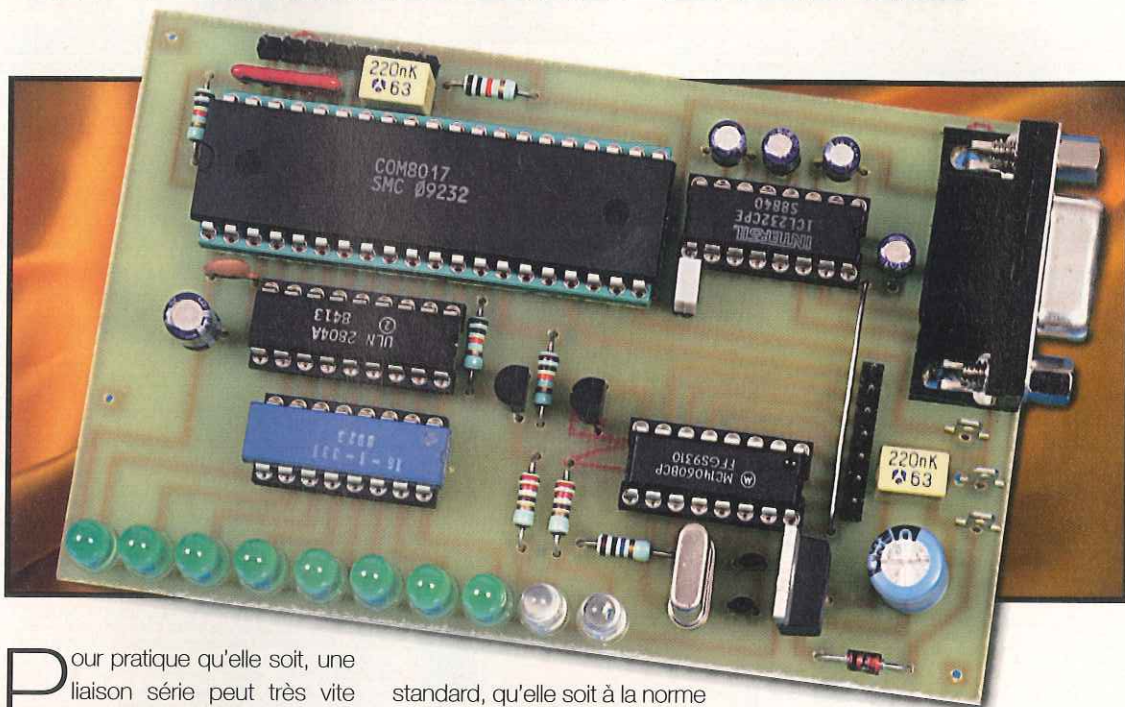
```

*****
;
; Init SYSTEME
*****
org $002 ;attention on démarre a cette adresse
;pour garder 2 octets de libre pour
;le debugueur ($000 + $001)

start clr rb
clr ra
mov M,$0F ;*** Mode 0Fh pour direction E/S
mov w,$01 ;RBO en entrées
mov lrb,w ;RB1,RB2,RB3,RB4,RB5,RB6 et RB7 en sorties
mov w,$00 ;
mov lra,w ;RA0,RA1,RA2,RA3 en sorties
mov M,$0E ;*** Mode 0Eh pour config pull-up
mov w,$00
mov lra,w ;enabled tous les pull-up du port A
mov w,$00
mov lrb,w ;enabled tous les pull-up du port B
clr ra
clr rb
mov w,$0F
mov ra,w ;allume toutes les leds
call tempo1 ;tempo
mov w,$00
mov ra,w ;eteint toutes les leds
call tempo1 ;tempo
setb ra.0 ;allume D6
setb rb.4 ;commande des 4066 sur canal 1
mov w,$01
mov memo,w ;charge registre de memorisation
*****
;
; Programme Principal
*****
loop snb rb.0 ;test si BP enfonce
jmp pp1
jmp loop
pp1 clc
rl memo ;incrimente memo
sb memo.4 ;test si superieur a 4
jmp pp3
mov memo,$01
pp3 call tempo1
pp2 sb rb.0 ;test si bp toujours enfonce
jmp pp4
jmp pp2
pp4 mov w,memo
and w,$0F
mov ra,w ;mise a jour des leds
mov byte1,w
rl byte1
rl byte1
rl byte1
rl byte1
mov w,byte1
mov rb,w ;mise a jour de la commande des 4066
jmp loop
*****
tempo1 mov byte3,$05 ;sous-programme de temporisation
t4_1 mov byte2,$FF
t4_2 mov byte1,$FF
t4_3 djnz byte1,t4_3
djnz byte2,t4_2
djnz byte3,t4_1
ret
*****
;
; Fin de programme
*****
end

```


Moniteur de liaison série



Les liaisons séries asynchrones, appelées le plus souvent liaisons séries "tout court", sont de plus en plus répandues, que ce soit dans un environnement de type PC ou dans un environnement de type microcontrôleur. En effet, ces derniers sont de plus en plus nombreux à disposer d'un UART intégré et, donc, d'au moins un port série asynchrone. Les célèbres Basic Stamp ont donné l'exemple mais ils sont loin aujourd'hui d'être les seuls dans ce cas.

Pour pratique qu'elle soit, une liaison série peut très vite devenir un véritable casse-tête lorsqu'elle ne fonctionne pas comme prévu, et les outils classiques de l'électronicien sont alors désespérément impuissants. En effet, il faut un sacré coup d'œil pour vérifier, sur l'écran d'un oscilloscope, que la succession de niveaux logiques hauts et bas correspond à tel ou tel caractère, surtout si l'oscilloscope en question n'a pas de mémoire, ce qui est le cas de la majorité des appareils que nous possédons. Nous vous proposons aujourd'hui de remédier à cette situation, avec notre moniteur de liaison série qui est capable de lire n'importe quel caractère voyageant sur une liaison de ce type, ainsi que d'indiquer toute erreur de format de transmission ou bien, encore, toute erreur de parité. Son prix de revient dérisoire et l'absence de toute utilisation de composants programmables pour sa réalisation rendent cette dernière accessible à tous.

Présentation

Notre montage peut être raccordé à toute liaison série asynchrone

standard, qu'elle soit à la norme RS232, comme c'est le cas sur un compatible PC ou un micro-ordinateur en général, ou à la norme TTL comme c'est le cas en sortie d'un microcontrôleur.

Il affiche, sur 8 LED, le code binaire de chaque caractère transitant sur la liaison qu'il reconnaît. 2 autres LED, de couleur rouge celles-là, indiquent, quant à elles, toute erreur de format ou toute erreur de parité.

Notre moniteur est capable de travailler aux principales vitesses normalisées sur les liaisons de ce type à savoir : 300, 600, 1200, 2400, 4800 et 9600 bauds. Enfin, il s'alimente sous une tension de 9 à 12V qui n'a même pas besoin d'être stabilisée. Un bloc secteur "prise de courant", voire même une simple pile de 9V pour des utilisations de courte durée, conviennent ainsi à merveille.

Le schéma

Si l'on ne veut pas utiliser de microcontrôleur, la solution la plus simple pour décoder des caractères sur une liaison série consiste tout simplement à faire appel à un composant dont c'est la fonction première,

à savoir l'UART. Ce dernier est toujours double puisqu'il contient un émetteur et un récepteur mais nous n'utilisons ici que la partie réception. C'est cet UART qui est au cœur de notre schéma, visible sous la référence IC₂ en **figure 1**. Il reçoit les caractères émanant de la liaison série, soit directement via l'entrée ETL dans le cas d'une liaison au niveau TTL (S₆ n'est alors pas mis en place), soit après adaptation de niveau dans le cas d'une liaison RS232.

Cette adaptation de niveau est confiée à un classique MAX232, repéré IC₁, qui se charge, à partir de la seule alimentation 5V du montage, de générer de véritables niveaux RS232. En effet, notre moniteur ne se contente pas de recevoir des caractères ; il génère aussi les niveaux logiques adéquats sur les lignes de contrôle CTS et DSR de la liaison RS232 afin que celle-ci puisse fonctionner correctement dans tous les cas.

Les pattes 39 à 35 de l'UART sont reliées à des interrupteurs externes qui permettent de configurer son mode de fonctionnement comme

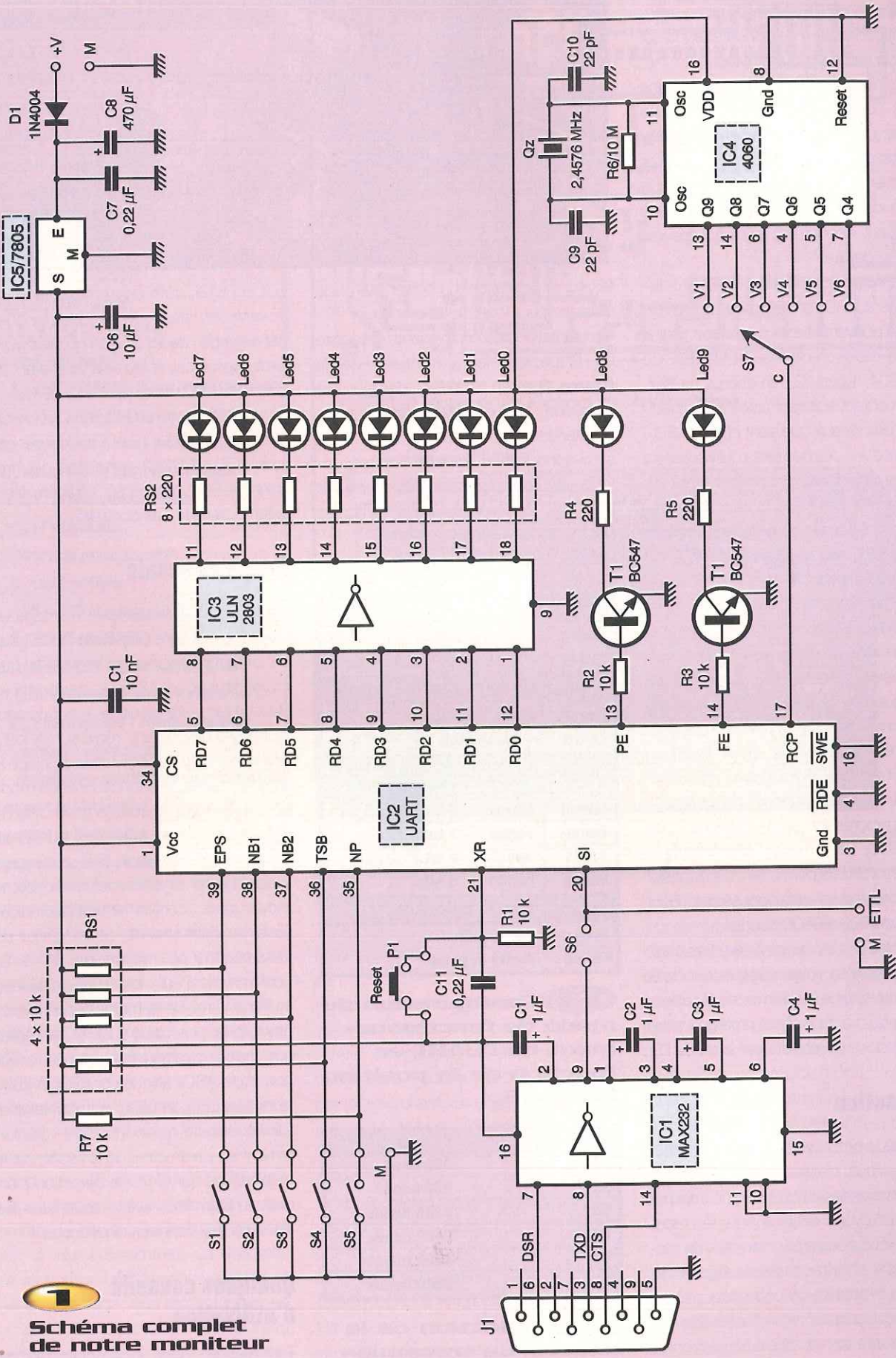


Schéma complet de notre moniteur

indiqué **tableau 1**. Il est ainsi possible de programmer ce dernier de façon exacte conforme à ce qui est attendu sur la liaison série en cours de test.

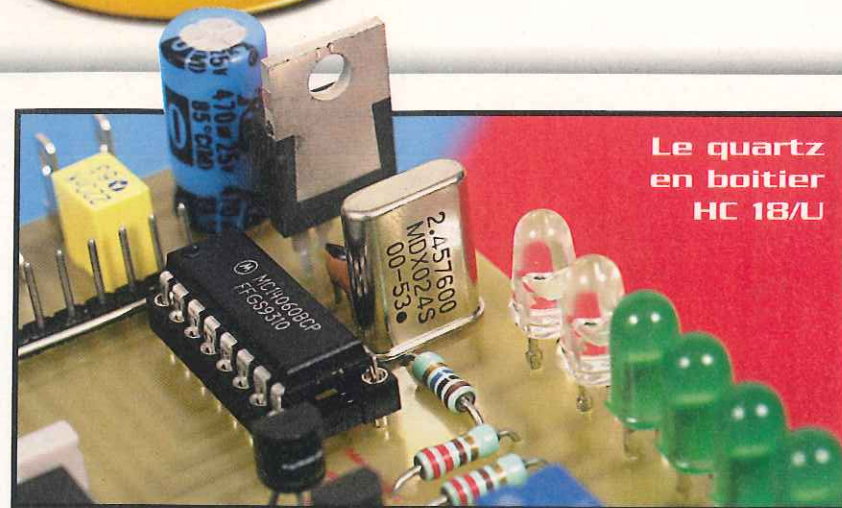
Même s'il est très difficile de "planter" une UART, nous avons prévu la possibilité d'utiliser un poussoir de reset, repéré P₁, pour les cas les plus difficiles. L'appui sur ce dernier permet de ré-initialiser la logique interne de l'UART après une réception totalement incohérente dont il n'arriverait pas à se défaire.

Les données reçues par l'UART sont disponibles sur RD0 à RD7 mais, comme le courant pouvant être fourni sur ces sorties est très faible, un classique ULN2803 ou ULN2804, repéré IC₃, se charge de fournir un courant suffisant aux 8 LED d'indication de niveaux que sont LED₀ à LED₇. Ce sont donc ces dernières qui visualisent le code des différents caractères reçus. La sortie PE de l'UART passe au niveau haut en cas d'erreur de parité et celle repérée FE fait de même en cas d'erreur de format. Chacune commande une LED rouge via les transistors amplificateurs T₁ et T₂. La vitesse de réception de l'UART est déterminée par l'horloge externe qui lui est appliquée à sa patte RCP. Cette horloge est évidemment pilotée par quartz afin d'assurer sa stabilité et se trouve entièrement réalisée avec un banal 4060 en technologie CMOS, repéré IC₄. En effet, compte tenu des fréquences nécessaires et de la progression par multiples de 2 des vitesses de transmission, ses nombreux diviseurs internes permettent de disposer de toutes les valeurs nécessaires pour couvrir de 300 à 9600 bauds.

L'alimentation est, quant à elle, très classique puisqu'un vulgaire régulateur "trois pattes" se charge de délivrer du 5V à l'ensemble du montage. Il est protégé contre les inversions de polarité par la diode D₁.

Réalisation

Tous les composants étant très classiques, vous n'aurez aucune difficulté d'approvisionnement. Le circuit imprimé que nous avons dessiné pour les recevoir ne vous posera pas non plus de problème car son tracé, visible **figure 2**, est très simple et ne comporte pas de piste fine. Ce circuit reçoit l'ensemble des composants du montage à l'exception de S₁ à S₅, de S₇ et de P₁ qui seront mon-



tés sur la face avant du boîtier que vous utiliserez pour loger notre moniteur. L'implantation des composants est à faire en respectant les indications de la **figure 3** et en travaillant dans l'ordre classique : supports de circuits intégrés, résistances, condensateurs et enfin semi-conducteurs. Veillez à bien respecter le sens des composants polarisés et plus particulièrement celui des chimiques placés autour du MAX232.

Les LED pourront être soudées sur le cir-

cuit imprimé, côté cuivre ou côté composants, en laissant leurs pattes longues par exemple, de façon à ce qu'elles puissent, elles aussi, affleurer la face avant du boîtier afin d'être visibles de l'extérieur. Le strap S₆ pourra être implanté sur le circuit imprimé pour le cas où le changement TTL - RS232 serait très peu fréquent ou être ramené sur un interrupteur externe dans le cas contraire.

Premiers essais

Il est très facile d'essayer le montage en utilisant le port série COM1 ou COM2 d'un PC et un programme de terminal tel que Hyperterminal de Windows par exemple. Raccordez notre moniteur à un bloc secteur "prise de courant" réglé sur 9V ou à une simple pile de 9V. Mettez en place le strap S₆, de façon à fonctionner en mode RS232, et reliez ensuite la prise DB9 de notre montage au port série du PC au moyen d'un câble droit. Sélectionnez la vitesse de transmission de votre choix au moyen de S₇, conformément aux indications du **tableau 2**, et le format de transmission au moyen de S₁ à S₅ conformément aux indications du **tableau 1**. Paramétrez de la même façon le programme terminal utilisé sur le PC et vérifiez que chaque caractère frappé au clavier voit son code ASCII affiché sur LED₀ à LED₇ alors que LED₈ et LED₉ restent éteintes. Vérifiez ensuite qu'en modifiant volontairement les paramètres de la liaison sur le moniteur ou sur le PC la détection d'erreur a bien lieu avec, toutefois, les réserves mentionnées ci-dessous.

Quelques conseils d'utilisation

Tout d'abord, nous vous conseillons de compléter notre moniteur par trois acces-

S ₅	Parité	
Ouvert	Pas de parité	
Fermé	Parité selon position de S ₁	
S ₄	Nombre de bits de stop	
Ouvert	2 bits de stop	
Fermé	1 bit de stop	
S ₃	S ₂	Nombre de bits par caractère
Ouvert	Ouvert	8 bits
Ouvert	Fermé	7 bits
Fermé	Ouvert	6 bits
Fermé	Fermé	5 bits
S ₁	Type de parité	
Ouvert	Parité paire	
Fermé	Parité impaire	

T1 Configuration du mode de fonctionnement de l'UART en fonction de la position de S₁ à S₅

C relié à :	Vitesse
V1	300 bauds
V2	600 bauds
V3	1200 bauds
V4	2400 bauds
V5	4800 bauds
V6	9600 bauds

T2 Sélection de la vitesse de transmission au moyen de S₇

Un "espion" de cartes SIM



Un module d'identification

Comme son nom l'indique (Subscriber Identification Module), la carte SIM présente dans tous les téléphones portables sert avant toute chose à identifier et authentifier l'utilisateur de celui-ci.

Abstraction faite des pratiques fort discutables de "Simlockage" destinées à forcer la fidélité envers les opérateurs "subventionnant" les téléphones, n'importe quel mobile peut, en principe, fonctionner avec n'importe quelle carte SIM. Moyennant donc l'insertion de la carte SIM du client de n'importe quel opérateur, "tous les téléphones portables deviennent un peu le sien", en somme.

La capacité de mémoire EEPROM des cartes SIM ne cessant d'augmenter (32 Koctets et plus), il est clair que celles-ci contiennent nécessairement bien autre chose qu'un simple numéro de client.

À côté de toutes sortes de données

personnelles (répertoires de numéros, mini-messages, préférences diverses, etc.), une carte SIM renferme surtout un puissant processeur informatique, souvent plus performant que les "ordinateurs familiaux" du bon vieux temps.

Le téléphone portable, de son côté, est doté de ressources encore plus considérables, nécessitant bien souvent la coopération d'un microprocesseur et d'un DSP (Digital Signal Processor).

Le dialogue entre carte SIM et téléphone portable est donc particulièrement intéressant à analyser, puisqu'il peut révéler beaucoup de "petits secrets", tant de la carte que du mobile, voire du réseau dont celui-ci est l'hôte.

Un protocole très simple

La norme GSM 11.11, véritable "bible" de la carte SIM, précise que la communication se fait selon le protocole "T=0", le plus courant (et aussi le plus simple) en matière de

cartes à puce asynchrones. Dans la plupart des cas, le flot bidirectionnel de données circule à 9600 bauds sur le contact "ISO7" de la carte. Moyennant son interception par un circuit électronique branché en parallèle, celui-ci peut facilement être décodé par l'UART d'un port série de PC, quitte à en "trafiquer" astucieusement le paramétrage.

Le montage de la **figure 1** utilise, pour ce faire, un très courant MAX232 chargé de corriger les incompatibilités de tension entre carte à puce et ligne RS232.

Peu gourmand en énergie, le circuit est directement alimenté par le téléphone portable, laissant même encore de quoi allumer une LED "haute luminosité", bien utile pour visualiser l'activation et la désactivation de la carte.

En plus des données émises et reçues par la carte et le téléphone, le MAX232 prélève aussi le signal de "reset" de la carte utilisé par le logiciel pour se synchroniser à coup sûr.

Que peuvent donc bien se dire les téléphones portables et les cartes à puce "SIM" qu'ils contiennent ? La puissance plus que significative des systèmes informatiques embarqués de part et d'autre laisse deviner l'existence d'un dialogue aussi riche que complexe, qui éveille forcément l'intérêt...

Aussi surprenant que cela puisse paraître, il suffit de fort peu d'électronique et de logiciels très simples pour satisfaire entièrement cette légitime curiosité.

Passons à la pratique !

La réalisation pratique de cette partie matérielle suppose tout d'abord la gravure d'un circuit imprimé conforme au tracé de la **figure 2** et son câblage selon le plan de la **figure 3**.

Il s'agira ensuite de raccorder "proprement" ce module d'interface au téléphone portable et à sa carte SIM, par quatre fils correspondant aux contacts ISO1 (Vcc), ISO2 (Reset), ISO5 (masse) et ISO7 (données), le signal d'horloge (ISO3) n'étant volontairement pas exploité.

En présence d'un "vieux" téléphone portable réservé à l'expérimentation, il ne serait pas déraisonnable de souder tout simplement ces fils en parallèle sur les pattes appropriées du connecteur de carte SIM.

Il est toutefois infiniment plus élégant de passer par l'intermédiaire d'adaptateurs de connectique permettant de varier davantage les plaisirs (manipulations sur différents modèles de téléphones ou de lecteurs autonomes de cartes SIM). Cela peut se faire en équipant le montage d'une embase pour connecteur HE10 à dix contacts (simple tronçon de barrette sécable à double rangée de picots carrés coudés). On fabriquera alors un cordon spécial consistant en un morceau d'environ 20 cm de câble méplat à dix conducteurs, sur lequel on sertira trois fiches HE10 toutes dans le même sens, c'est à dire strictement en parallèle fil à fil. La compatibilité sera ainsi automatiquement assurée avec tous les éléments de la "boîte à outils" décrite dans notre ouvrage "Cartes à puce initiation et applications", ainsi qu'avec les connecteurs de carte à puce décrits dans nos précédents articles. Notons à ce sujet que les cartes SIM, découpées au format "micro", peuvent s'insérer dans des adaptateurs du commerce les ramenant au plein format "ISO" ou, tout simplement, être provisoirement recollées avec un simple morceau de ruban adhésif (si on a eu la bonne idée de conserver le reste de la carte d'origine).

Le problème étant ainsi réglé côté carte, il reste à assurer le raccordement du téléphone au moyen d'une "fausse carte" gravée sur de l'époxy simple face de 8/10 mm. Bien entendu, la forme exacte de cet adaptateur variera d'un modèle de téléphone à l'autre. L'exemple de la **figure 4** est destiné à un certain nombre de modèles de chez SAGEM, à commencer par le RC712.

L'insertion de cette "fausse carte" nécessite la dépose/repose de la coquille arrière du boîtier, fixée par seulement quatre vis TORX n°6 compatibles notamment avec le jeu de mini-tournevis VTSET3 de VELLEMAN. S'ils sont suffisamment fins et souples, les fils de liaison peuvent facilement être pincés entre la batterie faisant office de couvercle et le boîtier ré-assemblé, ce qui permet le fonctionnement normal du téléphone dont la carte SIM est maintenant externe.

Le raccordement des fils (10 à 20 cm au maximum) pourra se faire au moyen d'un petit adaptateur réalisé à partir du circuit imprimé de la **figure 5** muni du même réceptacle HE10 que les autres maillons de la chaîne.

Mise en œuvre logicielle

Une fois tous ces raccordements effectués (et soigneusement vérifiés !), le téléphone doit fonctionner de façon parfaitement normale avec sa carte SIM habituelle ou avec une carte de test (voire avec n'importe quelle SIM si on a pu le "déverrouiller"). Si tel est bien le cas, le moment est venu de connecter l'interface au port série COM1 : de n'importe quel compatible PC.

Les deux logiciels fournis (SIMINV.EXE et SIMDIR.EXE) sont des applications MS-DOS, ce qui signifie qu'ils peuvent fonctionner même sur des PC résolument obsolètes (si possible 386SX25 ou supérieur). En présence d'un ordinateur plus récent, on quittera complètement Windows avant de lancer l'un ou l'autre de ces programmes.

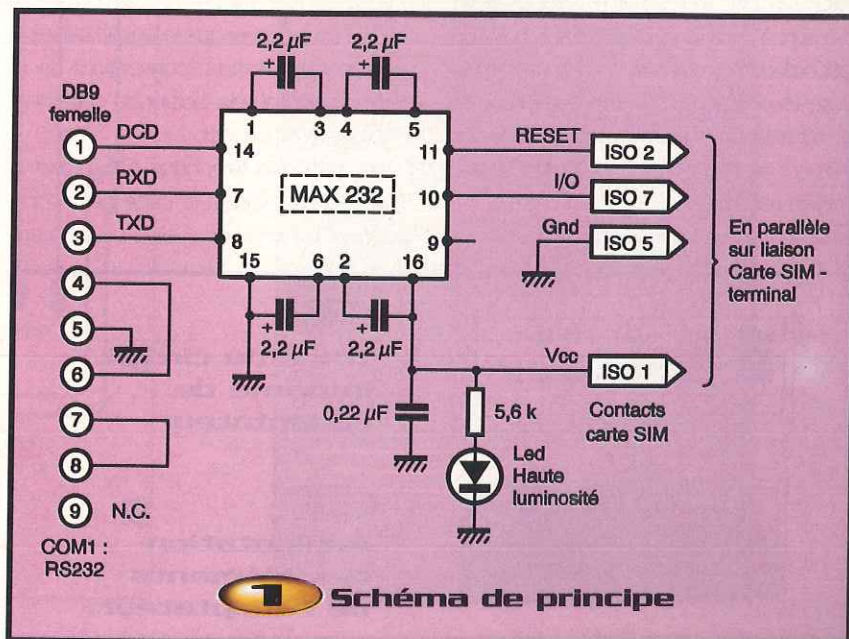
L'un de ceux-ci (SIMINV.EXE) est destiné aux seules cartes SIM à "convention inverse" (réponse au reset commençant par 3Fh), le second (SIMDIR.EXE) étant réservé aux cartes à "convention directe" (réponse au reset commençant par 3Bh). L'utilisation d'un programme inadapté à la carte ne risque nullement de détériorer quoi que ce soit, mais se traduira par l'affichage de données incorrectes puis l'arrêt plus ou moins rapide sur un code d'erreur.

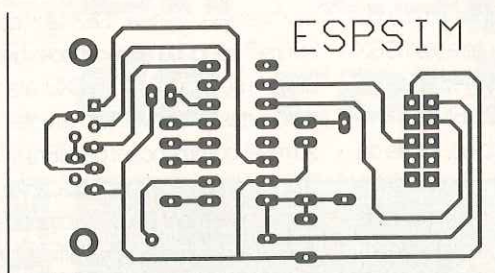
Notons que l'appui sur la touche ESCape permet de toute façon de quitter "proprement" le programme.

Dès leur lancement, l'un comme l'autre de ces deux logiciels attend deux précisions importantes.

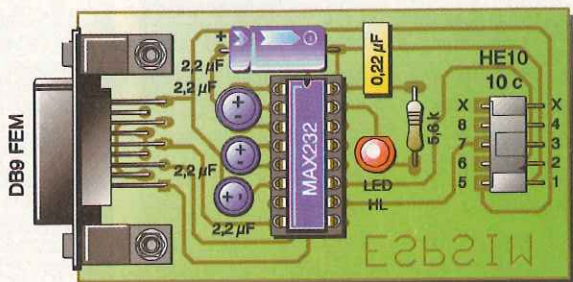
- D'abord, ce qu'il faut faire des données capturées. Par défaut (ou si on tape "CON"), elles seront affichées sur l'écran au fur et à mesure de leur interception, mais on peut aussi les imprimer (taper PRN) ou mieux les enregistrer dans un fichier ".LOG" (taper seulement son nom, sans aucune extension).

- Ensuite, s'il faut apporter une correction tenant compte d'une fréquence d'horloge différente des 3,58 MHz habituels, lesquels mènent à une vitesse de transmission "standard" de 9600 bauds (vérifier éventuellement avec un fréquencemètre numérique branché entre ISO3 et masse). Citons ainsi le cas d'un RC712 appliquant à la carte une fréquence d'horloge de 3,25 MHz et nécessitant donc l'utilisation d'un facteur d'ajustement de 13 au lieu de 12, valeur par défaut.





2
Tracé du circuit imprimé de l'interface



3
Implantation des éléments de l'interface

On peut légitimement se demander si cette entorse aux usages est destinée à "tromper l'ennemi" ou bien résulte tout simplement d'une surexploitation des ressources du processeur embarqué, comme d'autres indices peuvent le laisser supposer...

Si tout fonctionne normalement, les données capturées se présentent sous la forme d'un texte constitué de valeurs hexadécimales délimitées par des espaces. Toute pause bien nette dans le flot de données est matérialisée par un retour à la ligne, ce qui signale assez souvent (mais pas toujours) une inversion de la direction de communication (mobile vers SIM ou SIM vers mobile).

Lorsque la communication se fait à un rythme soutenu et que le PC utilisé est un peu lent à l'affichage, il peut se produire un débordement (overflow) du tampon du port série (code d'erreur 69). En pareil cas, on se rabattra avantageusement sur l'enregistrement dans un fichier .LOG que l'on examinera ultérieurement.

Notons qu'il est préférable que ce fichier

réside sur un "disque virtuel" à temps d'accès particulièrement court. L'interprétation des communications ainsi enregistrées suppose une bonne connaissance du protocole T=0 et de la spécification GSM 11.11, dont l'essentiel est expliqué dans notre ouvrage "Téléphones portables et PC" (collection ETSF). Retenons pour le moment que, si on lance le programme puis que l'on met le téléphone portable sous tension (opérer impérativement dans cet ordre), il doit d'abord s'afficher (une ou plusieurs fois) la "réponse au reset" de la carte SIM. Au cas où celle-ci ne commencerait pas par 3Fh ou 3Bh, il faudrait soupçonner que le logiciel utilisé ne correspond pas à la "convention" de la carte ou bien que le facteur de correction d'horloge est incorrect.

C'est ensuite le téléphone qui "prend la main" en envoyant à la carte des "commandes" normalisées. Celles-ci commen-

cent nécessairement par un "en-tête" de 5 octets dont le premier (A0h) est la "classe ISO" propre aux cartes SIM. L'octet qui suit est le code opératoire de la commande et, à ce stade, sera fréquemment A4h. Cela correspond à l'instruction "SELECT" utilisée pour accéder à un répertoire ou à un fichier de la carte.

L'adresse sélectionnée occupe les 2 octets qui sont émis juste après la répétition du code opératoire A4h par la carte, en guise d'accusé de réception de la commande (octet dit "de procédure").

Bien souvent, le téléphone commencera par sélectionner le répertoire 3F00h (le répertoire racine de la SIM). S'il considère, à juste titre, que celui-ci est sélectionné par défaut après un reset, il sélectionnera plutôt directement le sous-répertoire 7F20h (dit

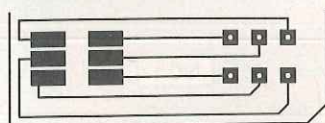


"GSM") ou 7F21h (dit "DCS") en attendant l'occasion d'aller dans le répertoire 7F10h (dit "Télécom").

Après les 2 octets de compte-rendu (9FXXh) que répond la SIM à toute commande SELECT valide, on trouve presque toujours une commande GET RESPONSE (COh) émise par le téléphone. Celle-ci réclame l'émission par la SIM d'un nombre d'octets au plus égal à la valeur XXh du compte-rendu précédent. L'interprétation de cette réponse renseigne en détail sur les caractéristiques du répertoire ou du fichier sélectionné : la taille de son contenu, son organisation logique, mais aussi les droits d'accès qui y sont attachés.

Dans certains cas, une commande STATUS (F2h) est utilisée dans le même but, ou bien à titre de confirmation. C'est notamment à ce stade que le téléphone saura si le code confidentiel (baptisé PIN ou CHV) de la carte est activé ou non.

Si la présentation du code est requise, le dialogue se trouvera suspendu pendant que le téléphone demande, sur son écran, que le PIN soit composé. C'est alors par l'entremise d'une commande VERIFY CHV



4
Tracé du circuit imprimé de l'adaptateur



5
Implantation des éléments de l'adaptateur

(code opératoire 20h) que le code tapé sur le clavier sera présenté à la carte, laquelle répondra 9000h si, et seulement si, le code est reconnu correct.

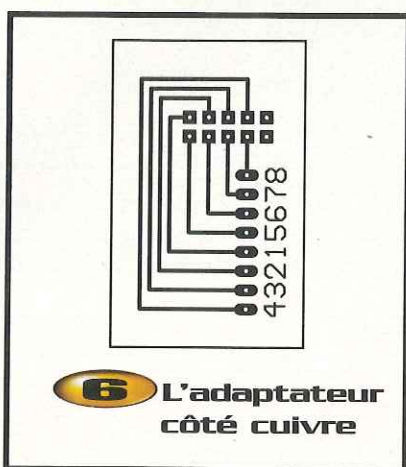
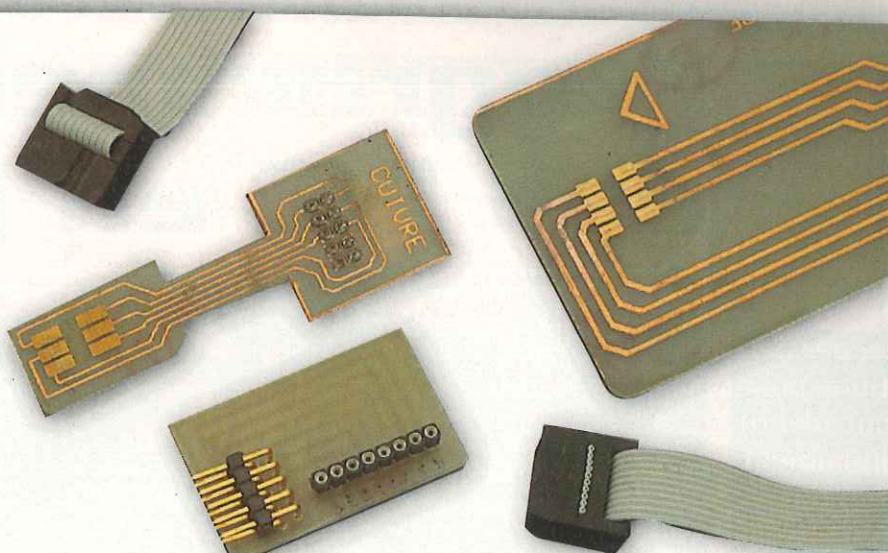
Voici, par exemple, les résultats obtenus en branchant ainsi notre montage entre un RC712 et une carte prépayée (périmée) dont le code PIN est actif

```

3B 82 00 55 19
A0 A4 00 00 02
A4 7F 20
9F 17
A0 A4 00 00 02
A4 7F 21
9F 17
A0 A4 00 00 02
A4 7F 20
9F 17
A0 C0 00 00 0E
C0 00 00 00 24 7F 20 02 00 00 44 44 01
09 13 90 00
A0 F2 00 00 14
F2 00 00 00 24 7F 20 02 00 00 44 44 01
09 13 00 14 04 00 83 8A 90 00
    
```

Il est bien évident que l'analyse de la suite du dialogue, qui peut se composer de milliers d'octets, sort largement du cadre de cet article simplement destiné à fournir à nos lecteurs les moyens de partir eux-mêmes à l'aventure. Qu'ils ne doutent pas un seul instant que celle-ci sera passionnante!

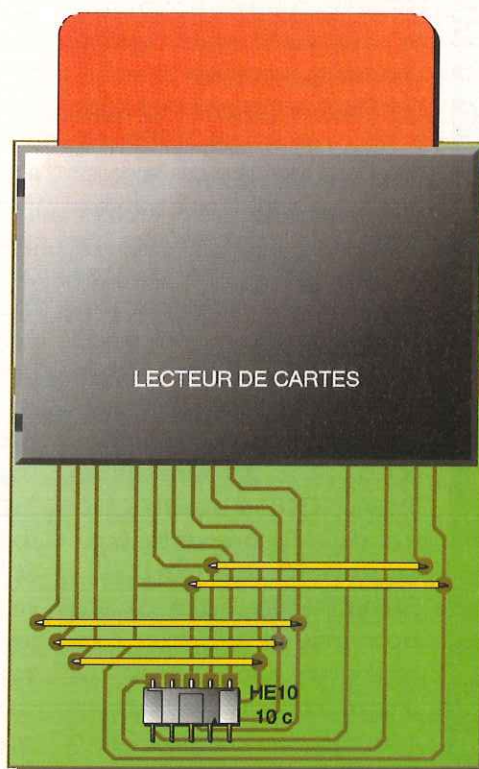
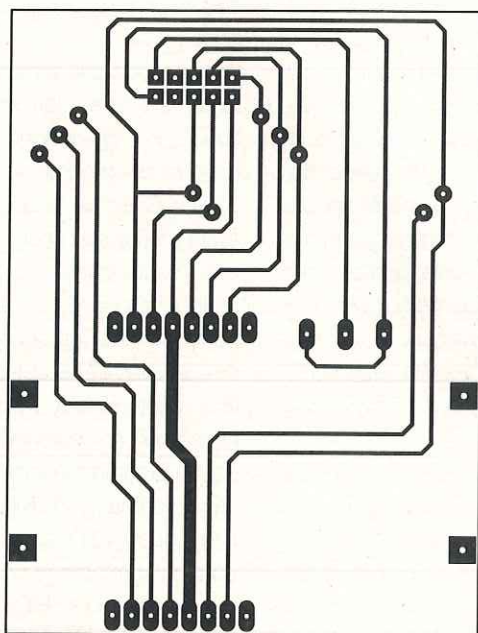
P. GUEULLE



Nomenclature

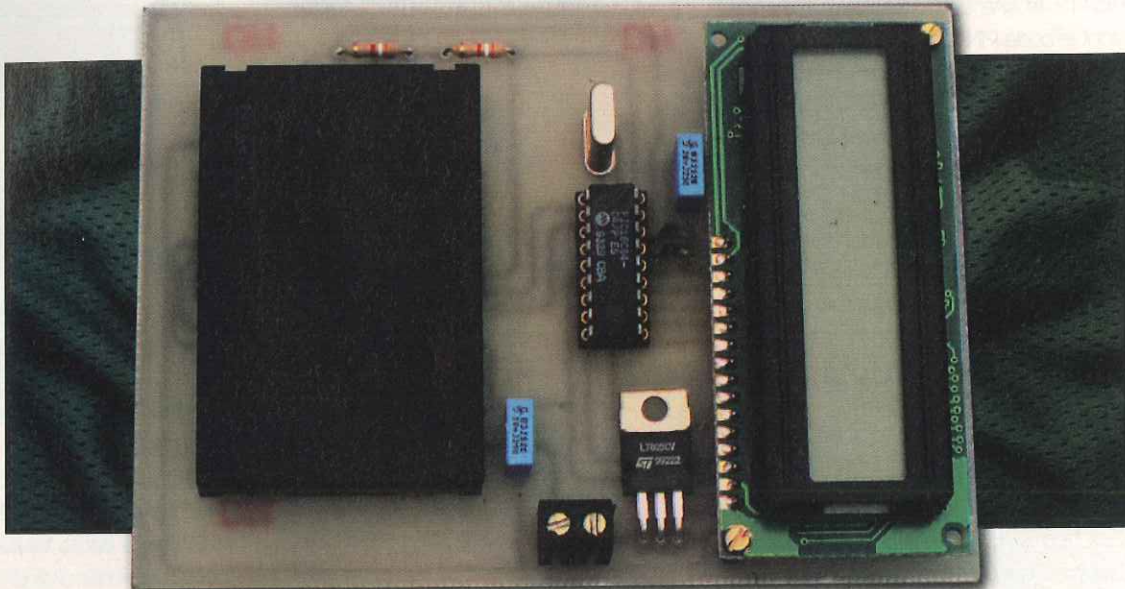
- 1 MAX 232
- 1 résistance 5,6 k Ω (vert, bleu, rouge)
- 1 LED rouge haute luminosité
- 3 condensateurs 2,2 μ F/16V radial
- 1 condensateur 2,2 μ F/16V axial
- 1 condensateur 0,22 μ F/63V
- 1 embase DB9 femelle pour circuit imprimé
- 1 embase HE10 10 contacts ou barrette sécable

7 8 Tracé du circuit imprimé et implantation des éléments du lecteur



Programmer des PIC en Basic

Développer des applications pour microcontrôleurs est infiniment plus simple en BASIC qu'en assembleur, le succès de produits comme le Basic Stamp en apportant d'ailleurs la preuve éclatante. Par rapport à l'implantation d'un interpréteur BASIC dans le PIC lui-même, le recours à un authentique compilateur, produisant du pur langage machine, se traduit pourtant par de bien meilleures performances. Cela sans même parler des bibliothèques de "périphériques virtuels", parfois mises à la disposition du programmeur. Nous allons montrer ici comment même une version limitée, gratuite permet de développer une application "carte à puce" déjà ambitieuse en... vingt lignes de code source.



Interpréteur ou Compilateur ?

