

# INTERFACES ET DÉVELOPPEMENTS PC

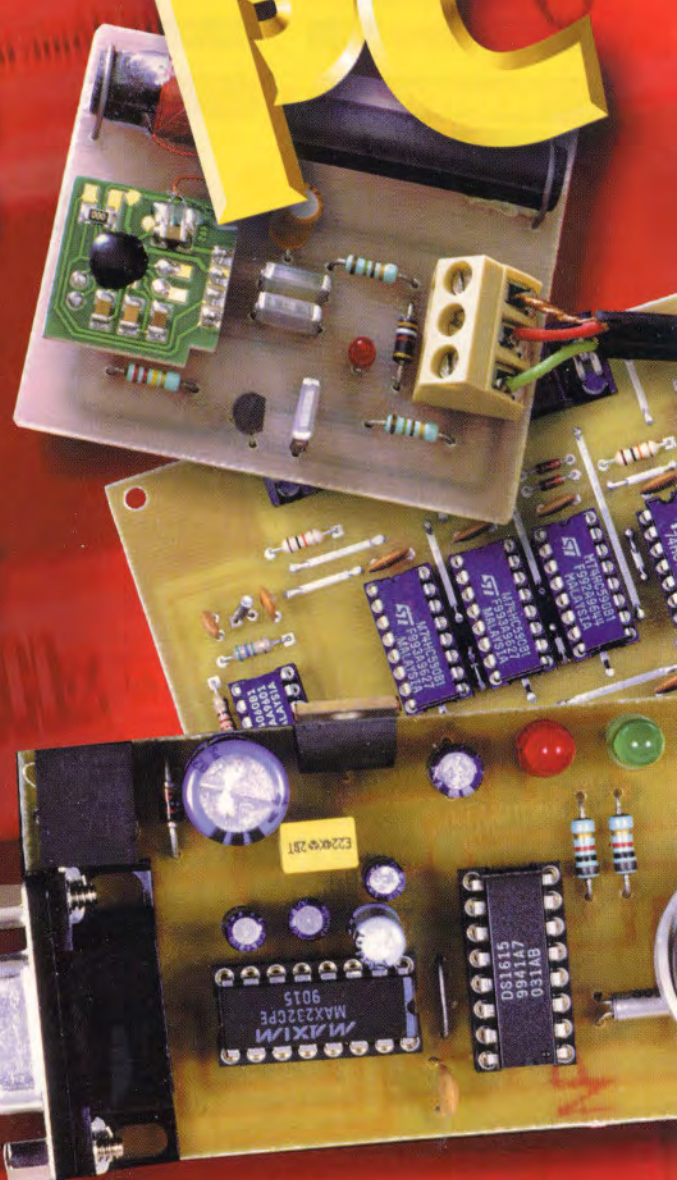
## Réalisez vous-même 16 CARTES

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



**CD ROM  
INCLUS**

- **Fréquencemètre 1GHz,**
- **Récepteur de signaux horaires**
- **Horloge Internet**
- **Enregistreur de température**
- **etc.**



# Le port parallèle du PC



Le port parallèle du PC est l'un des plus intéressants de par ses possibilités et sa simplicité de programmation, contrairement au port série qui nécessite un protocole bien précis. Bien que réservé à l'origine pour la connexion d'imprimantes, il est de plus en plus utilisé pour ses capacités d'entrées-sorties à très haute vitesse.

On peut ainsi lui connecter divers périphériques tels que les scanners, les lecteurs de disquettes à très haute capacité, les disques durs, mais aussi des instruments de mesures et des cartes électroniques diverses. Dans cet article nous vous proposons de découvrir les possibilités de cette interface.

Dans la première partie, nous décrivons les caractéristiques des différents ports existants sur le PC, tandis que dans la seconde nous décrivons le principe de cartes simples d'entrées-sorties.

Sans prétendre aborder tous les aspects de cette interface (ce qui nécessiterait un petit livre), nous pensons que ces quelques pages pourront aider les nombreux lecteurs désireux de la mettre en œuvre.

Le port parallèle dispose de nombreuses lignes : huit lignes de données, quatre lignes de contrôle et cinq lignes de statut qui sont des entrées et qui permettent de vérifier l'état de l'imprimante, ainsi que plusieurs lignes de masse.

Le **tableau 1** représenté ci-contre donne les noms et la position de ces différentes lignes :

Le dessin de la **figure 1** montre le câblage d'un cordon Centronics qui utilise un connecteur SUBD 25 broches et un connecteur Centronics de 36 broches.

Le tableau 1 est en fait valable pour le port parallèle de type compatible (unidirectionnel) et non bidirectionnel. Le rôle du port unidirectionnel du premier PC était uniquement d'envoyer des

données à l'imprimante, ce qui justifiait le fait que ces données ne circulent que dans un sens. Ce qui varie principalement entre les deux types de ports est que les lignes de données deviennent bidirectionnelles dans le second.