Quel que soit l'outil de développement utilisé, c'est pratiquement toujours sur un PC que l'on met au point les applications à base de microcontrôleurs PIC, puis que l'on programme les composants achetés vierges. L'écriture de programmes en assembleur demeurant passablement rébarbative, il a été imaginé d'effectuer le développement dans des langages évolués tels que le C ou le bon vieux Basic. Les microcontrôleurs ne sachant exécuter que du code machine, une "couche" logicielle intermédiaire s'impose, qui peut prendre la forme soit d'un interpréteur, soit d'un compilateur. Le principe de l'interpréteur est bien connu, puisqu'il fut à l'origine du succès des ordinateurs dits "familiaux" (ZX 81 et consorts !), mais aussi du GWBasic encore très populaire sur les PC utilisés sous DOS. En Basic interprété, le code source du programme est simplement compacté en une suite de "tokens", puis directement enregistré dans la mémoire du microcontrôleur. Celle-ci héberge également un logiciel "interpréteur" chargé de convertir les ins-

tructions Basic en code machine au fur et à mesure de leur exécution. Les composants de la famille "Basic Stamp", par exemple, sont vendus avec un interpréteur déjà "brûlé" dans leur mémoire "OTP", le Basic "tokenisé" venant pour sa part se télécharger dans une mémoire E2PROM. Très confortable au stade de la mise au point des programmes, cette approche a pour inconvénients une certaine lenteur de l'exécution et, aussi, un coût élevé dès que la maquette doit être reproduite à plusieurs exemplaires (pour chacun desquels il faut, en effet, payer le prix de l'interpréteur embarqué). Une solution élégante consiste à utiliser un outil logiciel permettant de programmer des PIC parfaitement ordinaires à partir du code source mis au point sur le Basic Stamp (voir notre ouvrage "BASIC pour microcontrôleurs et PC" dans la collection ETSF chez DUNOD). Improprement baptisés "compilateurs", ces outils de développement implantent, en réalité, un interpréteur dans le PIC avant d'y charger le programme Basic tokenisé. L'avantage est évident en matière de coût des composants, mais aussi de rentabi-

lisation de l'espace mémoire disponible, car l'interpréteur peut être taillé sur mesures pour chaque application. Entendons, par-là, qu'il peut être débarrassé du code correspondant aux instructions Basic que l'application ne met pas à contribution, libérant ainsi de la place pour davantage de Basic tokenisé. Il ne faut toutefois espérer aucun gain de rapidité palpable, puisqu'il s'agit encore et toujours de Basic interprété. Les choses sont entièrement différentes si l'on se sert d'un authentique compilateur, logiciel capable de traduire intégralement le code source Basic en langage machine avant même son chargement dans le microcontrôleur. Pour peu que cette traduction soit élégante, les performances obtenues peuvent rivaliser avec celles obtenues par un bon développeur, voire excéder sensiblement celles de code écrit par un programmeur malhabile. Les compilateurs de Micro Engineering Labs (MEL) sont considérés comme la référence en matière de microcontrôleurs PIC (du moins ceux à 14 bits), mais il faut aujourd'hui compter avec la concurrence

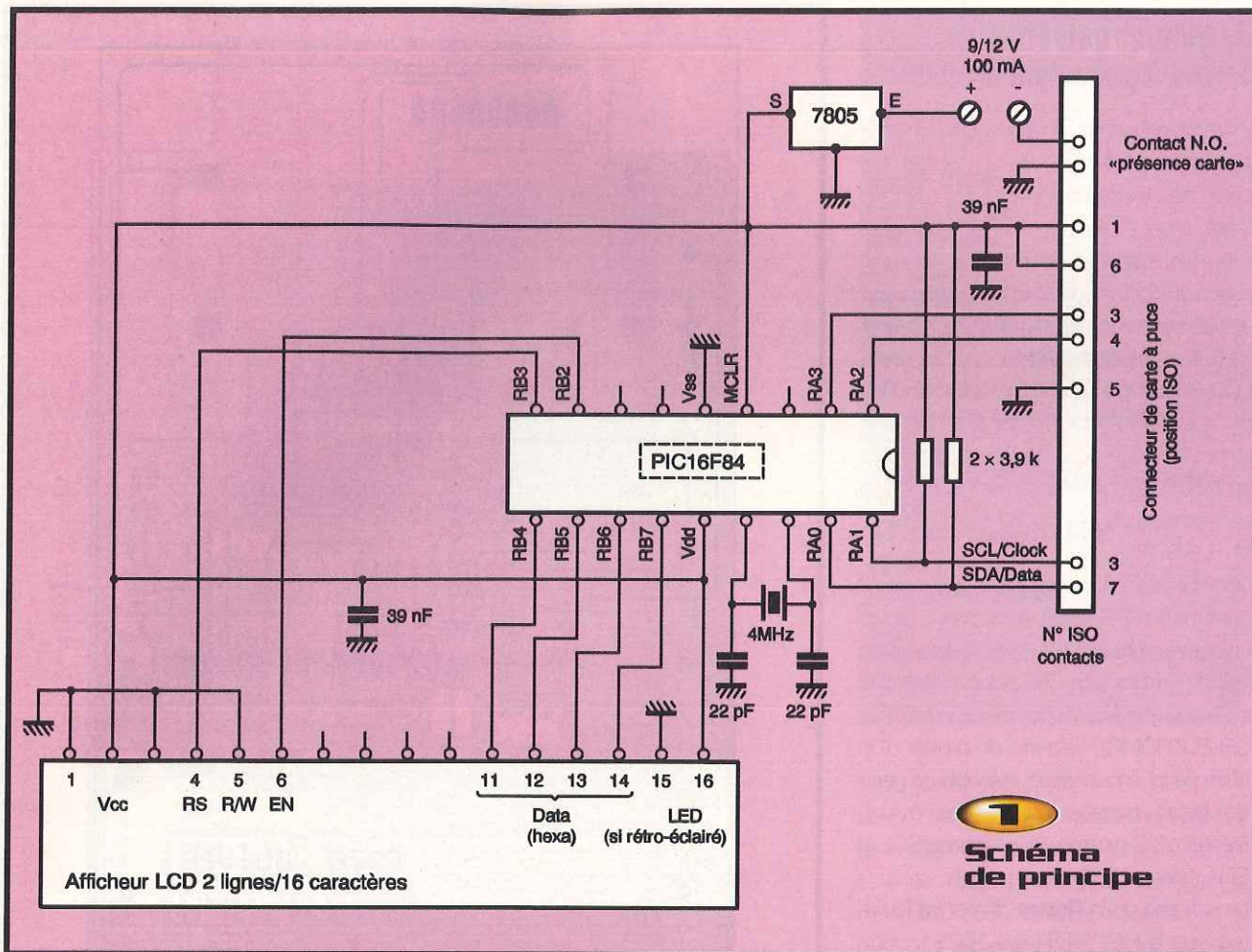


Schéma de principe

d'un logiciel britannique dont les ambitions méritent d'être prises au sérieux.

Le compilateur LET Basic

Ce tout nouveau compilateur offre ainsi la possibilité rare de produire, entièrement sous environnement Windows, du code pour les très populaires PIC 12C508 et PIC 12C509 (à noyau 12 bits, rappelons-le). Fonctionnant en tandem avec l'assembleur "officiel" MPASM de MICROCHIP, il permet d'examiner le code assembleur intermédiaire et, par conséquent, d'en apprécier objectivement la qualité.

C'est la firme de la région de Cambridge "Crownhill Associates" qui se charge de la diffusion de ce compilateur, selon une formule pour le moins originale : en attendant la version commerciale définitive, des versions "Lite" intermédiaires sont offertes gratuitement en téléchargement, assorties toutefois de limitations d'autant plus importantes que l'on s'approche de la version finale. Le code source, par exemple, ne peut excéder une trentaine de lignes (mais pourquoi ne pas placer plusieurs ins-

tructions par ligne ?). Le support technique est assuré uniquement sous la forme d'un forum sur Internet, dont l'auteur du compilateur ne se prive pas de tirer parti pour parfaire progressivement son œuvre.

Deux versions assez représentatives de cette évolution sont reproduites sur le CD-ROM joint à ce numéro :

- La version 7.00 (setup700.exe) supporte de nombreuses références de PIC (dont les 12C50x), mais souffre encore de diverses imperfections parfois délicates à contourner.

- La version 7.10 (setup710.exe), très sensiblement améliorée, ne supporte plus, en revanche, que le PIC 16F84. Les versions ultérieures pourront être téléchargées sur le site <http://www.letbasic.com> sous la forme d'un unique fichier nommé setup.exe. L'exécution de ce fichier entraîne l'installation du logiciel et d'une version récente de MPASM pour Windows, le tout étant accompagné d'un manuel très complet en format PDF. Celui-ci est également reproduit (dans sa version 7) sur notre CD-ROM et ses versions successives seront téléchargeables séparément sur

le même site (ou sur son "miroir" <http://www.cambsnet.net/letbasic>).

On s'apercevra immédiatement que cet outil se démarque clairement du PBasic (Parallax Basic) hérité du Basic Stamp, pour revenir à un dialecte beaucoup plus généraliste de ce langage à tout faire. L'effort d'adaptation sera ainsi nettement moindre pour les habitués du GWBasic ou du QBasic (voire de Visual Basic !), bien que de multiples instructions nouvelles fassent leur apparition. Bon nombre d'entre elles concernent les "périphériques virtuels" fournis sous la forme de bibliothèques à inclure et initialiser si, et seulement si, on en a besoin :

- gestion d'afficheurs LCD,
- support du bus I2C,
- support d'une ligne série 9600 bauds,
- gestion d'un clavier à 16 touches,
- conversion analogique/numérique avec le PIC 16C71.

Précisons enfin qu'il est possible de "mélanger" hardiment Basic et assembleur dans le code source, ce qui démultiplie encore les possibilités offertes au programmeur maîtrisant les deux langages.

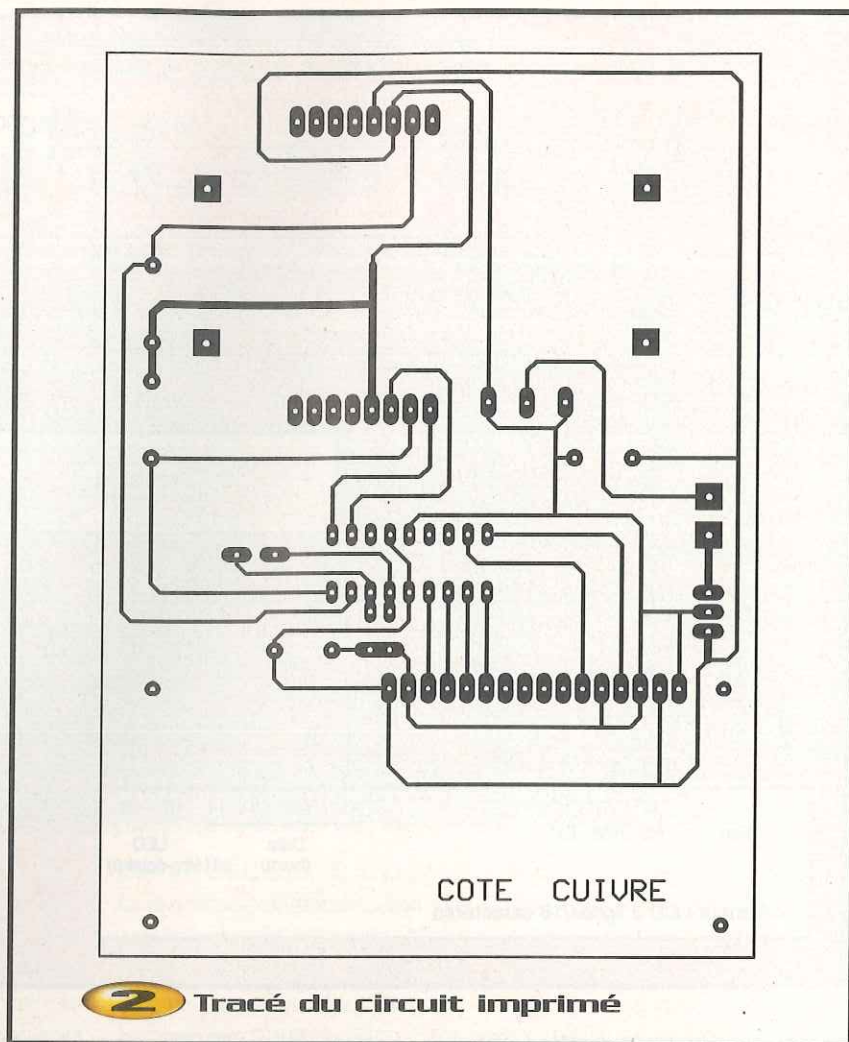
Un lecteur universel de cartes à puce synchrones

L'exemple que nous avons choisi, pour illustrer les possibilités du LET Basic, fait partie de ces très populaires applications "carte à puce". Il s'agit d'un lecteur matériellement capable de prendre connaissance du contenu de n'importe quelle carte à puce synchrone (famille des télécartes) et de l'afficher séquentiellement sur un écran LCD à deux lignes de 16 caractères. Cela, on l'aura deviné, pour mettre à contribution la fonctionnalité de commande d'afficheur du LET Basic, très facilement accessible par les habituelles instructions CLS, PRINT, et CURSOR.

Dans la version logicielle présentée ici (programme I2CPIC.BAS), il est prévu de lire uniquement le contenu de cartes à puce I2C de 2 Kbits (soit 256 octets) telles que la D2000 PHILIPS (disponible au détail chez SELECTRONIC), cela afin de profiter d'un autre périphérique virtuel, capable de gérer de façon transparente l'accès à une mémoire I2C externe par ses lignes SCL et SDA (horloge et données).

Le schéma de la **figure 1** montre toutefois que le PIC 16F84 (ou 16C84) communique avec les autres contacts du connecteur de carte, d'où la possibilité de reprogrammer le montage pour lire à peu près n'importe quelle sorte de cartes synchrones. Cela au prix de l'écriture d'un logiciel forcément plus long et exigeant donc vraisemblablement le recours à la version complète du compilateur.

Pour l'heure, le code source que voici affiche le résultat de la lecture sous la forme



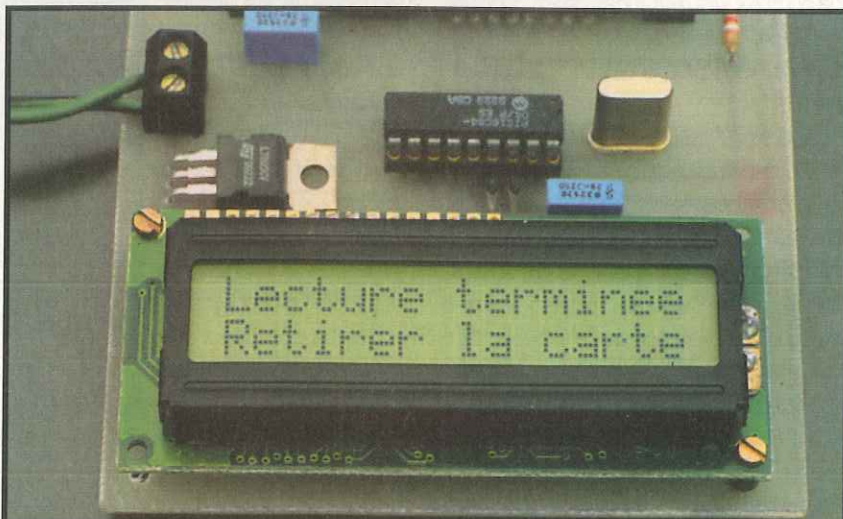
de seize groupes de seize octets, convertis chacun en un caractère déduit du code ASCII correspondant. L'expérience montre, en effet, que bien des cartes I2C contiennent en réalité des codes ASCII : cartes-jeu distribuées par GEMPLUS au salon CARTES, "Password Card" de TOWITOKO utilisée pour mémoriser les

mots de passe d'un utilisateur de PC, etc. Chaque bloc de caractères est affiché pendant une seconde et demie (six fois le maximum de 255 ms autorisés par le compilateur), surmonté d'une ligne rappelant ses adresses de début et de fin en hexadécimal (ce qui fournit l'occasion d'expérimenter avec plusieurs formes possibles de l'instruction PRINT).

Le cas échéant, il serait d'ailleurs extrêmement facile de modifier le programme pour que l'affichage se fasse en décimal ou en hexa.

Réalisation pratique

Toute la complexité du montage étant prise en charge par sa partie logicielle, la réalisation pratique se limite à l'interconnexion de fort peu de composants sur un circuit imprimé gravé d'après la **figure 2**. Le plan de câblage de la **figure 3** est prévu pour un afficheur rétro-éclairé, ce qui suppose l'emploi d'un régulateur 7805 pour subvenir à sa consommation non négligeable (au moins 100mA sous 5V). En présence d'un



Emploi d'un afficheur LCD 2 x 16 caractères

Listing

```

DEVICE 16F84
INCLUDE LCD
INCLUDE I2CBUS
DIM A, B, C, F, G, H
INIT LCD,PortB
INIT I2CBUS
for g=0 to 15
CLS : H = 16*G : C = H+15
PRINT $H," ..... ",SC : CURSOR 1,2
FOR F=0 to 15
B=H+F : A=MEMREAD(B) : PRINT #A
NEXT F
GOSUB tempo : GOSUB tempo : GOSUB tempo
NEXT G
CLS : PRINT "Lecture terminee"
CURSOR 1,2
PRINT "Retirer la carte"
STOP
tempo: delays(255) : delays(255) : return
END

```

l'afficheur et du bus I2C.

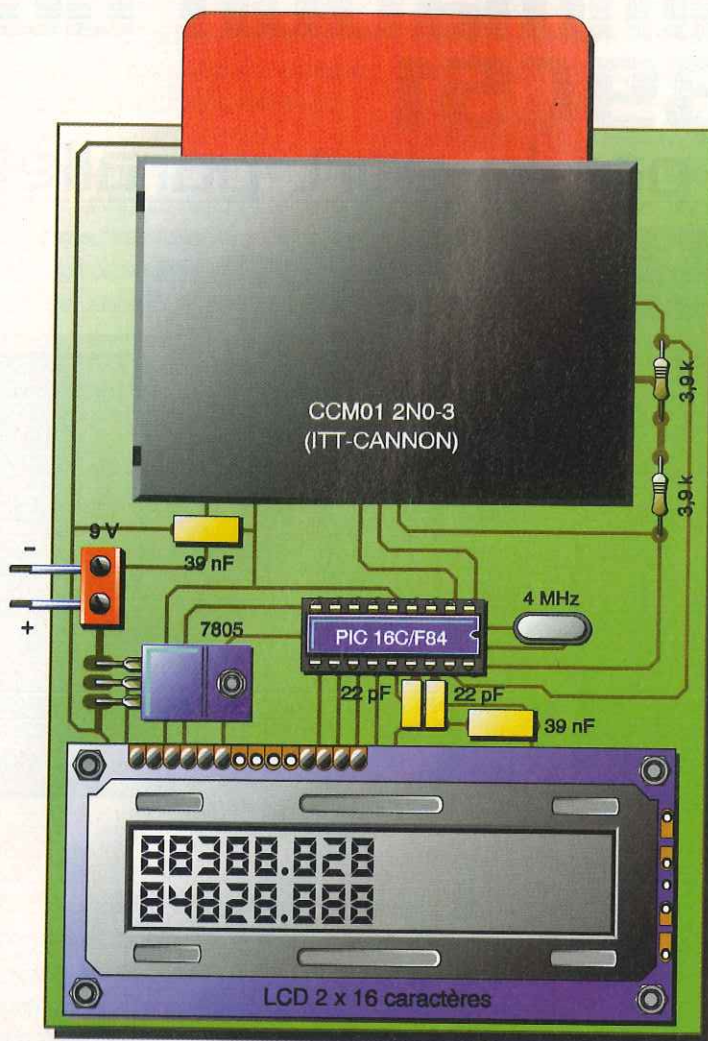
Le PIC sera monté de préférence par l'intermédiaire d'un support à contacts "tulipe" et il ne restera plus qu'à alimenter la maquette sous une tension continue de 9 ou 12V. L'alimentation se faisant à travers le contact "présence carte" du connecteur de carte à puce, rien ne doit se passer tant qu'une carte n'est pas introduite. A ce moment seulement, l'afficheur doit s'éclairer et la lecture peut commencer.

Après les seize étapes de l'affichage (adresses 00h à F0h), un message invitant à retirer la carte s'affichera jusqu'à son extraction effective. Bien entendu, la carte doit contenir des données pour que quelque chose de significatif s'affiche !

Une application pratique de ce montage pourrait être la lecture de données enregistrées par la carte alors qu'elle était introduite dans un autre équipement (par exemple une suite de résultats de mesures ou bien de noms de personnes ayant introduit leur propre carte dans un lecteur). En tant que cartes à lecture et écriture libres, les cartes I2C se prêtent en effet idéalement au transport de petits volumes de données en tous genres.

Bien entendu, le compilateur LET Basic peut servir à développer des programmes gérant ainsi l'écriture dans ces cartes : c'est quasiment aussi simple que pour la lecture !

P. GUEULLE



3 Implantation des éléments

Nomenclature

PIC 16F84 (ou PIC 16C84) à programmer

Régulateur 7805

Quartz 4 MHz (environ)

Afficheur LCD 2x16 caractères

2 résistances 3,9 kΩ

(orange, blanc, rouge)

2 condensateurs 39 nF

2 condensateurs 22 pF

1 connecteur de carte à puce (CCM01

2N0-3 ITT-CANNON)

1 bornier 2 circuits 5,08mm

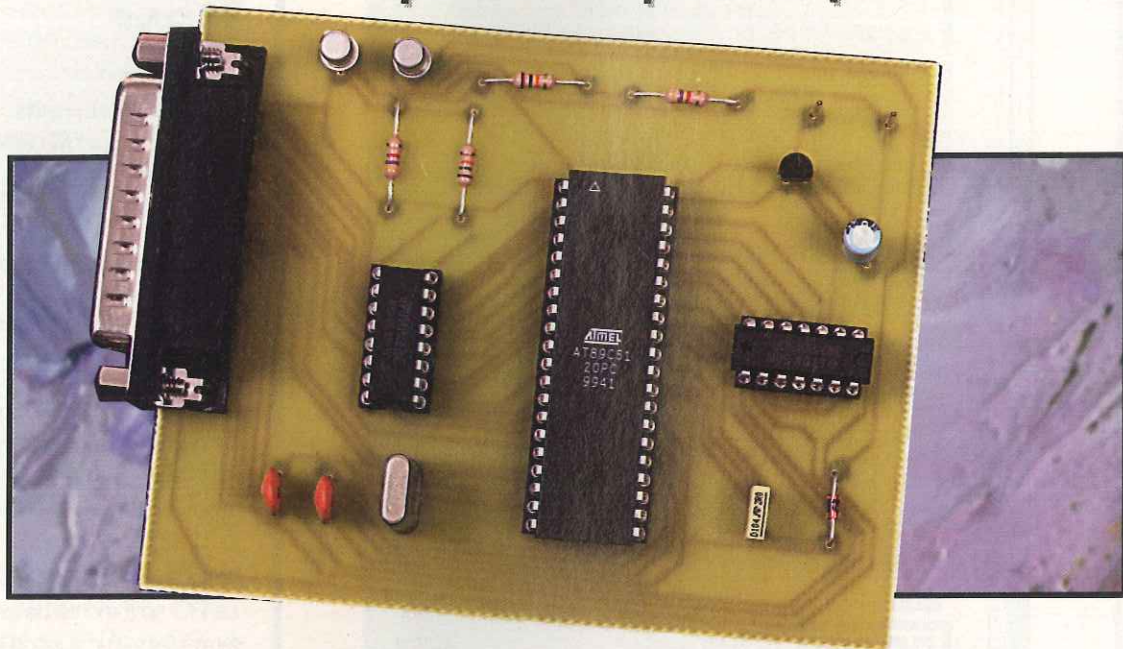
modèle uniquement réflexif, un 78L05 suffirait, tandis que l'afficheur ne serait muni que de 14 connexions au lieu de 16.

On notera que celui-ci fonctionne en mode "hexa", c'est à dire sur des blocs de données de seulement quatre bits. Cela minimise le nombre de broches du PIC utilisées pour la communication, ménageant ainsi la possibilité d'éventuelles extensions du montage à partir des lignes d'entrée/sortie restant disponibles.

Le PIC sera programmé au moyen du fichier I2CPIC.HEX produit par le compilateur et dont notre CD-ROM contient une version "enrichie" des positions des fusibles de configuration (oscillateur en mode XT, WDT OFF, PWRT OFF, et CP OFF). On remarquera au passage que l'opération ne consomme que 391 mots de mémoire (soit un gros tiers des 1024 disponibles), dont l'essentiel correspond d'ailleurs aux bibliothèques de gestion de

Programmation du μ C AT89C51

par le port parallèle



Le montage décrit dans cet article permet de programmer le microcontrôleur AT89C51 de chez ATMEL à partir du port parallèle d'un micro-ordinateur de type PC. Le circuit représenté à la figure 1 utilise les huit broches de données (de 2 à 9) et trois lignes de contrôle de cette interface (1, 14 et 16).

Étude du montage

Les lignes de contrôle ainsi que D0 et D1 agissent respectivement comme signaux de remise à zéro et d'horloge pour le compteur binaire sur 12 bits qui est un 74HC4040. La sortie de ce compteur est connectée aux lignes d'adresses du AT89C51. Les lignes D6 et D7 correspondent aux signaux d'horloge et de données du 74HC164 qui est un registre à décalage avec entrée série et sorties parallèles. La sortie de ce registre à décalage est connectée aux lignes de données du AT89C51. Le reste des lignes du port parallèle est utilisé comme signaux de contrôle pour la programmation de l'AT89C51. C'est un microcontrôleur 8 bits en technologie CMOS à faible consommation et à haute performance : il comporte en interne une mémoire flash de 4 koctets programmable et une mémoire à lecture seule effaçable (EPROM). Ce composant utilise une technologie de mémoire non volatile à haute densité développée par ATMEL et est compatible avec les instructions

ainsi qu'avec le brochage du standard de l'industrie qu'est le MCS-51. La mémoire Flash intégrée sur la puce permet de programmer la mémoire programme sur le système lui-même ou par un programmeur conventionnel non volatile. L'AT89C51, dont le schéma bloc interne est représenté à la **figure 2**, fournit les caractéristiques standards : 4 Koctets de mémoire Flash, 128 octets de mémoire RAM, 32 lignes d'entrées-sorties, 2 chronomètres/compteurs sur 16 bits, une architecture à 2 niveaux et 5 vecteurs d'interruption, 1 port série full-duplex, 1 oscillateur intégré sur la puce et 1 circuit d'horloge. De plus, l'AT89C51 a été conçu avec de la logique statique pour opérer jusqu'à des fréquences qui peuvent descendre à zéro et supporter deux modes de sauvegarde d'énergie sélectionnables par programmation. Le mode inoccupé, appelé idle mode, arrête le microcontrôleur tout en permettant le fonctionnement de la mémoire RAM, du chronomètre/compteur, du port série et du

système d'interruption.

Le mode de réduction de la consommation, encore appelé power-down mode, sauvegarde le contenu de la mémoire RAM mais gèle l'oscillateur, dévalidant toutes les fonctions du composant jusqu'à la prochaine remise à zéro matérielle.

Voici une brève description de ses différentes broches. VCC et GND sont les broches d'alimentation et de masse.

Le Port 0 est un port d'entrées-sorties bidirectionnelles à drain ouvert sur 8 bits. En tant que port de sortie, chaque broche peut alimenter huit entrées TTL. Le Port 0 peut aussi être configuré pour être le bus d'adresses/données multiplexé de faible poids durant l'accès à un programme externe et à la mémoire des données : dans ce mode, le Port 0 possède des résistances de rappel en interne. Lorsque des niveaux logiques hauts sont écrits sur les broches du port 0, ces broches peuvent être utilisées comme entrées à haute impédance. Le Port 0 reçoit aussi les octets de code durant la program-

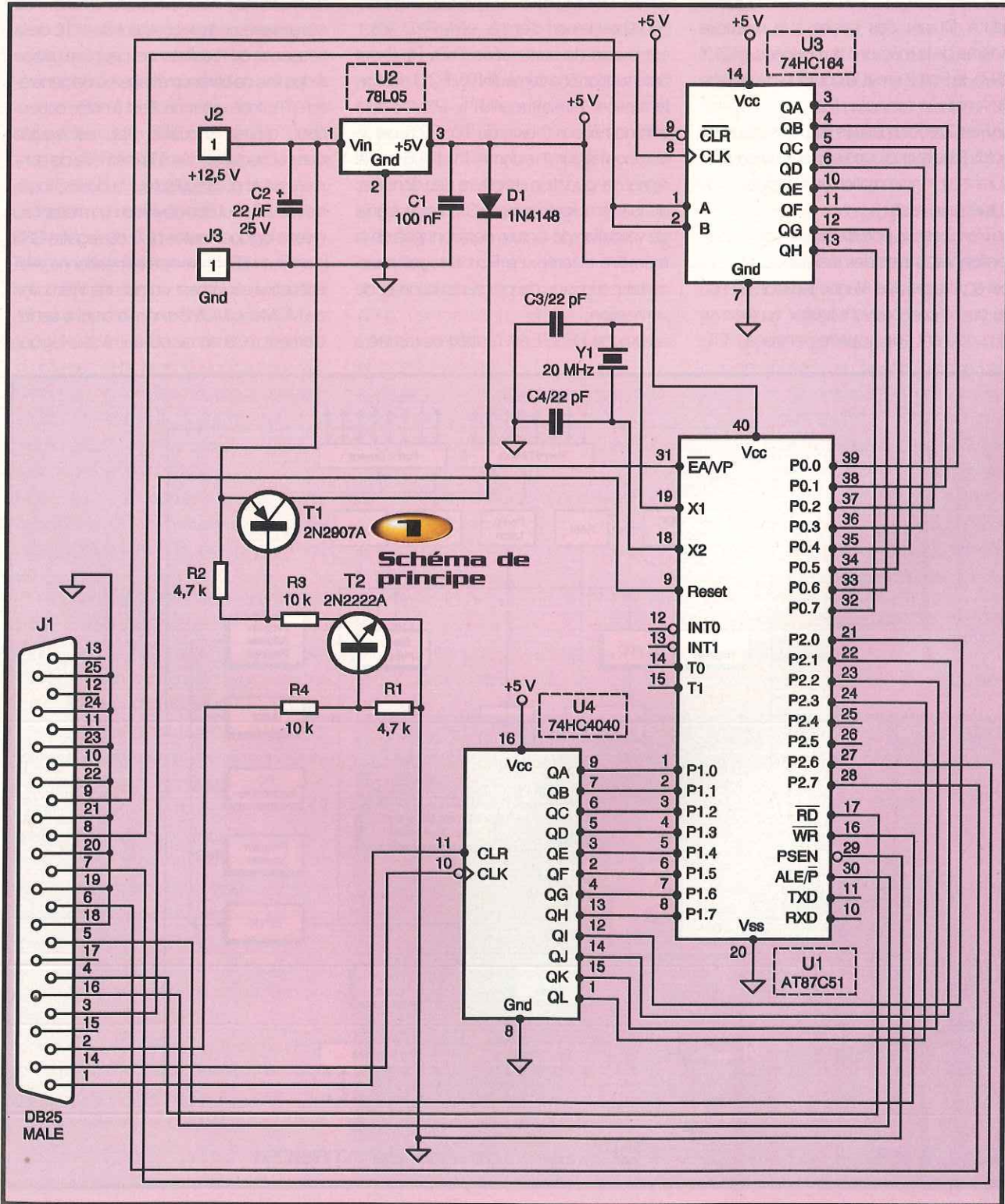
mation de la mémoire Flash et ressort ces octets de code durant la vérification de la programmation ; des résistances de rappel sont requises durant la vérification de la programmation.

Le Port 1 est un port d'entrées-sorties bidirectionnelles avec des résistances de rappel en interne ; les étages tampon de sortie de ce port peuvent fournir ou recevoir du courant de quatre entrées TTL. Lorsque

des niveaux logiques hauts sont écrits sur les broches du Port 1, ces broches sont mises au rappel par des résistances internes et peuvent alors être utilisées comme des entrées. En tant qu'entrées, les broches du Port 1 qui sont extérieurement mises à l'état bas, peuvent fournir du courant grâce à leur résistance interne de rappel.

Le Port 2 est un port d'entrées-sorties bidi-

rectionnelles avec des résistances de rappel en interne ; les étages tampon de sortie de ce port peuvent fournir ou recevoir du courant de quatre entrées TTL. Lorsque des niveaux logiques hauts sont écrits sur les broches du Port 2, ces broches sont mises au rappel par des résistances internes et peuvent alors être utilisées comme des entrées. En tant qu'entrées, les broches du Port 2 qui sont extérieurement



ment mises à l'état bas, peuvent fournir du courant grâce à leur résistance interne de rappel. Le Port 2 émet l'octet d'adresse de poids fort durant des recherches à partir de la mémoire programme externe et durant des accès vers la mémoire de données externe qui utilise un adressage sur 16 bits (MOVX @ DPTR). Dans cette dernière application, il utilise fortement les résistances de rappel internes lorsque ces broches émettent des niveaux logiques hauts. Durant des accès à la mémoire externe de la mémoire de données (MOVX @ RI), le Port 2 émet le contenu du registre de fonctions spéciales P2.

Le Port 2 reçoit aussi les bits d'adresses de poids forts et quelques signaux de contrôle durant la programmation et la vérification de la mémoire flash.

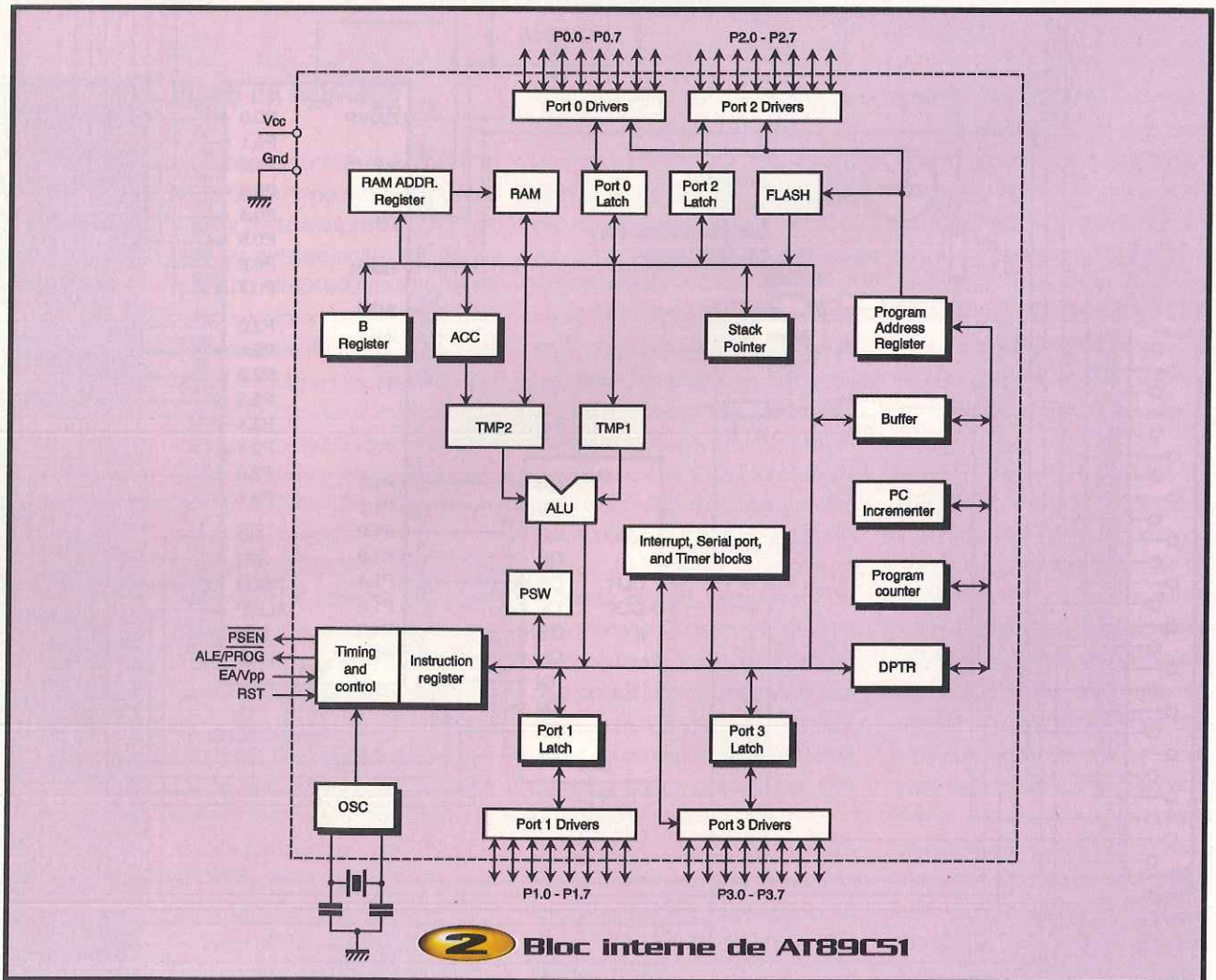
Le Port 3 est un port d'entrées-sorties bidirectionnelles avec des résistances de rappel en interne ; les étages tampon de sortie de ce port peuvent fournir ou recevoir du courant de quatre entrées TTL.

Lorsque des niveaux logiques hauts sont écrits sur les broches du Port 3, ces broches sont mises au rappel par des résistances internes et peuvent alors être utilisées comme des entrées. En tant qu'entrées, les broches du Port 3 qui sont extérieurement mises au rappel à l'état bas, peuvent fournir du courant grâce aux résistances de rappel en interne. Le Port 3 sert aussi à différentes fonctions de caractéristiques particulières du AT89C51 : P3.0 est le port d'entrée série RXD, P3.1 est le port de sortie série TXD, P3.2 est l'interruption 0 externe /INT0, P3.3 est l'interruption 1 externe /INT1, P3.4 est le chronomètre 0 externe T0, P3.5 est le chronomètre 1 externe T1, P3.6 est le signal de validation d'écriture des données de la mémoire externe, P3.7 est le signal de validation de lecture des données de la mémoire externe. Le Port 3 reçoit aussi certains signaux de programmation et de vérification.

La broche RESET est l'entrée de remise à

zéro : un niveau logique haut sur cette broche durant deux cycles machine pendant que l'oscillateur fonctionne remet à zéro le composant.

La broche ALE-/P : l'impulsion de la sortie validation du verrouillage des adresses (Address Latch Enable) sert à verrouiller l'octet d'adresse de poids faible durant l'accès à la mémoire externe ; cette broche est aussi l'entrée d'impulsion de programmation de la mémoire Flash. En opération normale, ALE est émise à un taux constant de 1/16 de la fréquence de l'oscillateur et peut être utilisée à des fins de chronométrage ou de génération d'horloge externes. Il est à noter, cependant, qu'une impulsion ALE est sautée durant chaque accès à la mémoire de données externe. Si l'utilisateur le désire, l'opération ALE peut être dévalidée en mettant au niveau logique haut le bit 0 du registre SFR d'adresse 8EH ; avec ce bit positionné, ALE est active seulement durant une instruction de MOVX ou MOVC sinon, la broche est faiblement mise en rappel au niveau logique



haut. Le positionnement du bit de dévalidation de ALE n'a pas d'effet si le microcontrôleur est en mode d'exécution externe.

La broche /PSEN est la validation de l'enregistrement du programme (Program Store Enable) qui valide la lecture vers la mémoire programme externe. Lorsque l'AT89C51 exécute du code de la mémoire programme externe, /PSEN est active deux fois à chaque cycle machine, excepté que deux activations de /PSEN sont sautées durant chaque accès à la mémoire de données externes.

La broche /EA-VPP : la validation de l'accès externe (External Access Enable) doit être mise à la masse GND de manière à permettre à l'AT89C51 de rechercher le code à partir des emplacements de la mémoire programme externe qui partent de 0000H pour aller vers FFFFH. Il est à noter, cependant, que si le bit 1 de verrouillage, /EA est verrouillée en interne sur la remise à zéro. /EA doit être reliée à la tension d'alimentation VCC pour des exécutions du programme en interne ; cette broche reçoit aussi la validation de la tension de programmation +12V (VPP) durant la programmation de la mémoire flash pour les parties du composant qui nécessitent +12V.

La broche XTAL1 est l'entrée vers l'amplificateur inverseur de l'oscillateur et l'entrée vers le circuit interne de génération d'horloge.

La broche XTAL2 est la sortie de l'amplificateur inverseur de l'oscillateur.

XTAL1 et XTAL2 sont respectivement l'entrée et la sortie de l'amplificateur inverseur qui peut être configuré pour être utilisé en tant qu'oscillateur sur la puce, ou encore utiliser un résonateur en cristal de quartz ou céramique. Pour s'interfacer à une source d'horloge externe, XTAL2 doit restée non connectée tandis que XTAL1 doit accueillir la source de l'horloge ; il n'y a aucune condition particulière sur le rapport cyclique du signal d'horloge externe, puisque l'entrée du circuit d'horloge interne passe à travers une bascule qui divise par deux, mais une tension basse minimale et haute maximale doit être observée.

L'AT89C51 est normalement vendu avec son champ de mémoire flash en état effacé, c'est-à-dire avec un contenu de FFH, et prête à être programmée. L'interface de programmation accepte soit une haute tension de +12V ou un signal de vali-

ation de programmation basse tension VCC. Le mode de programmation basse tension fournit un moyen pratique de programmer l'AT89C51 sur le système utilisateur lui-même, tandis que le mode de programmation haute tension est compatible avec les programmeurs conventionnels de mémoires flash ou EPROM.

Ce circuit intégré est vendu soit avec le mode de programmation basse ou haute tension validée. Le champ mémoire de code du 89C51 est programmé octet par octet dans chaque mode de programmation. Pour programmer n'importe quel octet non effacé sur la mémoire flash interne sur la puce, la mémoire entière doit être effacée en utilisant la mode d'effacement de la puce. Expliquons à présent l'algorithme de programmation. Avant la programmation de l'AT89C51, les signaux d'adresses, de données et de contrôle doivent être positionnés suivant le mode de programmation de la mémoire flash. Afin de programmer ce microcontrôleur, l'utilisateur doit suivre les étapes suivantes :

- 1) Entrer les emplacements mémoire désirés sur les lignes d'adresses,
- 2) Entrer les octets de données appropriés sur les lignes de données,
- 3) Activer la combinaison correcte des signaux de contrôle,
- 4) Monter /EA-VPP à 12V pour la programmation haute tension,
- 5) Envoyer une impulsion ALE-/PROG une fois pour programmer un octet dans le champ de la mémoire Flash ; le cycle d'écriture d'un octet s'auto-calibre dans le temps

et ne prend typiquement pas plus que 1,5 ms.

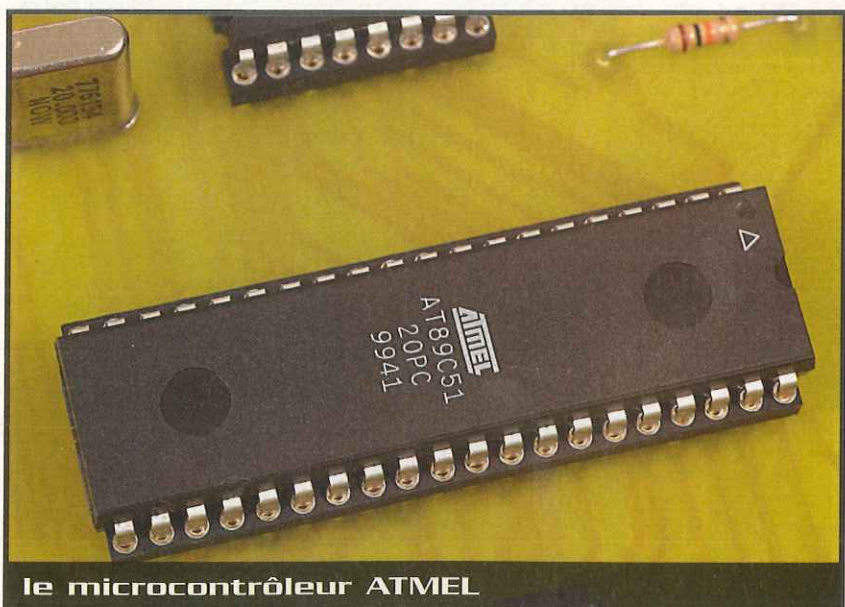
Il faut répéter les étapes 1 à 5 en changeant les adresses et les données pour le champ complet ou jusqu'à ce que la fin du fichier objet soit atteinte.

La caractéristique de l'AT89C51 du contrôle des données (Data Polling) sert à indiquer la fin d'un cycle d'écriture. Durant un cycle d'écriture, une tentative de lecture du dernier octet écrit résulte dans le complément de la donnée écrite sur P0.7. Une fois que le cycle d'écriture est terminé, la vraie donnée est valide sur toutes les sorties et le prochain cycle peut commencer. Le contrôle des données peut commencer n'importe quand après l'initialisation d'un cycle d'écriture.

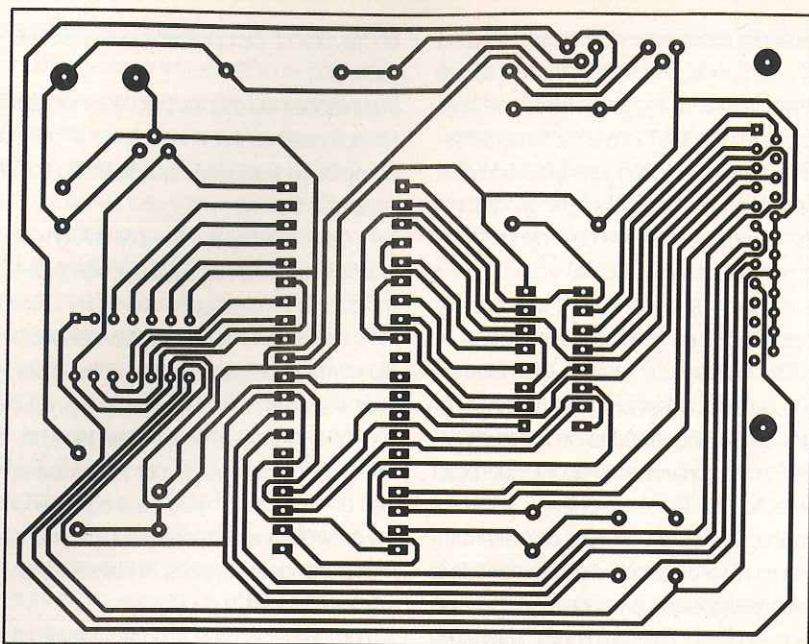
La progression de la programmation d'un octet peut aussi être contrôlée par le signal de sortie RDY-/BSY. P3.4 est mis au niveau logique bas après que ALE soit au niveau logique haut durant la programmation pour indiquer BUSY. P3.4 est mis de nouveau au niveau logique haut lorsque la programmation est terminée pour indiquer READY.

Si les bits de verrouillage LB1 et LB2 n'ont pas été programmés, le code de données programmées peut être relu par les lignes d'adresses et de données pour une vérification. Les bits de verrouillage ne peuvent pas être vérifiés directement. La vérification des bits de verrouillage est accomplie en observant que leur caractéristique est validée.

Le champ entier de la flash est effacé élec-



le microcontrôleur ATMEL



3 Tracé du circuit imprimé

triquement en utilisant la combinaison correcte des signaux de contrôle et en maintenant le signal ALE-/PROG au niveau logique bas pendant 10 ms. La zone de code est écrite avec que des 1. L'opération d'effacement de la puce doit être exécutée avant que le code mémoire soit reprogrammé. Les octets de signature sont lus avec la même procédure qu'une vérification normale des emplacements 030H (valeur = 1EH indiquant le fabricant

ATMEL), 031H (valeur = 51H indiquant 89C51) et 032H (valeur = FFH pour programmation +12V et 05H pour programmation +5V) excepté que P3.6 et P3.7 doivent être portés au niveau logique bas.

Réalisation pratique

Le câblage de notre circuit ne pose aucune difficulté particulière. Il est, bien sûr, recommandé de mettre les trois circuits intégrés

sur des supports au cas où ces derniers doivent être changés si une mauvaise manipulation survient. La **figure 3** représente le circuit imprimé côté soudures et la **figure 4** représente le circuit imprimé côté composants.

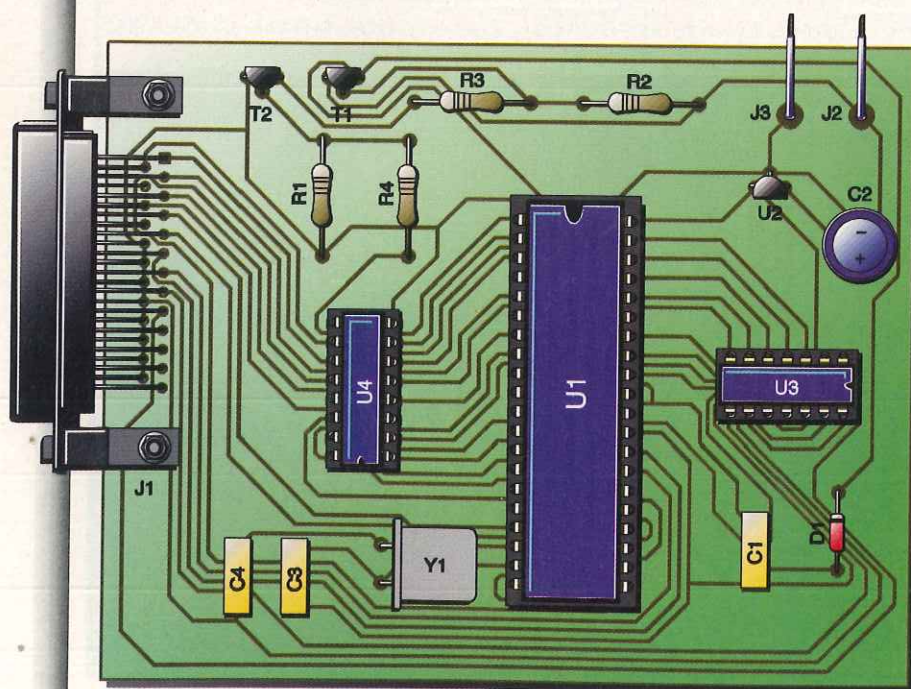
Le programme en langage Turbo C associé à ce montage est disponible sur le CD Rom joint à la revue : il efface dans un premier temps l'AT89C51 et envoie ensuite le programme du microcontrôleur qui est contenu dans un fichier binaire nommé AT89C51.BIN vers le programmeur.

Conclusion

L'utilisateur qui désire approfondir la description de l'AT89C51, ou encore sa programmation, peut se référer à la documentation du constructeur ATMEL très bien documentée. En combinant un microcontrôleur polyvalent 8 bits avec une mémoire Flash sur la puce monolithique, l'AT89C51 est un circuit intégré puissant qui fournit une très haute flexibilité et une solution à faible prix pour beaucoup d'applications de contrôles embarquées.

M. LAURY

4 Implantation des éléments



Nomenclature

- U₁ : AT89C51 + support DIP 40
- U₂ : 78L05
- U₃ : 74HC164 + support DIP 14
- U₄ : 74HC4040 + support DIP 18
- R₁, R₂ : 4,7 kΩ 1/4W (jaune, violet, rouge)
- R₃, R₄ : 10 kΩ 1/4W (marron, noir, orange)
- C₁ : 100 nF
- C₂ : 22 µF/25V radial
- C₃, C₄ : 22 pF
- D₁ : diode 1N4148
- T₁ : transistor 2N2907A
- T₂ : transistor 2N2222A
- Y₁ : quartz 20 MHz
- J₁ : connecteur DB25 mâle pour circuit imprimé
- J₂, J₃ : connecteurs 1 point

INTERFACES PC

Interfaces PC n°8

AU SOMMAIRE :
 Introduction : Mesure de grandeurs physiques par ordinateur - Utilisation du CDROM - Les cartes à puces et Windows® - **Les 16 cartes à réaliser :** Interrupteurs programmables intelligents - Anémomètre sur PC - 8 entrées parallèles vers 1 sortie RS232 - Emulateur d'EPROM - Interface pour Bus 1 fil - Commutateur pour port série - Platine d'essai pour µC 68HC811E2 - Potentiomètre numérique sur port série - Interface parallèle polyvalente sur port série - Contrôle de gain d'un amplificateur opérationnel - Convertisseur décimal/hexa/binaire - Interface série sur port parallèle

avec CD-ROM inclus

de tous les PCB et programmes du numéro + des centaines de pages de catalogues produits, des démos gratuites....



Interfaces PC n°7 Au sommaire :

Les convertisseurs AN/NA - Utilisation du CD-ROM - Tina Pro - Kit Velleman K8016 - Afficheur LCD sur port parallèle - **Les 15 cartes à réaliser :** Thermomètre pour Windows® - Interface RS232 pour téléphone portable - Testeur de télécommande IR - Implémentation d'un contrôleur de souris par le port série - Décodeur DTMF - Jeux sur minitel® - Contrôleur de moteurs pas à pas sur le port série - Programmeur d'arrosage - Interface écran et clavier sur port parallèle - Girouette électronique - capacimètre piloté par liaison RS232 - Bus I2C sur le port parallèle - Parafoudre - Projet multimédia : montage de commande de perceuse à PIC16F84 avec simulation logicielle, uniquement sur le CD-ROM.

avec CD-ROM inclus

de tous les PCB et programmes du numéro + des centaines de pages de catalogues produits, des démos gratuites....



Interfaces PC n°6 Au sommaire :

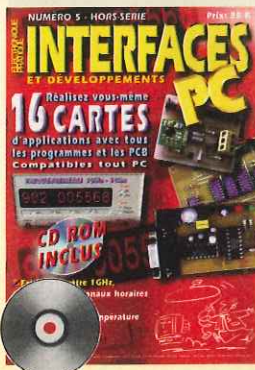
Les mémoires du PC - Utilisation du CDROM - Interconnexion par câble de 2 ordinateurs sous Windows - Le clavier PC et son interface - Un PC de 200 grammes - **Les 15 cartes à réaliser :** Convertisseur A/N sur 12 bits par le port série - Mini programme sur site pour Basic Stamp 2 - Gradateur piloté par PC - Traceur de courbes courant/tension par le port parallèle - Interface d'expérimentations haute protection pour port parallèle - Programmeur CYPRESS CY7C6300 - Système d'entrées/sorties pour port parallèle EPP - Programmeur d'ispGAL 22V10 - Registres à décalage sur PC - Voltmètre à mémoire - Dump d'une cartouche SNES - Sur le CDROM, un montage en multimédia - Identificateur et testeur de câbles - 2 adaptateurs pour entrée micro - Carte à convertisseur A/N pour port parallèle

avec CD-ROM inclus

de tous les PCB et programmes du numéro + des centaines de pages de catalogues produits, des démos gratuites....

PRÉCÉDENTS NUMÉROS DISPONIBLES PAR CORRESPONDANCE

(VOIR LE BON CI-DESSOUS)



Interfaces PC n°5 Au sommaire :

Le port parallèle du PC • Commutateur pour clavier • Horloge internet • Fréquencemètre 1 GHz sur port parallèle • Enregistreur de température autonome • Récepteur de signaux horaires DCF77 • Programmeur 27(C)64/ 27(C)128 • Précis-Volt • Lecteur cartes magnétiques pour PC • Carte 8E analogique sur port série • Fréquencemètre par le port parallèle • Carte 32 E/S sur port série • Convertisseur A/N 4 canaux • Convertisseur RS232 pour bus I2C • Sonde de mesure pour PC • Programmeur pour mC AVR d'ATMEL • Analyseur de protocole par RS232...

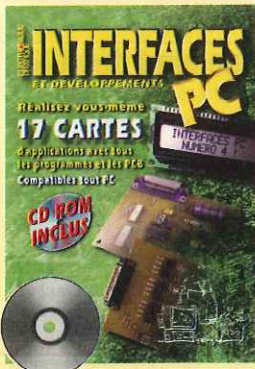
avec CD-ROM des programmes et PCB des réalisations du numéro plus de nombreux sharewares et démonstrations gratuites



Interfaces PC n°3 Au sommaire :

L'évolution du PC Utilisation du CD-ROM
 Les 16 cartes à réaliser : Alimentation de laboratoire - Programmeur d'EEPROM Microwire - Lecteur de cartes à puce - Télécommande téléphonique - Testeur de port // et série - Répartiteur RS232 8 canaux - Convertisseur série // sur port RS232 - Convertisseur RS232 Centronics - Isolateur UV commandé par le port // - Interface RS232 - TTL Thermomètre/Thermostat piloté par PC - Interface de télescope locale - Programmeur de PIC 12C508/509 - Convertisseur analogique 11 canaux - Contrôleur de moteur pas à pas opto-isolé - Interface domotique déportée

avec CD-ROM des programmes et PCB des réalisations du numéro plus de nombreux sharewares et démonstrations gratuites



Interfaces PC n°4 Au sommaire :

L'USB - Utilisation du CD-ROM - Les 17 cartes à réaliser : Interface XY - Espion USB - Liaison laser RS232 - Alimentation programmable - Convertisseur série-parallèle pour imprimante - RS232 vers 8 entrées - RS232 relais - RS232 vers 8 sorties - Analyseur logique 4 canaux - Lecteur de cartes à puce asynchrone - Table de mixage - Thermomètre sans fil - Journal lumineux - Interface série pour afficheurs - Voltmètre 8 voies - Convertisseur RS232/RS422 - Protecteur port Centronics

avec CD-ROM des programmes et PCB des réalisations du numéro plus de nombreux sharewares et démonstrations gratuites



Interfaces PC n°2 Au sommaire :

Les bus et les connecteurs - Commutateur automatique - Carte interface de bus PC - Carte 8 entrées/8 sorties pour bus PC - Carte 8 entrées analogiques à convertisseur A/D - Carte 24 entrées/sorties pour bus PC - Contrôleur de moteur pas à pas - Programmeur de PIC 16C84 par le port parallèle - Isolateur galvanique - Chiffage téléphonique - Convertisseur RS232 boucle de courant passive - Convertisseur N/A 8 voies - Prolongateur RS232 - Espion RS232 - Fréquencemètre 0 à 1 MHz - Verrouillage pour PC - Compteur horaire pour internet - Interface pour moteur à courant continu - Triple alimentation - Télécommande IR par le port série - Répartiteur port Centronics

avec disquette des programmes et PCB ainsi que la version light du logiciel de CAO Quickroute version 4 100% en français

Interfaces PC n°1 épuisé

BON DE COMMANDE

Oui, veuillez me faire parvenir

- Interfaces PC n°2 au prix franco de 40 F
- Interfaces PC n°3 au prix franco de 40 F
- Interfaces PC n°4 au prix franco de 40 F
- Interfaces PC n°5 au prix franco de 40 F
- Interfaces PC n°6 au prix franco de 40 F
- Interfaces PC n°7 au prix franco de 40 F
- Interfaces PC n°8 au prix franco de 40 F
- Interfaces PC n° 2 + 3 au prix spécial franco de 65 F
- Interfaces PC n° 2 + 3 + 4 au prix spécial franco de 100 F
- Interfaces PC n° 2 + 3 + 4 + 5 au prix spécial franco de 140 F
- Interfaces PC n° 2 + 3 + 4 + 5 + 6 au prix spécial franco de 180 F
- Interfaces PC n° 2 + 3 + 4 + 5 + 6 + 7 au prix spécial franco de 200 F
- Interfaces PC n° 2 + 3 + 4 + 5 + 6 + 7 + 8 au prix spécial franco de 220 F

Nom : Prénom :

Adresse :

Code postal : Ville :

Ci-joint mon règlement par chèque mandat CB*

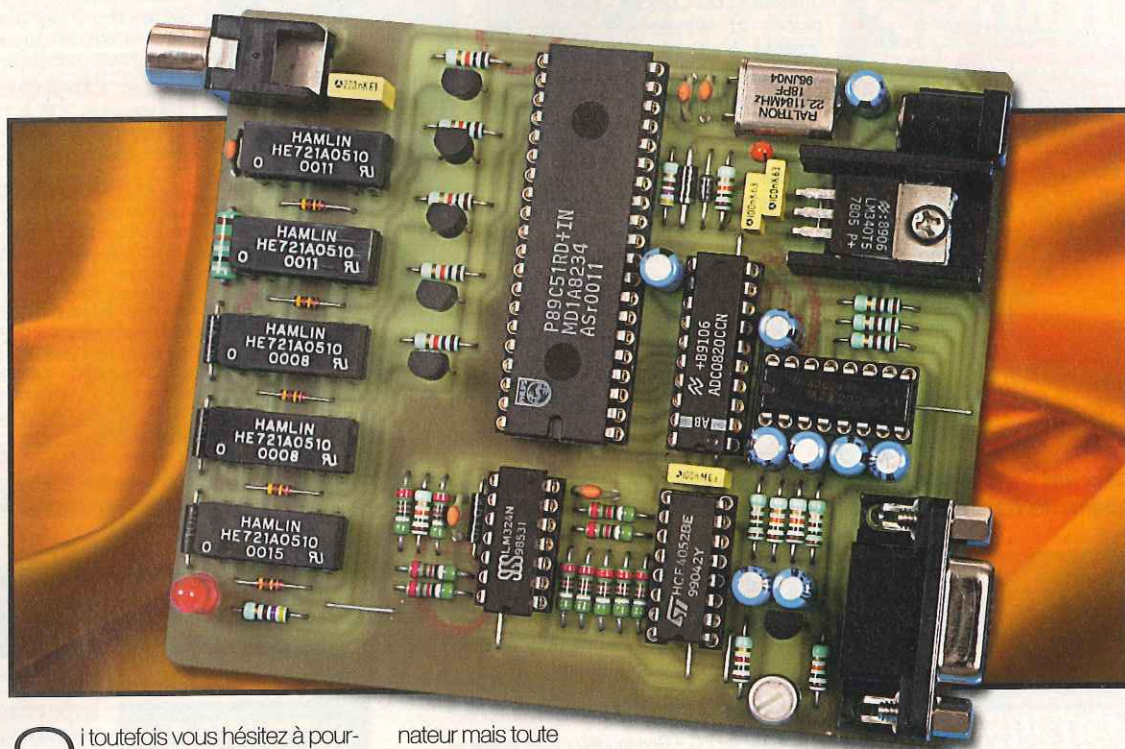
CB n° _____

expire le : _____ signature : _____

*(pour un règlement supérieur ou égal à 100 F)

à l'ordre de **PGV**, Service Abonnements 2 à 12, rue de Bellevue, 75019 Paris ou par internet : <http://www.eprat.com>

Oscilloscope numérique pour PC



Vous avez sans doute rêvé devant les pages des oscilloscopes numériques des revendeurs d'électronique mais le Père Noël est passé sans déposer de cadeaux pour les grands enfants. La description de la carte objet de cet article vous permettra, sans rêveries, de réaliser un outil de mesure performant, facile d'emploi et qui ne surchargera pas la colonne "dépenses" visée par le banquier du ménage.

Si toutefois vous hésitez à poursuivre cette aventure, la lecture des caractéristiques suivantes va, sans conteste, vous convaincre :

- 200 K échantillons/seconde, résolution 8 bits,
 - base de temps de 100 μ s/Carreau à 100 s/Carreau avec 20 échantillons/Carreau,
 - calibres amplitude de 5mV/Carreau à 20V/Carreau en continu et alternatif avec 20 échantillons/Carreau,
 - fonctions mémoire, mesure d'amplitude, durée d'événements, trigger, enregistrement et impression des courbes avec choix des couleurs, sauvegarde pour tableur...
 - Impédance d'entrée normalisée 1 M Ω /27 pF pour sondes 1/10.
- Tout ceci est accessible d'un clic de souris.

Principe de fonctionnement

L'oscilloscope est composé de deux parties, l'une bien visible qu'est la carte électronique, l'autre immergée dans les Méga octets de votre ordi-

nateur mais toute aussi importante qu'est le programme sous Windows. Les deux parties de l'iceberg communiquent par la liaison série RS232 du PC à une vitesse de 115200 bauds. Le programme PC transmet toutes les configurations de calibre base de temps, de tension ainsi que des commandes de mode de fonctionnement tandis que la carte électronique acquiert les échantillons et les transmet.

Les plus attentifs ont déjà bondi sur leur calculette pour vérifier le nombre d'octets qu'il est possible de transmettre en une seconde à cette vitesse de communication pour constater une petite erreur. Le calcul théorique indique qu'il est possible de transmettre un octet (en réalité 8 bits + 2 bits de "start" et "stop") à chaque intervalle de 86,8 μ s alors que le plus petit pas d'échantillonnage est de 5 μ s! En fait, à partir de 5 ms/carreau, soit un échantillonnage toutes les 250 μ s et jusqu'à 5 μ s, le microcontrôleur emmagasine des tranches de temps composées de 600 échantillons qu'il retransmet par paquet au PC à pleine vitesse. Par ce procédé,

la précision entre chaque intervalle de temps est garantie. Pour les calibres de base de temps 10 ms à 100 s/carreau, les mesures sont transmises après chaque échantillonnage. La courbe ainsi formée par la succession de points (échantillons) s'affiche sur l'écran du PC à mesure que remontent les données.

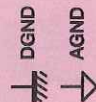
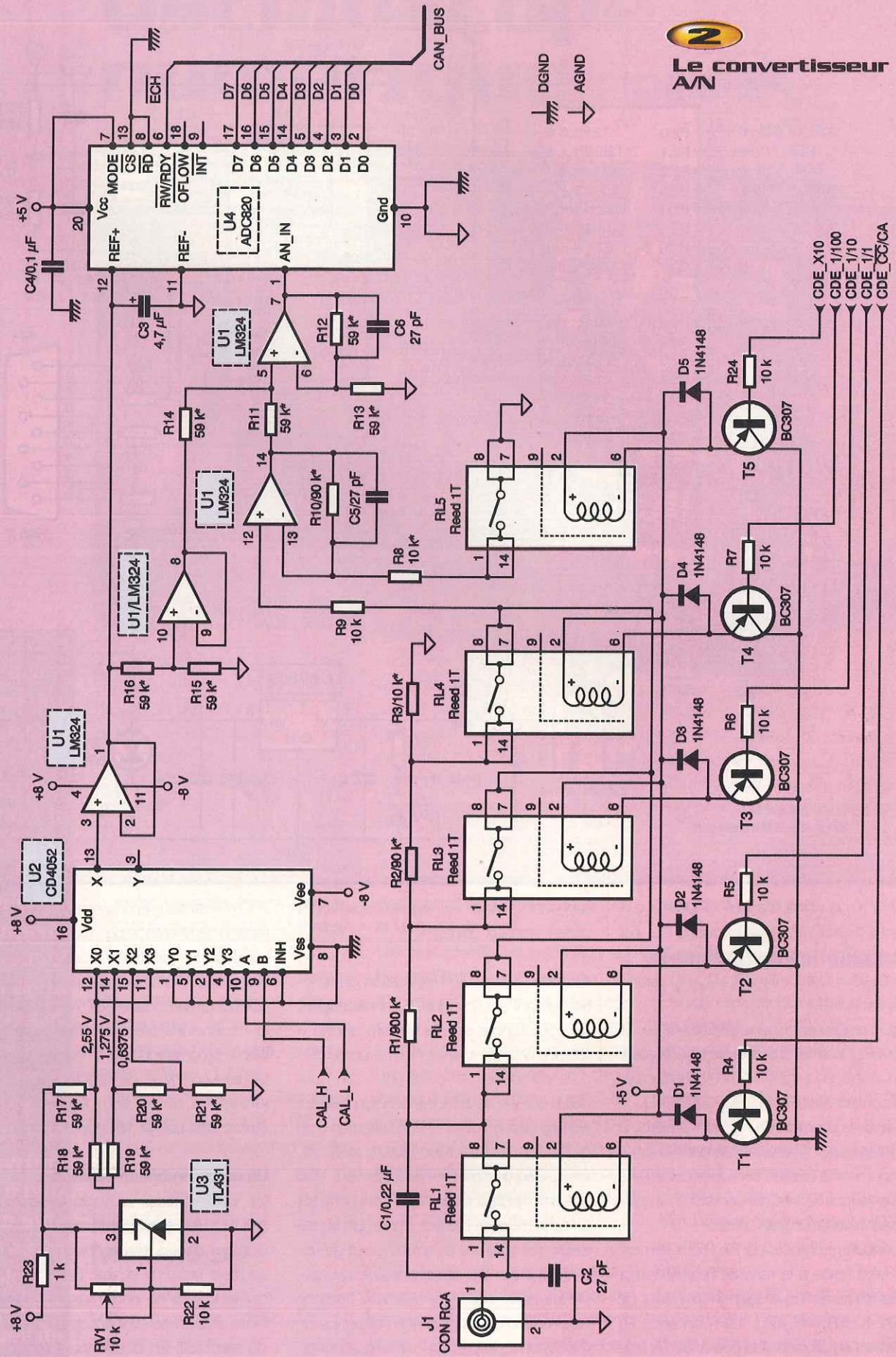
Fonctionnement détaillé

Le signal à mesurer est injecté par le connecteur J₁ (RCA ou BNC), emprunte au choix la liaison à couplage alternatif par C₁ ou la liaison à couplage direct par le relais RL₁. Lorsque les contacts du relais RL₁ sont ouverts, le signal passe par C₁. Le réseau de résistances R₁ à R₃ constitue un atténuateur à 2 décades à impédance constante de 1 M Ω . Le fonctionnement est très simple à appréhender :

Les relais RL₂, RL₃ et RL₄ prélèvent sur les prises intermédiaires du pont diviseur, soit la totalité du signal appliqué à l'entrée quand RL₂ est activé,

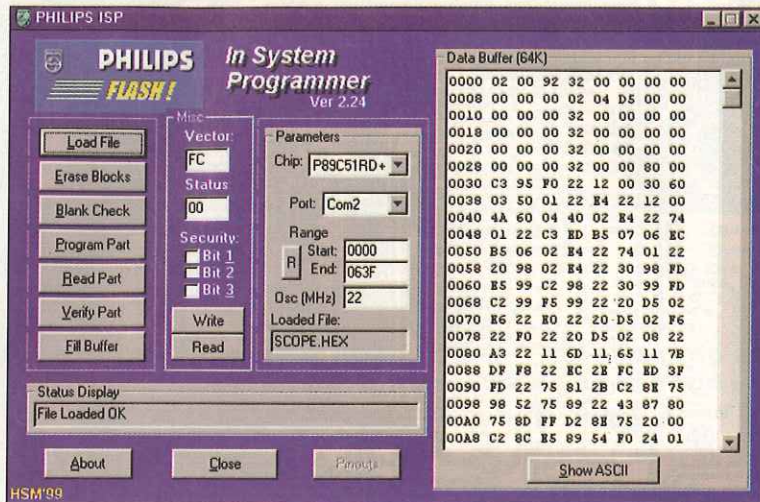


Le convertisseur AN

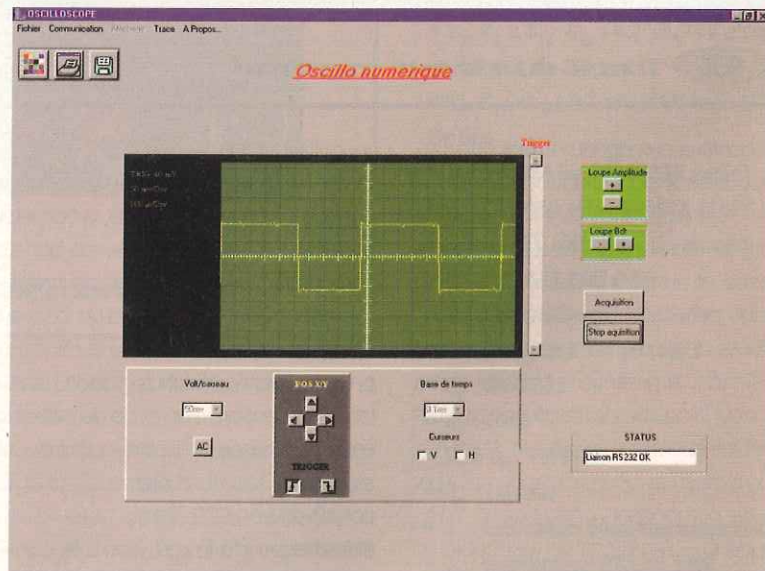


référence et en y ajoutant le signal à mesurer, la trace de l'oscilloscope se positionne naturellement à moitié de l'écran que l'on assimile habituellement à la position du zéro volt. Avec ces quelques explications, le fonctionnement du reste du circuit coule de source.

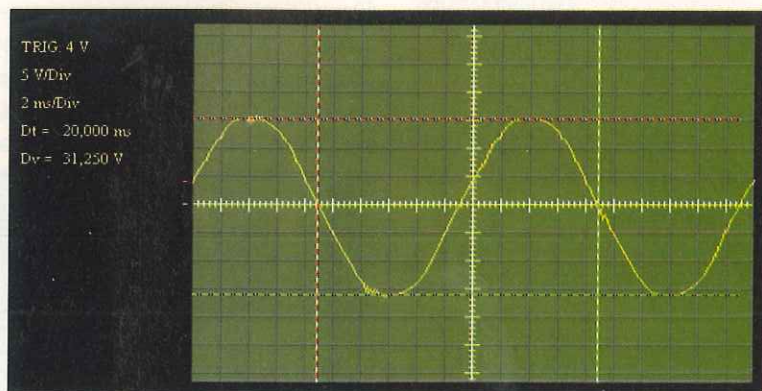
Les tensions de références sont produites par U_3 , RV_1 et R_{17} à R_{23} . Le circuit intégré U_3 de type TL431 se comporte comme une diode zéner programmable dont la tension répond à la formule suivante : $U^{\circ}2,495(1+RV_1/R_{22})$. RV_1 permet un ajustage de la tension zéner entre 2,5V et 5V. Cette dernière tension est appliquée au pont diviseur constitué de cinq résistances identiques R_{17} à R_{21} . Le calcul des tensions aux points intermédiaires est très simple car les cinq résistances vont se répartir équitablement le potentiel total appliqué ($1/5^{\text{ème}}$ de Vref). En soumettant le pont à une tension de 3,1875V, le nœud inférieur est au potentiel 0,6375V, le nœud central à 1,275V et le nœud supérieur à 2,55V, un cas d'école que l'on aurait bien aimé rencontrer. Le multiplexeur analogique U_2 (CD4052) établit une liaison électrique entre la broche X et les broches X0 à X3 en accord avec le mot de commande appliqué sur les entrées A et B. Si A et B sont soumises à deux états bas, l'entrée X0 est sélectionnée, à l'opposé si ces deux comportent un état haut, l'entrée X3 est désignée. Normalement l'entrée X3 n'a pas à être activée, cependant, à la mise sous tension et tant que le logiciel du microcontrôleur n'a pas pris le contrôle des broches de port, ces dernières présentent un état haut sélectionnant ainsi l'entrée X3. En reliant l'entrée X3 à X0, la tension de référence la plus élevée est sélectionnée par défaut. La deuxième partie du multiplexeur n'est pas utilisée, toutes les entrées/sorties sont tirées à la masse. Notez un petit détail concernant les masses : Le circuit U_2 (4052) a une vocation analogique, cependant ne vous y méprenez pas, la broche 8 (VSS) est reliée à DGND qui est la masse numérique. En effet cette broche sert de référence pour exprimer les états logiques présents sur les entrées A et B. Les signaux analogiques (tensions de référence) évoluent, quant à eux, entre les potentiels VDD et VEE à $\pm 8V$. La chaîne analogique est constituée de U_1 , un quadruple amplificateur opérationnel. Deux des quatre sections sont montées en amplificateur suiveur (U_{1A} et U_{1B}) dont le but



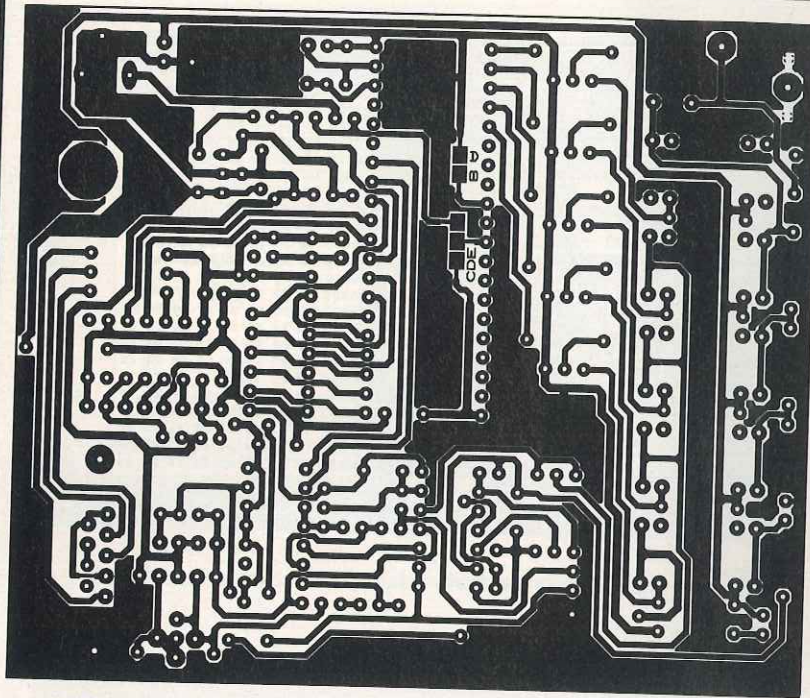
3 Programmation du microcontrôleur



4 Ecran de contrôle



5 Déplacement des barres de mesure avec la souris



6 Tracé du circuit imprimé

est de conférer aux signaux traités une certaine robustesse pour attaquer les sections avalées. U_{1A} injecte à U_4 (CAN ADC820) la tension de référence dont il a besoin pour sa pesée et au pont diviseur de tension constitué par R_{15} et R_{16} . Ces résistances de valeurs identiques vont se partager V_{ref} et ainsi créer le potentiel $_Vref$ qui sert à positionner la courbe de l'oscilloscope à mi-écran. Dernière étape, ajouter le $_Vref$ à la tension à mesurer, opération accomplie par U_{1C} et les composants connexes R_{11} à R_{14} ainsi que C_6 . L'amplificateur est câblé en additionneur non-inverseur comportant deux entrées, donc son gain (amplification I) doit être de deux. S'il avait comporté cinq entrées, son gain aurait été de cinq. Le gain de deux est fixé par R_{12} et R_{13} et répond à la formule :

$$Av = (R_{12} + R_{13}) / R_{13}$$

et comme $R_{12} = R_{13}$, $Av = 2$.

Choisir $R_{11} = R_{14} = R_{13}$.

Vous avez sans doute noté que onze résistances comportent la même valeur (presque arbitraire). Ces dernières seront choisies dans un lot à 1%, voire mieux, leur valeur importe peu et devra être comprise entre 50 k Ω et 100 k Ω . A vos fonds de tiroir !

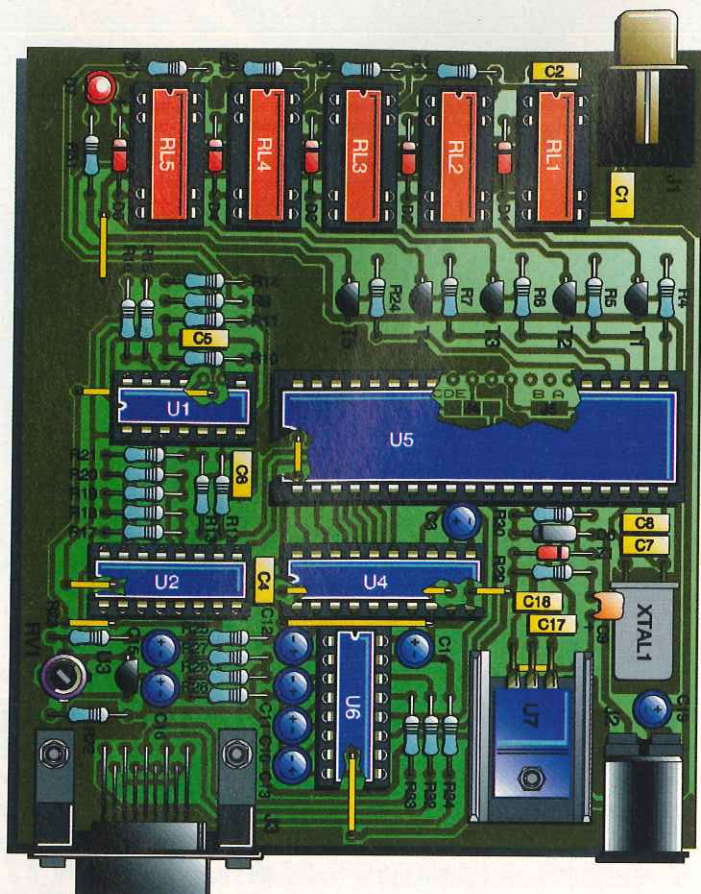
Le circuit U_4 est un convertisseur analogique/numérique semi-flash à 8 bits. Il tient son nom au fait que la conversion est effectuée en deux passes par deux convertis-

seurs flash à 4 bits entre autre, un convertisseur N/A à 4 bits et un soustracteur. Dans le mode de conversion utilisé, la conversion s'effectue en environ 2 μ s, ce qui reste compatible avec les temps de travail du microcontrôleur. Le choix d'un convertisseur (beaucoup) plus rapide aurait été une perte d'argent car la plus grande partie du temps nécessaire à un cycle de conversion est monopolisée par le microcontrôleur. Il y a encore 10 ans (une éternité...), l'ADC820 coûtait environ 200 Francs, aujourd'hui on le trouve pour 5 Euros. L'obsolescence a parfois du bon ! Pour initier la traduction de la tension AN_IN en son équivalent numérique, le microcontrôleur passe momentanément à l'état bas la broche WR/RDY, attend 2 μ s et récupère le résultat de la conversion sur le bus D0 à D7. Afin de ne pas entacher d'erreurs le résultat de la conversion, surtout pour les calibres où la référence de tension de 0,6375V (la plus



faible) est utilisée, un soin particulier a été apporté dans la séparation des masses analogiques (AGND) et numériques (DGND). La jonction a lieu au niveau de U_4 . En effet, un parasite d'une amplitude de 2,5mV sur le rail d'alimentation GND suffit à produire une erreur de 1 bit. Malgré ces précautions, l'ADC820 semble éprouver des difficultés (à peine perceptibles) à travailler avec une référence aussi basse. Pas d'inquiétude cependant.

Du côté du microcontrôleur, rien de bien innovant, le câblage est du plus classique à première vue, mais à y regarder de plus près, on voit roder autour du bloc de cavaliers J_4 et J_5 un signal nommé V_PGM comme pour traduire une intention de programmation. En effet, en achetant un 89C51RD+ de Philips, vous venez d'acquérir l'un des derniers rejets 8051 de l'écurie. Quel cheval ! Avec ses 64 Ko de mémoire flash, 1 Ko de mémoire RAM, son aptitude à s'auto-programmer par le port série et bien d'autres performances, il a de quoi faire pâlir les utilisateurs d'autres marques dans la même gamme de "chip". A l'aide du petit programme fourni gratuitement par le fabricant, vous pourrez débrouiller seul pour la (re)programmation. Reportez-vous au paragraphe "Programmation du microcontrôleur". Pour ce qui est du reste, ce 8051 requiert, comme ses congénères, un réseau RC d'initialisation à la mise sous tension constitué de R_{29} et C_9 . L'horloge est tirée d'un quartz de 22,1184 MHz. Cette valeur étrange correspond au double d'un habituel quartz de 11,0592 MHz duquel on peut extraire une vitesse de communication exacte de 57600 bauds. Logiquement, en doublant la valeur du quartz, on double la vitesse de communication pour atteindre 115200 bauds. La transmission des données entre le PC et la carte s'effectue dans les deux sens par U_6 (MAX232). Trois résistances R_{32} à R_{34} , insérées entre U_6 et la prise DB9 (J_3), tendent à limiter la casse en cas de connexion de la sonde de l'oscilloscope à un montage relié au secteur. Il ne faut pas oublier que le fil de terre du PC est connecté à la masse et, par voie de conséquence, à la masse de notre petit montage. Ces trois résistances ne sauraient se substituer à votre vigilance... Attention donc ! L'alimentation est confiée à un bloc (cube) mural dont le sélecteur de tension est positionné sur 7,5V ou 9V en fonctionnement.



7 Implantation des éléments

normal et sur 12V la première fois pour effectuer la programmation. Le pôle positif est relié à la broche centrale de la fiche d'alimentation. Toute surchauffe du régulateur témoigne d'une trop grande tension d'alimentation. Sur la carte électronique, la tension principale de 5V est fournie par U₇ (7805 + dissipateur) accompagné des condensateurs C₁₇ à C₁₉. D₆ évite une inversion de la polarité de la tension d'alimentation, son prix est sans commune mesure en rapport avec la sécurité qu'elle apporte. La diode zéner Z₁, polarisée par R₃₀, fournit les 12V (V_PGM) nécessaires à la programmation du 89C51RD+. Pour témoigner de l'activité de la carte et afin de ne pas l'oublier sous tension durant des heures (des jours parfois), la LED D₇ visualise la présence de la tension +5V.

Certains lecteurs notent avec raison la présence d'une tension négative sur la carte. Cette dernière provient d'une petite astuce de câblage réalisée autour de U₆ (MAX232). Ce dernier est apte à fournir, grâce à ses pompes de charge, des potentiels de ±8V conformes à la norme

RS232. Le lien est vite établi de l'idée à la connexion.

Les circuits analogiques tireront leur source d'alimentation de ces potentiels leur permettant de jongler sans distorsion près du zéro volt. R₂₅ et C₁₅ constituent un filtre passe bas pour la tension positive, de même que R₂₆ et C₁₆ pour la tension négative. R₂₇ et R₂₈ équilibrent les potentiels symétriques, le point central devient la masse analogique (AGND). Les puristes auraient préféré remplacer ces deux résistances par une masse virtuelle type TLE2425, mais ici les consommations sont si faibles que l'intérêt ne s'en fait pas ressentir.

Réalisation

Le circuit imprimé est très simple à réaliser et n'appelle aucune remarque particulière. Commencez par implanter les 10 straps puis les composants de faible épaisseur comme les résistances, diodes, condensateurs et terminez par les plus épais.

La semelle du régulateur de tension, préa-

Nomenclature

- R₁ : 900 kΩ 0,1% MPR24 ou 909 kΩ 1% MRS25
- R₂, R₁₀ : 90 kΩ 0,1% MPR24 ou 90,9 kΩ 1% MRS25
- R₃, R₆ : 10 kΩ 0,1% MPR24 ou 10 kΩ 1% MRS25
- R₄ à R₇, R₉, R₂₂, R₂₄, R₂₇ à R₂₉ : 10 kΩ 5% SFR25
- R₁₁ à R₂₁ : 51 kΩ à 100 kΩ 1% (toutes identiques)
- R₂₃ : 1 kΩ 5% SFR25
- R₂₅, R₂₆, R₃₂ à R₃₄ : 100 Ω 5% SFR25
- R₃₀, R₃₁ : 470 Ω 5% SFR25
- C₁ : 0,22 μF à 0,47 μF/100V type LCC Philips
- C₂, C₅ à C₆ : 27 pF céramique pas de 5,08 mm
- C₃, C₉ à C₁₆, C₁₉ : 4,7 μF/25V chimique radial pas de 2,5 mm
- C₄, C₁₇, C₁₈ : 0,1 μF/100V type LCC Philips
- U₁ : LM324 ou (OP4906P riche)
- U₂ : CD4052 MUX-DEMUX CMOS
- U₃ : TL431 référence de tension
- U₄ : ADC820 CAN 8 bits
- U₅ : 89C51RD+IN (FARNELL réf. 316-5942 à 114,40 FHT)
- U₆ : MAX232
- U₇ : 7805
- D₁ à D₅ : 1N4148
- D₆ : 1N4007
- D₇ : LED rouge 5mm (ou autre couleur)
- J₁ : connecteur RCA ou BNC
- J₂ : connecteur basse tension à cosses 2,1 mm
- J₃ : prise DB9 Femelle coudée pour circuit imprimé
- J₄, J₅ : ponts de soudure à effectuer sur la carte de circuit imprimé
- RL₁ à RL₅ : relais REED 1T bobine 5V (environ 500 Ω en boîtier DIL)
- RV₁ : ajustable miniature CERMET 10 kΩ /1T ou 10T boîtier type 67W ou 67X
- T₁ à T₅ : transistors PNP type BC307, BC557
- XTAL₁ : quartz 22,1184 MHz
- Z₁ : diode zéner 12V/0,5W type BZX55C12
- 1 dissipateur thermique type ML26 + vis + pâte thermique

lablement enduite d'une fine couche de graisse silicone, sera fixée sur le dissipateur thermique à l'aide d'une vis M3X10. Soudez seulement ses pattes de connexion à l'issue du positionnement et du serrage. Terminez par une inspection visuelle, les pistes sont larges mais un pont de soudure peut toujours se former malencontreusement.