Il existe différents types de liaison parallèle, celles-ci étant définies dans le standard IEEE 1284 :

- mode compatible (SPP ou Standard Parallel Port),
- mode 4 bits ou NIBBLE MODE (unidirectionnel),
- mode 8 bits ou BYTE MODE (bidirectionnel),
- mode EPP (Enhanced Parallel Port ou port parallèle étendu),
- mode ECP (Extended Capability Port ou port à capacités étendues).

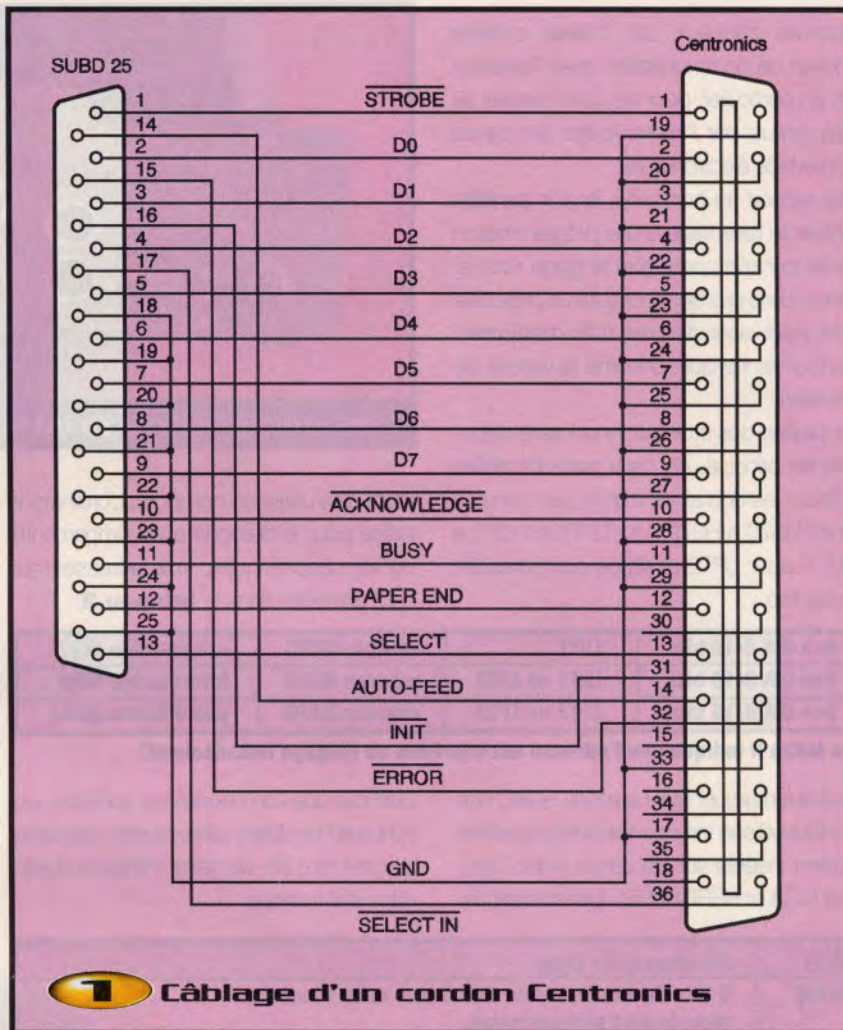
Broche SUBD 25	Broche Centronics	Fonction	Direction	Registre
1	1	STROBE	sortie	contrôle
2	2	donnée D0	sortie	donnée
3	3	donnée D1	sortie	donnée
4	4	donnée D2	sortie	donnée
5	5	donnée D3	sortie	donnée
6	6	donnée D4	sortie	donnée
7	7	donnée D5	sortie	donnée
8	8	donnée D6	sortie	donnée
9	9	donnée D7	sortie	donnée
10	10	ACKNOWLEDGE	entrée	statut
11	11	BUSY	entrée	statut
12	12	PAPER OUT	entrée	statut
13	13	SELECT	entrée	statut
14	14	AUTO FEED	sortie	contrôle
15	32	ERROR	entrée	statut
16	31	INITIALIZE	sortie	contrôle
17	36	SELECT PRINTER	sortie	contrôle
18-25	19-30	GROUND	masse	

**T1** Tableau 1

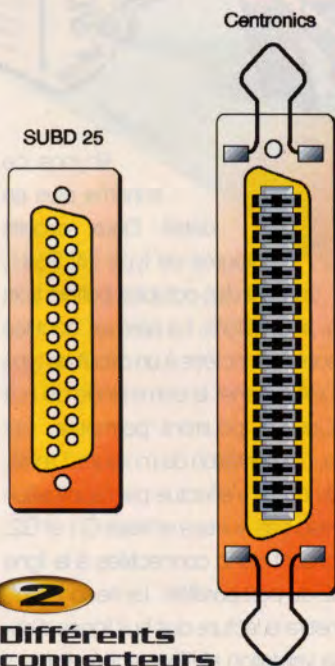
## Caractéristiques des ports parallèles

Nous donnons maintenant les caractéristiques des différents ports, en ne citant que les principales, puisque nous verrons plus loin que les données nécessaires à leur programmation ne varient pas. Lorsque nous disons données, nous parlons des adresses qu'ils occupent dans les machines, aussi bien dans la mémoire des entrées/sorties que dans le BIOS.

- le type unidirectionnel en 4 bits qui permet une entrée de données sur 4 bits, la sortie se faisant en 8 bits comme pour le port standard,
- le type 1 bidirectionnel en huit bits (type PS2 bidirectionnel) conçu par IBM pour son ordinateur PS2 qui permet à cette machine de communiquer avec des périphériques grâce aux huit lignes de données, un bit supplémentaire permettant de signaler dans quelle direction se fait l'échange. Cet échange atteint des vitesses comprises entre 60 ko/s et 300 ko/s,
- le type 3 bidirectionnel en huit bits (DMA



**1** Câblage d'un cordon Centronics



## 2 Différents connecteurs

8 bits ou Direct Access Memory 8 bits) qui a équipé les PS2 IBM de type 57, 90 et 95. Ce port est d'une rapidité élevée puisqu'il utilise le DMA ou accès direct à la mémoire,

- le type étendu EPP qui a été développé par plusieurs sociétés dont INTEL et qui offre une vitesse de fonctionnement très

élevée. Il permet des transferts d'environ dix fois plus importants qu'un port parallèle de l'ancienne génération, puisqu'il permet des débits atteignant la vitesse de 1 Mo à 2 Mo par seconde. C'est pour cette raison que l'on a pu y connecter divers périphériques dont les disques durs,



- le type à capacités étendues ECP conçu, entre autres, par Microsoft et qui offre également une très grande vitesse, nécessite l'utilisation d'un canal DMA (accès direct à la mémoire) non utilisé ordinairement par le port parallèle. Cependant, les PC vendus actuellement permettent la configuration du ou des ports parallèles en mode EPP ou ECP. On choisira de préférence, en pénétrant dans le BIOS (setup), la configuration EPP qui est en principe la plus compatible avec les périphériques existants sur le marché. Il est à signaler que ces deux derniers

ports (EPP et ECP) requièrent l'utilisation de circuits intégrés logiques spéciaux afin d'assurer l'envoi et la réception de données.

Le standard IEE 1284 concernant les signaux échangés entre PC et périphériques définit les caractéristiques physiques des ports parallèles bidirectionnels des PC. Cette norme définit l'échange entre l'ordinateur et les différents périphériques qui lui sont connectables, aussi bien les imprimantes que les divers matériels comme les disques durs, périphériques de réseaux, lecteurs de bandes, etc. La **figure 2** montre les différents connecteurs utilisés pour la connexion d'imprimantes ou la connexion entre deux ordinateurs selon le standard IEEE 1284. On peut ainsi espérer obtenir des débits d'environ 100 Mbits/s.

## L'utilisation du port parallèle

Ce paragraphe tentera d'aborder la programmation simple du port parallèle par les

lecteurs désireux de l'utiliser comme moyen de communication avec l'extérieur et, en particulier, pour les commandes de processus par l'intermédiaire de cartes d'interface électroniques.

Par rapport au port série, le port parallèle présente l'avantage d'une programmation facile par n'importe quel langage évolué, alors que le premier nécessite un protocole très strict pour des résultats médiocres, surtout en ce qui concerne la vitesse de transfert.

La plupart des ordinateurs peuvent disposer, en principe, de deux ports parallèles utilisant les domaines d'adresses compris entre H370 à H37F pour LPT1 et H270 à H27F pour LPT2 et situés aux adresses suivantes :

bus ISA 8/16 bits	LPT1	adresse H3BC	interruption IRQ7
bus ISA 8/16 bits	LPT1 ou LPT2	adresse H378	interruption IRQ5
bus ISA 8/16 bits	LPT2 ou LPT3	adresse H278	pas d'interruption

(la lettre H indique que l'adresse est exprimée en langage hexadécimal)

Il est à remarquer que l'adresse H3BC était surtout utilisée lorsque les ports parallèles étaient insérés sur les cartes vidéo (couleur CGA et HERCULES). Il n'est pratique-

ment plus utilisé de nos jours. L'ordinateur utilise pour le dialogue avec l'imprimante ou les périphériques, trois adresses qui sont données dans le **tableau 2** :

Dès lors que l'on connaît les adresses du port que l'on désire utiliser, il sera aisé de le programmer afin de gérer n'importe quelle carte électronique.