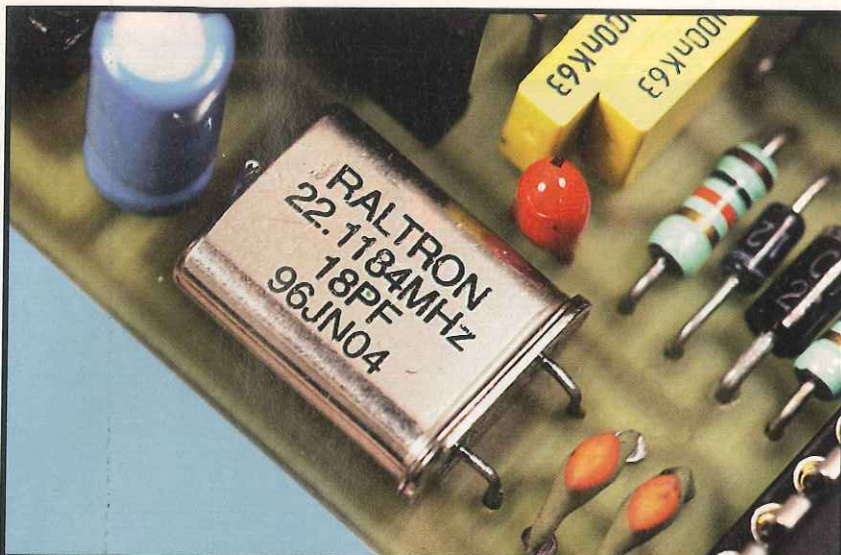
Retirez tous les circuits intégrés de leur support et effectuez une première mise sous tension afin de vérifier les paramètres vitaux de la carte :

- présence de la tension +5V entre les broches d'alimentation des circuits intégrés U_4 , U_5 et U_6 ,
- présence de la tension +12V sur la broche V_PGM du connecteur J_4 (sélectionneur d'alimentation sur 12V !),
- coupez l'alimentation, implantez U_6 (MAX232), remettez sous tension et vérifiez la présence du +8V \pm 2V ainsi que du -8V \pm 2V aux bornes de U_1 et U_2 ,
- procédez au réglage de la tension de référence en agissant sur RV_1 , la tension sera ajustée au plus fin à 2,55V et mesurée sur la broche 12 de U_1 . Vérifiez par la même occasion les tensions 1,275V sur la broche 14 et 0,6375V sur la 15. Ne soyez pas trop exigeant car le chiffre annoncé est théorique et tient pour une grande part à la tolérance de 1% des résistances. De plus la résistance ajustable RV_1 agit dans de grandes proportions.
- coupez l'alimentation et implantez (dans le bon sens) les circuits manquants.

La sérénité vous envahit, les quelques minutes que vous venez de perdre vous distinguent du débutant qui aurait succombé à la tentation d'essayer précipitamment son dernier jouet.

Programmation du microcontrôleur

- 1)** Commencez par installer le programme "WINISP" sur votre ordinateur. Ce dernier prendra en charge toutes les opérations de programmation du 89C51RD+.
- 2)** Déposez une goutte de soudure (sous le 89C51RD) en J_5 (contacts A et B) et J_4 (contacts D et E). Mettez sous tension (cube réglé sur la position 12V) la carte électronique préalablement reliée au PC par le port RS232.
- 3)** Lancez WINISP et remplissez les différents champs de données propres au type



Le quartz et ses condensateurs 27 pF

de micro, au port de communication, à la fréquence du quartz... comme la recopie d'écran reprise en **figure 3** (fenêtre "Parameters"). Chargez le fichier source "SCOPE.HEX" (commande "Load File").

4) Testez la liaison avec la commande de virginité, la fenêtre du bas doit indiquer la vitesse de communication (19200 bauds) retenue. Si le 89C51RD+ n'est pas vierge, alors l'effacer avec la commande "Erase Blocks".

5) Lancez la commande de programmation, attendez quelques instants, c'est déjà programmé !

6) Dans la fenêtre "Misc", modifiez le contenu de la case "Status" avec la valeur 00 sans toucher à la valeur de la case "Vector" qui doit toujours être FC, cela permettra à votre micro de booter normalement à la prochaine mise sous tension. Attention, ne tentez jamais de modifier les valeurs inscrites dans les cases "Vector" et "Status" par autre chose que FC et 00 si vous ne possédez pas de programmeur spécialisé. Vous risqueriez de ne plus pouvoir accéder au petit programme interne de chargement, donc votre composant ne pourrait plus s'auto-programmer. Après vérification du contenu de ces cases, appuyez sur "Write". Par la suite, si vous deviez programmer, effacer, lire... il n'est plus nécessaire de modifier le contenu de la fenêtre "Misc".

7) Cette opération terminée, coupez l'alimentation de la carte, retirez le pont de soudure de J_5 et déplacez celui de J_4 entre les contacts C et D. L'alimentation de la carte se fera désormais sous une tension de 7,5V à 9V.

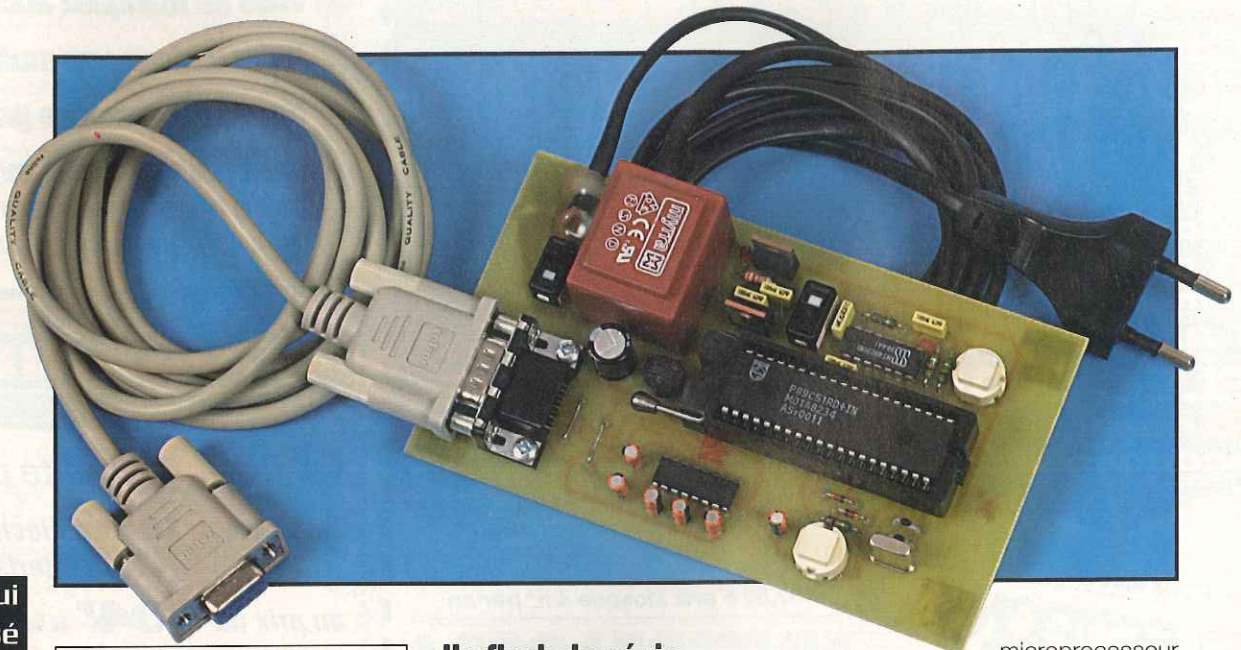
Le programme du 89C51RD+ (1,5 Ko) a été écrit à l'aide du compilateur basic BASICOM 8051 distribué par la société OPTIMINFO (voir pages des annonceurs). Les puristes du langage électronique n'apprécient guère le Basic, mais cette réalisation démontre l'efficacité du compilateur où codes Basic et assembleur se côtoient sans difficulté.

Logiciel d'exploitation

Rien de plus simple ici aussi, il suffit de copier les fichiers du CD sur le disque dur dans un répertoire de votre choix. Le lancement s'effectue par la commande "Osc1.exe", mettez sous tension, sélectionnez le port de communication et appuyez sur le bouton "acquisition". Il est inutile de vous commenter l'ensemble des fonctions tellement l'usage est instinctif. Une description détaillée des commandes viendrait plus encombrer les pages de cette revue qu'éclairer l'utilisateur. Un petit fichier d'aide sera livré dans le futur en fonction de vos questions. Si vous portez beaucoup d'intérêt à ce (petit) montage, contactez la revue et des améliorations (soft PC ou microcontrôleur) pourront être apportées sans aucune modification du montage. En raison des ressources PC utilisées, une machine disposant d'un PII 350 MHz, voire plus, semble être un minimum pour une bonne fluidité du rendu. Cette caractéristique minimale semble bien éloignée du standard actuel !

Merci à Ch. LE BIGOT,
J-Ph. PITOLLET

Programmateur Flash 8051



Le montage qui vous est proposé aujourd'hui concerne, non seulement les nombreux utilisateurs de microcontrôleurs famille 8051, mais aussi tous ceux qui n'ont pas encore franchi le pas pour des raisons pratiques (trop compliqué à programmer) ou financières.

Cette réalisation va vous permettre de travailler avec les outils et les micros de dernière génération, avec une fonction supplémentaire permettant de ranimer un composant mal programmé. Elle est à la portée de tout amateur qui possède un PC et un fer à souder.

Complexité : minimale.

Coût du montage : de 0 Francs en version de base si vous faites de la récupération (tous les composants sont très courants) à moins de 200 Francs en version classique.

Un flash de génie

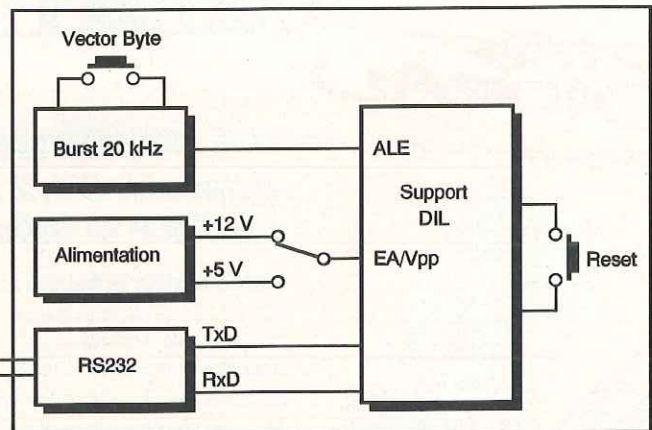
Amis du 8051, vous pouvez envoyer sans arrière pensée tous vos microcontrôleurs à fenêtre (UV PROM dans le jargon) au grand cimetière du silicium. L'association de la technologie FLASH et de l'ISP est en passe de révolutionner le matériel de programmation des microcontrôleurs.

Explication :

Avant, pour développer une application à micro, que ce soit en version microcontrôleur UV PROM ou

microprocesseur à EPROM externe, il vous fallait disposer d'un programmeur spécifique, d'un effaceur d'EPROM, d'un peu de temps (pause café pendant le passage aux U.V.) ou de quelques circuits intégrés à fenêtre d'avance. Depuis quelques temps déjà, la technologie FLASH EPROM permet de programmer et d'effacer électriquement certains composants dont les microcontrôleurs (les tubes UV ne lui disent pas merci). Mais la vraie révolution vient de l'ISP (In System Programming) qui permet de program-

1
Synoptique



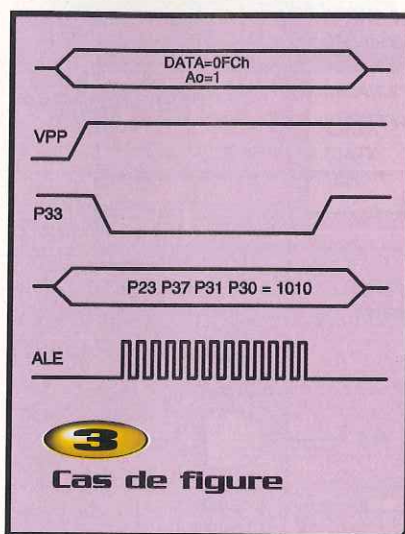
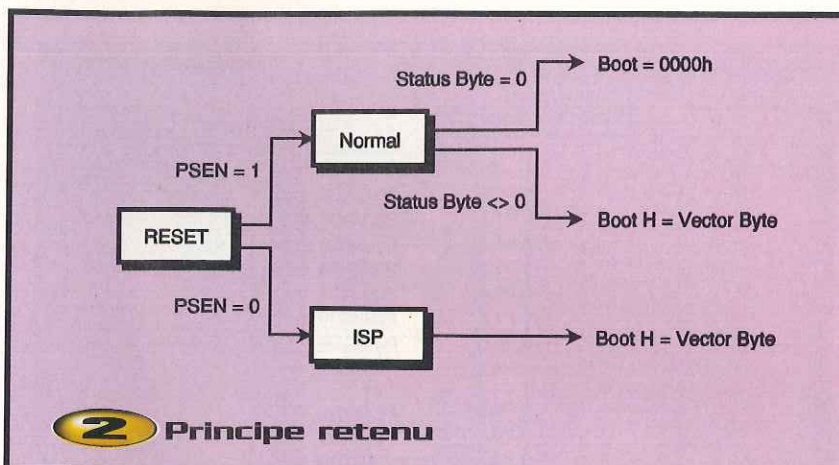
mer et d'effacer électriquement des micro-contrôleurs de la famille 8051 sans recourir à un programmeur spécifique et, pour être plus précis, avec juste une poignée de composants courants.

Comment ça marche ?

Très simplement : Le fabricant a intégré une ROM de 1Ko dans le 8051 à l'adresse FC00h. Cette ROM contient les quelques routines ISP nécessaires au test et à la programmation série du composant. Elle n'est sélectionnée que lors de la phase de programmation et n'interfère pas avec l'espace d'adressage du 8051 qui reste toujours égal à 64Ko. Encore plus fort, ces routines sont accessibles directement par dialogue sur la liaison série Rx/Tx utilisant un protocole très simple. Et pour décider les plus réticents, il est inutile de se pencher sur ledit protocole, car la société PHILIPS vous offre un logiciel complet d'exploitation sur son site Internet.

Comme un malheur n'arrive jamais seul, les micros FLASH ont suivi une cure de jouvence. Au cœur de 8051 standard a été rajouté un timer2 configurable en mode compteur, timer, liaison RS232, chien de garde, générateur PWM, capture, horloge programmable (rien que ça !). Le séquençement a été réduit sur certains modèles à 6 cycles d'horloge par instruction à 20 MHz (au lieu des 12 cycles rencontrés habituellement). On voit également apparaître un deuxième DPTR et quelques fonctions annexes qui concernent la gestion des interruptions et le mode Power Down.

Les nouveaux modèles FLASH (préfixe P89C51) existent également en OTP (préfixe P87C51). La lettre R suivie de A, B, C ou D indique la taille de la FLASH (de RA à RD pour 8Ko à 64Ko). Le suffixe "+" indique un micro à cycle d'horloge standard, tandis que le suffixe "2" est réservé aux micros à cycle d'horloge réduit. Dans la version la plus complète (P89C51RD2), PHILIPS propose donc un micro de 64K FLASH EPROM avec 1Ko RAM, reprogrammable plus de 10000 fois, capable de tourner à 20 MHz sur 6 cycles (soit un équivalent 40 MHz), pour un prix unitaire inférieur à celui d'une cartouche de (mauvaises) cigarettes. Plusieurs constructeurs proposent également des micros FLASH ISP très performants à base de cœur 8051, comme



ATMEL/TEMIC et DALLAS, avec une version à 50 MHz exécutant une instruction en un seul cycle machine !

Le programmeur qui vous est proposé aujourd'hui (schéma synoptique **figure 1**) vous donne accès à une nouvelle famille de composants à cœur de 8051, ainsi qu'à une nouvelle façon de programmer. La version FLASH sera avant tout très efficace lors des phases de développement. Le micro définitif pourra tout à fait être un OTP, qui est moins cher qu'un FLASH. Si celui-ci ne comporte pas l'ISP (références en 87C51FA à FC), vous pourrez toujours effectuer une programmation parallèle classique, par exemple à l'aide du programmeur décrit dans le n°213 d'Électronique Pratique.

Status Byte et Vector Byte

Le 8051 FLASH contient deux registres spécifiques qui déterminent son mode de fonctionnement. En fonctionnement normal (PSEN non forcé au niveau bas), lors du front descendant sur la patte de Reset, la

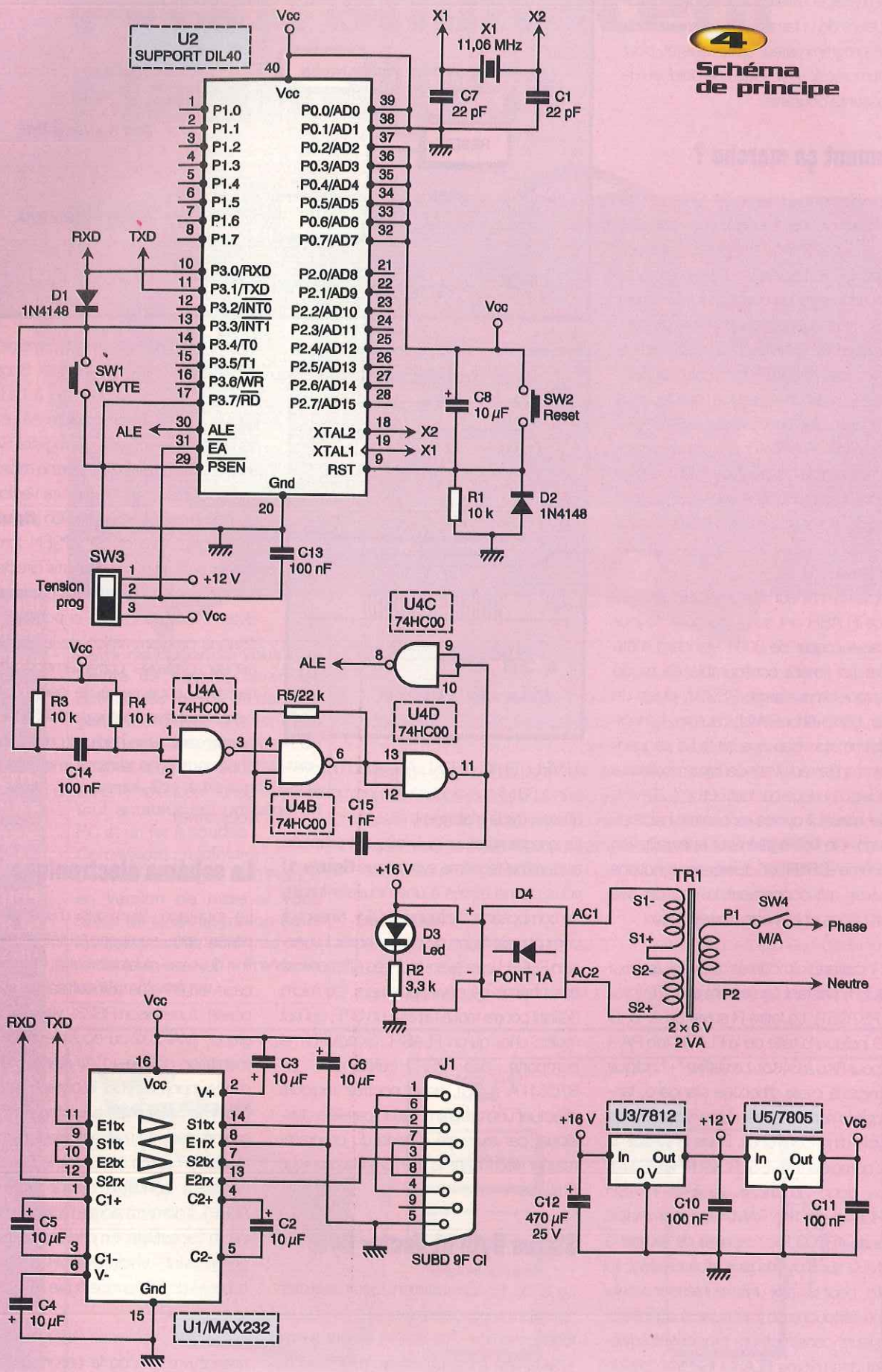
logique interne du micro examine le contenu du registre StatusByte. Si celui-ci vaut 00h, le micro démarre à l'adresse 0000h, c'est à dire à son adresse normale de boot. Si le contenu du registre Status-Byte est différent de 00h, alors le micro examine le contenu XXh du registre VectorByte, et il démarre à l'adresse XX00h (**figure 2**). En fonctionnement ISP (PSEN forcé au niveau bas), le micro démarre directement à l'adresse XX00h indiquée par le Vector-Byte. Il est donc très important, après chaque programmation de la FLASH, de penser à initialiser correctement ces deux registres, sous peine de perdre plus ou moins définitivement le contact avec votre composant. Une partie du montage que nous vous proposons permet néanmoins de le "récupérer" si une telle mésaventure vous arrivait.

Le schéma électronique

Le montage, bien que d'une simplicité désarmante, comporte toutefois une fonction qui va se révéler fort utile. La version de base, en principe autosuffisante, est composée d'une liaison RS232 via le circuit intégré U₁ (MAX232 ou équivalent), d'une alimentation double +12V/+5V (U₃ et U₂) et d'un support U₂ pour le composant à programmer. A la mise sous tension, le programmeur se met automatiquement en mode ISP après appui sur le poussoir de Reset, à condition d'avoir PSEN=0 et ALE=1. Il démarre alors à l'adresse indiquée par le VectorByte. En principe, à l'achat du composant, VectorByte=FCh, soit une adresse de démarrage égale à FC00h pour le mode ISP.

Oui mais... si vous faites un fausse manœuvre lors de la programmation du VectorByte (par exemple, vous remettez à

4
Schéma de principe



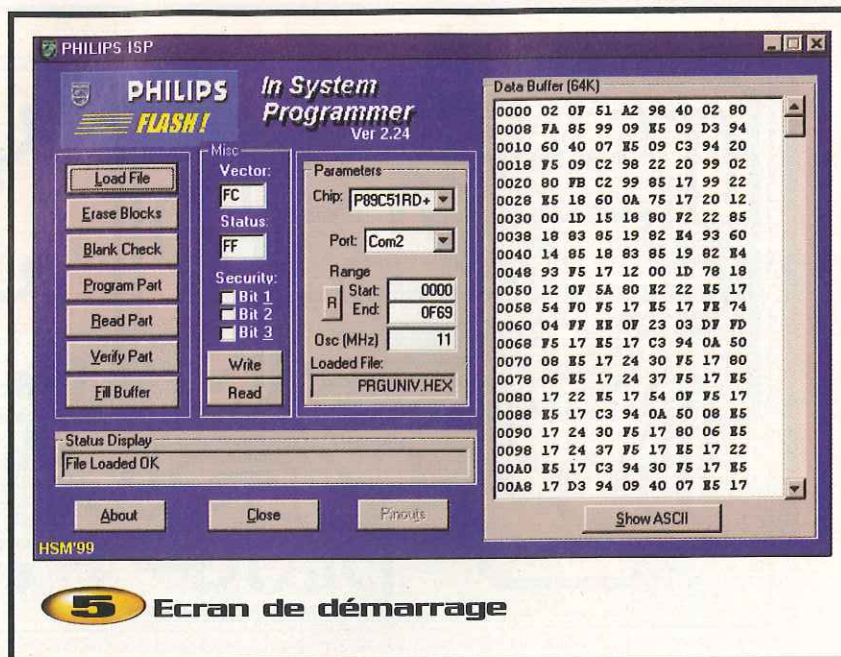
zéro le contenu de StatusByte et Vector-Byte, mais vous oubliez de reprogrammer le VectorByte), alors votre composant va démarrer à l'adresse 0000h lors de la prochaine séance de programmation ISP. Inutile de vous dire qu'il aura du mal à rejoindre son adresse de démarrage normale 00FCh. Dans la pratique, cela signifie que vous ne pouvez pas accéder aux routines ISP, donc vous ne pouvez plus reprogrammer votre composant par l'ISP. Votre FLASH vient de se transformer (temporairement, rassurez-vous) en OTP.

La seule possibilité qui vous reste pour "décoincer" votre micro consiste à utiliser un programmeur du commerce pour reprogrammer le VectorByte. Sauf si vous utilisez la fonction supplémentaire de notre montage (circuit intégré U_4). Lorsque SW_1 est enfoncé, un astable ($U_{4B}/U_{4D}/U_{4C}$) produit des signaux sur ALE à 20 kHz pendant une fenêtre de temps de 1ms fixée par $C_{14}/R_4/U_{4A}$. Vous remarquerez que le mot de 8 bits, présent en permanence sur le port P0, est égal à FCh. Si ces deux événements ont lieu simultanément alors que les signaux P37, P33, P30 et RESET sont au niveau bas (figure 3), le mode programmation parallèle du VectorByte est actif. En clair, maintenez le poussoir RESET enfoncé et appuyez sur SW_1 , le VectorByte reprendra sa valeur d'origine FCh.

Le commutateur SW_3 sert à déterminer la tension de programmation : +5V pour les références qui ont le suffixe "2" et +12V pour les références qui ont le suffixe "+". Cette tension doit être fixée avant la mise sous tension et rester présente pendant toute la phase de travail avec l'ISP.

La réalisation

Vous remarquerez que le dessin du circuit imprimé (pistes larges et généreuses) a été étudié pour vous permettre de réaliser la gravure sans trop de difficultés, même à l'aide d'un matériel sommaire. Un petit rappel : la qualité du circuit imprimé qui sortira du bain de perchlorure de fer dépend essentiellement du soin apporté à l'insolation (transparent bien contrasté et plaqué fermement contre le circuit imprimé, temps d'insolation correct). Pour obtenir un bon transparent, une technique très simple, mais qui donne d'excellents résultats, consiste à imbiber d'huile le papier support du dessin des pistes du circuit imprimé. Le



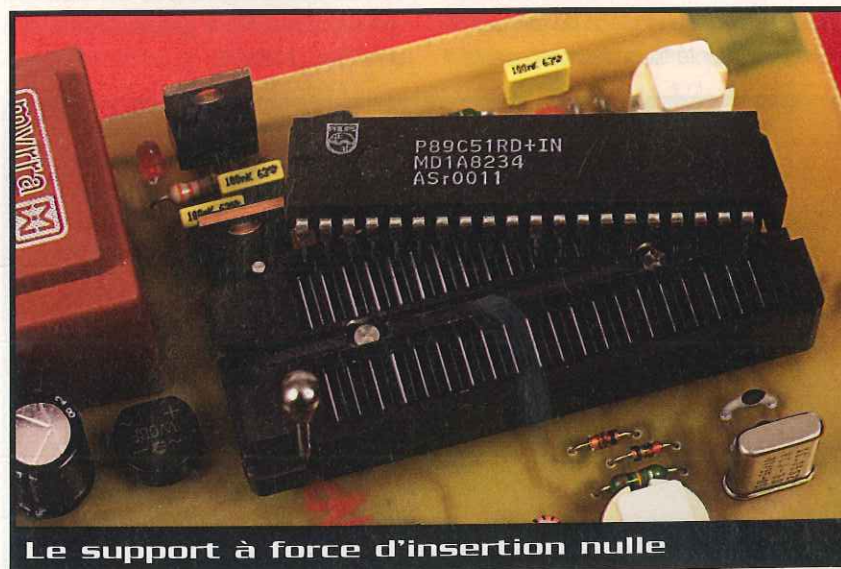
surplus d'huile doit ensuite être parfaitement éliminé afin d'avoir un papier exempt de toute tâche de gras. Vous obtenez alors un transparent aux UV de qualité tout à fait respectable.

Que vous souhaitiez mettre le montage dans un coffret ou le laisser à l'air libre, rajoutez au minimum 1 cm sur chacun des côtés par rapport au contour d'origine pour vous permettre de fixer des vis, des entretoises ou des pattes sous le circuit imprimé. Percez tous les trous à un diamètre de 0,8mm. Suivant la provenance des composants, certains trous devront être agrandis à 1mm (en général le transformateur, les poussoirs, le pont de diode et les régulateurs).

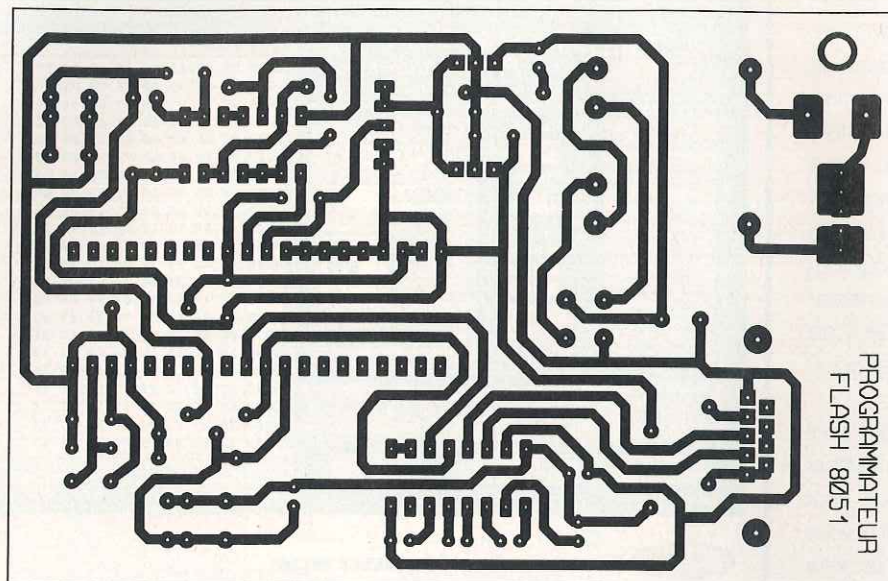
Les condensateurs, jusqu'à une valeur de 100 nF, pourront être de type céramique

ou plastique. Ceux de 10 μ F peuvent être remplacés par des 2,2 ou des 4,7 μ F. Pour le support de circuit intégré U_2 , choisissez impérativement un support tulipe de bonne qualité. Vous remarquerez que le condensateur C_9 , qui se trouve sous le support U_2 , peut se monter couché sur le circuit imprimé. L'utilisation d'un support à force d'insertion nulle est également très fortement conseillée.

A noter : seuls certains supports (par exemple le modèle ARIES de couleur noire) peuvent s'insérer directement dans un support tulipe. Les poussoirs ont été dessinés en double implantation, ce qui permet de monter plusieurs modèles de poussoirs standards. Les régulateurs n'ont pas besoin de refroidisseur si vous respectez les caractéristiques du transforma-



Le support à force d'insertion nulle



6

Tracé
du circuit
imprimé

teur (2x6V/2 à 2,5VA, par exemple un modèle MYRRA 2,3VA). L'interrupteur SW₄ doit impérativement être homologué pour la tension secteur 230V. La liaison au réseau électrique se fera par un cordon et une fiche 2P standard. Si vous laissez votre montage à l'air libre, il faut nécessairement isoler la partie reliée au secteur 230V, ce qui peut être réalisé facilement en enrobant cette zone avec de la colle à chaud isolante. Le circuit imprimé a été dessiné pour accueillir une prise SUBD 9 points femelle qui sera reliée au PC par un cordon standard. Choisissez plutôt une SUBD avec équerre de fixation, les pastilles côté cuivre vous remercieront un jour. Si voulez faire l'économie de ces deux pièces ou dans le cas d'une intégration dans un boîtier, il est toujours possible de souder directement un câble 3 fils en lieu et place de la SUBD9. Vous avez monté tous les composants ?

Passons au test de la carte.

Ne montez pas le microprocesseur sur le support pour la première mise sous tension. Vous devez obtenir du +5V sur la patte 40 du support U₂. Si vous mesurez la tension sur la patte 31, elle doit être de +5V ou de +12V en fonction de la position du commutateur SW₃. Si vous avez un oscilloscope, vous pouvez vérifier que l'appui sur le poussoir SW₁ provoque un burst unique de signaux carrés de fréquence 20 kHz pendant une durée d'environ 1 ms (soit une vingtaine d'impulsions de programmation). C'est tout bon ? Vous pouvez monter un micro sur le support, connecter le montage au port série de votre PC et passer à la partie programmation.

Vérifiez quand même une dernière fois vos soudures, vous gagnerez peut-être un temps précieux.

Le soft de programmation WINISP

Pour vous procurer ce logiciel gratuitement, rien de plus simple : vous vous connectez sur le site Internet de PHILIPS à l'adresse <http://www.semiconductors.philips.com/mcu/download> et vous téléchargez les fichiers WINISP disk1 et disk2 dans un répertoire unique de votre PC. Enfin, lancez le SETUP. Bonne surprise : le logiciel WINISP a la bonne idée d'être compatible avec toutes les versions de Windows. Lancez l'application, l'écran de démarrage s'affiche (**figure 5**). On y trouve plusieurs zones de travail :

- A droite, le buffer qui contient le code exécutable visualisable sous forme ASCII ou HEXA.

- Au centre, des paramètres à renseigner obligatoirement : la référence du microcontrôleur, le port série sur lequel est branché votre montage, et la fréquence du quartz : 11 MHz.

- A gauche, les commandes : charger un fichier, effacement, test de virginité, programmation / lecture / vérification du micro, programmation et lecture des mots "VectorByte" et "StatusByte" dont nous avons déjà longuement parlé, ainsi que la programmation des bits de sécurisation.

- En bas, une fenêtre "StatusDisplay" vous renvoie une information sur le résultat des commandes.

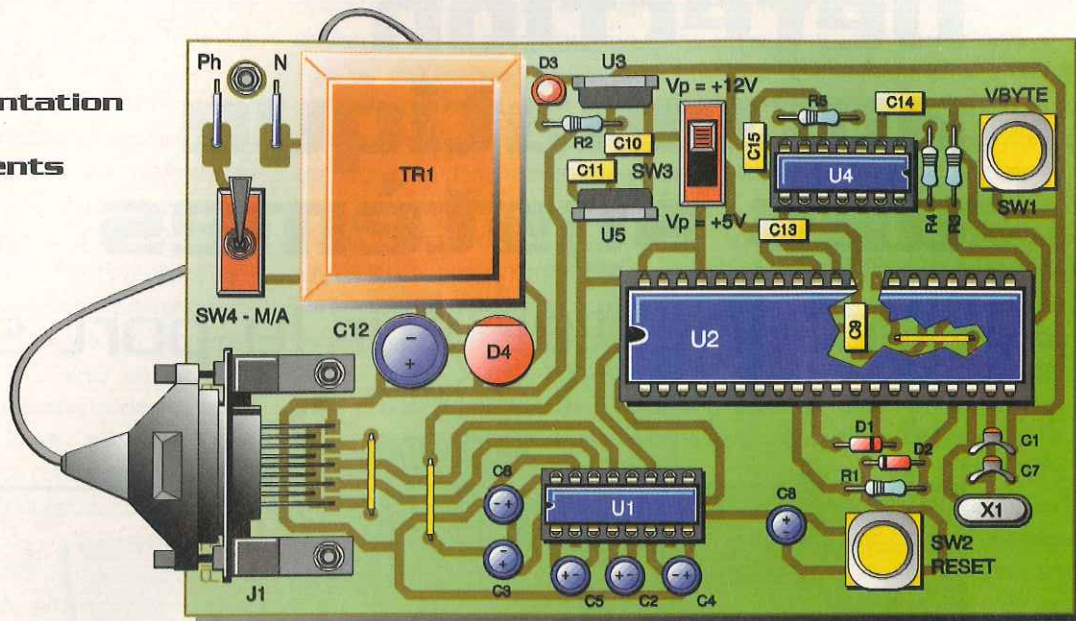
Connectez votre programmeur au PC, mettez le sous tension, appuyer sur le poussoir de Reset. Vous pouvez lancer la



Le connecteur SUBD 9 femelle pour CI

7

Implantation des éléments



Nomenclature

3 straps

R_1, R_3, R_4 : 10 k Ω 1/4W

R_2 : 3,3 k Ω 1/4W

R_5 : 22 k Ω 1/4W

C_1, C_7 : 22 pF céramique ou plastique

C_2 à C_6, C_8 : 10 μ F chimiques

C_9 à C_{11}, C_{13}, C_{14} : 100 nF céramique ou plastique

C_{12} : 470 μ F/25V chimiques

C_{15} : 1 nF céramique ou plastique

U_1 : MAX232 ou ICL232 ou HIN232

U_2 : support tulipe DIL40 + support à force d'insertion nulle

U_3 : régulateur +12V type 7812

U_4 : portes NAND 74HC00 ou équivalent

U_5 : régulateur +5V type 7805

D_1, D_2 : diodes de signal type 1N4148 (ou équivalent)

D_3 : LED 3mm rouge ou verte

D_4 : pont de diodes 1A standard

J_1 : connecteur SUBD 9 Femelle pour CI

SW_1, SW_2 : poussoirs standards ITT (ou équivalent)

SW_3 : commutateur 2 positions faible puissance

SW_4 : interrupteur marche/arrêt 230V alternatifs

TR_1 : transformateur 2x6V/2,2VA

(ou 12V/2,2VA)

X_1 : quartz 11,0592 MHz

(ou 22,1184 MHz)

Hors schéma

Cordon secteur + prise 230V

Circuit imprimé

Cordon prolongateur 9 points M/F

Pieds adhésifs + colle à chaud

(ou boîtier)

commande test de virginité "Blank Check" sur votre micro FLASH pour tester la liaison série. WINISP doit vous informer du résultat dans la fenêtre "StatusDisplay". Sur un micro vierge, la réponse normale est "Selected Range is Blank". Si le micro contient un programme, la réponse est "Part not Erased at xxxh". Si le soft vous renvoie des horreurs du genre "Invalid Response" ou "Initializing Device", soit la liaison série fonctionne mal (vérifiez que vous êtes bien connectés sur le bon port RS232), soit le micro est planté (appuyez sur le poussoir de Reset).

Ce type de message apparaît également lorsque le "VectorByte" est mal programmé (valeur différente de FCh) ce qui ne peut pas être le cas sur un micro tout juste sorti du magasin. Pour déverrouiller le micro, et c'est là tout l'intérêt du montage, maintenez le poussoir de Reset enfoncé et appuyez sur SW_1 . Votre micro est prêt pour une nouvelle vie.

Pour charger le fichier à programmer, utilisez la commande "Load File". Celui-ci doit être au format standard HexIntel qui est délivré par tous les compilateurs pour 8051. La commande d'effacement du micro "Erase Blocks" permet d'effacer tout ou partie du composant par morceaux de 8Ko ou 16Ko, ce qui peut être utile si plusieurs programmes différents cohabitent dans le même microcontrôleur. L'effacement se fait en quelques secondes seulement !

La commande annexe "Fill Buffer" vous permet de remplir complètement le buffer avec

une même valeur. Les commandes de programmation, lecture et vérification n'appellent pas de commentaires particuliers. Le temps de programmation est d'environ 1Ko de programme toutes les 3 secondes. Après avoir programmé le composant, n'oubliez jamais d'utiliser la commande "Write" pour programmer le "VectorByte" à FCh et le "StatusByte" à 00h, ce qui vous permettra de pouvoir utiliser et reprogrammer normalement votre 8051. Si vous voulez éviter les déboires, faites-le systématiquement sans vous fier aux valeurs affichées dans la fenêtre concernée.

L'environnement de travail tel que nous venons de le décrire correspond à la version 2.24 de WINISP. Si vous voulez de la documentation sur les micros FLASH de Philips, vous pouvez consulter le site <http://www.semiconductors.philips.com> rubrique "microcontrollers". La conception de ce programmeur a pu être élaborée grâce aux documents techniques suivants :

Flash Memory Microcontrollers Program/Erase Specifications (1999 Jun 17)

In-circuit and In-application programming of the 89C51Rx+/Rx2/66x microcontrollers (AN461)

8XC54/58 8XC51FA/FB/FC

8XC51RA+/RB+/RC+/RD+ (DATA SHEET 2000 Aug 07)

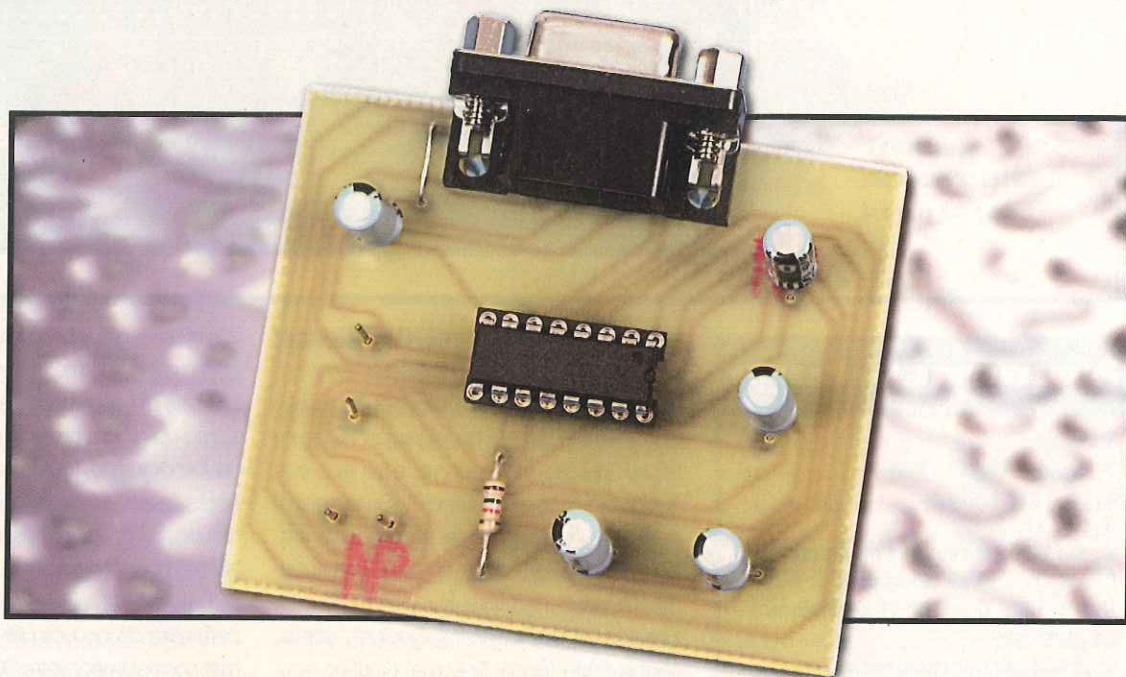
P89C51RB2/P89C51RC2/P89C51R

D2 (DATA SHEET Preliminary Specifications 2000 Aug 21)

E. MIGOT

Détection automatique des systèmes

connectés sur le port série



L'idée de base du montage présenté ici est de générer un signal sonore qui peut être détecté par l'application utilisée par un ordinateur de type PC. De cette manière, l'application en question peut détecter automatiquement le port série auquel le périphérique externe est connecté

Étude du montage

La communication série asynchrone est peut-être une des plus populaires méthodes de communication entre un ordinateur personnel et un périphérique externe, basée sur le standard qu'est le RS232. Ces périphériques externes peuvent être : des outils de développement, des outils de mise au point, l'acquisition de données, etc. qui sont seulement occasionnellement connectés à l'ordinateur. C'est parce que d'autres ressources beaucoup plus fréquemment utilisées, telles que la souris, un modem externe ou une manette de jeu, ont besoin d'accéder à un nombre limité de port série.