Port	Fonction de la ligne
H378	8 bits de données pour l'émission ou la réception avec le port bidirectionnel
H379	lecture des lignes de contrôle d'état Bit 3 : ERROR (erreur) Bit 4 : ON LINE (imprimante connectée) Bit 5 : PAPER END (plus de papier) Bit 6 : ACKNOWLEDGE (acquiescement) Bit 7 : BUSY (imprimante occupée)
H37A	Bit 0 : STROBE Bit 1 : AUTOFEED Bit 2 : INITIALISATION Bit 3 : SELECT Bit 4 : AUTORISATION D'INTERRUPTION Bit 5 : mis à 0, écriture des données et mis à 1, lecture des données (ce bit n'étant adressable que sur les ports bidirectionnels)
H278	8 bits de données pour l'émission ou la réception avec le port bidirectionnel
H279	lecture des lignes de contrôle d'état Bit 3 : ERROR (erreur) Bit 4 : ON LINE (imprimante connectée) Bit 5 : PAPER END (plus de papier) Bit 6 : ACKNOWLEDGE (acquiescement) Bit 7 : BUSY (imprimante occupée)
H27A	Bit 0 : STROBE Bit 1 : AUTOFEED Bit 2 : INITIALISATION Bit 3 : SELECT Bit 4 : AUTORISATION D'INTERRUPTION Bit 5 : mis à 0, écriture des données et mis à 1, lecture des données (ce bit n'étant adressable que sur les ports bidirectionnels)

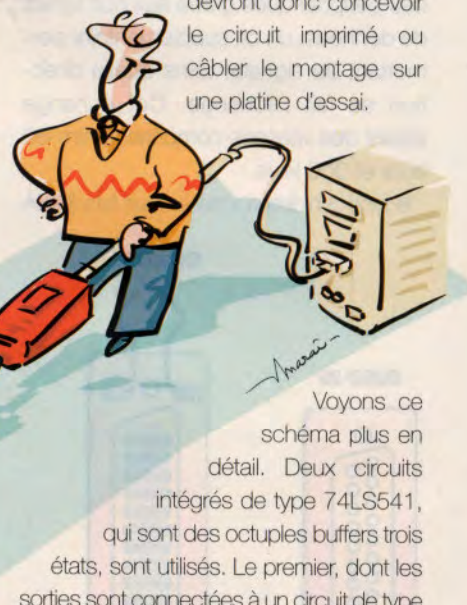
**T2** Tableau 2



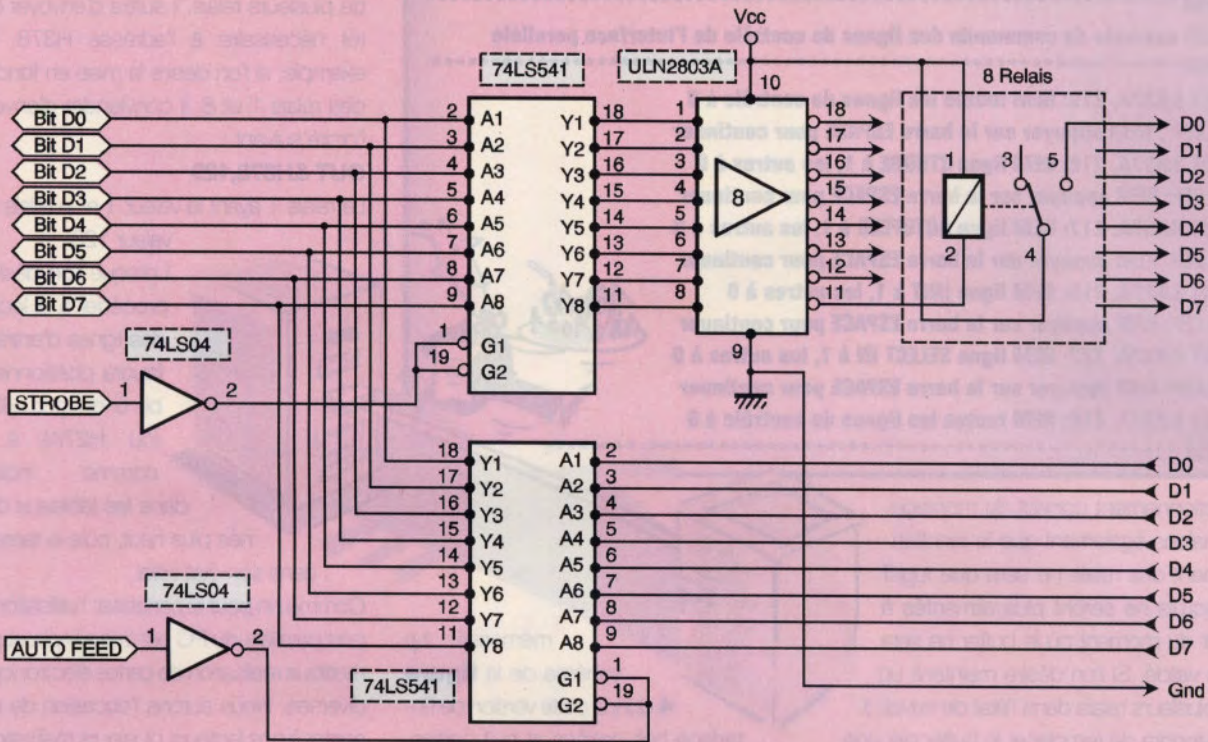
Les différentes prises de raccordement

Le dessin de la **figure 3** donne un exemple simple d'interface permettant la commande de huit relais (de puissance ou non) et la lecture de huit entrées. Signalons que nous ne donnons ce schéma qu'à titre d'exemple et que les lecteurs intéressés

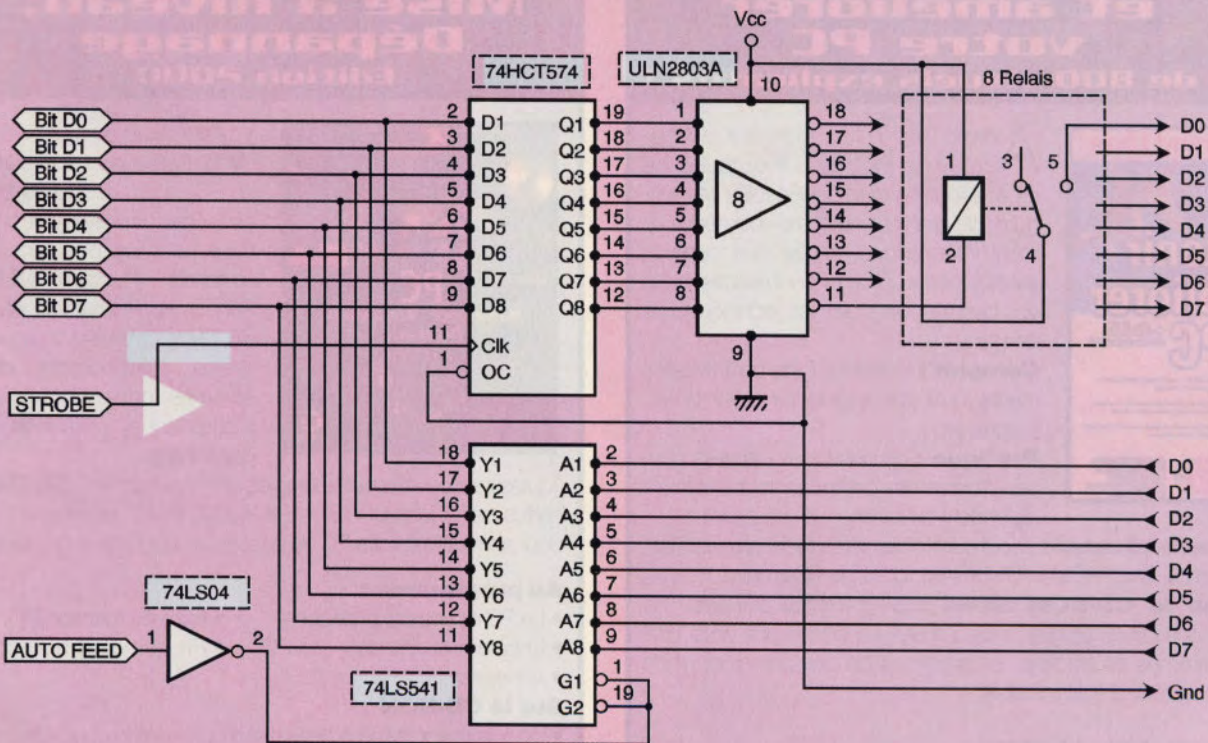
devront donc concevoir le circuit imprimé ou câbler le montage sur une platine d'essai.



Voyons ce schéma plus en détail. Deux circuits intégrés de type 74LS541, qui sont des octuples buffers trois états, sont utilisés. Le premier, dont les sorties sont connectées à un circuit de type ULN2803A, permet la commande de huit relais. Ceux-ci pourront permettre, par exemple, l'alimentation de moteurs. La validation du circuit s'effectue par l'application d'un niveau bas sur ses entrées G1 et G2. Celles-ci sont donc connectées à la ligne STROBE du port parallèle. Le second buffer permettra la lecture des huit lignes d'entrées, sa validation s'effectuant également par l'application d'un niveau bas au moyen de la ligne AUTO FEED sur ses entrées de contrôle. Il conviendra, ce qui n'est pas représenté sur le schéma, de placer des résistances de rappel d'une valeur de 10 kΩ à la masse en sortie du premier buffer. On connectera des résistances de même valeur entre les entrées du second et le + de l'alimentation afin d'obtenir un



**3** Exemple d'interface permettant la commande de huit relais



**4** Interface 8 entrées / 8 sorties

```

REM *****
REM exemple de commande des lignes de contrôle de l'interface parallèle
REM *****
OUT &H37A, 219: REM toutes les lignes de contrôle à 0
SLEEP : REM appuyer sur la barre ESPACE pour continuer
OUT &H37A, 218: REM ligne STROBE à 1, les autres à 0
SLEEP: REM appuyer sur la barre ESPACE pour continuer
OUT &H37A, 217: REM ligne AUTOFEED à 1, les autres à 0
SLEEP: REM appuyer sur la barre ESPACE pour continuer
OUT &H37A, 215: REM ligne INIT à 1, les autres à 0
SLEEP: REM appuyer sur la barre ESPACE pour continuer
OUT &H37A, 227: REM ligne SELECT IN à 1, les autres à 0
SLEEP: REM appuyer sur la barre ESPACE pour continuer
OUT &H37A, 219: REM toutes les lignes de contrôle à 0
REM *****
    
```

fonctionnement correct du montage. Signalons également que le fonctionnement des relais ne sera que fugitif puisqu'ils ne seront plus alimentés à partir du moment où le buffer ne sera plus validé. Si l'on désire maintenir un ou plusieurs relais dans l'état de travail, il conviendra de remplacer le buffer par une octuple bascule de type 74LS574 dont l'entrée CLOCK sera connectée à la ligne STROBE. L'ordre envoyé sur le bus de données de l'interface parallèle sera ainsi



mémorisé. Le schéma de la **figure 4** donne cette version de l'interface huit entrées et huit sorties.