Lorsqu'il est temps de connecter un de ces périphériques, qui possèdent une faible priorité, que la confusion s'installe : sur quel port COM le périphérique en question

doit être configuré ? Cet article présente une solution intéressante à prendre en considération pour de nouvelles conceptions.

Le montage implique un peu d'électronique supplémentaire ainsi qu'un petit programme qui le pilote et qui est attaché à l'application qui tourne sur l'ordinateur utilisant le port série en question. La communication typique entre un ordinateur de type PC et un périphérique sur le port série utilise seulement trois fils : RX (pour la réception), TX (pour la transmission) et la masse.

Le circuit intégré MAX232 possède quatre translateurs de niveau. Seulement deux sont utilisés dans notre montage : un pour la réception des données (translateur RS-232 vers TTL) et un pour la transmission des données (translateur TTL vers RS-232). Ainsi, deux translateurs restent disponibles. Nous allons étudier, dans une première partie, le

fonctionnement du MAX232 ainsi que du MAX213 du constructeur MAXIM pour expliquer, ensuite dans une seconde partie, son utilisation dans notre application.

Le MAX232 est un émetteur/récepteur de ligne conçu pour toutes les interfaces de communication correspondant aux spécifications EIA/TIA-232E et V.28/V.24 jusqu'à un taux de transfert des données de 120 kbauds avec une charge qui correspond à la spécification EIA/TIA-232E. Chaque sortie d'émetteur et entrée de récepteur est protégée contre les décharges électrostatiques de +/-15 KV, sans blocage. Ces circuits intégrés sont spécialement utiles dans les dispositifs alimentés par batterie, puisque leur mode de faible consommation disponible sur le MAX213 réduit la dissipation de la puissance à moins de 5 µW. Le MAX232 est tout particulièrement recommandé pour les.

applications dans lesquelles l'espace sur le circuit imprimé est critique, ainsi que pour un faible coût. Ce composant est disponible en boîtier DIP de 16 broches et fonctionne avec quatre capacités externes de $1 \mu\text{F}$. Il comprend trois sections : les convertisseurs à pompe à charge en tension, les émetteurs et les récepteurs. Cette version fournit des protections supplémentaires contre les décharges électrostatiques ; elle survit à des décharges de l'ordre de $\pm 15 \text{ kV}$ sur les entrées et les sorties, en utilisant le modèle du corps humain. Lors du test suivant la norme IEC1000-4-2, ce composant a résisté à des décharges par contact de $\pm 8 \text{ kV}$ et des décharges de $\pm 15 \text{ kV}$ par transmission dans l'espace ; il a été conçu pour résister dans des environnements rudes ou dans des applications dans lesquelles la connexion RS232 est fréquemment utilisée (comme dans le cas des ordinateurs portables par exemple).

La conversion de $+5\text{V}$ à $\pm 10\text{V}$ est effectuée par une double pompe à charge en tension représentée à la **figure 1**. Le premier convertisseur à pompe à charge utilise la capacité C_1 pour doubler le $+5\text{V}$ en $+10\text{V}$, stockant le $+10\text{V}$ sur le V_+ de la capacité du filtre de sortie C_3 . Le second convertisseur à pompe à charge utilise C_2 pour inverser le $+10\text{V}$ en -10V , stockant le $+10\text{V}$ sur le V_- de la capacité du filtre de sortie C_4 . Dans le mode basse consommation, V_+ est connectée en interne au V_{CC} par une résistance de rappel de $1 \text{ k}\Omega$ et V_- est connectée en interne à la masse par une résistance interne de $1 \text{ k}\Omega$. Avec une tension d'alimentation de $+5\text{V}$, l'oscillation typique de la commande de sortie en tension est de $\pm 8\text{V}$ lorsque cette dernière est chargée avec un récepteur

RS232 de valeur nominale de $5 \text{ k}\Omega$. L'oscillation en sortie est garantie pour satisfaire les spécifications EIA/TIA-232E et V.28/V.24 qui imposent des niveaux de sortie au minimum de $\pm 5\text{V}$ sous les pires conditions de fonctionnement, ce qui inclut une charge de $3 \text{ k}\Omega$, une tension d'alimentation minimale et une température de fonctionnement maximale.

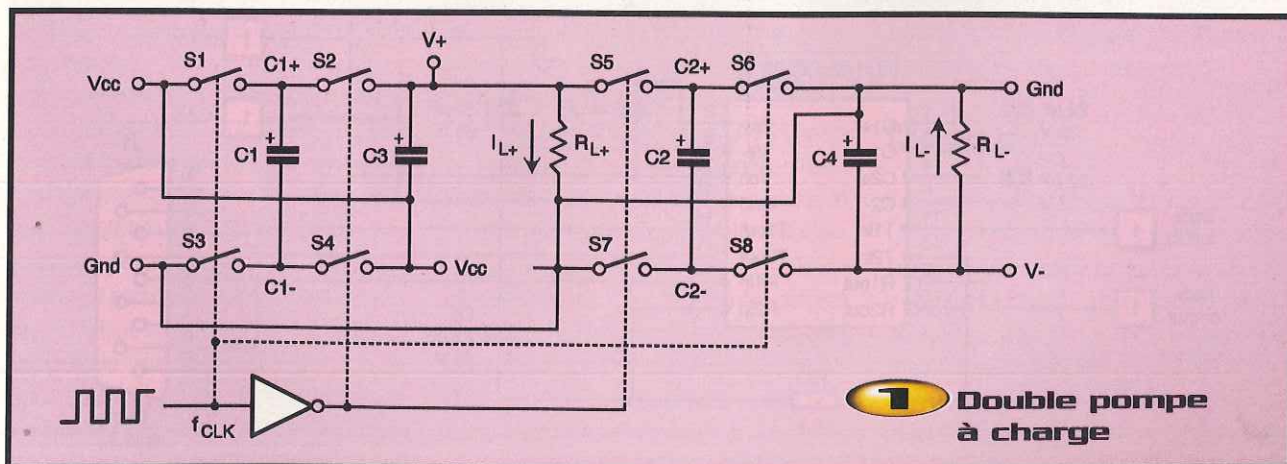
La tension de sortie en circuit ouvert oscille entre $(V_+ - 0,6\text{V})$ et V_- . La tension de seuil en entrée est compatible avec les technologies CMOS et TTL. Les entrées de commande non utilisées peuvent être laissées non connectées car une résistance de rappel à la tension positive de l'alimentation de $400 \text{ k}\Omega$ est incluse sur la puce du composant.

Puisque tous les émetteurs ou récepteurs inversent, les résistances de rappel internes forcent les sorties non utilisées au niveau bas. En mode de basse consommation, les sorties du MAX232 sont coupées et utilisent seulement les courants de fuite, même si elles sont reliées à des tensions entre 0 et 12V . En dessous de $-0,5\text{V}$, la sortie de l'émetteur est fixée à la masse par une diode avec une impédance série de $1 \text{ k}\Omega$. Les récepteurs convertissent les signaux en niveaux logiques CMOS. Les seuils d'entrée des récepteurs garantis à $0,8$ et $2,4\text{V}$ sont plus ajustés que les seuils de $\pm 3\text{V}$ requis par la spécification EIA/TIA-232E. Ceci permet aux entrées des récepteurs de répondre aux niveaux logiques TTL et CMOS, aussi bien qu'aux niveaux RS232. La tension de seuil au niveau bas garantie de $0,8\text{V}$ assure que le récepteur, dont l'entrée est court-circuitée à la masse, a une sortie avec un niveau logique haut. Les entrées des récepteurs

possèdent un hystérésis de $0,5\text{V}$ environ ; ceci fournit des transitions en sorties propres, même avec des signaux avec des fronts de montée et de descente lents avec une quantité modérée de bruit et de sur-modulation.

Dans le mode faible consommation, les pompes à charge sont coupées, V_+ est rappelé à V_{CC} , V_- est rappelé à la masse et les sorties des émetteurs sont dévalidées. Ceci réduit le courant d'alimentation typique à $15 \mu\text{A}$. Le temps nécessaire pour sortir du mode basse consommation est inférieure à 1 ms . Tous les récepteurs du MAX213 sont mis en état de haute impédance ; sa broche de contrôle de validation est utilisée pour mettre les sorties du récepteur en état de haute impédance et permet une connexion en OU câblée entre deux ports EIA/TIA-232E ou des ports de types différents à l'UART, mais elle n'a pas d'effet sur les commandes RS232 ou les pompes à charge.

La broche de contrôle est active au niveau haut tandis que la broche de mise dans le mode basse consommation est active au niveau bas pour le MAX213. Le temps de propagation du récepteur du MAX213 est de $0,5 \mu\text{s}$ en opération normale. Dans le mode basse consommation, le temps de propagation augmente à $4 \mu\text{s}$ à la fois pour les transitions montantes et descendantes. Les entrées des récepteurs du MAX213 ont un hystérésis d'environ de $0,5\text{V}$, sauf dans le mode de basse consommation. Lorsqu'il entre dans le mode faible consommation avec des récepteurs actifs, ce mode ne devient actif que $80 \mu\text{s}$ après que le mode basse consommation soit validé. Lorsque le mode basse consommation

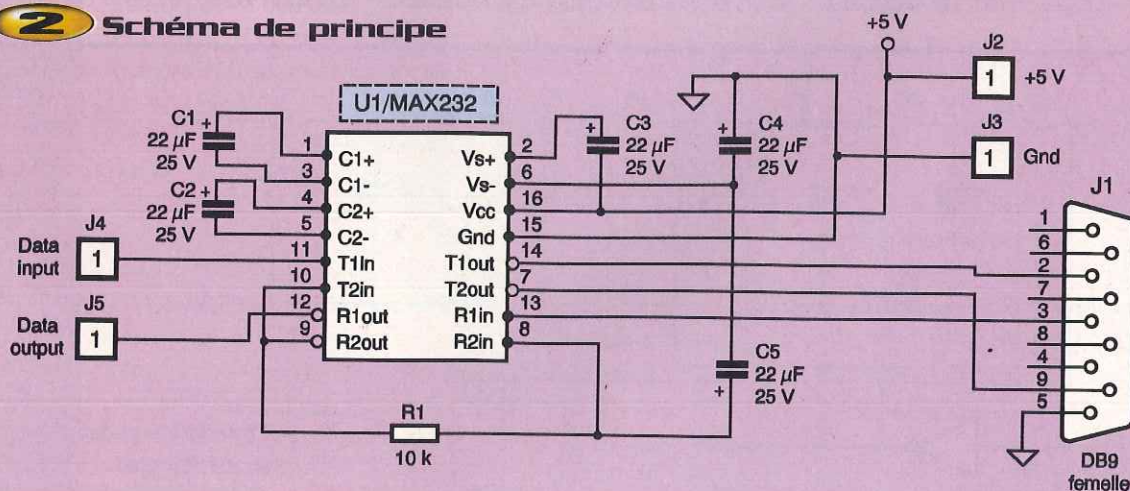


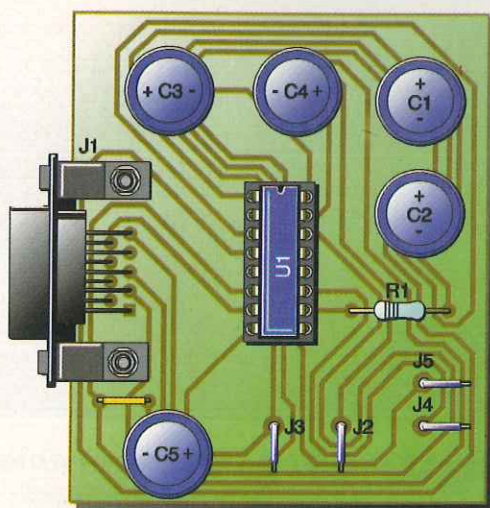
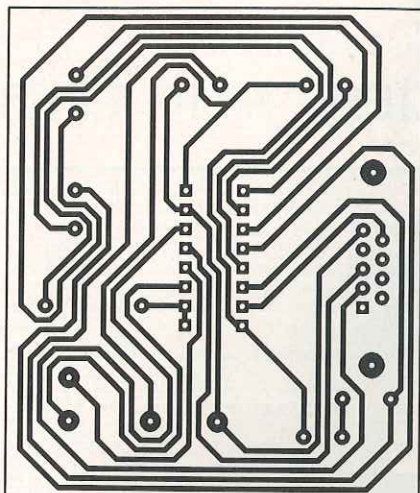
est terminé, toutes les sorties des récepteurs sont invalides jusqu'à ce que les pompes à charge atteignent les niveaux de tension nominale (moins de 2 ms). Comme avec tous les composants du constructeur MAXIM, les structures de protection contre les décharges électrostatiques sont incorporées sur toutes les broches pour se protéger contre les ESD rencontrées durant le maniement et l'assemblage. Les sorties des émetteurs et l'entrée des récepteurs ont des protections supplémentaires contre l'électricité statique. Les ingénieurs de chez MAXIM développent des structures de l'état de l'art pour protéger ces broches contre les décharges électrostatiques de +/- 15 kV sans dommage. Les structures ESD résistent aux décharges dans les cas suivants : opération normale, mode basse consommation et mode dévalidation des sorties. Après un événement électrostatique, ces circuits intégrés restent en fonctionnement sans blocage, là où des produits RS232 de la concurrence peuvent se verrouiller et doivent être mis hors tension pour enlever ce blocage. La protection ESD peut être testée de plusieurs manières différentes. Les sorties des émetteurs et les entrées des récepteurs de ce produit sont caractérisées pour la protection des limites suivantes : +/- 15 kV en utilisant le modèle du corps humain ; +/- 8 kV en utilisant la méthode de décharge par contact spécifié par la norme IEC1000-4-2 ; +/- 15 kV en utilisant la méthode de transmission par l'air spécifié par la norme IEC1000-4-2. La performance de l'ESD dépend d'une variété de conditions. Le modèle du

corps humain consiste en une capacité de 100 pF chargée à la tension ESD désirée, qui est ensuite déchargée dans le circuit intégré à travers une résistance de 1,5 k Ω . Le standard IEC1000-4-2 couvre le test ESD et les performances de l'équipement final ; il ne se réfère pas spécifiquement au composant lui-même. Le composant de chez MAXIM aide le concepteur à concevoir des équipements qui répondent au niveau 4 (le plus élevé) de la spécification IEC1000-4-2, sans avoir à ajouter d'autres composants pour la protection ESD. La différence majeure entre les tests effectués avec le modèle du corps humain et le standard IEC1000-4-2 est un courant crête plus élevé pour ce dernier, car les résistances en série sont plus faibles dans ce cas. Ainsi, la tension de régime ESD mesurée avec la spécification IEC1000-4-2 est généralement plus basse que celle mesurée en utilisant le modèle du corps humain. Le test, par transmission par l'air, consiste à l'approche de composant avec une sonde chargée. La méthode de décharge par contact connecte la sonde au composant avant que la sonde soit stimulée. Le modèle de la machine pour les tests ESD sur les broches utilise une capacité de stockage de 200 pF et une résistance de décharge de valeur zéro. Son objectif est d'émuler le stress causé par le contact qui se produit avec le maniement et l'assemblage durant la fabrication. Bien sûr, toutes les broches nécessitent cette protection durant la fabrication, pas seulement les entrées et sorties RS232. Par conséquent, après l'assemblage d'une carte pour ordinateur,

le modèle de la machine est moins pertinent sur les ports d'entrée/sortie. Le type de capacité utilisée pour C₁-C₄ n'est pas critique pour un fonctionnement correct. Le MAX232 nécessite des capacités de valeur 0,1 μ F tandis que la MAX232 nécessite des capacités de valeur 1 μ F, bien que dans tous les cas des capacités de valeurs supérieures à 10 μ F peuvent être utilisées sans nuire au bon fonctionnement. Des capacités en céramique, aluminium, électrolytique ou tantale sont suggérées pour la valeur de 1 μ F. Des diélectriques en céramique sont suggérés pour la valeur de 0,1 μ F. Lorsque le concepteur utilise les valeurs minimales recommandées, il faut être sûr que la valeur de la capacité n'entraîne pas de dégradations excessives lorsque les températures de fonctionnement varient. Dans le doute, il faut utiliser des capacités avec une valeur plus grande que la valeur nominale (le double au moins). La résistance série effective de la capacité, qui augmente habituellement aux températures basses, influence la quantité d'ondulations sur V+ et V-. Il faut utiliser des valeurs de capacité plus élevées pour réduire l'impédance de sortie sur V+ et V-. Il est recommandé de découpler VCC à la masse avec une capacité de valeur au moins égale à 0,1 μ F. Dans les applications sensibles au bruit dans les tensions d'alimentation générées par les pompes à charge, il faut découpler VCC par rapport à la masse avec une capacité de la même valeur que celle des capacités des pompes à charge. Une petite quantité de puissance peut être tirée de V+ et V-, bien que ceci réduit à la fois l'oscillation des

2 Schéma de principe





Nomenclature

- U₁ : MAX232 + support CI DIP16
- J₁ : connecteur DB9 femelle pour circuit imprimé
- R₁ : 10 kΩ 1/4W (marron, noir, orange)
- C₁ à C₅ : 22 µF/25V radial
- J₂ à J₅ : connecteurs 1 point

3 4

Tracé du circuit imprimé et Implantation des éléments

émetteurs de sortie et les marges de bruit. L'augmentation des valeurs des capacités des pompes de charge aide à maintenir de bonnes performances lorsque de la puissance est tirée de V+ et V-. Chaque émetteur est conçu pour commander un seul récepteur. Des émetteurs peuvent être mis en parallèle pour commander de multiples récepteurs. Ces émetteurs maintiennent les tensions de sortie minimale RS232 +/-5,0V avec un taux de transfert de 120 kbauds. Le translateur RS232 vers TTL employé dans notre application possède une caractéristique spéciale qui est une hystérésis de, approximativement, 0,5V. Avec cette particularité, une porte classique avec oscillation peut être créée. Cet oscillateur peut être construit en utilisant les translateurs disponibles du MAX232. Comme le représente le schéma de notre montage à la **figure 2**, la sortie de l'oscillateur est connectée à la broche 9 de la prise série

DB9 ; cette dernière broche remplit la fonction de signal d'indicateur de sonnerie sur la prise RS232.

Maintenant, nous allons nous intéresser au programme à ajouter à l'application qui tourne sur l'ordinateur. La fonction à remplir nécessite de détecter si l'oscillation est présente sur l'entrée d'indicateur de sonnerie du port série qui est scruté. Pour réaliser cette opération, le programme commence à scruter le contrôleur de chaque port série (qui est en général un circuit intégré basé sur le 8250 ou un UART), teste si l'oscillation décrite est présente en lisant le bit d'indicateur de sonnerie dans le registre d'état du modem (MSR : Modem Status Register). Lorsque cette oscillation est détectée, le périphérique a été détecté et est prêt. La fonction Search Oscillator() compte le nombre de secondes avant qu'un front sur le signal d'entrée d'indicateur de sonnerie soit produit en testant le bit 2 du registre MSR. Ce bit, appelé TERI

(Tailing Edge of Ring Indicator ou encore arrière du front de l'indicateur de sonnerie), indique que l'entrée d'indicateur de sonnerie est passée du niveau bas au niveau haut. Ce comptage est sauvegardé dans la variable ringCount et doit être égal à la fréquence de l'oscillation.

Une bande de détection de fréquence (F_MAX, F_MIN) est créée pour réaliser la détection de l'oscillation de manière plus

efficace. L'intervalle de temps pour le comptage de ce front est indiqué en utilisant la variable globale de drapeau searchTimeout ; cette dernière est mise à TRUE dans la routine d'interruption du compteur lorsque le temps désiré (une seconde approximativement) est passé. Cette fonction est aussi utile pour vérifier si le périphérique série est alimenté ou non. La fonction DetectSerialDevice() est utilisée pour recueillir les ports séries possibles qui sont disponibles sur l'ordinateur ; cette fonction retourne l'adresse de base d'entrée/sortie du périphérique qui a été détecté à travers la signature de la fréquence de l'oscillateur. Des protocoles de plus hauts niveaux peuvent être utilisés (avec les lignes RX et TX) lorsque le périphérique série est identifié, créant une simulation de système Plug & Play.

Réalisation pratique

Le câblage de notre circuit ne pose aucune difficulté particulière. Il ne faut pas oublier de commencer par câbler l'unique strap de ce montage. Il est bien sûr recommandé de mettre le MAX232 sur support au cas où une mauvaise manipulation interviendrait. La **figure 3** représente le circuit imprimé côté cuivre et la **figure 4** l'implantation des composants.

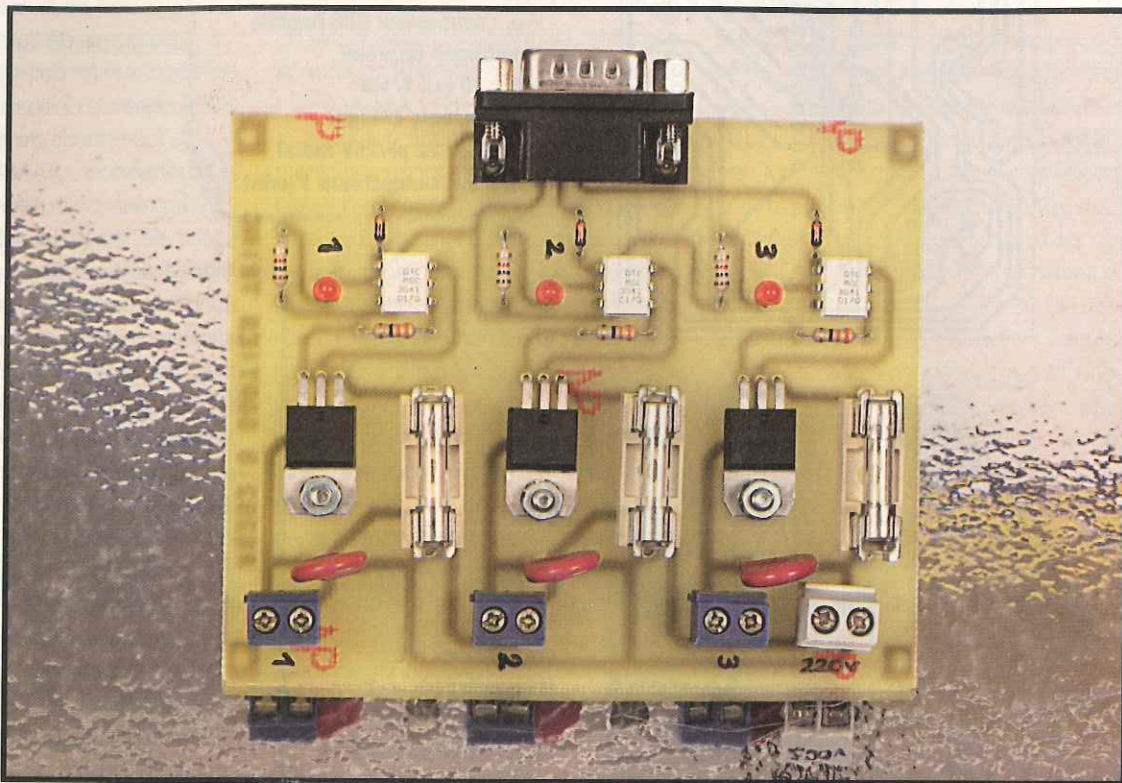
Conclusion

L'approche présentée dans cet article a été utilisée avec succès dans plusieurs réalisations. Cette application est facile à implémenter, apporte peu de pénalités au niveau du prix de revient et élimine la confusion pour l'utilisateur et les erreurs lors de la configuration des périphériques séries aux ports COM d'un ordinateur de type PC.

M. LAURY

Port série

3 sorties sur triacs



Voici un petit montage sans prétention mais qui vous rendra sûrement de précieux services. Il permet facilement de piloter trois charges alimentées en courant alternatif, à partir du port série disponible sur n'importe quel ordinateur. Deux logiciels développés sous DELPHI vous sont proposés, permettant de piloter le montage à partir de WINDOWS.

Schéma électrique (figure 1)

La commande de notre carte s'effectue par trois sorties du port RS232 : DTR (broche n° 4), TxD (broche n° 3) et RTS (broche n° 7). L'utilisation de la tension du secteur nous conduit à réaliser une isolation galvanique afin de protéger correctement l'ordinateur. Cette isolation est effectuée à l'aide d'un optocoupleur (du type MOC3041), un tel circuit se compose de deux parties distinctes (isolation galvanique de 7500V) : la première est constituée d'une diode infrarouge qui va venir mettre en conduction le triac contenu dans la deuxième partie. Il est également muni d'un dispositif qui détecte le passage à zéro de la tension du secteur afin d'éviter de générer des parasites lors de l'alimentation de la charge. Le courant de l'ordre de 10mA, nécessaire à l'activation de la diode infrarouge, est généré par la sortie du port série. La limitation de l'intensité est assurée par

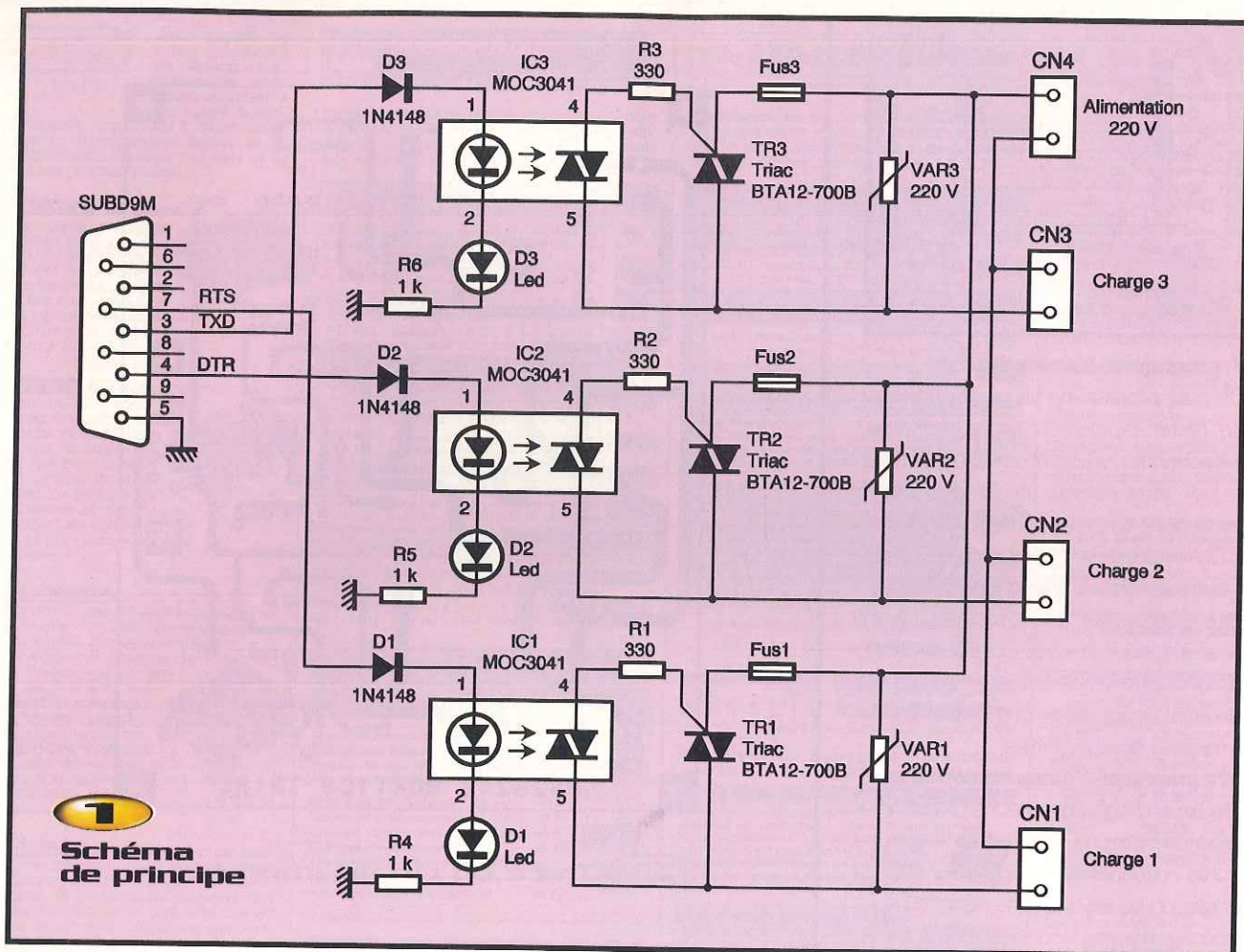
une résistance de 1 k Ω . La diode électroluminescente mise en série permet de valider visuellement l'état de la diode interne au circuit. Par précaution, la diode 1N4148 permet d'annuler le courant inverse bien que la LED de l'optocoupleur et la LED de signalisation peuvent sans problème supporter une tension inverse de 10V. Outre la sécurité offerte par les optocoupleurs, leur utilisation nous permet de se passer d'une alimentation en courant continu. En effet, la diode de commutation est alimentée par le port série et le triac par la tension secteur. Donc, nul besoin de transformateur et autre régulateur de tension.

La faible puissance du triac interne à l'optocoupleur ($I_{MAX}=100mA$) ne permet pas une alimentation directe d'une charge importante. Un deuxième triac, mis en cascade, permet de disposer d'une puissance beaucoup plus importante. Celui utilisé ici est un BTA 08-700B en boîtier TO220. Il peut fonctionner sous une tension alternative max. de 700V et débiter un cou-

rant de 40A !. Si vous utilisez le montage pour piloter des charges inférieures à 100W (ex : ampoules électriques), on peut se passer d'un dissipateur, dans le cas contraire celui-ci devient obligatoire afin d'éviter la destruction du triac.

Le composant référencé VAR est une varistance qui permet de protéger le montage lors du pilotage d'une charge inductive, les phénomènes d'auto-induction, lors de l'établissement et la coupure du courant, peuvent détériorer le triac. Chaque triac possède donc une varistance montée en parallèle. Ce 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.

Pour finir, un fusible vient compléter la protection de notre montage, son calibre sera déterminé en fonction du courant maximum absorbé par la charge. Par exemple, avec une charge de 100W alimentée sous une



1
Schéma de principe

tension de 220V, vous devez utiliser un fusible ayant un calibre supérieur à 0,5A.

Réalisation

Vous trouverez **figures 2 et 3** le tracé et le schéma d'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.

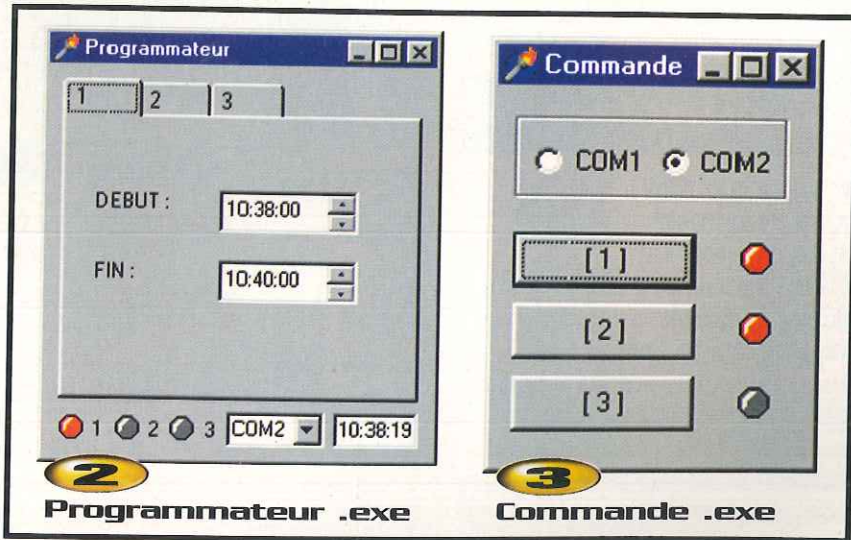
Avant de réaliser la carte, assurez-vous que la taille des supports des fusibles correspond à l'implantation du tracé.

Si vous décidez d'utiliser des dissipateurs thermiques, vérifiez avant de les acheter qu'ils ont la place nécessaire pour être placé sur le circuit. Cependant, vous avez la possibilité de les déporter, un cordon de 3 conducteurs permet alors d'effectuer la liaison au montage. La connexion de la carte au port série pourra s'effectuer à l'aide d'un cordon comportant 4 conducteurs auquel on viendra souder deux connecteurs au format DB9 femelle (**voir figure 4**).

Logiciels

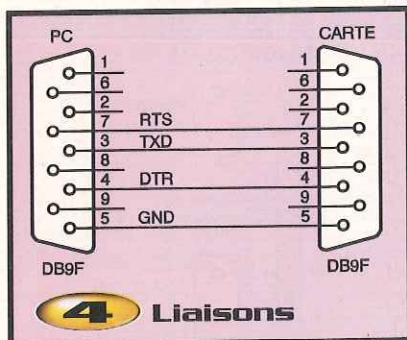
Deux logiciels, développés sous DELPHI 4, vont vous permettre de piloter les trois sorties d'un simple click de souris. Le protocole de dialogue RS232 n'est pas utilisé. On vient piloter chaque sortie en écrivant directement dans les registres du port série. La ligne TxD correspond au bit n°6 du registre de commande de

ligne (adresse COM+3). La ligne DTR (bit 0) et la ligne RTS (bit 1) sont accessibles dans le registre de commande du modem (adresse COM+4). La partie du programme pilotant les différents registres du port série est directement écrite en assembleur. En effet, contrairement aux premières versions de DELPHI, la commande PORT[adresse_registre] :=valeur n'existe plus.



2
Programmeur .exe

3
Commande .exe



1^{er} programme (commande.exe)

Ce petit programme va vous permettre de tester rapidement votre montage. Sélectionnez tout d'abord le port sur lequel vous désirez utiliser la carte. A l'ouverture du programme, c'est le port COM1 qui est par défaut activé. Ensuite, il suffit de cliquer sur un des trois boutons pour activer (ou désactiver suivant l'état antérieur) la sortie correspondante, un petit voyant signale la prise en compte de votre action (gris = charge désactivée / rouge = charge activée).

2^{ème} programme (programmeur.exe)

Ce second logiciel permet de programmer l'état de chacune des sorties. A chaque sortie correspond un onglet. Sur chaque onglet on trouve une zone début où vous spécifiez l'heure à laquelle la sortie doit s'activer, une zone fin où vous spécifiez l'heure à laquelle la sortie doit se désactiver. L'heure du système est affichée au bas de la fenêtre ainsi que trois voyants qui signalent l'état de chacune des sorties. Une zone liste vous permet de sélectionner le port de communication.

Remarque : pour obtenir des informations complémentaires, l'auteur vous invite à visiter son site Internet :

http://site.voila.fr/David_REY

D. REY

Nomenclature

R₁ à R₃ : 330 Ω 1/4W

R₄ à R₆ : 1 kΩ 1/4W

d₁ à d₃ : 1N4148

L₁ à L₃ : LED Ø3mm

IC₁ à IC₃ : MOC3041

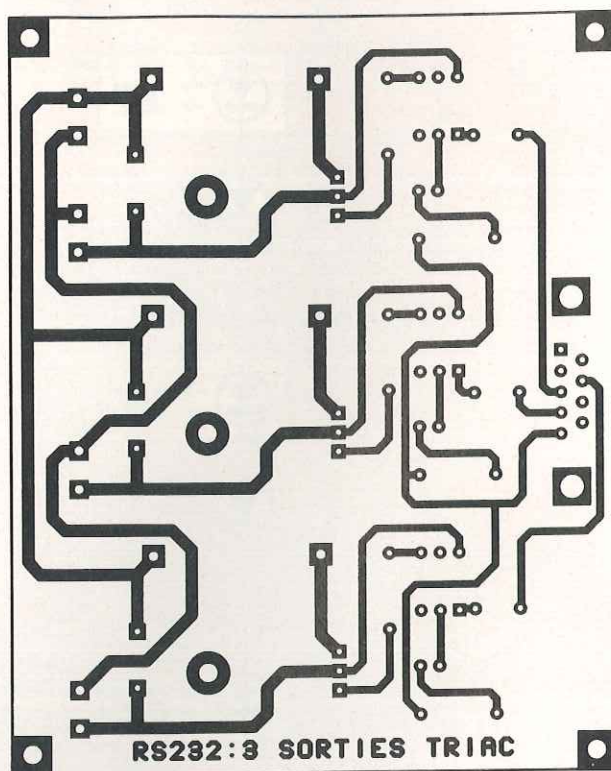
TR₁ à TR₃ : BTA 12-700B

Fus₁ à Fus₃ : porte-fusibles + fusibles

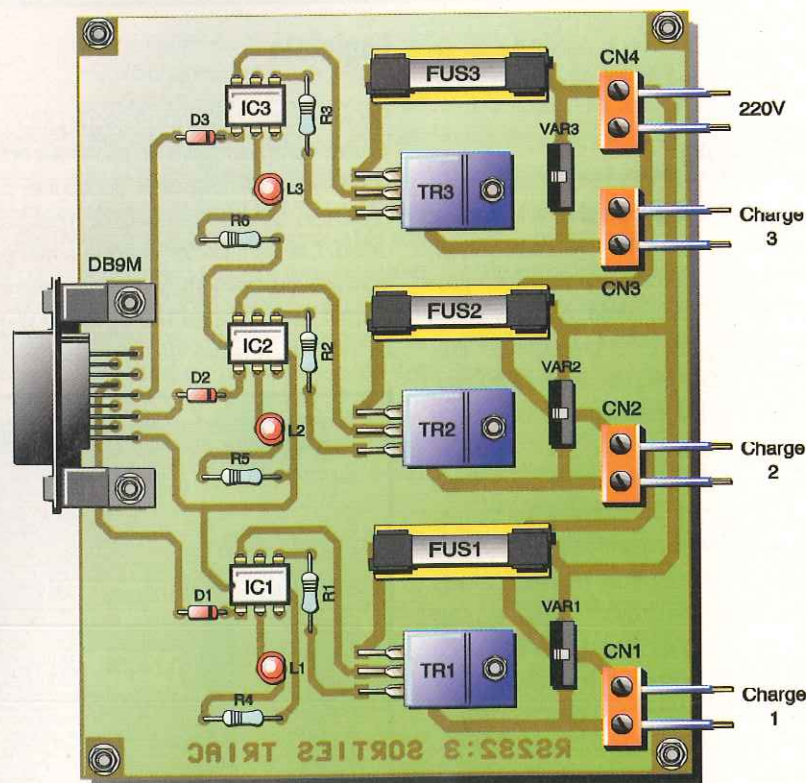
VAR₁ à VAR₃ : varistances 220V

CN₁ à CN₄ : bornier à vis 2 plots

SUBD9M : connecteur SUBD 9 brochs mâle, coudé 90°

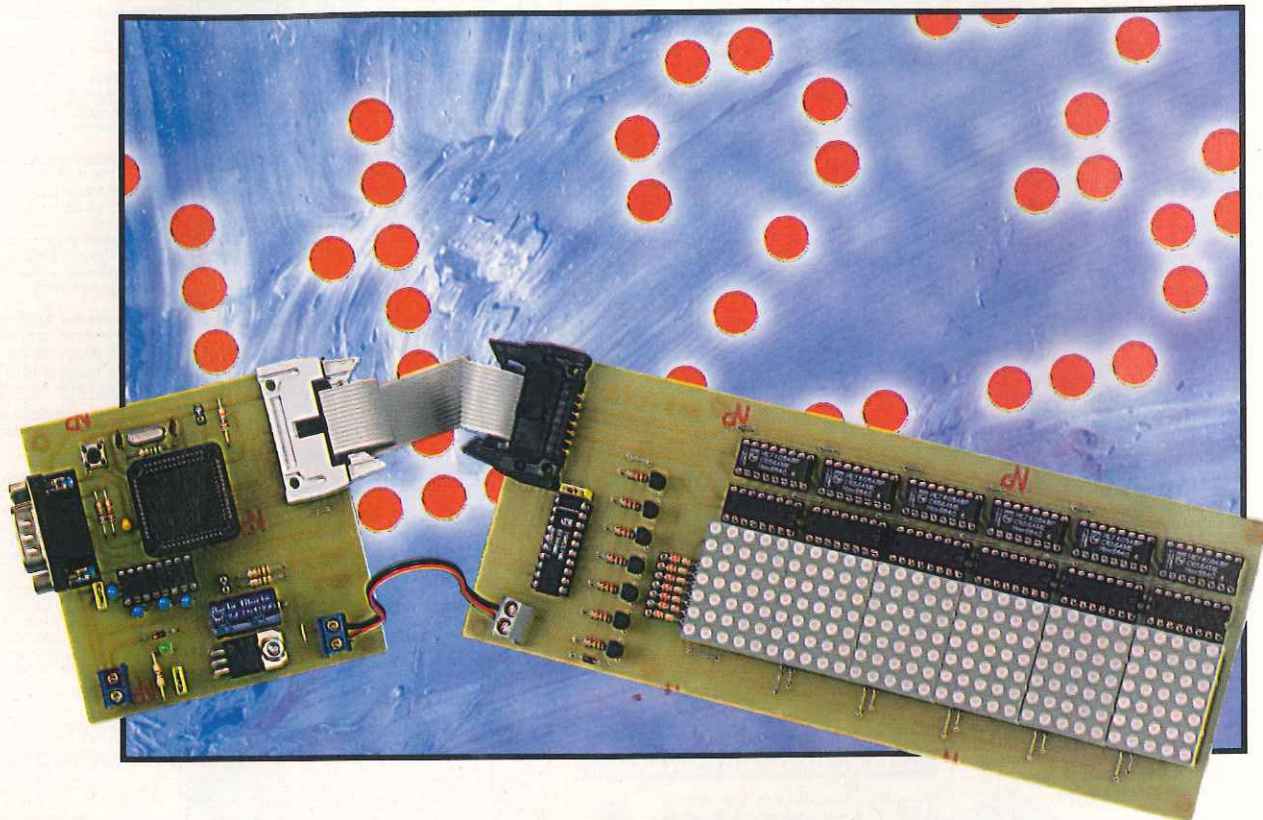


2 Tracé du circuit imprimé



3 Implantation des éléments

Afficheur de message à LED



Les afficheurs à LED permettent la diffusion attractive d'un message en continu. Celui proposé ici est capable d'afficher simultanément 6 caractères, dispose d'un défilement horizontal dont la vitesse est paramétrable, ce qui permet une lecture aisée du message. Le texte à diffuser, pouvant atteindre 378 caractères, est sauvegardé même en cas de coupure de l'alimentation.

Les afficheurs à LED permettent la diffusion attractive d'un message en continu. Celui proposé ici est capable d'afficher simultanément 6 caractères, dispose d'un défilement horizontal dont la vitesse est paramétrable, ce qui permet une lecture aisée du message. Le texte à diffuser, pouvant atteindre 378 caractères, est sauvegardé même en cas de coupure de l'alimentation.

Un logiciel utilisable sous WINDOWS permet de définir votre propre police de caractère et de composer le message à afficher qui est ensuite transmis au montage par une liaison série utilisant le protocole RS232.

Le montage se compose de deux modules distincts : le module d'affichage proprement dit et le module de commande architecturé autour d'un microcontrôleur de type 68HC11.

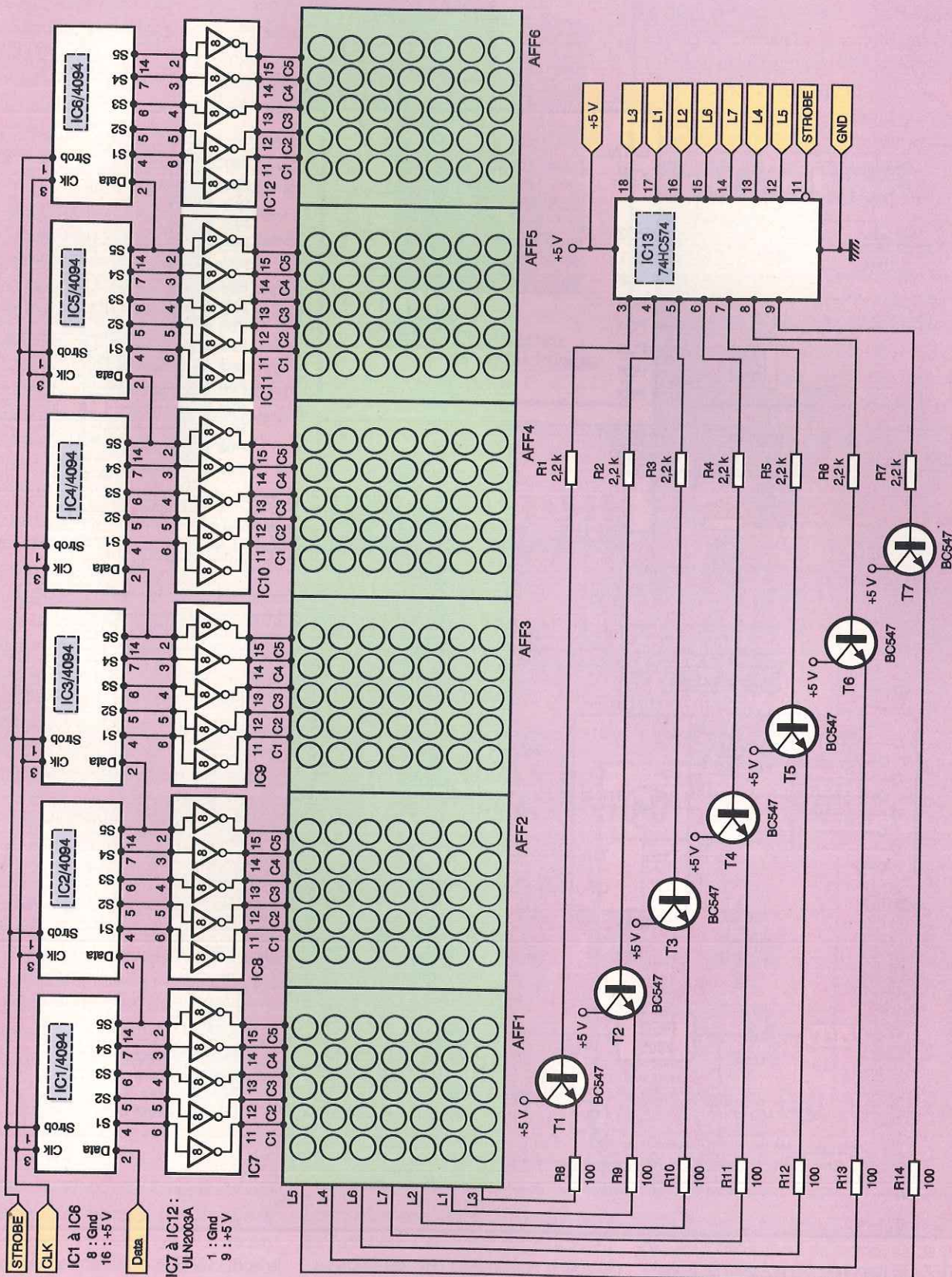
Module d'affichage (figure 1)


Pour simplifier la réalisation, des afficheurs matriciels à LED 7x5 sont utilisés, ce qui évite de souder les LED une par une. Le module comporte 6 afficheurs de 35 LED chacun, soit un total de 210 LED, imaginez s'il fallait les souder à la main ! Les matrices utilisées sont des LTP-1557AE de couleur rouge, les lignes sont à cathode commune. La hauteur du boîtier est de 30,4mm ce qui permet une lecture confortable jusqu'à une dizaine de mètres. Pour piloter notre matrice d'affichage qui comporte donc au total : 30 colonnes et 7 lignes, il faut théoriquement disposer de 37 sorties, allez donc trouver un ordinateur ou un microcontrôleur disposant de 37 sorties ! L'astuce consiste à multiplexer l'affichage, en d'autres termes il suffit d'afficher chacune des colonnes une par une, mais

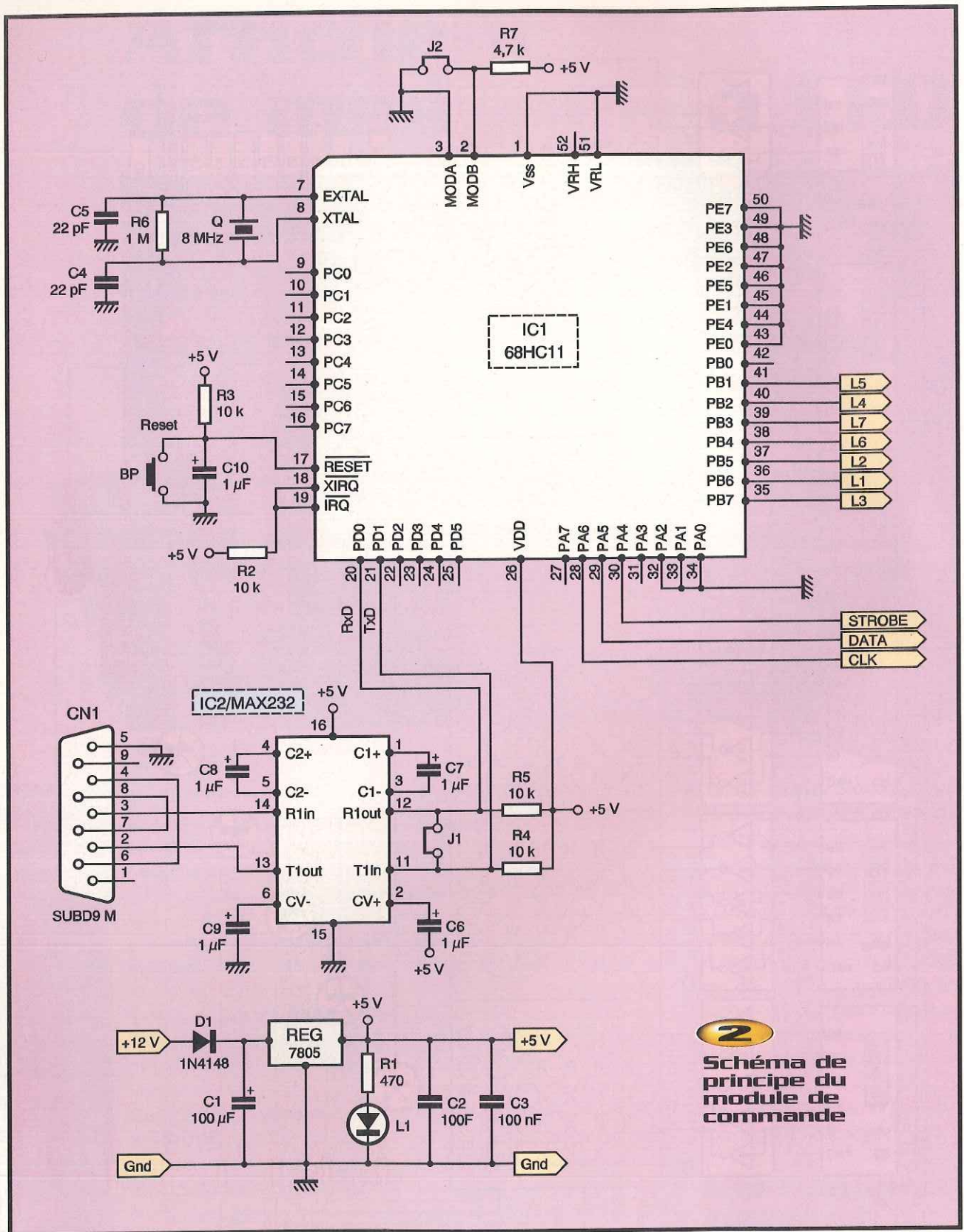
dans un délai de quelques millisecondes de sorte que l'œil humain à l'impression d'un affichage continu. C'est le rôle attribué aux 6 registres à décalages de la carte (IC₁ à IC₆ : CD4094). Sachant qu'un seul registre comporte 8 sorties cela nous donne un total de 48 sorties, ce qui est plus que suffisant sachant que 4 auraient suffi, mais ce choix de conception permet de simplifier grandement le tracé du circuit. Ce choix est d'autant plus justifié vu le prix relativement peu élevé d'un registre à décalage.

Rappel sur les registres à décalage

Le principe d'un registre à décalage est de transformer des données série sous forme parallèle. Les trois entrées qui permettent de piloter un tel circuit sont : CLOCK (broche n°3) qui est l'entrée d'horloge, DATA (broche n°2) l'entrée de donnée et



 Schéma du module d'affichage



2
Schéma de principe du module de commande

STROBE (broche n°1) qui permet le verrouillage des sorties. A chaque front montant présenté sur l'entrée d'horloge, l'ensemble des bits contenus dans un registre interne au 4094 sont décalés d'un bit vers la droite, et l'état logique alors appliqué sur l'entrée DATA est reco-

pié dans le premier bit du registre. Comme il s'agit d'un registre 8 bits, au bout de 8 fronts montants l'entrée DATA aura été lue 8 fois et le registre sera alors plein. Si on continue le processus, le registre déborde et le bit D7 passe alors sur la sortie CARRY. Il suffit donc de relier

la sortie CARRY sur l'entrée DATA d'un second registre à décalage pour passer le nombre de sorties à 16. Les entrées CLOCK des deux circuits sont alors mises en commun, de même que les entrées STROBE qui servent à basculer les données présentes dans le registre

interne au 4094 vers le tampon de sortie. Les sorties garderont leur niveau logique jusqu'à la prochaine action sur l'entrée STROBE.

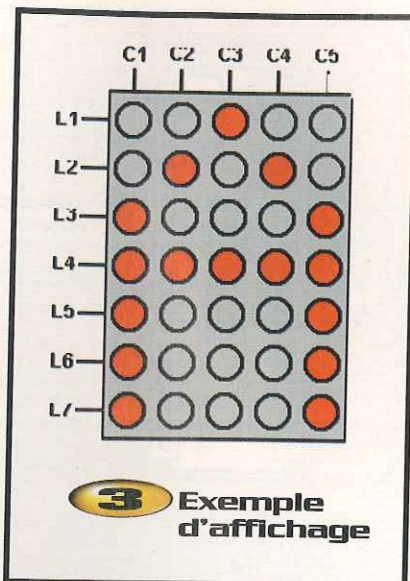
On peut ainsi mettre en série autant de registres à décalage que de sorties désirées, sachant que l'ajout d'un 4094 augmente de 8 le nombre de sorties.

Dans notre application, un registre à décalage pilote un afficheur soit 5 sorties. En conséquence, chaque registre déborde sur sa sortie n°5, celle-ci est donc reliée en série sur l'entrée DATA du registre suivant (les sorties 6, 7, 8 et carry ne sont pas utilisées).

Le processus se déroule ainsi : au premier front montant sur l'entrée CLOCK, un niveau logique haut est injecté sur l'entrée DATA du premier registre (IC1). Après l'envoi d'un front montant sur STROBE, la colonne 1 (C1) de l'afficheur est active. Ensuite l'entrée DATA est remise à zéro et reste dans cet état durant toute la période d'affichage. Chaque nouveau front montant sur CLOCK fait décaler le niveau logique haut sur les sorties des 6 registres à décalages (une seule ligne sur les 30 est active simultanément). Il faudra donc au total 30 fronts montants sur CLOCK pour balayer les 30 colonnes que comporte l'afficheur. Rappelons qu'à chaque coup d'horloge, il faut un front montant sur l'entrée STROBE pour réaliser la mise à jour des sorties du registre à décalage. Lorsqu'une colonne est active, il faut simultanément présenter sur les 7 lignes des états logiques cohérents pour faire afficher un caractère. Un caractère est affiché lorsque 5 colonnes sont soumises une par une au niveau haut. Chaque colonne est codée sur un octet (le bit n°7 n'est pas utilisé), le bit 0 définit l'état de la ligne 1, le bit 1 l'état de la ligne 2... le bit 6 celui de la ligne n°7. Un caractère nécessitant 5 colonnes, il faut donc 5 octets pour sa codification.

Par exemple pour faire afficher la lettre A (voir **figure 3**) sur la première matrice, il faut envoyer la donnée 124 lorsque la première colonne est active, 10 lorsque la colonne 2 est active, 9 pour la colonne 3, 10 pour la 4 et 124 pour la 5ème. Si ce principe est valable pour un caractère, il est valable pour l'affichage simultané des 6 caractères.

Le processus que nous venons de décrire concerne l'affichage statique,



3 Exemple d'affichage

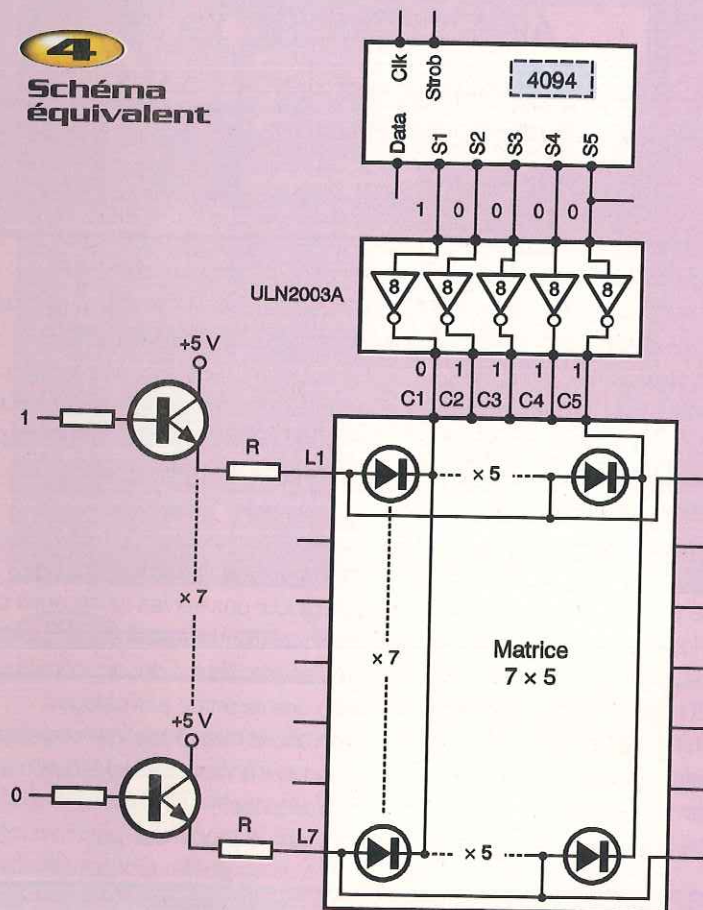
colonne devient la 2ème, la première étant remplacée par la suite du texte à afficher. On réalise le processus d'affichage et de décalage autant de fois qu'il y a de colonnes à afficher.

Si le texte à afficher comporte 80 caractères, il faudra coder 80x5=400 colonnes pour pouvoir l'afficher et réaliser 400 fois le décalage pour faire défiler la totalité du texte.

Ce principe de défilement est très fluide car l'affichage ne se décale pas caractère par caractère mais par morceau de caractère (un morceau correspondant à une colonne).

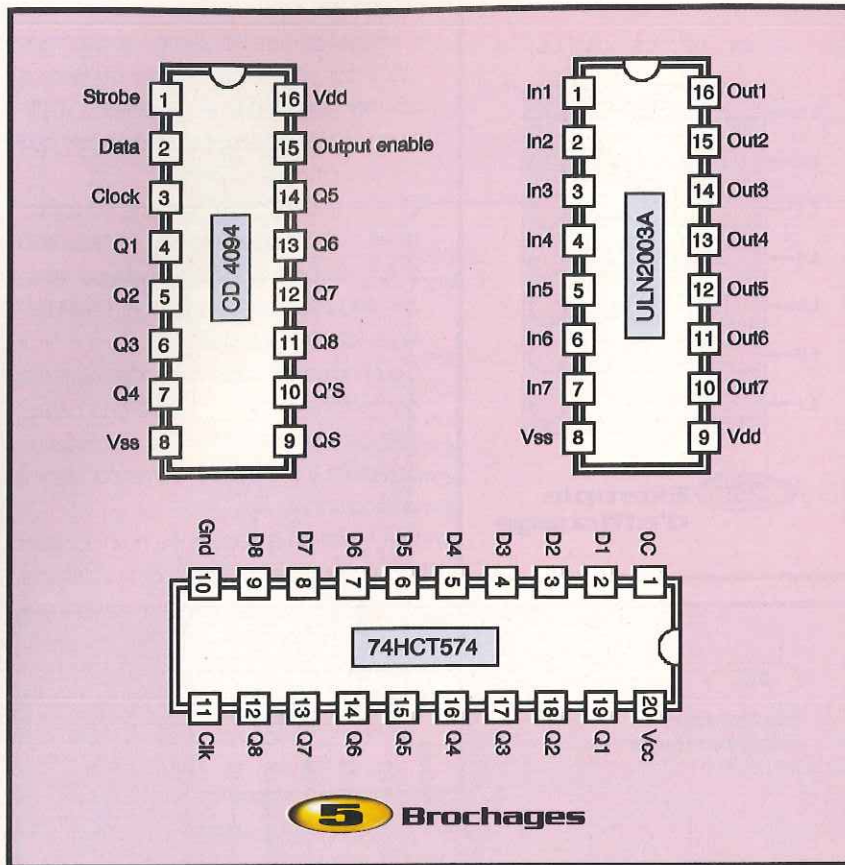
Les registres à décalages ne disposant pas d'une puissance de sortie suffisante

4 Schéma équivalent



mais il reste encore à faire défiler le texte sur l'afficheur. Ceci n'est pas plus compliqué : après l'affichage des 6 premiers caractères, soit un total de 30 colonnes affichant chacune un octet, il suffit de réaliser un décalage : le texte est décalé d'une colonne vers la gauche. La dernière colonne est éjectée, la 29ème devient la dernière, ainsi de suite, la première

pour alimenter les LED, il faut utiliser un circuit ULM7803 qui se compose de 8 transistors (seuls 5 sont utilisés ici), mais point important : ils sont inverseurs. La puissance délivrée par chacune des 7 lignes est assurée par un transistor du type BC547. La base de chacun des transistors est associée à une résistance de 2,2 kΩ connectée à la sortie d'un cir-



cuit 74HC547 qui joue le rôle de tampon. Pour comprendre le fonctionnement, regardons le schéma équivalent présenté sur la **figure 4**.

Seule la sortie S1 du registre à décalage est au niveau haut. Après passage dans le circuit ULN2003A, les niveaux sont inversés. En conséquence, seule la colonne C1 est au niveau bas, les colonnes C2 à C5 étant au niveau haut. Sur la base du transistor de la ligne L1, un niveau logique haut est appliqué ce qui rend le transistor passant. Ainsi, seule la LED placée entre la colonne C1 et la ligne L1 est allumée.

On constate, d'après le schéma, que seules des matrices d'affichage avec des anodes en lignes peuvent fonction-

ner avec le principe adopté ici. Les résistances R permettent de limiter le courant envoyé à chaque LED. L'intensité nominale pour chaque LED est de 20mA, mais peut monter à 80mA avec un courant modulé et un rapport cyclique de 1/16^{ème}.

Le circuit IC₁₃ permet de mémoriser les niveaux logiques des lignes L1 à L7. La mise à jour des sorties se faisant à chacun des fronts descendants envoyés sur l'entrée Strobe qui est commune à celle des registres à décalages.

Bien que la mise à jour des registres se faisant sur un front montant, cela n'a pas d'importance.

La mise à jour n'est pas simultanée

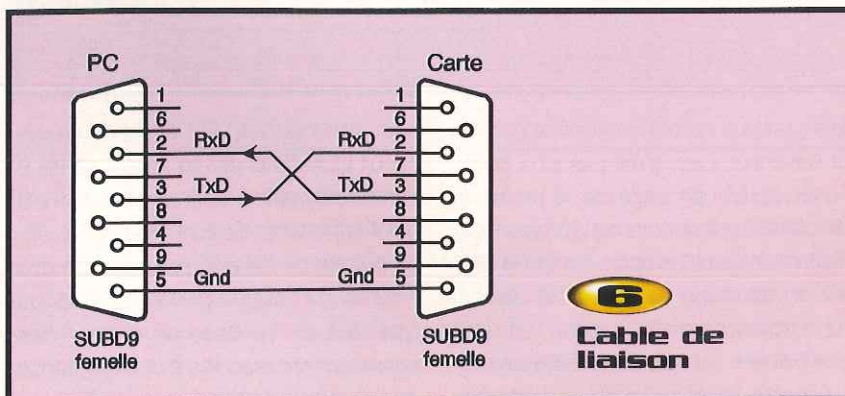
mais suffisamment faite rapidement pour que cela ne gêne en rien le fonctionnement du montage (ceci évite la mise en place d'une logique inverseuse supplémentaire).

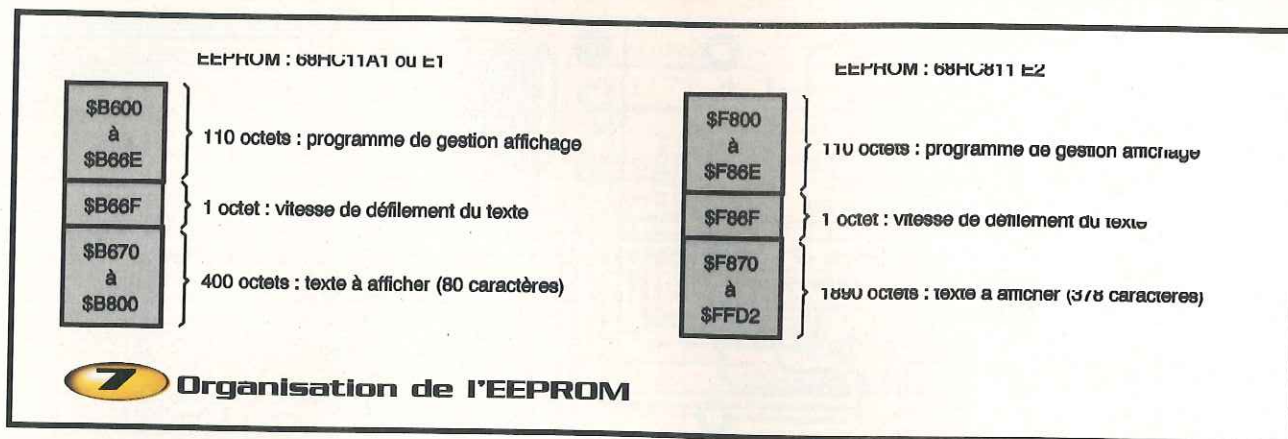
Module de commande (figure 2)

Le module de commande basé sur l'utilisation d'un microcontrôleur 68HC11 permet le pilotage de notre afficheur. Le stockage du texte à afficher est assuré par l'EEPROM interne. Avec un 68HC11A1, il est possible de stocker 80 caractères. Mais le 68HC811E2 est plus intéressant car, avec ces 2 Ko de mémoire, il est possible de stocker 378 caractères. Les deux composants sont parfaitement interchangeables, il suffit juste d'apporter une petite modification au programme. Il est même possible d'utiliser le 68HC11E1 qui dispose aussi de 512 octets d'EEPROM (comme le 68HC11A1) la seule différence, mais qui n'a aucune influence sur le fonctionnement du montage, est qu'il dispose de 512 octets de RAM alors que le A1 et E2 en disposent 256 et que sa patte PA3 dispose d'un fonctionnement particulier, mais elle n'est pas utilisée ici.

Le 68HC11 fonctionne en mode Bootstrap (MODA=0 et MODB=0), ce fonctionnement particulier permet la programmation de l'EEPROM, qui s'effectue par une liaison du type RS232. N'importe quel ordinateur peut faire l'affaire. On retrouve donc un MAX232 qui permet la mise à niveau des tensions TTL <-> RS232 avec ces 4 condensateurs indispensables au fonctionnement de la pompe de charge interne. Le cavalier J₁, lorsqu'il est en place, permet de faire redémarrer le µC après un reset lorsque le module n'est plus relié au port série du PC, ce qui rend le montage totalement autonome. En effet le cavalier relie les lignes TxD et RxD ainsi, après chaque reset, le µC s'envoie à lui-même la donnée \$00, ce qui permet de le faire exécuter le programme. En phase de programmation, il faut bien entendu retirer ce cavalier, c'est l'ordinateur qui enverra alors cette donnée d'amorçage.

Remarque : pour ceux qui voudraient utiliser le 68HC11 en mode SINGLE CHIP (MODA=0 et MODB=1), il leur suffit de





retirer le cavalier J₂. Attention ce mode d'utilisation implique une modification des programmes, il faut prévoir l'initialisation du vecteur RESET situé à l'adresse \$FFFE.

Les sorties chargées de piloter les lignes L1 à L7 sont prélevées sur le port B du µC. La ligne CLOCK est pilotée par la sortie PA6, la ligne DATA par PA5 et STROBE par PA4.

Le programme chargé de piloter l'afficheur est très condensé, seulement 110 octets. Un octet supplémentaire est utilisé pour stocker la vitesse de défilement. Pour une mémoire de 512 octets (68HC11A1), il reste donc 400 octets de libres pour stocker le texte à afficher. Sachant que pour former un caractère il faut 5 octets, il nous est possible d'afficher un texte comportant 80 caractères. Avec une mémoire de 2 Ko (68HC811E2) le nombre de caractères passe à 378 (voir organisation de l'EEPROM **figure 7**).

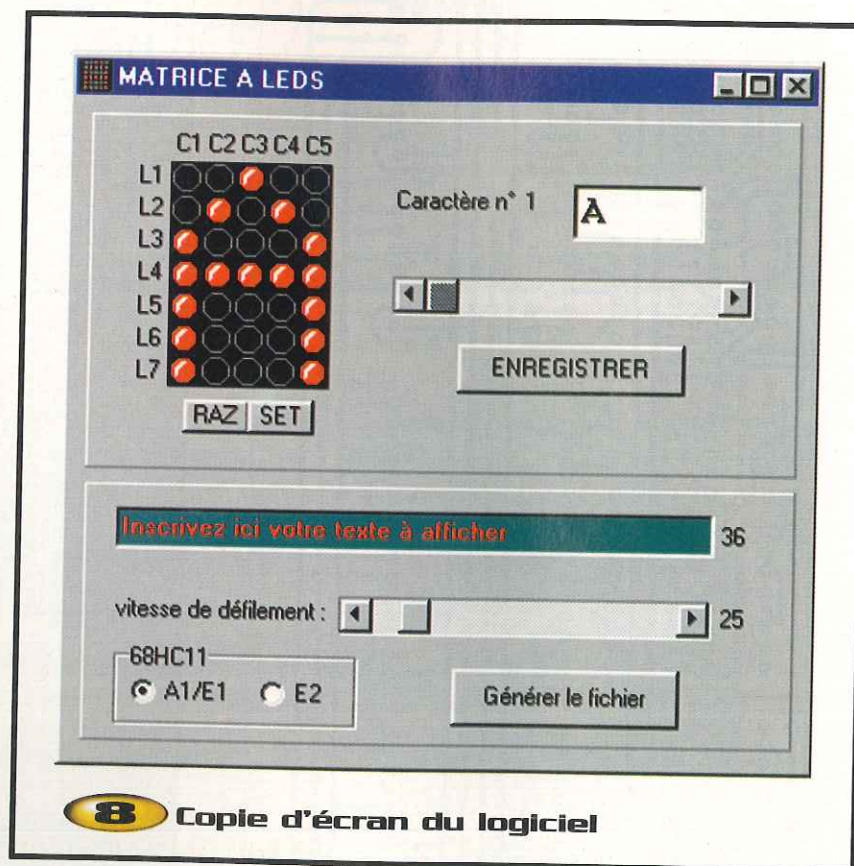
Le transfert du texte dans l'EEPROM est effectué en même temps que la programmation du µC et tout ceci grâce au logiciel PCBUG11 fourni gracieusement par MOTOROLA.

Mais avant de programmer le 68HC11, il faut bien entendu réaliser les deux modules.

Réalisation pratique-

Le tracé des modules d'affichage et de commande vous sont présentés **figures 9 et 10** et l'implantation des composants **figures 11 et 12**.

La réalisation du module d'affichage ne devrait pas poser de problème. Concernant la mise en place des composants, commencez par les straps qui sont relativement nombreux. Les affi-

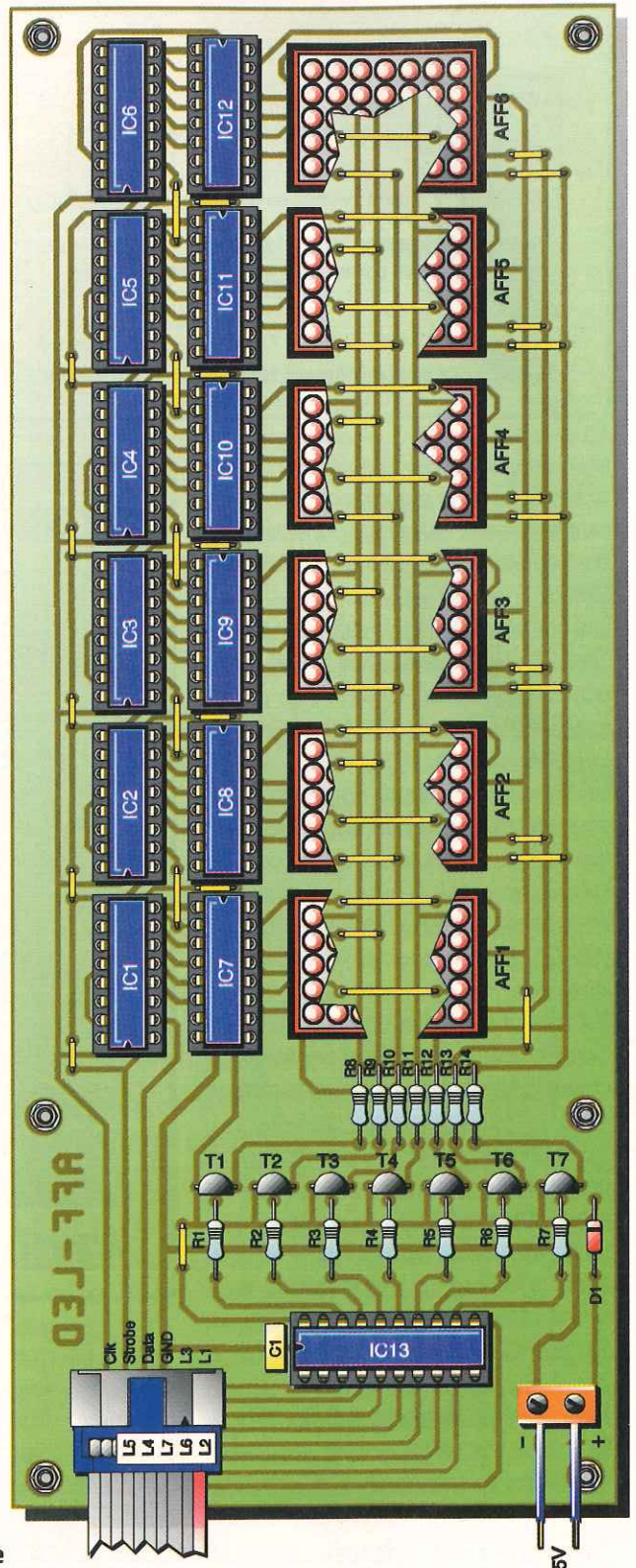
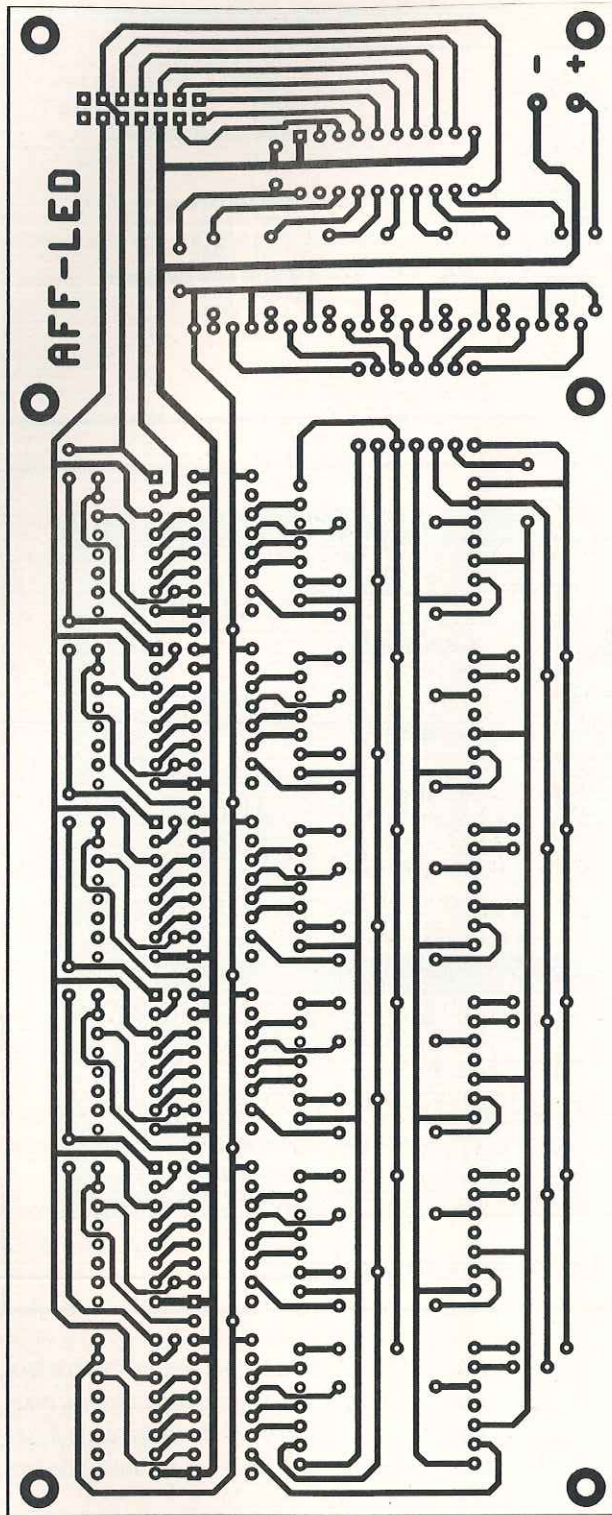


8 Copie d'écran du logiciel

cheurs pourront être soudés directement, ou mieux, positionnés sur des morceaux de barrettes tulipes sécables, la face comportant l'inscription mentionnant la référence doit être orientée vers les circuits intégrés. Les résistances R₈ à R₁₄ pourront aussi être positionnées sur deux morceaux de barrettes 7 points, ainsi il vous sera possible de les changer facilement pour des valeurs plus faibles. On peut théoriquement descendre à 50 Ω, mais attention si vous stoppez le programme en cours de fonctionnement (reset), certaines LED restent allumées et le courant n'est plus modulé ce qui peut les détériorer. Pour éviter ceci, la solu-

tion consiste à intercaler un interrupteur sur la ligne d'alimentation entre les deux modules. Avant de faire un reset, il suffit alors de couper l'alimentation de l'afficheur.

Concernant le module de commande, la partie délicate reste l'implantation du support PLCC destiné à recevoir le 68HC11. Pour le perçage, utilisez impérativement un foret neuf, dans le cas contraire vous risquez d'arracher les pastilles. Pour la mise en place du support, repérez le coin biseauté et assurez-vous ensuite que les 52 pattes traversent simultanément la plaque d'époxy, ne forcez surtout pas ! Pour la liaison des deux modules, il



9 11 Tracé du circuit imprimé et implantation des éléments du module d'affichage

vous faut un câble en nappe 14 conducteurs munis de deux connecteurs femelles au format HE10. La liaison entre le PC et le module de commande sera réalisée avec un câble comportant 3 conducteurs munis de deux connecteurs au format SUBD9 femelle (voir **figure 6**).

Programmation du 68HC11

Le programme "testA1E1.rec" va permettre à ceux qui ont choisi d'utiliser le 68HC11A1 ou le 68HC11E1 de tester l'afficheur. Le programme "testE2.rec" est réservé aux utilisateurs du 68HC811E2.

Tout d'abord, copiez les fichiers relatifs à PCBUG11 ainsi que les fichiers "testA1E1.rec" et "testE2.rec" dans un répertoire de votre disque dur, par exemple c:\afficheur. Relier le module de commande au port COM1 de votre PC. Ensuite, retirez le cavalier J₁, le cavalier J₂ doit être en placé

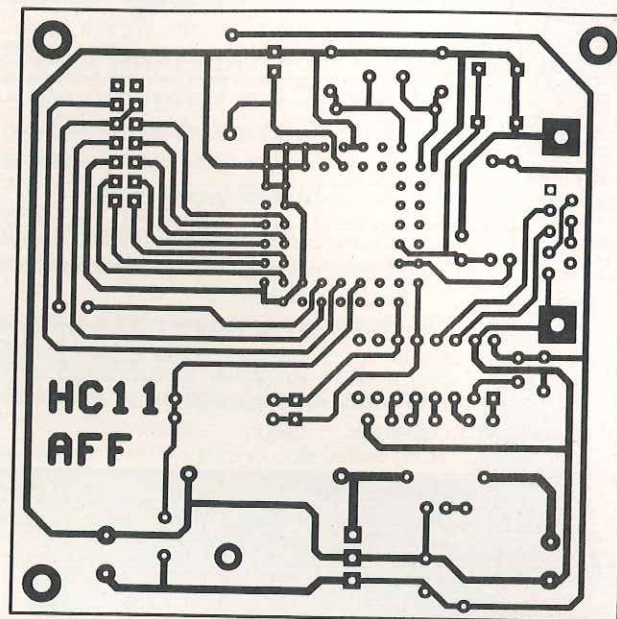
Nomenclature

Module d'affichage

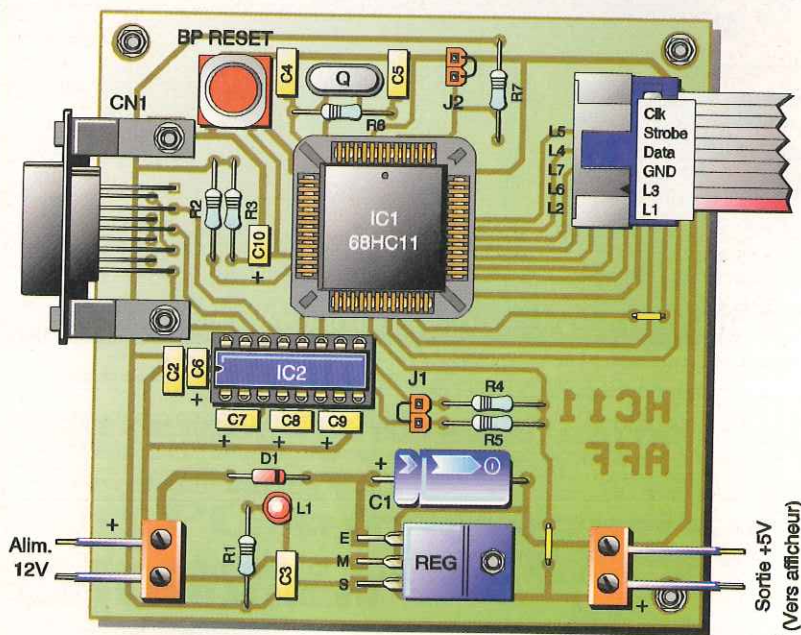
- R₁ à R₇ : 2,2 kΩ
- R₈ à R₁₄ : 100 Ω
- C₁ : 100 nF LCC jaune
- D₁ : diode 1N4148
- T₁ à T₇ : transistors BC547
- IC₁ à IC₆ : CD4094 + support 16 broches
- IC₇ à IC₁₂ : ULN2003A + support 16 broches
- IC₁₃ : 74HC574 + support 20 broches
- AFF₁ à AFF₆ : afficheurs matriciels 5x7 points (réf. : LTP-1557AE)
- 1 bornier à vis 2 plots
- 1 connecteur HE10 mâle 2x7 broches pour CI, soudé à 90°

Module de commande

- R₁ : 470 Ω
- R₂ à R₅ : 10 kΩ
- R₆ : 1 MΩ
- R₇ : 4,7 kΩ
- C₁ : 100 µF/15V horizontal
- C₂, C₃ : 100 nF LCC jaunes
- C₄, C₅ : 22 pF céramique
- C₆ à C₁₀ : 1 µF tantale
- D₁ : diode 1N4148
- L₁ : LED Ø3mm
- Q : quartz 8 MHz
- BP : bouton poussoir miniature
- IC₁ : 68HC11A1 ou 68HC11E1 ou 68HC811E2 (voir article) + support PLCC 52cts
- IC₂ : MAX232 + support 16 broches
- CN₁ : connecteur DB9 mâle pour CI, soudé à 90°
- REG : régulateur 7805
- 2 borniers à vis 2 plots
- J₁, J₂ : barrettes HE10 2 broches + cavalier
- 1 connecteur HE10 mâle 2x7 broches pour CI, soudé à 90°



10 Tracé du circuit imprimé du module de commande



12 Implantation des éléments

pour être en mode Bootstrap (MODB=0), puis mettez la carte sous tension.

68HC11A1 (testA1E1.rec)

Lancez l'exécution du logiciel par la commande PCBUG11 -A
 Pour tester la liaison série entre le µC et le PC : tapez CTRL+R, si le message "Communications synchronised" s'affiche tout

va bien. Dans le cas contraire, on obtient le message "communication faults", faites un reset manuel et relancez PCBUG11 en entrant la commande restart.
 Définissez la location de l'EEPROM par la commande : eeprom \$B600 \$B7FF
 Effacez ensuite l'EEPROM du HC11 en entrant la commande : eeprom erase bulk.
 Chargez le programme par la commande : loads c:\afficheur\testA1E1.rec (dans la

fenêtre principale s'affiche le nombre d'octets programmés)

Il ne reste plus qu'à lancer l'exécution du programme grâce à la commande : g \$B600.

68HC11E1 (testA1E1.rec)

Lancez l'exécution du logiciel par la commande PCBUG11 -A

Pour tester la liaison série entre le μ C et le PC : tapez CTRL+R, si le message "Communications synchronised" s'affiche tout va bien. Dans le cas contraire, on obtient le message "communication faults", faites un reset manuel et relancez PCBUG11 en entrant la commande restart.