Le petit programme donné ci-dessous permet de commander les lignes de contrôle de la sortie imprimante du PC. Afin de commander l'alimentation d'un ou

de plusieurs relais, il suffira d'envoyer l'octet nécessaire à l'adresse H378. Par exemple, si l'on désire la mise en fonction des relais 1 et 8, il conviendra d'envoyer l'ordre suivant :

**OUT &H378,129**

Le relais 1 ayant la valeur 1 et le relais 8 la valeur 128.

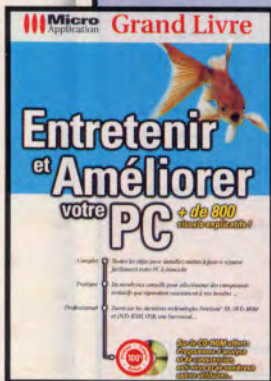
Lorsque l'on désirera procéder à la lecture des lignes d'entrée, il faudra positionner le bit 5 du port H37A (ou H27A) à 1, comme indiqué dans les tableaux donnés plus haut, puis le remettre dans son état initial.

Comme on peut le constater, l'utilisation du port parallèle du PC est très simple et permettra la réalisation de cartes électroniques diverses. Nous aurons l'occasion de présenter à nos lecteurs plusieurs réalisations, des plus simples (entrées/sorties) aux plus complexes (cartes de mesures à convertisseurs, etc.).

P. OGUIC

## Entretien et améliorer votre PC

+ de 800 visuels explicatifs !



Avec le Grand Livre "Entretien et Améliorer votre PC", vous avez toutes les informations en main pour installer et configurer de manière optimale votre équipement informatique, choisir des composants évolutifs qui répondent exactement à vos besoins, découvrir les techniques de demain...

**Complet** : toutes les infos pour installer, mettre à jour et réparer facilement votre PC à domicile,

**Pratique** : de nombreux conseils pour sélectionner des composants évolutifs qui répondent exactement à vos besoins,

**Professionnel** : Zoom sur les dernières technologies Pentium, III, DVD-ROM et DVD-RAM, USB, son Surround...

**Sur le CD-ROM offert** : programmes d'analyse de disques durs, programmes d'analyse système, anti-virus, programmes de backup, programmes de compression, et de nombreux autres utilitaires...

J. ADAMS - M. KOCH  
 MICRO Application  
 20/22, rue des Petits-Hôtels - 75010 PARIS  
 T. : 01.53.34.20.20 - F. : 01.53.34.20.00  
 www.microapp.com

## PC Maintenance, Mise à niveau, Dépannage Edition 2000



Votre micro au top niveau ! Le livre que tout utilisateur de PC se doit absolument de posséder ! Entièrement optimisé pour les plus récents matériels (Pentium MMX, Pentium II, Pentium III, AMD K6, Cyrix...), ce livre vous permettra d'économiser des sommes considérables en maintenance, réparation et dépannage.

Toutes les nouvelles technologies enfin couvertes : USB, DVD, cartes mères Socket 7, Slot 1 et Slot M. Toutes les informations pour permettre à votre PC de passer l'an 2000 sans problème.

### Au programme :

- Le PC et ses périphériques,
- Entretien et optimiser son PC,
- Annexes.
- Faire évoluer son PC,
- Problèmes et solutions,

### Sur le CD-ROM :

- De nombreux utilitaires de test et d'optimisation pour votre PC.