Déverrouillez la mémoire EEPROM par la commande : mm \$1035 entrez la valeur \$10

Définissez la location de l'EEPROM par la commande : eeprom \$B600 \$B7FF

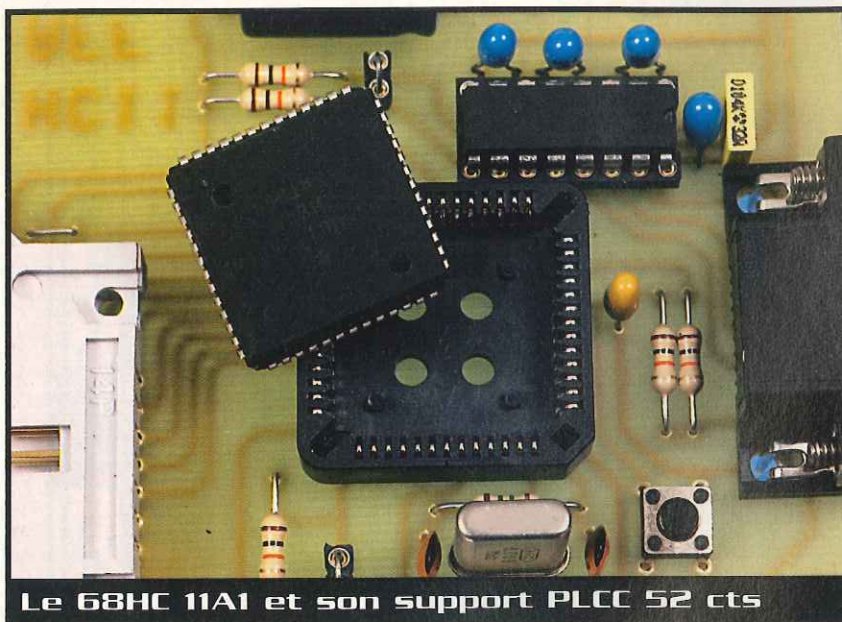
Effacez ensuite l'EEPROM du HC11 en entrant la commande : eeprom erase bulk.

commande : eeprom \$F800 \$FFFF

Effacez ensuite l'EEPROM du HC11 en entrant la commande : eeprom erase bulk. Chargez le programme par la commande : loads c:\afficheur\testE2.rec (dans la fenêtre principale s'affiche le nombre d'octets programmés)

Il ne reste plus qu'à lancer l'exécution du programme grâce à la commande : g \$F800.

Si le 68HC11 est correctement programmé, vous devez voir défiler sur le module d'affichage un texte de démonstration.



Le 68HC 11A1 et son support PLCC 52 cts

Chargez le programme par la commande : loads c:\afficheur\testA1E1.rec (dans la fenêtre principale s'affiche le nombre d'octets programmés)

Il ne reste plus qu'à lancer l'exécution du programme grâce à la commande : g \$B600.

68HC811E2 (testE2.rec)

Lancez l'exécution du logiciel par la commande PCBUG11 -A

Pour tester la liaison série entre le μ C et le PC : tapez CTRL+R, si le message "Communications synchronised" s'affiche tout va bien. Dans le cas contraire, on obtient le message "communication faults", faites un reset manuel et relancez PCBUG11 en entrant la commande restart.

Déverrouillez la mémoire EEPROM par la commande : mm \$1035 entrez la valeur \$10

Définissez la location de l'EEPROM par la

commande : eeprom \$F800 \$FFFF. Vous voulez désormais voir défiler votre propre texte, dans ce cas vous devez utiliser le logiciel "AFF.EXE" spécifiquement développé pour l'utilisation de notre afficheur.

Création du texte à afficher

Le programme "AFF.EXE" dédié à notre montage a été développé sous DELPHI 4. Il permet, d'une part de créer graphiquement sa propre police de caractère, d'autre part de taper le texte et de générer le fichier qui sera transmis au 68HC11 lors de sa reprogrammation (à la place de celui précédemment utilisé). Pour l'installer sur votre ordinateur, il suffit de copier les fichiers du programme situés sur le CD (voir page 15 ou notre site internet www.eprat.com) (matrice.txt, progA1E1.asc, progE2.asc, ledoff.bmp, ledon.bmp et l'exécutable AFF.exe) dans le même répertoire que celui cité précédemment : c:\afficheur.

Créer sa propre police de caractère

A l'exécution du logiciel, le fichier matrice.txt est automatiquement lu. Sur la fenêtre, on peut voir une représentation graphique d'une matrice 7x5 points comme celles utilisées sur le module d'affichage. Lorsque vous cliquez sur la barre de défilement, des caractères apparaissent sur cette matrice. Il vous est possible de les modifier en cliquant directement sur la matrice pour faire allumer ou éteindre les LED désirées. Une fois que le caractère vous convient, cliquez sur le bouton enregistrer, ce qui va le sauvegarder (écrasant l'ancien). Pour créer de nouveaux caractères, vous devez attendre un emplacement de libre, créer le nouveau caractère et renseigner la fenêtre caractère avant d'effectuer la sauvegarde.

Entrer le texte à afficher

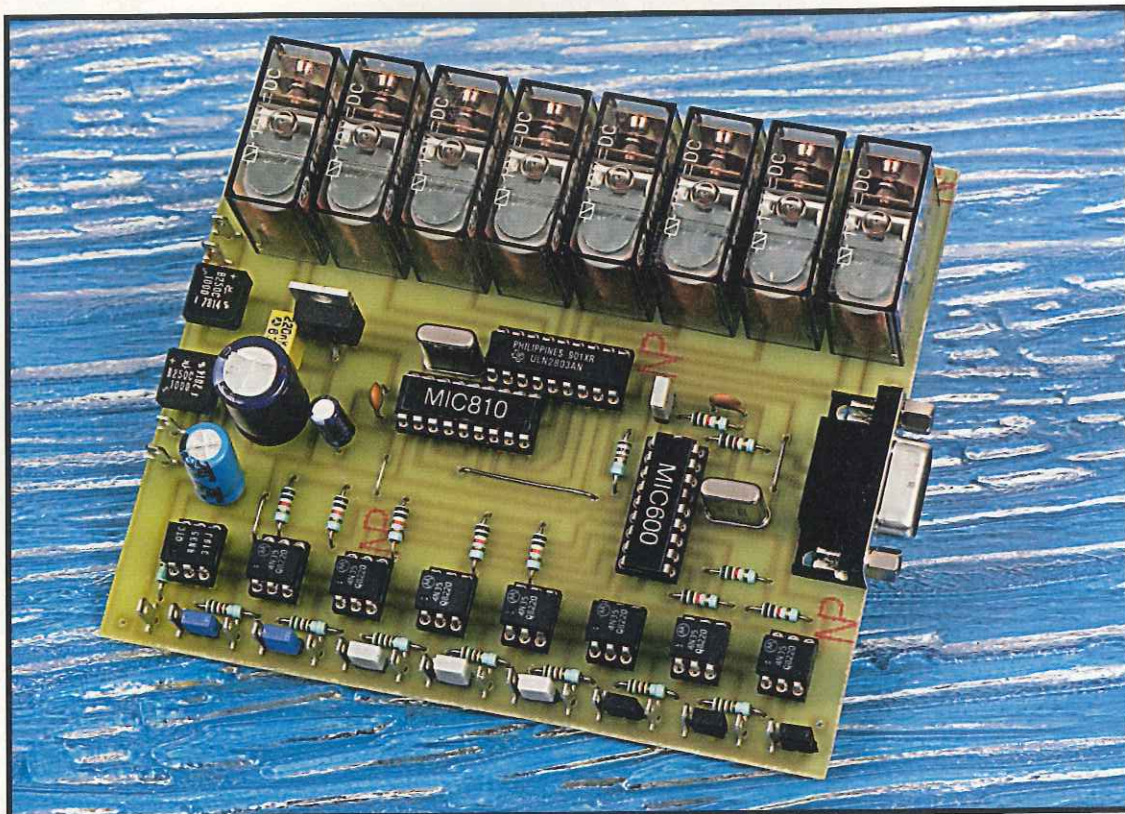
Dans le panneau inférieur, sélectionnez en premier lieu le type de 68HC11 utilisé : A1/E1 ou E2. Ensuite, entrez votre texte à afficher en respectant le nombre maximum de caractères. Sélectionnez une vitesse de défilement (entre 0 et 255), par défaut elle est de 25. Cliquez alors sur le bouton "générer le fichier". Une nouvelle fenêtre s'affiche, vous devez alors spécifier un nom pour sauvegarder ce nouveau fichier par exemple "perso.asc", veillez à utiliser l'extension ".asc". Le fichier "perso.asc" est créé à partir du fichier "progA1E1.asc" (ou "progE2.asc" pour le 68HC11E2), il ne faut surtout pas effacer ou renommer ce fichier, c'est lui qui contient le programme d'affichage. Attention, veillez à enregistrer le fichier "perso.asc" dans le même répertoire que celui du programme, soit c:\afficheur. Il faut ensuite compiler ce nouveau programme : sous MS DOS tapez la commande asmhc11 perso.asc ce qui va générer le fichier "perso.rec" c'est avec celui-ci qu'il faut désormais reprogrammer le 68HC11 (reprendre la partie "programmation 68HC11" vue précédemment).

Remarque : pour obtenir des informations complémentaires, l'auteur vous invite à visiter son site Internet : http://site.voila.fr/David_REY

D. REY

Interface d'automatisation polyvalente

Vous êtes de plus en plus nombreux à posséder un "vieux" PC, pas toujours si vieux que ça dans l'absolu d'ailleurs, mais incapable de faire "tourner" Windows et tous les logiciels associés, tous gros consommateurs de puissance et de ressources machine. Plutôt que de le laisser dormir sous la poussière d'un grenier, nous vous proposons de l'utiliser, avec notre montage aujourd'hui, pour automatiser ce que bon vous semblera.



Notre réalisation vous permet, en effet, de contrôler indépendamment l'un de l'autre jusqu'à 8 relais et de prendre en compte l'état de 8 entrées ce qui lui ouvre la porte à une très large gamme d'applications, depuis l'automatisation de votre maison ou de votre atelier d'électronicien jusqu'à celle de votre réseau de train électrique miniature par exemple. Et pour pouvoir se contenter du "vieux" PC évoqué ci-dessus, notre montage n'a besoin d'aucun langage de programmation récent mais se contente de simples caractères ASCII envoyés et reçus via une liaison série RS232 tout à fait ordinaire. Malgré ces apparentes limitations, ses performances sont tout à fait intéressantes comme nous vous invitons à le découvrir sans plus tarder.

Étude du schéma

Même s'il peut sembler plus naturel d'utiliser un port parallèle plutôt qu'un port série pour commander des relais et lire l'état de 8 entrées parallèles ; ce n'est pas ce que nous avons fait. En effet, d'une part le port parallèle des "vieux" PC n'est que rarement bidirectionnel, ce qui complique sérieusement les problèmes d'entrées de données, d'autre part, une liaison parallèle est toujours nécessairement courte, ce qui impose de placer les montages qui l'utilisent à faible distance du PC.

Nous avons donc fait appel à un port série tout à fait classique ce qui, comme notre montage ne travaille qu'à 2400 bits/seconde, permet de l'éloigner du PC de plusieurs mètres sans aucun problème. De plus, pour que notre montage reste très simple

et peu coûteux, nous avons utilisé deux nouveaux circuits intégrés de chez MICTRONICS spécialement conçus pour ce type d'application. La **figure 1** présente ainsi la partie commande des relais tandis que la **figure 2** présente la partie entrées parallèles. Comme vous pouvez le constater, elles sont aussi simples l'une que l'autre.

La commande des relais a lieu via la sortie série de la liaison RS232 qui est reliée directement au circuit intégré MIC 810, repéré IC₁ sur la figure 1. Ce circuit délivre sur ses huit sorties parallèles des niveaux logiques correspondant aux caractères qu'il reçoit sur son entrée série selon un mode de codage fort simple que nous verrons dans un instant. Ses sorties ne pouvant délivrer un courant suffisant pour commander des relais, nous avons fait appel à un classique

ULN2803, repéré IC₂, pour amplifier ce courant. Afin que notre montage puisse être utilisé dans des applications domestiques où les consommations de courant sont parfois très importantes, nous avons fait appel à des relais FINDER capables de couper jusqu'à 10A en 220V.

En fait, le MIC 810 dispose d'autres possibilités, telles que l'adressage du circuit ce qui permet de relier jusqu'à 8 MIC 810 sur la même liaison série permettant ainsi de commander jusqu'à 64 sorties parallèles. Elles ne sont pas utilisées ici mais vous pouvez les découvrir sur le site du fabricant d'adresse www.micronics.net où les fiches techniques complètes des circuits sont disponibles en téléchargement et en français de surcroît !

L'alimentation du MIC 810 a lieu sous une tension stabilisée à 5V par IC₃ tandis que les relais utilisent du 12V, qui n'a pas besoin

de stabilisation et qui se trouve donc prélevé avant IC₃.

Notez l'utilisation d'un transformateur à deux enroulements secondaires indépendants ; en effet le deuxième secondaire sert à délivrer, via le pont PT₂, une tension continue isolée du reste du montage pour alimenter les photocoupleurs de l'étage d'entrée que nous allons étudier maintenant.

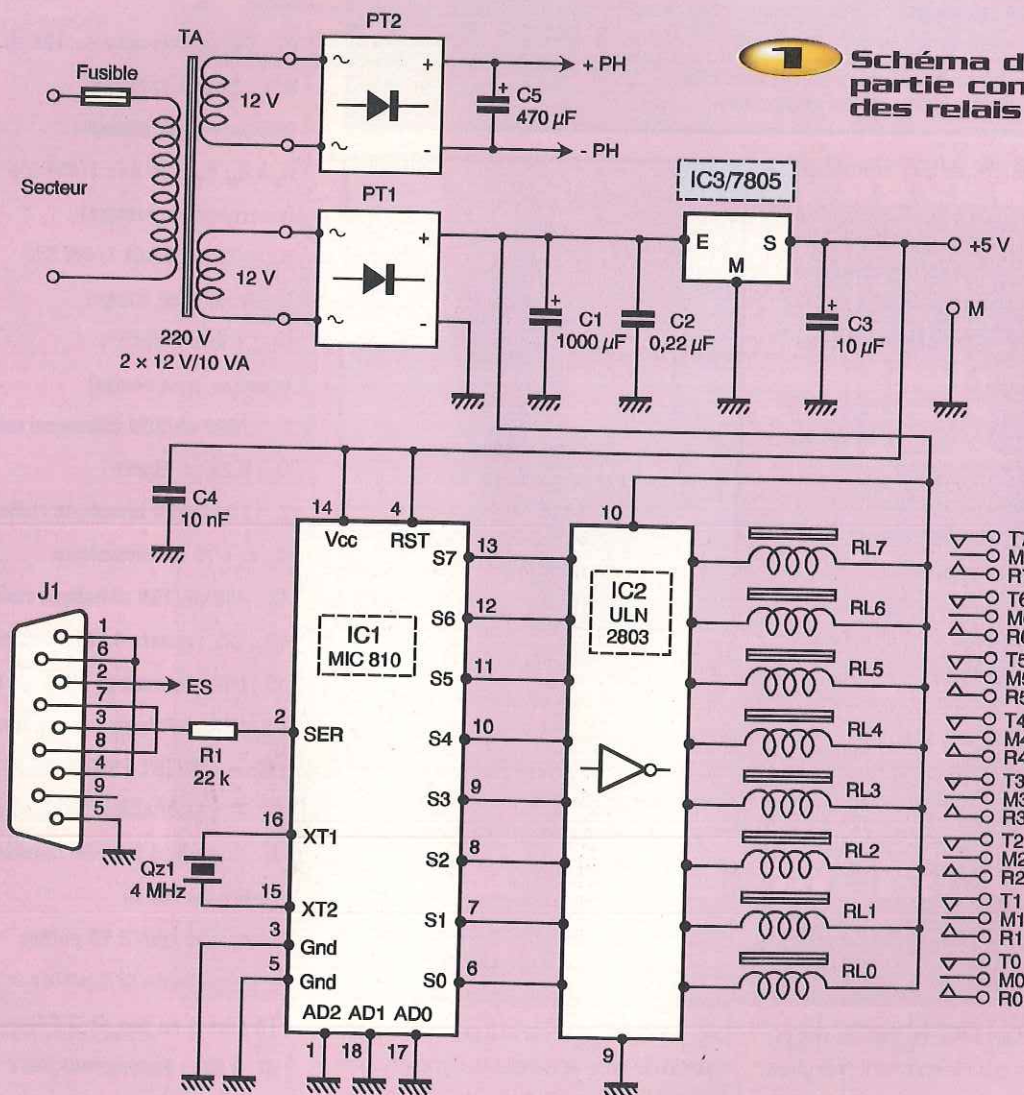
En effet, afin d'offrir un maximum de sécurité tant à notre montage qu'au PC qui lui est relié, nous avons fait appel à des entrées opto-isolées comme vous pouvez le constater à l'examen de la figure 2.

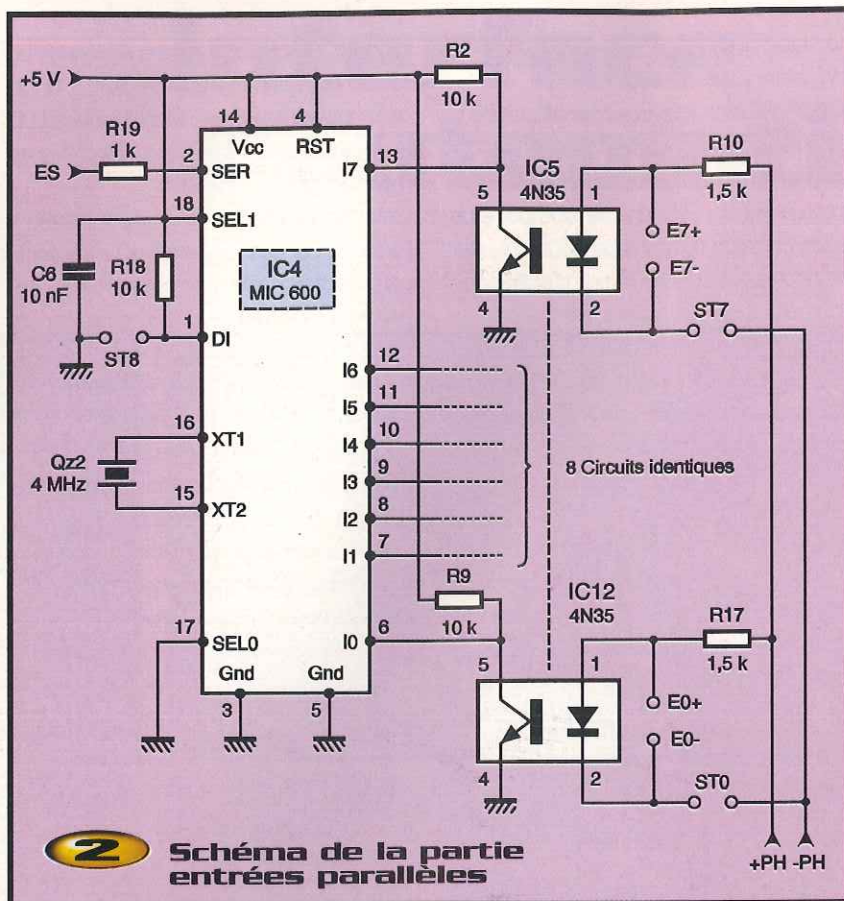
Lorsque les straps ST₀ à ST₇ sont en place, la tension délivrée par le secondaire indépendant du transformateur alimente les LED des optocoupleurs. Il est alors possible de connecter entre les points Ex+ et Ex- les interrupteurs de votre choix. Un interrupteur ouvert laissera la LED du photocoupleur

allumée ce qui conduira à appliquer un niveau logique bas à l'entrée de IC₄, tandis qu'un interrupteur fermé fera éteindre la LED et appliquera un niveau logique haut à cette même entrée.

Cette façon de faire n'étant pas suffisante si l'on veut, par exemple, prendre en compte une information telle que la présence ou l'absence du secteur, nous avons prévu pour cela les straps ST₀ à ST₇. Si l'on veut utiliser l'entrée E0 pour détecter la présence du secteur, il suffit de ne pas câbler R₁₇ et d'ouvrir le strap ST₀. Il ne reste plus alors qu'à relier à E0+ et E0- le montage dont le schéma vous est proposé figure 3 pour disposer d'un système très fiable et totalement isolé de détection de présence du secteur. Cette façon de faire peut, bien sûr, être utilisée sur n'importe laquelle des entrées E0 à E7. Il suffit juste pour cela de ne pas câbler la ou les résistances correspondantes et

1 Schéma de la partie commande des relais





2

Schéma de la partie entrées parallèles

d'ouvrir les straps ST_0 à ST_7 correspondants.

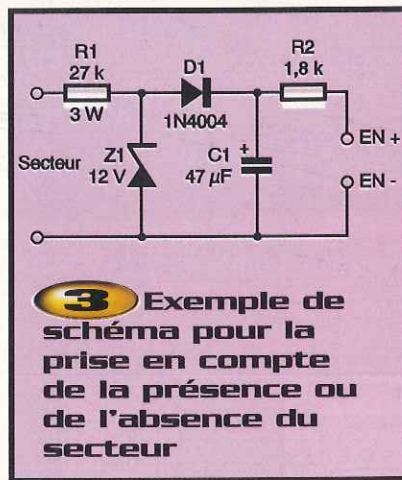
Le reste du schéma de la figure 2 est d'une extrême simplicité grâce, ici encore, à l'utilisation d'un circuit intégré spécialisé, le MIC 600 de MICTRONICS, repéré IC₄ sur cette même figure.

Ce circuit dispose de huit entrées parallèles I0 à I7, reliées ici à la sortie de nos optocoupleurs. Il transmet, sur sa sortie série disponible en patte 2, un caractère correspondant à l'état de ses entrées. Cette transmission a lieu automatiquement dès qu'une entrée au moins change d'état ce qui facilite évidemment grandement l'utilisation du montage puisqu'il n'est pas nécessaire de l'interroger en permanence pour connaître l'état de ses entrées.

Afin de faciliter encore la gestion des informations transmises par le MIC 600, le strap ST_8 permet de choisir d'envoyer le niveau réel des entrées I0 à I7 (strap en place) ou le niveau contraire (strap enlevé).

Réalisation

Hormis les circuits MICTRONICS, qui ne sont disponibles pour le moment que chez SELECTRONIC, les autres composants de notre réalisation sont des classiques que



3

Exemple de schéma pour la prise en compte de la présence ou de l'absence du secteur

l'on trouve partout. Notez que vous n'êtes pas tenus d'approvisionner l'intégralité des composants figurant dans notre nomenclature et que vous pouvez vous limiter, tant au niveau des relais qu'au niveau des optocoupleurs, au nombre d'entrées et de sortie dont vous avez réellement besoin, quitte à en ajouter ensuite si nécessaire.

Attention au niveau des relais utilisés. Nous avons choisi des modèles FINDER type 40.31 ou équivalents car ils peuvent couper 10A en 220V ce qui permet la commande de gros appareils électroménagers. Ces relais existent avec deux pas d'implantation des pattes : un pas de 3,5mm

qui est celui retenu ici et un pas de 2,5mm qui ne conviendrait pas au dessin que nous avons réalisé. Vérifiez bien le modèle que vous achetez ou corrigez si nécessaire le dessin du circuit imprimé à ce niveau. Attention également au choix du transformateur TA qui doit avoir deux enroulements secondaires séparés. C'est le cas de tous les modèles moulés à double secondaire (appelés parfois abusivement modèles à point milieu) mais pas toujours

Nomenclature

IC₁ : MIC 810 (SELECTRONIC)

IC₂ : ULN2803

IC₃ : régulateur +5V/1A, boîtier TO220 (7805)

IC₄ : MIC 600 (SELECTRONIC)

IC₅ à IC₁₂ (ou moins selon besoins) : 4N35

PT₁, PT₂ : ponts moulés 100 V/1A

R*1 : 22 kΩ 1/4W 5%

(rouge, rouge, orange)

R₂ à R₉, R₁₈ : 10 kΩ 1/4W 5%

(marron, noir, orange)

R₁₀ à R₁₇ : 1,5 kΩ 1/4W 5%

(marron, vert, rouge)

R₁₉ : 1 kΩ 1/4W 5%

(marron, noir, rouge)

C₁ : 1000 µF/25V chimique radial

C₂ : 0,22 µF Mylar

C₃ : 10 µF/25V chimique radial

C₄, C₆ : 10 nF céramique

C₅ : 470 µF/25V chimique radial

QZ₁, QZ₂ : quartz 4 MHz, boîtier HC 18/U

TA : transformateur 220V/2x12V/10VA

RL₁ à RL₇ (ou moins selon besoins) :

relais 12V/1RT, FINDER 40.31, ZETTLER

AZ 692, SCHRACK RP 412 ou équivalent

J₁ : prise DB9 femelle coudée à 90° à implanter sur CI

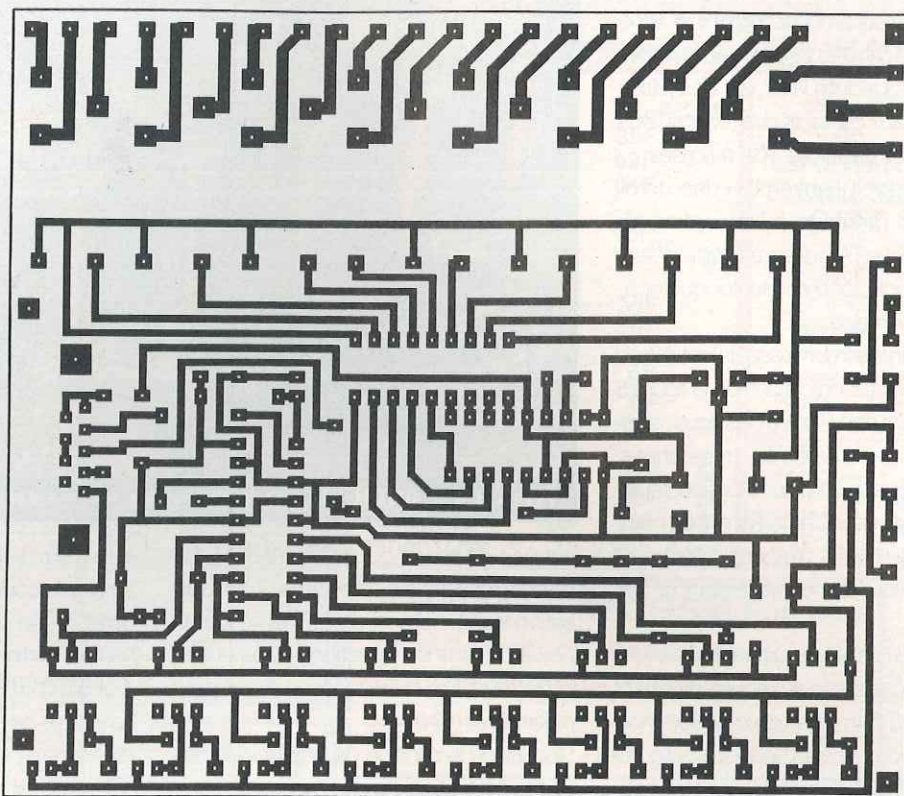
3 supports de CI 18 pattes

8 supports de CI 6 pattes

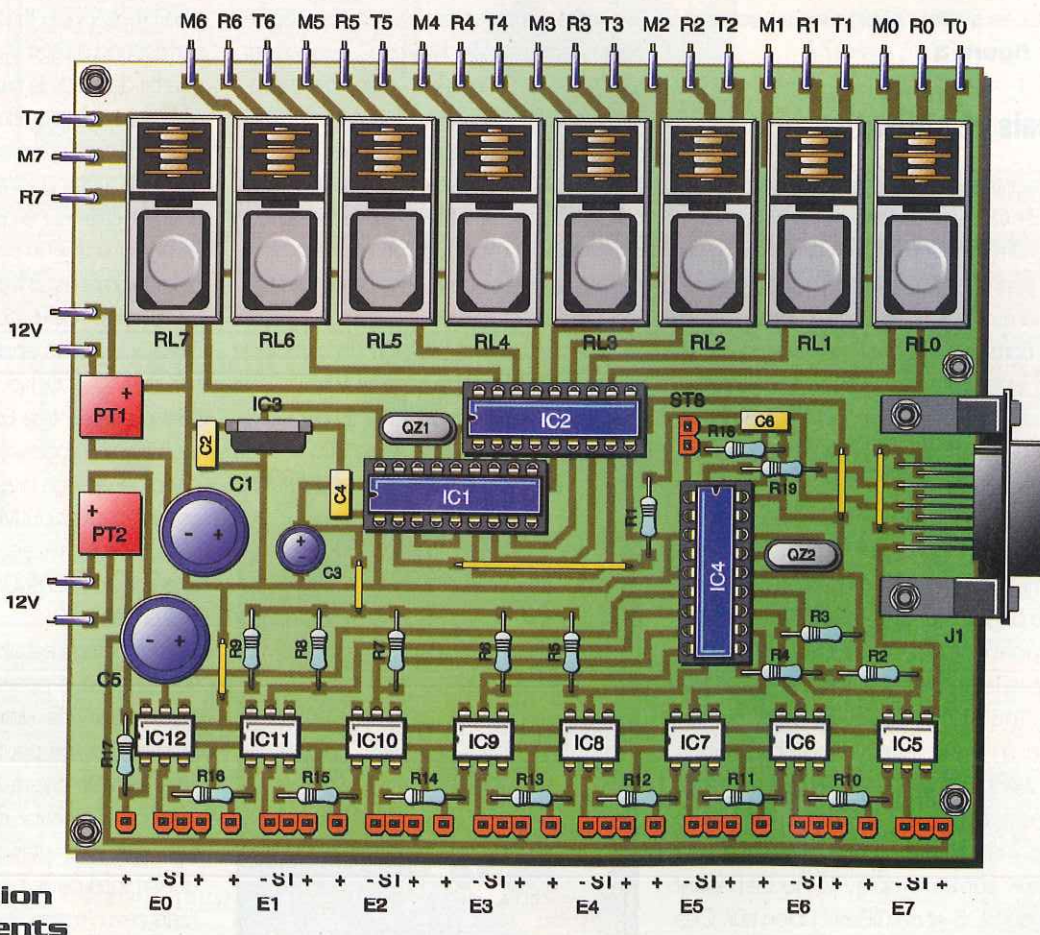
18 picots au pas de 2,54mm

ST₀ à ST₈ : 9 straps au pas

de 2,54mm



4
Tracé du
circuit
imprimé



5
Implantation
des éléments

des modèles à étrier. Le tracé du circuit imprimé vous est proposé **figure 4** et ne présente pas de difficulté particulière. Veillez juste à ne pas réduire la largeur des pistes aboutissant aux contacts des relais si vous voulez pouvoir utiliser les 10A autorisés.

Ce circuit reçoit l'ensemble des composants du montage à l'exception du transformateur TA qui sera un modèle à étrier fixé sur le fond du boîtier qui recevra notre montage.

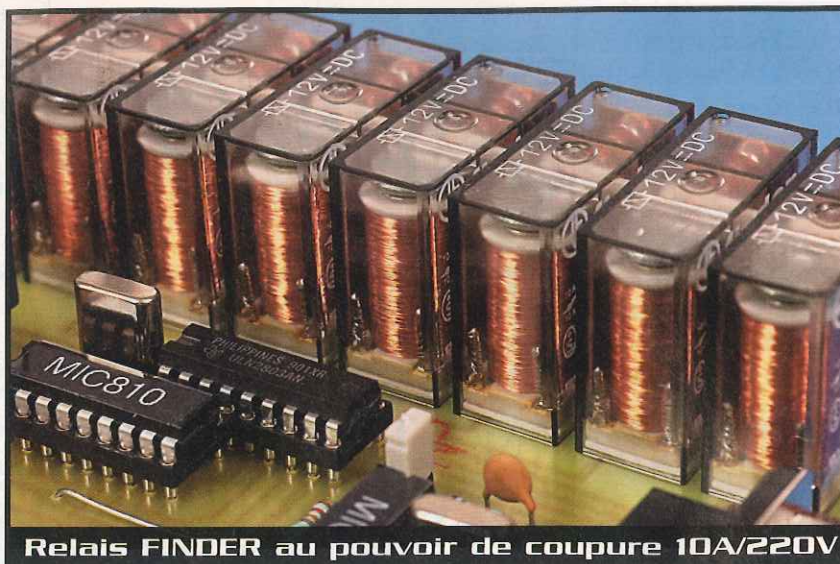
L'implantation des composants est à faire en respectant les indications de la figure 5 et en travaillant dans l'ordre classique : supports de circuits intégrés, résistances, condensateurs et enfin semi-conducteurs. Veillez à bien respecter le sens des composants polarisés et plus particulièrement celui des ponts de redressement et des chimiques.

Les straps ST_0 à ST_7 pourront être de véritables straps réalisés au moyen de picots au pas de 2,54mm et de cavaliers normalisés si vous voulez pouvoir configurer votre circuit à la demande, ou être soudés ou laissés libre à demeure selon que vous utiliserez ou non une ou plusieurs interfaces secteur analogues au schéma de la **figure 3**.

Essais et utilisation

Le montage ayant été conçu pour ne nécessiter qu'un minimum de programmation, voire même pas de programmation du tout, son utilisation est fort simple. Raccordez-le au secteur ainsi qu'à un port série d'un compatible PC au moyen d'un câble droit. Programmez le port du PC pour fonctionner à 2400 bits/seconde, sur 8 bits et sans parité ce qui est le format d'émission et de réception de notre montage.

Utilisez alors, au moins dans un premier temps, un logiciel de terminal capable d'émettre des caractères sur le port série pour tester la commande des relais et capable de recevoir des caractères pour la lecture des entrées. Sur un PC muni de Windows, vous pouvez faire appel à l'hyper terminal mais nous vous proposons deux petits programmes en QBasic capable de réaliser ces tests avec un maximum de facilité et de confort sur un simple PC ne disposant que du DOS et de QBasic, bien sûr. Ces deux programmes ont pour noms : TESTINTE.BAS pour le test de la partie



entrées du montage et TESTINTS.BAS pour le test de la partie sortie. Leur utilisation se comprend d'elle-même car ils posent toutes les questions utiles et affichent leurs résultats de façon on ne peut plus claire.

Ceci étant, le principe de commande des sorties est très simple et ne nécessite d'envoyer au MIC 810 que deux caractères ASCII selon le format suivant : MN. M est une lettre comprise entre S et Z qui représente l'adresse du MIC 810 auquel est destiné l'ordre. Dans le cas de notre montage cette lettre est fixée à S par mise à la masse des pattes AD0, AD1 et AD2 du MIC 810. N est un caractère ASCII de code compris entre 00 et FF (hexadécimal) qui correspond à l'état que l'on souhaite donner aux sorties selon le principe de codage logique suivant :

- Le bit de poids le plus fort du caractère correspond à l'état de la sortie S7,
- Le bit de poids le plus faible du caractère correspond à l'état de la sortie S0.

Pour faire coller les relais 0, 4 et 5 il suffira

ainsi d'envoyer au montage S1 puisque 1 a comme code ASCII exprimé en hexadécimal 31, ce qui positionne bien au niveau haut les sorties correspondant aux relais 0, 4 et 5 (0011001).

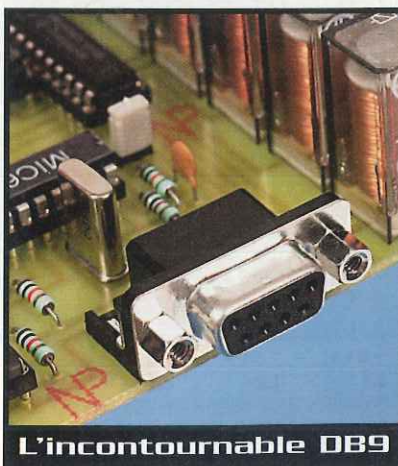
La lecture de l'état des entrées est encore plus simple ; en effet, le MIC 600 transmet automatiquement un caractère lorsqu'une au moins de ses entrées change d'état. Ce caractère est codé comme suit :

- Le bit de poids le plus fort du caractère correspond à l'état de l'entrée I7,
- Le bit de poids le plus faible du caractère correspond à l'état de l'entrée I0.

Si le strap ST_8 est en place et que, par exemple, vous laissez ouvertes toutes les entrées sauf E0 et E7, vous recevrez le caractère ü dont le code ASCII exprimé en hexadécimal est 81 ou encore 1000001 en binaire. Cela correspond bien aux niveaux logiques appliqués aux entrées I0 à I7 du MIC 600. Nous vous rappelons en effet qu'une entrée Ex+ Ex- ouverte laisse la LED de l'optocoupleur allumée ce qui fait donc générer un niveau bas sur l'entrée Ix correspondant du MIC 600 et comme le strap ST_8 est en place, dans notre exemple, le MIC 600 transmet l'état réel de ses entrées.

Votre montage est alors pleinement opérationnel et vous pouvez donc l'utiliser pour l'application de votre choix. Il se programme, si l'on peut dire, avec n'importe quel langage puisqu'il suffit juste de savoir émettre et recevoir des caractères ASCII. Vous pouvez utiliser, comme base de départ, nos petits programmes en QBasic mais cela n'a rien d'impératif vu la simplicité du dialogue à établir avec le montage.

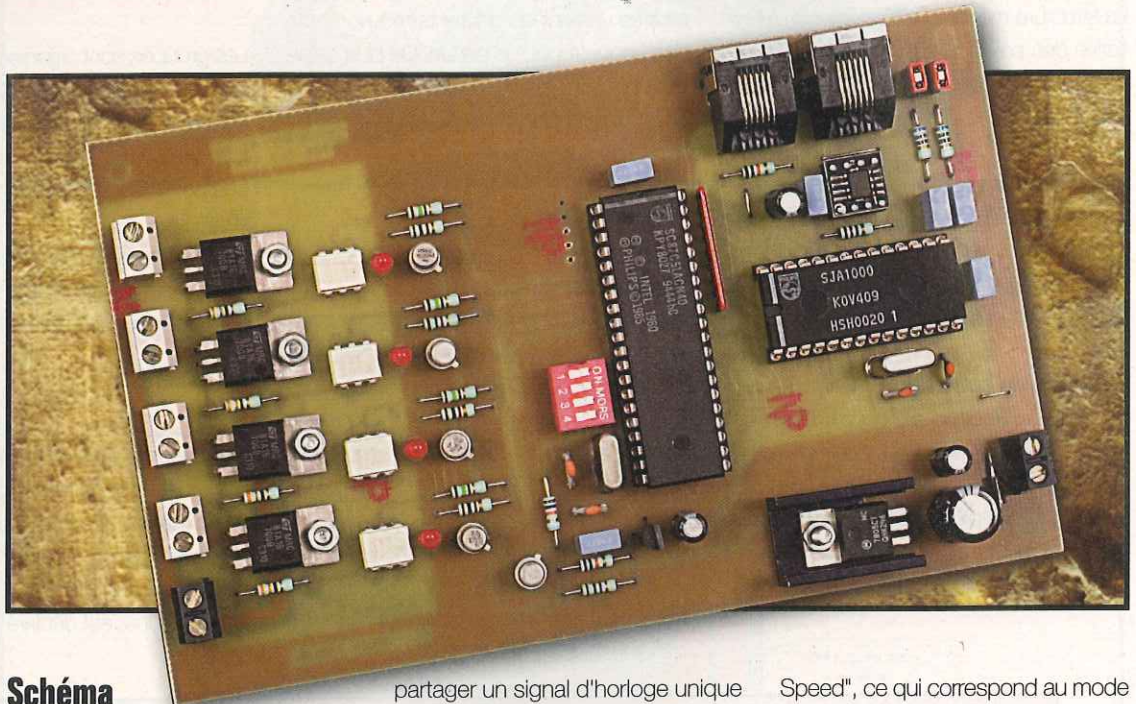
C. TAVERNIER



Interface 4 entrées/4 sorties triacs

pour bus CAN

Le bus CAN étant particulièrement bien adapté aux applications domotiques, nous vous proposons de réaliser une première petite maquette qui deviendra vite le complément idéal de l'interface RS232 / bus CAN que nous avons déjà proposé. Afin de découvrir les possibilités du bus CAN, nous avons préféré limiter les possibilités de ce montage afin d'en préserver la simplicité. Le montage est prévu pour piloter 4 sorties à triacs capables d'alimenter des charges de faibles puissances. Il dispose également de 4 entrées de type "contact" capables de relire l'état d'un interrupteur ou d'un bouton-poussoir.



Schéma

Le schéma de notre montage est reproduit en **figure 1**. Comme cela apparaît sur cette figure, le schéma n'est pas vraiment compliqué. Un microcontrôleur 87C51 est au cœur de ce montage. Ce choix est particulièrement adapté à notre cas de figure car l'interface, avec le circuit spécialisé SJA1000, se réduit à sa plus simple expression. Le décodage de l'espace adressé par le microcontrôleur est réduit au strict minimum puisque nous avons simplement utilisé la ligne d'adresse A15 pour piloter la broche CS du circuit U₄. La broche 11 de U₄ détermine le mode de fonctionnement du bus qui est fixé dans le mode "Intel" sur notre schéma.

L'horloge interne du microcontrôleur U₂ est mise en œuvre au moyen du quartz QZ2 et des condensateurs C₈ et C₉ qui lui sont associés. Par souci de simplicité, nous avons préféré utiliser un quartz supplémentaire pour mettre en œuvre l'oscillateur du circuit U₄, plutôt que de

partager un signal d'horloge unique entre les circuits U₂ et U₄.

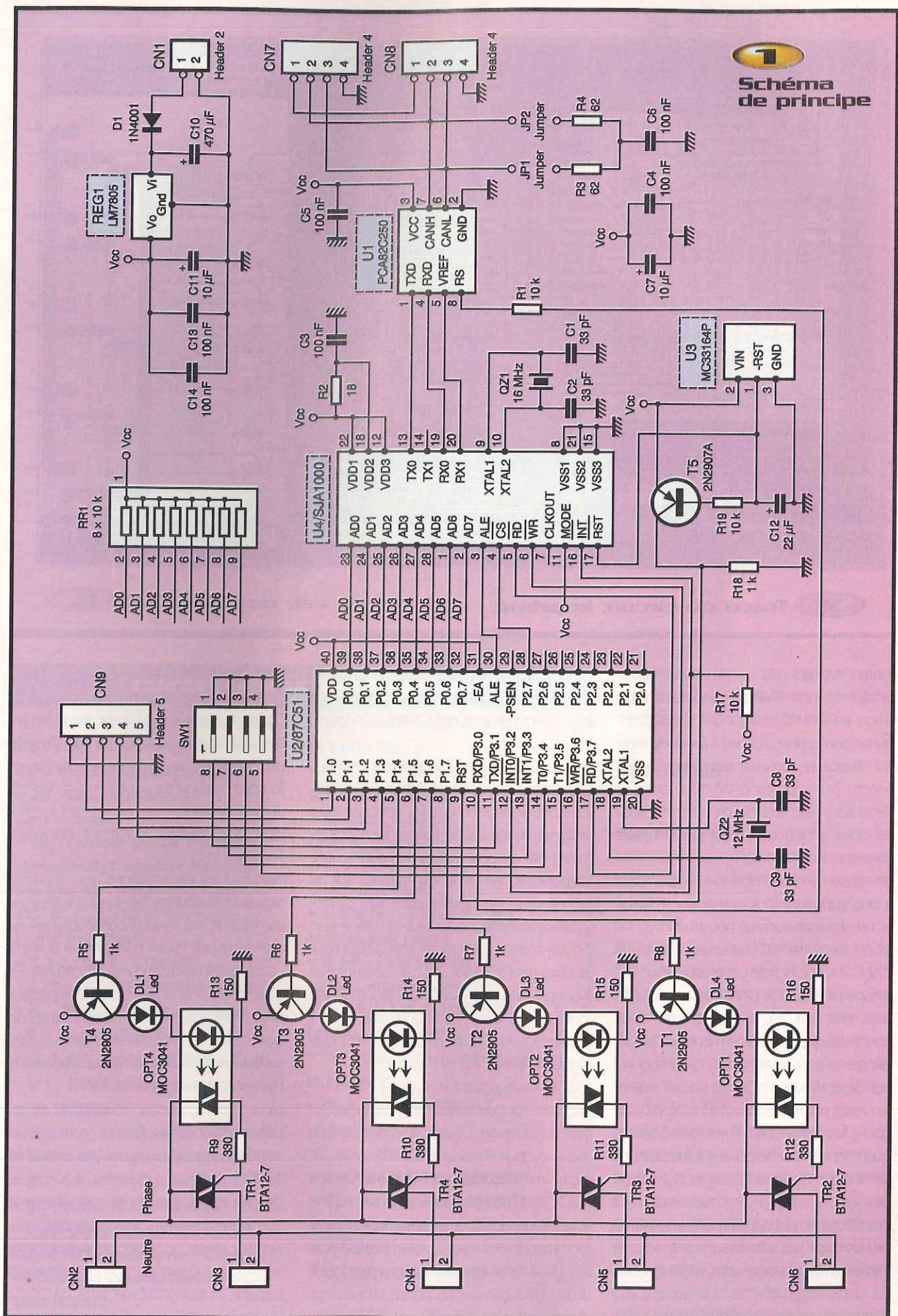
Pour gérer la remise à zéro de ce montage, nous avons préféré faire appel à un superviseur d'alimentation (U₃). Ce circuit permet de garantir le bon démarrage du montage même si la montée de la tension d'alimentation du montage est lente. Ceci est important dans le cas où le montage serait installé dans un endroit peu accessible, distant ou lorsqu'il n'est pas possible de procéder à une remise à zéro manuelle. Le signal -RST issu du circuit U₃ pilote directement la ligne de remise à zéro du circuit U₄, tandis que le transistor T₅ inverse le niveau pour remettre à zéro le microcontrôleur. En effet, pour ce dernier, l'état actif du signal de remise à zéro est un niveau haut. Le bloc d'interrupteurs SW, permet de choisir la valeur de base des identificateurs auxquels le montage va répondre sur le bus CAN. Nous y reviendrons à la fin de cet article.

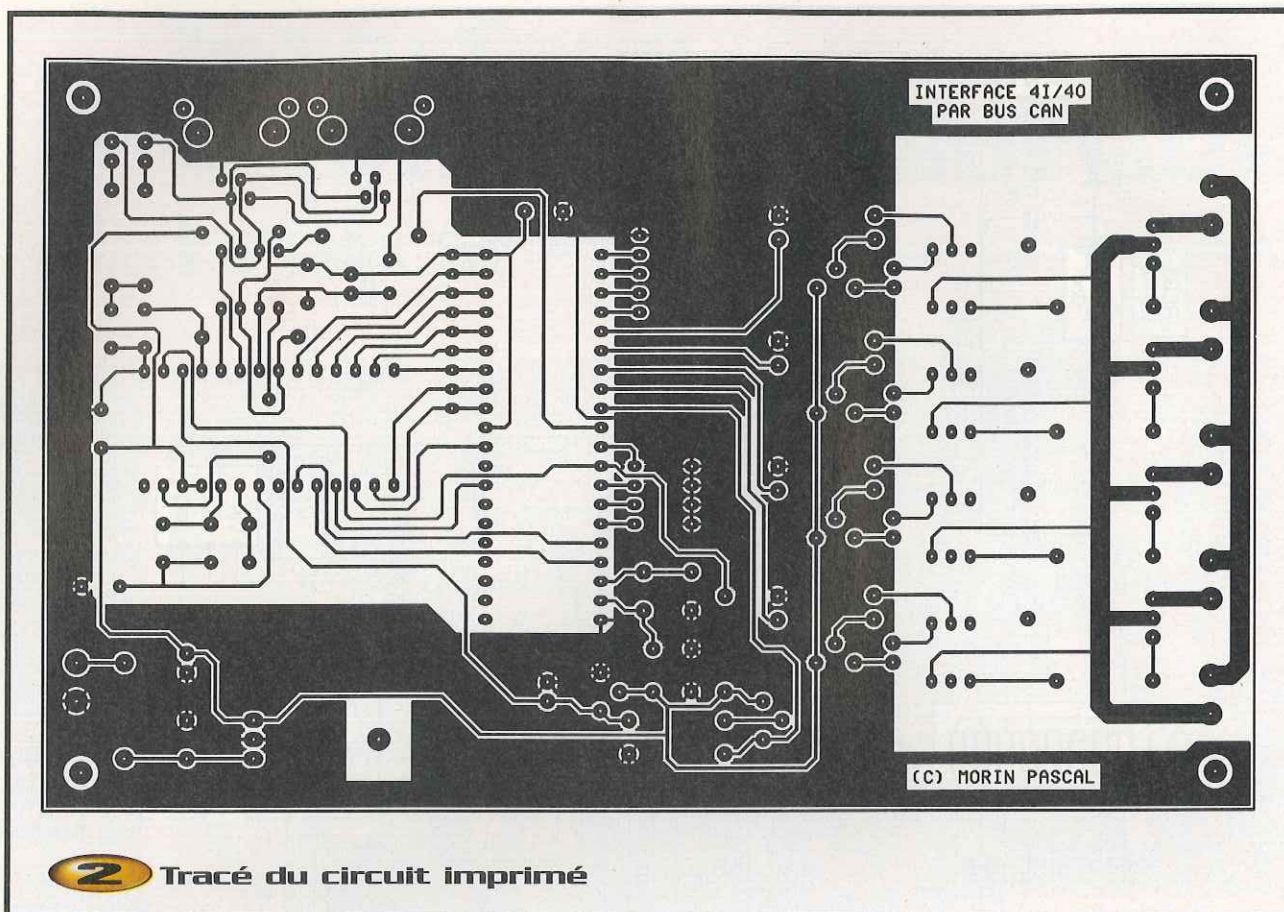
Les signaux TX0, RX0 et RX1 de U₄ sont confiés au circuit U₁ qui se chargera d'adapter les niveaux pour être compatibles avec un bus CAN "High

Speed", ce qui correspond au mode utilisé par l'interface RS232/Bus CAN que nous avons déjà proposé. Comme nous l'avons déjà expliqué, le mode CAN "High Speed" n'implique pas nécessairement un dialogue à des vitesses élevées. Pour augmenter la distance de communication entre les équipements du bus CAN, il est parfaitement possible d'utiliser le mode "High Speed" avec un débit relativement bas, tel que 20 kbits/s. Dans ce mode, ce qui compte c'est que la différence de potentiel des signaux soit nulle lorsque le bus présente un état récessif (CANH et CANL = 2,5V) tandis que la différence de potentiel sera de 2V lorsque le bus présente un état dominant (CANH=3,5V et CANL=2,5V). Pour communiquer dans le mode "High Speed", le circuit PCA82C250 est le "driver" de ligne le plus répandu actuellement. Ce circuit est totalement protégé contre les courts-circuits qui peuvent survenir sur les lignes CANH et CANL. Le circuit supporte même des courts-circuits au +12V, ce qui correspond à un cas de



Schéma de principe





2 Tracé du circuit imprimé

figure imposé dans les automobiles modernes. Il est d'ailleurs possible de remplacer le PCA82C250 par un PCA82C251 qui est compatible broches à broches, mais qui supporte des tensions anormales encore plus élevées (24V). Ces composants sont très répandus en version CMS, ce qui explique qu'il n'est pas toujours simple de s'en procurer en version DIP. Fort heureusement, il est possible de faire appel à des supports pour composants CMS. Nous en reparlerons un peu plus loin.

La connectique RJ11 retenue pour le bus CAN (CN₇ et CN₈) est compatible avec l'interface RS232/Bus CAN que nous avons déjà proposé. La présence de deux connecteurs permet à ce montage de s'intercaler facilement sur le bus CAN auquel il est destiné. Vous pouvez choisir indifféremment entre le connecteur CN₇ ou CN₈ pour jouer le rôle d'entrée ou de sortie. Ce qui compte c'est de penser à raccorder la terminaison de ligne (R₂, R₄ et C₆) à l'aide des jumpers JP₁ et JP₂, mais seulement pour l'équipement qui se trouve à l'extrémité du bus CAN. Nous insisterons une nouvelle fois sur le fait que, pour un bus CAN, seules les deux extrémités du bus doivent être chargées par des impédances de termi-

naison de ligne. Les équipements qui viennent se raccorder tout le long du bus CAN ne doivent pas posséder cette impédance de fin de ligne.

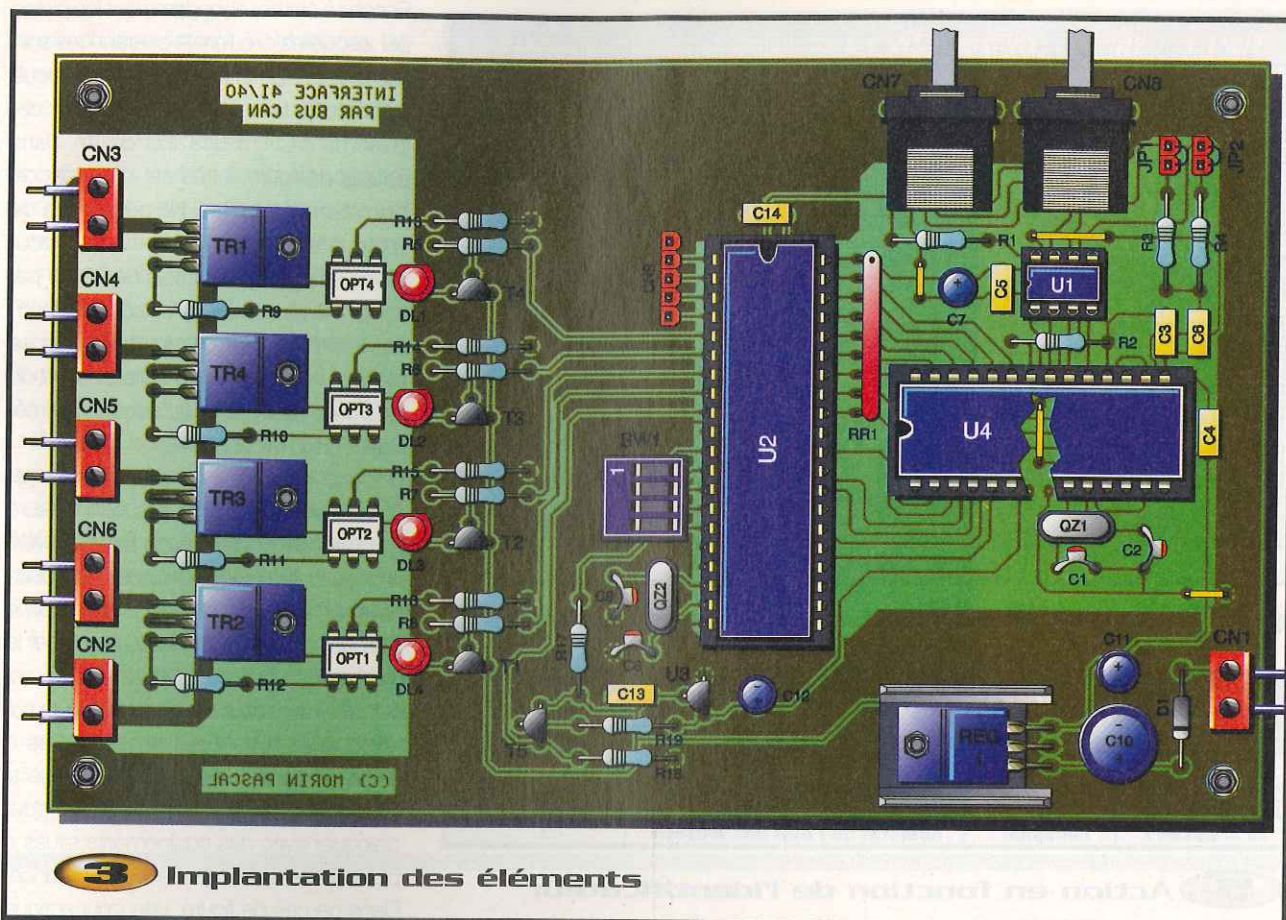
En ce qui concerne la partie purement "applicative" du montage, il n'y a rien de sorcier. Les signaux du port P1 sont partagés entre les entrées et les sorties du montage. Les 4 bits de poids faible du port P1 sont disponibles en tant qu'entrées sur le connecteur CN₉. Le microcontrôleur U₂ dispose de résistances de "Pull-Up" sur le port P1, de sorte que les lignes sont à l'état haut si elles sont laissées en l'air. Les entrées sont prévues pour relire l'état d'un contact qui effectue une mise à la masse, ce qui explique pourquoi la masse est distribuée également sur CN₉.

Les 4 bits de poids fort du port P1 sont utilisés en tant que sorties pour piloter les triacs du montage. L'adaptation des signaux issus du port P1 pour commander les triacs est confiée à des optotriacs (MOC3041). Ces derniers disposent d'une logique de détection pour synchroniser la commande des triacs avec le passage par 0V de la tension secteur. Ceci permet de limiter efficacement les perturbations électromagnétiques émises par le montage

(neige sur la TV, grésillement perceptibles lors de l'écoute des émissions radios, etc.). Ce mode de déclenchement des triacs est parfaitement adapté lorsque le triac alimente des charges essentiellement résistives (lampes, etc.). Par contre, ce mode de déclenchement n'est pas adapté au pilotage des charges fortement inductives (moteurs, etc.).

Bien que les triacs retenus pour ce montage soient capables de piloter des charges de 1200W, nous vous conseillons de ne pas dépasser 100W à 200W par sortie, à cause des limitations du circuit imprimé. En effet, avec 4 charges de 200W pilotées simultanément, l'intensité du courant qui circulera dans CN₂ pourra atteindre 4A. Cela fait beaucoup de courant pour les pistes et pour ce pauvre connecteur.

Nous terminerons la description de ce schéma par l'alimentation. Le montage sera alimenté de préférence par une tension de 12VDC. La tension d'alimentation n'a pas besoin d'être stabilisée, mais elle devra être correctement filtrée. Par exemple, vous pourrez utiliser un petit bloc d'alimentation surmoulé à condition que ce dernier soit capable de fournir 250mA sous 12VDC. La diode D₁ protège le montage contre les



3 Implantation des éléments

Nomenclature

- CN₁ à CN₆ : borniers de connexion à vis 2 plots, au pas de 5,08mm, à souder sur circuit imprimé, profil bas
- CN₇, CN₈ : embases RJ11 (voir le texte)
- CN₉ : barrette mini-KK 5 contacts, sorties droites, à souder sur circuit imprimé, réf. : MOLEX 22-27-2051
- C₁, C₂, C₈, C₉ : 33 pF céramique au pas de 5,08mm
- C₃ à C₆, C₁₃, C₁₄ : 100 nF
- C₇, C₁₁ : 10 µF/25V sorties radiales
- C₁₀ : 470 µF/25V sorties radiales
- C₁₂ : 22 µF/25V sorties radiales
- DL₁ à DL₄ : diodes LED rouges 3mm
- D*1 : 1N4001 (diode de redressement 1A/100V)
- JP₁, JP₂ : jumpers au pas de 2,54mm
- OPT1 à OPT4 : optotriacs MOC3041
- QZ₁ : quartz 16 MHz en boîtier HC49/U
- QZ₂ : quartz 12 MHz en boîtier HC49/U
- REG₁ : régulateur LM7805 (5V) en boîtier TO220 + dissipateur thermique 18°C/W (par exemple SHAFFNER réf. : RAWA 400 9P)
- RR₁ : réseau résistif 8x10 kΩ en boîtier SIL
- R₁, R₁₇, R₁₉ : 10 kΩ 1/4W 5% (marron, noir, orange)
- R₂ : 18 Ω 1/4W 5% (marron, gris, noir)
- R₃, R₄ : 62 Ω 1/2W 5% (bleu, rouge, noir)
- R₅ à R₈, R₁₈ : 1 kΩ 1/4W 5% (marron, noir, rouge)
- R₉ à R₁₂ : 330 Ω 1/4W 5% (orange, orange, marron)
- R₁₃ à R₁₆ : 150 Ω 1/4W 5% (marron, vert, marron)
- SW₁ : bloc de 4 micro-interrupteurs en boîtier DIL
- TR₁ à TR₄ : triacs BTA12-7 ou équivalent
- T₁ à T₅ : 2N2905A
- U₁ : PCA82C250 ou PCA82C251
- U₂ : microcontrôleur 87C51 avec EPROM interne
- U₃ : MC33164P
- U₄ : SJA1000 ou 82C200

inversions de polarité, ce qui est parfois utile pour les étourdis (dont l'auteur fait également partie de temps en temps).

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 CN₁ à CN₆, REG₁, D₁, TR₁ à TR₄, JP₁ et JP₂, il faudra percer avec un foret de 1mm de diamètre. Pour les connecteurs CN₇ et CN₈, n'oubliez pas de percer le passage nécessaire aux ergots du connecteur RJ11 avec un foret de 3,5mm de diamètre. Cette dernière remarque concerne également le trou de passage pour les vis de fixation de TR₁ à TR₄ et REG₁.

Avant de réaliser le circuit imprimé, il est préférable de vous procurer les composants pour vous assurer qu'ils s'implanteront correctement. Cette remarque concerne particulièrement le driver de ligne PCA82C250 (ou PCA82C251). Ces composants sont très répandus en version CMS, mais ils ne sont pas toujours disponibles en boîtier DIP. Si vous avez du mal à

SW*1-1	SW*1-2	SW*1-3	SW*1-4	Valeur de l'identificateur de base
ON	ON	ON	ON	000H
OFF	ON	ON	ON	008H
ON	OFF	ON	ON	010H
OFF	OFF	ON	ON	018H
ON	ON	OFF	ON	020H
OFF	ON	OFF	ON	028H
ON	OFF	OFF	ON	030H
OFF	OFF	OFF	ON	038H
ON	ON	ON	OFF	040H
OFF	ON	ON	OFF	048H
ON	OFF	ON	OFF	050H
OFF	OFF	ON	OFF	058H
ON	ON	OFF	OFF	060H
OFF	ON	OFF	OFF	068H
ON	OFF	OFF	OFF	070H
OFF	OFF	OFF	OFF	078H

4 Identificateur de base en fonction de SW₁

Identificateur de la trame	Émission / Réception	Action du montage	Nb d'octets de donnée (DLC)
ID = Base	Réception	Commande des sorties	1
ID = Base+1	Réception	Demande de l'état des entrées	0
ID = Base+2	Émission	Résultat de l'état des entrées	1

5 Action en fonction de l'identificateur

Commande des sorties	Sortie sur CN*3	Sortie sur CN*4	Sortie sur CN*5	Sortie sur CN*6
FOH	ON	ON	ON	ON
EOH	OFF	ON	ON	ON
DOH	ON	OFF	ON	ON
COH	OFF	OFF	ON	ON
BOH	ON	ON	OFF	ON
A0H	OFF	ON	OFF	ON
90H	ON	OFF	OFF	ON
80H	OFF	OFF	OFF	ON
70H	ON	ON	ON	OFF
60H	OFF	ON	ON	OFF
50H	ON	OFF	ON	OFF
40H	OFF	OFF	ON	OFF
30H	ON	ON	OFF	OFF
20H	OFF	ON	OFF	OFF
10H	ON	OFF	OFF	OFF
00H	OFF	OFF	OFF	OFF

6 Valeur à transmettre pour la commande des sorties

trouver un modèle DIP, vous pourrez utiliser le modèle CMS en faisant appel à adaptateur pour boîtier SOB. Pour le reste, l'implantation du circuit imprimé est relativement simple. Cependant, respectez scrupuleusement le découplage des lignes d'alimentations si vous voulez éviter les mauvaises surprises. Vous noterez la présence de trois straps

qu'il est préférable d'implanter en premier pour des raisons de commodité (en particulier pour le strap situé sur U₄). Le régulateur REG, sera monté sur un petit dissipateur thermique pour limiter la température de fonctionnement à une valeur acceptable au touché. Si vous montez le montage dans un petit boîtier, cela n'a pas beaucoup d'importance.

Pensez à étamer abondamment les pistes par lesquelles circulera le courant des sorties à triacs. En effet, comme nous vous l'avons déjà fait remarquer, le courant circulant dans CN₂ pourra atteindre 4A. Dans ce cas de figure, il très est important de bien étamer les pistes. Ne perdez pas de vue que le montage n'est pas prévu pour piloter des charges de plus de 200W par sortie, même si les triacs utilisés sont théoriquement capables de piloter des charges bien plus puissantes. La limitation vient de la largeur des pistes du circuit imprimé. Soyez donc raisonnables.

Comme nous l'avons déjà mentionné, nous avons fait appel à des connecteurs RJ11 pour la connexion du bus CAN. Ce choix est très pratique pour des manipulations sur table mais ce n'est pas le choix le plus judicieux si l'on souhaite raccorder le montage à une paire différentielle (ce qui est préférable pour établir un dialogue sur de longues distances). N'hésitez pas à revoir l'implantation du circuit imprimé si vous souhaitez utiliser ce montage pour dialoguer avec des équipements situés à plusieurs centaines de mètres de distance. Dans ce cas de figure, une connectique de type SubD est à conseiller.

Le microcontrôleur U₂ sera programmé avec le contenu d'un fichier qui est inclus sur le CD Rom qui accompagne ce numéro. Le fichier "CANIO.BIN" est le reflet binaire du programme du microcontrôleur tandis que le fichier "CANIO.HEX" correspond au format HEXA INTEL. Selon le modèle de programmeur dont vous disposez, vous utiliserez l'un ou l'autre des fichiers.

Le montage est relativement simple à utiliser, en particulier avec l'interface RS232 / Bus CAN publiée dans ces pages. La vitesse de transmission du montage est fixée par le logiciel du microcontrôleur à 20 kbits/s, afin de pouvoir dialoguer sur un bus CAN de plusieurs centaines de mètres de longueur. Cette valeur ne peut pas être modifiée pour ce montage, ce qui ne devrait pas être vraiment gênant dans le cadre d'une utilisation domotique, telle que nous l'avons conçue (le volume des données à échanger est faible et il n'est pas nécessaire d'avoir une fréquence de rafraîchissement élevée).

Selon la position des interrupteurs de SW₁, le montage répond à différents identificateurs sur le bus CAN. La figure 4 indique les 16 possibilités offertes par SW₁, ce qui

permet d'utiliser jusqu'à 16 montages de ce type sur le même bus. L'identificateur de base, selon la position de SW₁, correspond à une trame de donnée interprétée en réception par le montage. Seul le 1er octet de donnée de cette trame est interprété par le montage. La donnée reçue est tout simplement placée sur le port P1, sachant que les 4 bits de poids faible sont masqués automatiquement par le programme du montage pour ne pas interférer avec l'état des interrupteurs qui seront reliés à CN₉. La **figure 5** indique les différentes actions reconnues par le montage en fonction de l'identificateur de la trame tandis que la **figure 6** indique la valeur de la donnée à transmettre au montage pour obtenir la commande des sorties. Notez que, lorsque le montage reçoit une trame ayant pour identificateur la valeur Base + 1 (selon SW₁), le montage acquiert l'état de ses entrées et transmet le résultat dans une trame ayant pour identificateur la valeur de base + 2. Les données éventuelles de la trame qui



Le PCA 82C250 "driver" de ligne

déclenche la lecture de l'état des entrées ne sont pas utilisées. Le plus simple est d'émettre une trame sans aucune donnée associée (DLC=0). Pour l'opération de lecture, nous aurions pu utiliser une trame de réponse en exploitant le bit RTR de l'identificateur, mais nous avons préféré utiliser des identificateurs dis-

tincts pour être compatible avec une application particulière que l'auteur utilise pour réaliser ses tests. Avec le circuit SJA1000, ceci n'est pas pénalisant car cela revient à utiliser une partie des trames interceptées par le SJA1000 en raison des règles de filtrage qui sont appliquées.

P. MORIN

MICROS ROBOTS

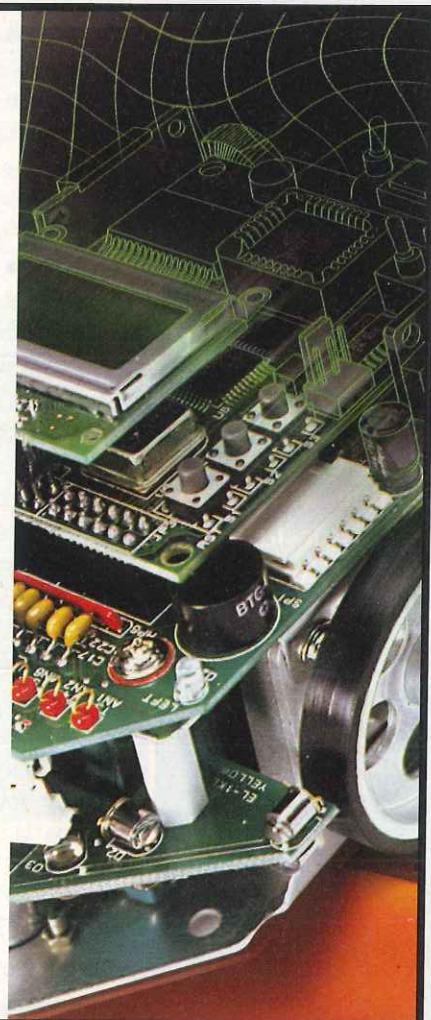
Entreprises, clubs, enseignants, étudiants, particuliers, créateurs d'événementiels... la robotique est votre métier **et /ou** votre passion.

Notre rubrique «sur le marché» est faite pour vous. N'hésitez pas à communiquer vos nouveaux produits, manifestations, événements... ayant trait à la robotique auprès de :

Rédaction MICROS & ROBOTS :
Bernard Fighiera - tél. : 01 44 84 84 65

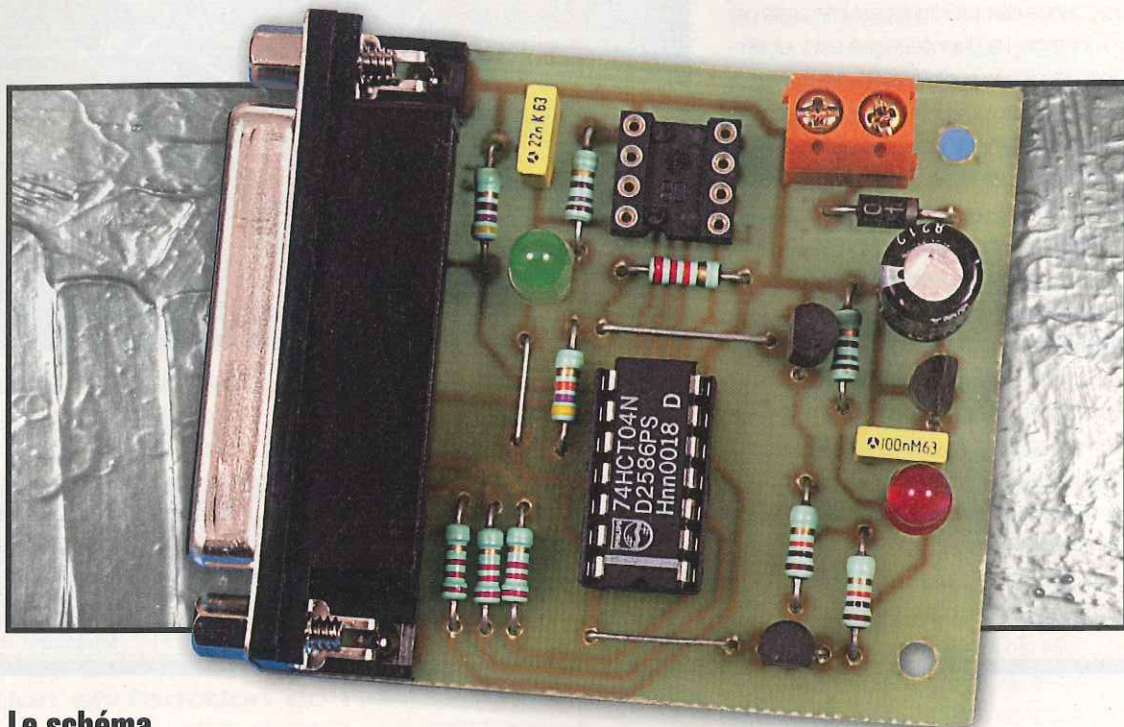
Annonces, vous souhaitez promouvoir vos produits ou services auprès d'un lectorat passionné :

Publicité MICROS & ROBOTS :
Pascal Declerck - tél. : 01 44 84 84 92
e-mail : lehp@le-hp.com



Programmateur

de mémoire EEPROM



Bien que rudimentaire, ce petit programmeur de mémoire EEPROM, 93C06 ou 93C46, n'en est pas moins efficace. De plus, il est très économique et facile à réaliser. Il se connecte au port parallèle LPT1 d'un PC. Son logiciel, utilisable sous DOS, est disponible sur le site de la revue (<http://www.eprat.com>).

Le schéma

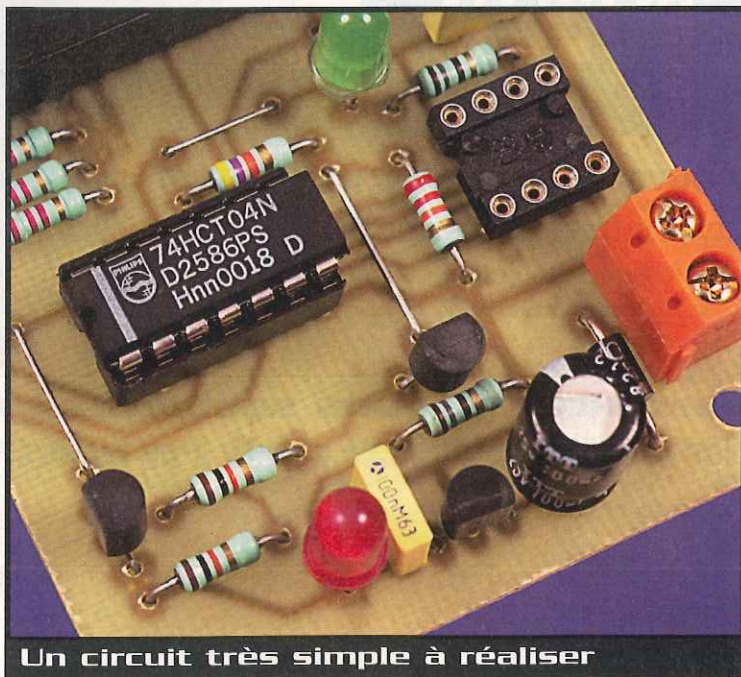
Quatre lignes du port parallèle sont nécessaires au dialogue avec la mémoire. Un tampon inverseur est placé sur chacune de ces lignes pour remettre en forme les signaux. Ces inverseurs sont contenus dans un circuit intégré TTL, de technologie HCT (**figure 1**). Les résistances R_7 à R_9 protègent quelque peu le port parallèle. Lorsque le programmeur n'est pas alimenté ou lorsque le logiciel n'est pas utilisé, la résistance R_{1C} abaisse le potentiel de la ligne ACKNOWLEDGE, broche 10 du port parallèle, pour le cas où une résistance de rappel serait placée sur la ligne.

L'alimentation de Cl_1 et de la mémoire à programmer est de 5V, tension régulée par Cl_2 , un régulateur de tension fixe 78L05. Ce dernier dispose d'une limitation en courant, laquelle protégera la mémoire en cas d'insertion incorrecte. La diode D_1 protège le montage contre une inversion de la source d'alimentation, laquelle peut être un adaptateur secteur ou une pile de 9V. Lorsque la carte est sous tension, la diode électroluminescente de couleur rouge est

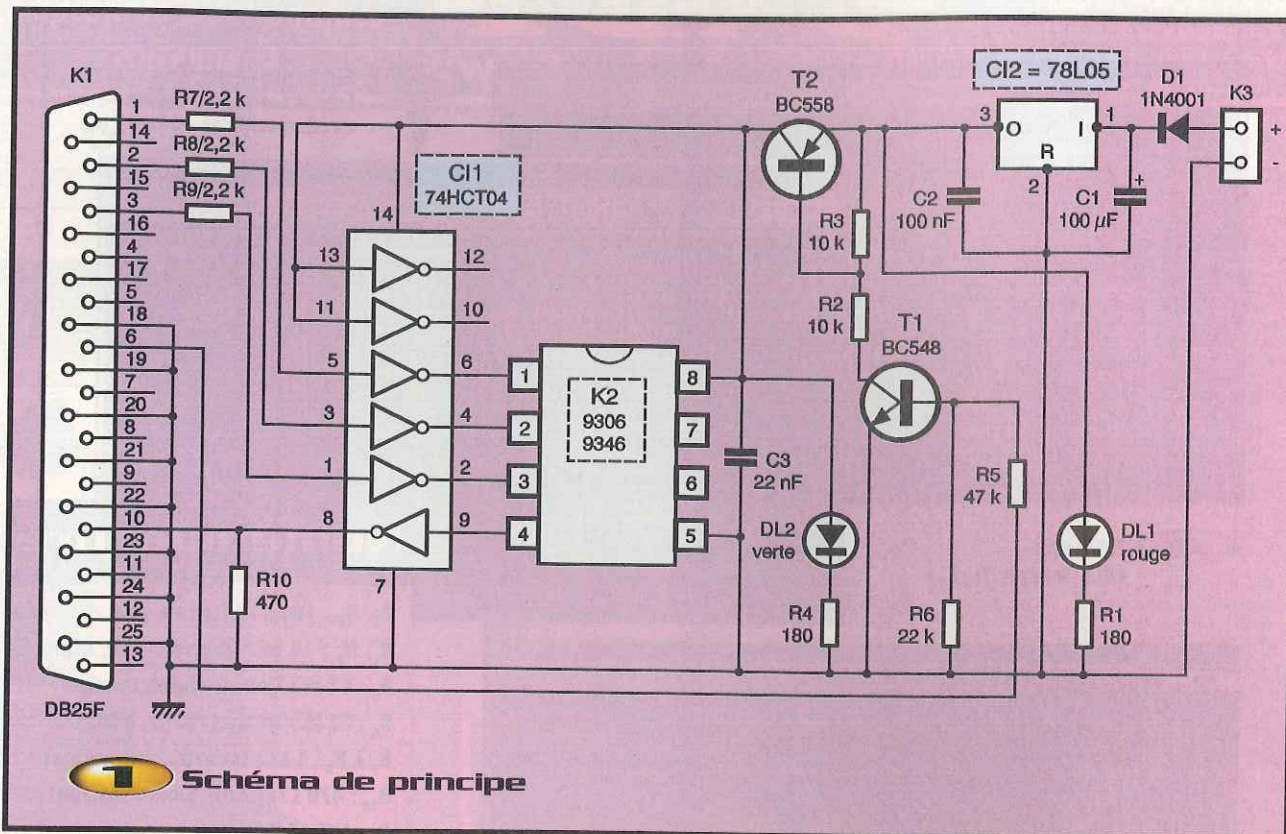
allumée et son courant de polarisation est limité à environ 20mA par la résistance R_5 .

La mémoire est mise sous tension par le logiciel. En effet, dès son lancement, la ligne DATA 5, broche 6 de K_1 (embase DB25 du port parallèle) est mise à un niveau logique haut. Dès lors, le transistor T_1 est passant,

ce qui entraîne la saturation du transistor T_2 , utilisé en commutation. La mémoire et Cl_1 reçoivent alors la tension régulée par le régulateur. De même, lorsque l'on quitte le logiciel, la ligne DATA 5 repasse à un état bas logique et le transistor T_1 n'est alors plus passant. Le transistor T_2 se



Un circuit très simple à réaliser



bloque. Il est ainsi équivalent à un interrupteur ouvert et les circuits intégrés ne sont plus alimentés.

La réalisation

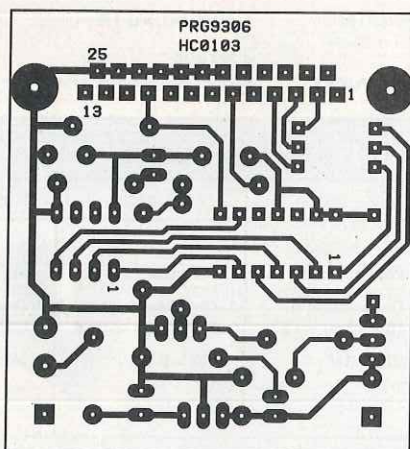
Le dessin du circuit imprimé (figure 2) peut facilement être réalisé par une quelconque méthode. Veillez surtout à bien net-

toyer le circuit imprimé après sa gravure et à le vernir après avoir achevé l'implantation des composants de la figure 3.

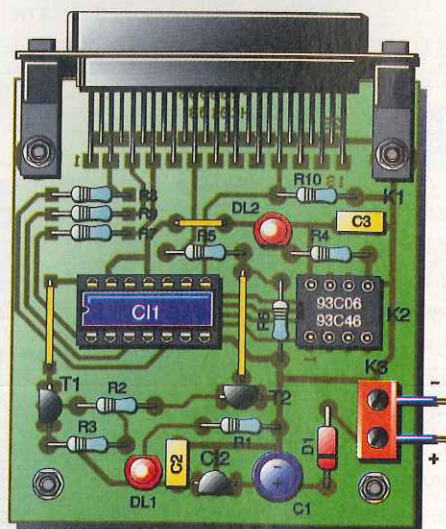
Tests de la carte

A la mise sous tension, la DEL rouge doit s'allumer, signalant ainsi que la régulation 5V de la carte est opérationnelle. Toutefois, sa

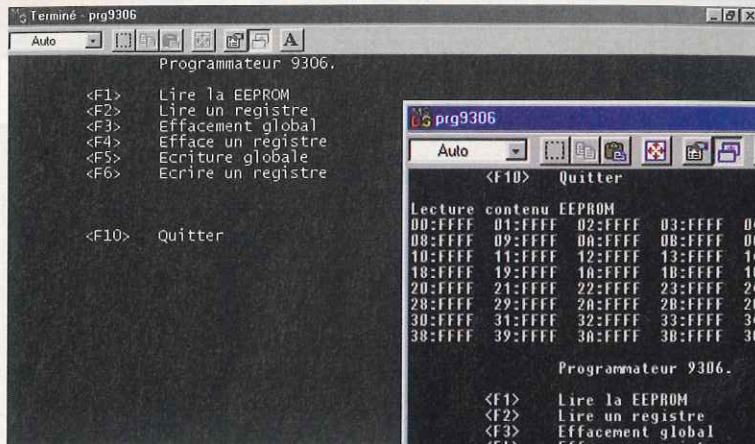
brillance doit être correcte. Si elle semble faible, la source de tension connectée au connecteur K₃ est sans doute trop faible. Pour rappel, un régulateur tel qu'un 78L05 ou 7805 nécessite, pour que sa régulation soit efficace, une tension d'entrée supérieure de 3V à sa tension de sortie. Dans le cas présent et compte tenu de la présence en série de la diode D₁, la tension minimale



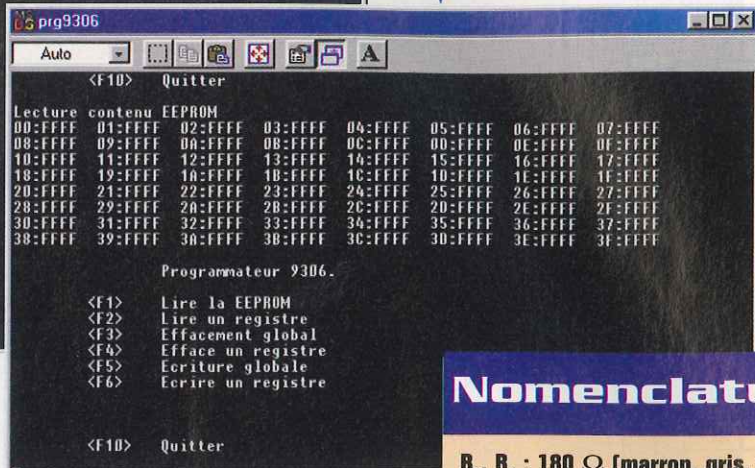
2 Tracé du circuit imprimé



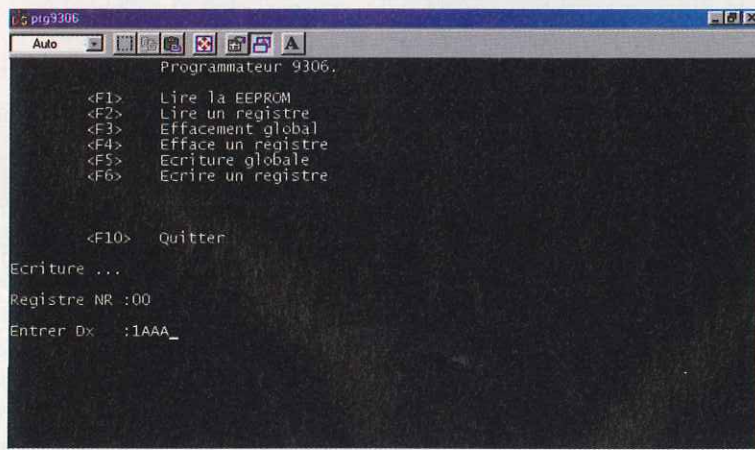
3 Implantation des éléments



4 Menu du logiciel



5 Écran après une lecture d'une mémoire vierge



6 Écran lors de la programmation d'un registre

Nomenclature

- R₁, R₄ : 180 Ω (marron, gris, marron)
- R₂, R₃ : 10 kΩ (marron, noir, orange)
- R₅ : 47 kΩ (jaune, violet, orange)
- R₆ : 22 kΩ (rouge, rouge, orange)
- R₇, R₈ : 1 kΩ (marron, noir, rouge)
- R₁₀ : 470 Ω (jaune, violet, marron)
- C₁ : 100 µF/25V
- C₂ : 100 nF
- C₃ : 22 nF
- D₁ : 1N4001..07
- DL₁ : DEL rouge Ø5
- DL₂ : DEL verte Ø5
- T₁ : BC548, BC547
- T₂ : BC558, BC557
- CI₁ : 74HCT04
- CI₂ : 78L05
- K₁ : embase DB25 femelle
- K₂ : support tulipe DIL8
- K₃ : bornier 2 plots
- 1 support DIL14
- 3 straps

de la source d'alimentation doit être d'environ 9V.

Le second test consiste à constater que la DEL verte s'allume, dès le lancement du logiciel "prg9306.exe" et qu'elle s'éteint lorsque l'on quitte le programme (touche F10).

Ensuite, placez la mémoire sur le support DIL8 et lancez à nouveau le logiciel, l'écran est alors celui de la **figure 4**. Effectuez une lecture de la mémoire et si votre mémoire est vierge, vous devez alors observer l'écran de la **figure 5**.

En appuyant sur la touche F6, vous pourrez procéder à la programmation d'un registre, par exemple le 00 avec la valeur hexadécimale 1AAAh (**figure 6**).

Lors d'une lecture d'une mémoire, l'ensemble des lignes à l'écran donne le contenu d'une mémoire 9346, de capacité plus importante qu'une 9306. Dès lors, avec les mémoires 9306, seules les deux

premières lignes, concernant les registres 00 à 0F doivent être pris en considération. Les principales caractéristiques de ces mémoires sont rappelées par le résumé de la **figure 7**.

H. CADINOT

		EEPROM (Mémoire programmable et effaçable électriquement) 1 million de cycles d'écriture / effacement		
Brochage		9306, 93C06	9346, 93C46	9347, 93C47
S	1		- 256 bits	- 1024 bits
C	2		- Organisation : 16 x 16 ou 8 x 32	- Organisation : 64 x 16 ou 128 x 8
D	3		- Vcc = 5 V	- Vcc = 5 V
Q	4			- Vcc = 3 à 5 V
		- S = Entrée de sélection de la mémoire	- Vcc = tension d'alimentation	
		- C = Horloge	- DU : ne pas utiliser	
		- D = Entrée des données séries	- ORG : organisation par octets (ORG = 0) ou mots longs (ORG = 1 ou libre).	
		- Q = Sortie série des données	- Vss = Masse	

7 Caractéristiques des mémoires 9306 et 9346