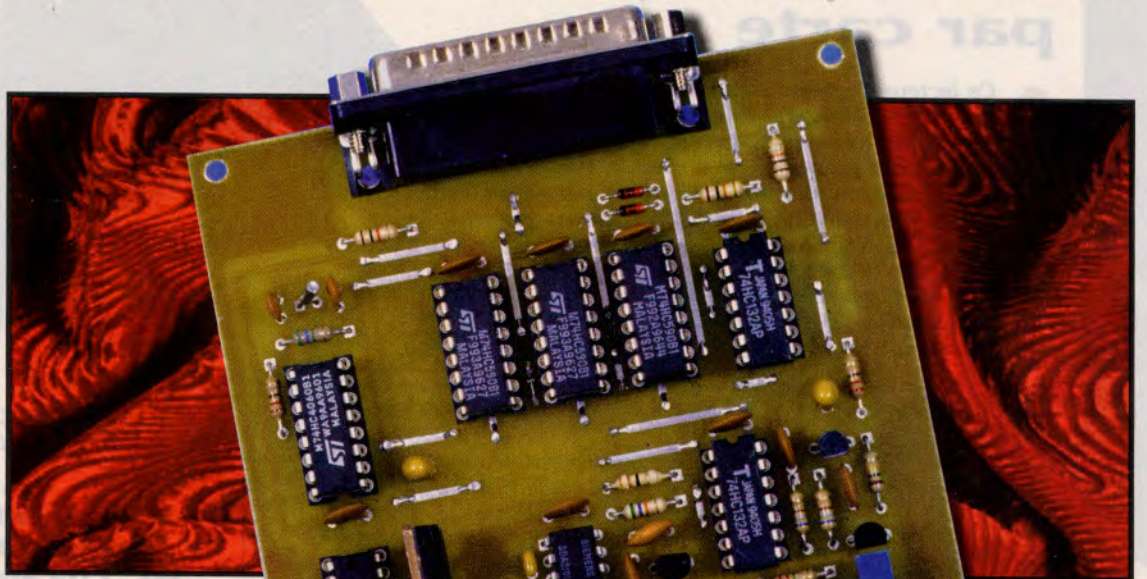
**Niveau** : Initié/Expérimenté

**Matériel** : PC/Windows 95/98 et NT4

M. MARTIN / SYBEX - www.sybex.fr

# Fréquencemètre 1 GHz

sur port parallèle



Certes, plusieurs réalisations de fréquencemètres ont déjà été décrites dans la revue. Abondance de choix ne nuisant point, surtout en matière de moyens de mesure, le fréquencemètre décrit dans ces lignes est intéressant par son faible coût, eu égard à ses possibilités, car il n'utilise que des composants bon marché. Pas de microcontrôleur et pas d'afficheurs, puisqu'il se connecte sur le port parallèle d'un PC.

En effet, de plus en plus, le PC trône dans l'atelier de l'amateur électronique et les occasions ne manquent pas de mettre à profit la polyvalence de cet outil omniprésent.

C'est une fois de plus le port parallèle du PC qui est mis à contribution dans cette application, car son utilisation est aisée et qu'il peut facilement être doublé ou triplé, pour un prix modique, grâce aux diverses cartes d'extension que l'on peut trouver sur le marché de l'informatique, permettant ainsi de réserver un port à l'imprimante et un autre aux expérimentations.

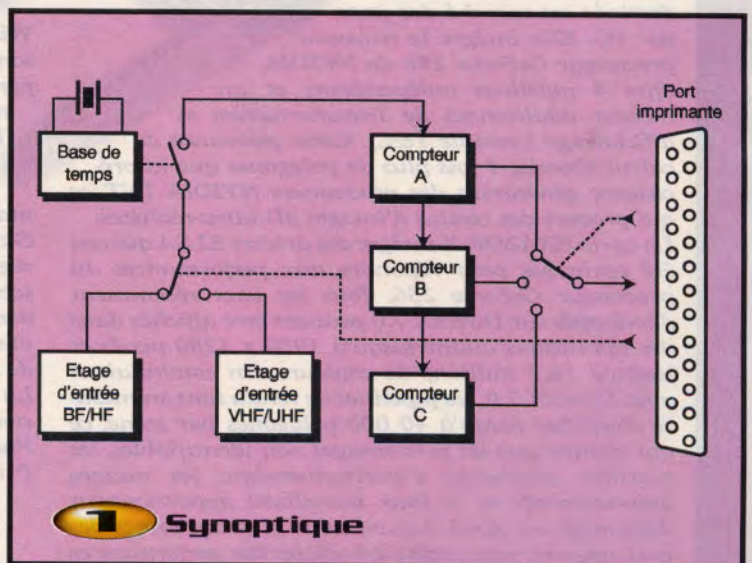
Le principe de fonctionnement du fréquencemètre, des plus classiques, consiste à compter le nombre de périodes d'un signal dans un temps donné. Il est illustré par le synoptique de la **figure 1**. L'étendue des fréquences mesurables se partage en deux entrées. Une entrée HF à haute impédance permet la mesure des fréquences de quelques hertz à une quinzaine de MHz. Une autre entrée, qui met en œuvre un pré-diviseur UHF par 64, permet de mesurer des fréquences jusqu'au GHz. Ces deux

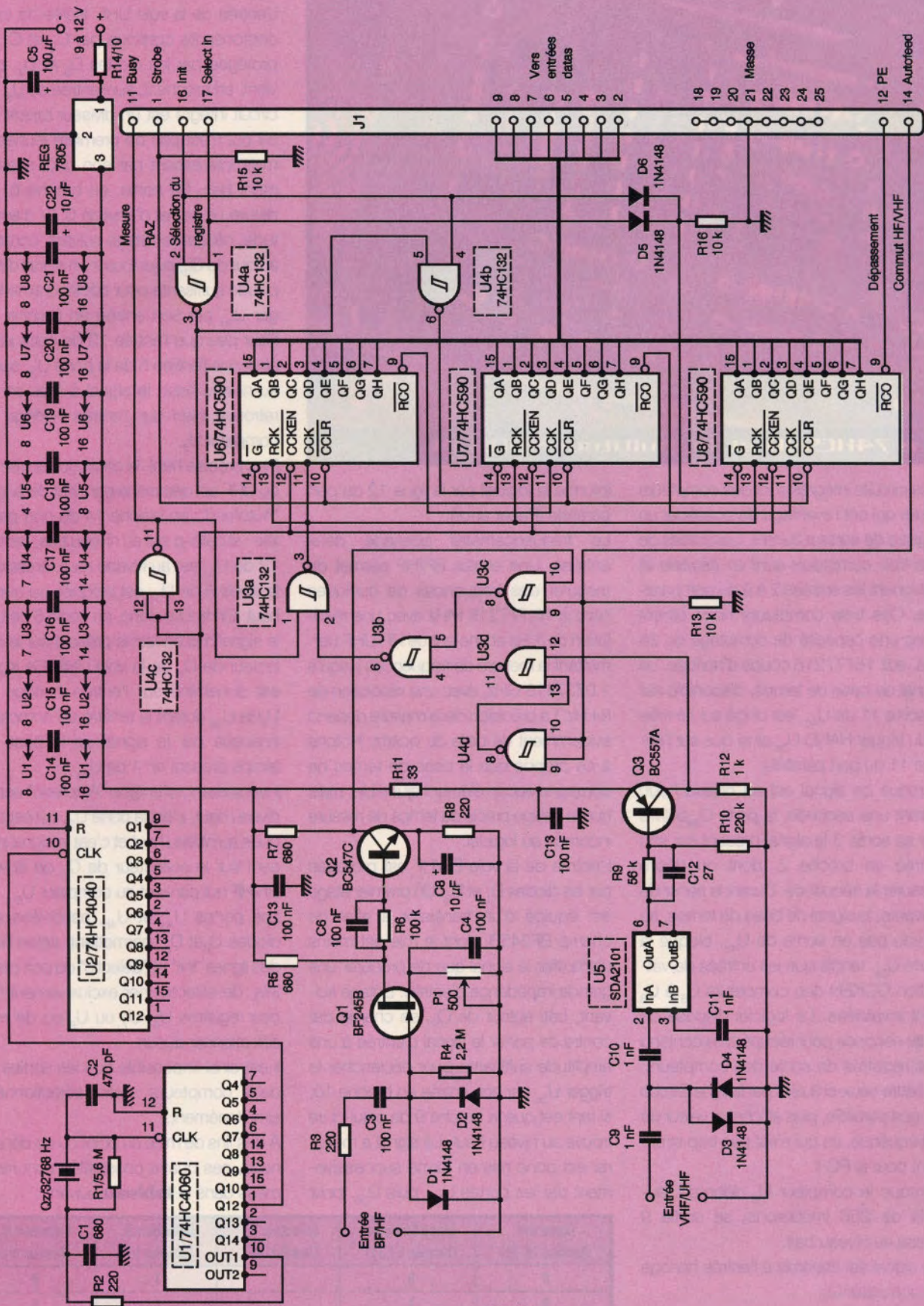
entrées sont commutées par logiciel. L'alimentation du montage sera assurée par un bloc secteur 9V/0,3A.

## Schéma électrique (figure 2)

La base de temps est élaborée à partir d'un quartz à 32768 Hz que met en résonance l'étage oscillateur

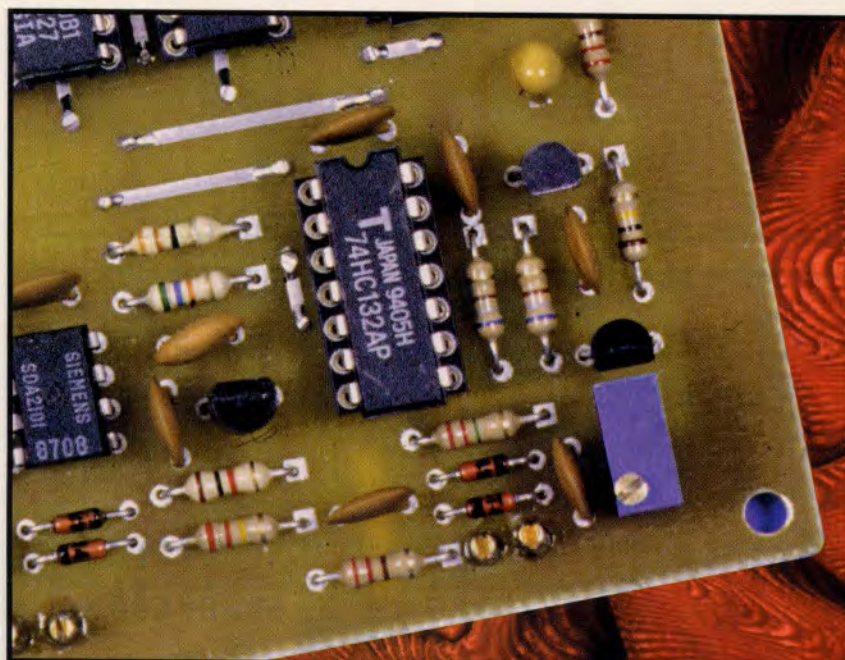
de  $U_1$  (74HC4060) dont la sortie Q12 est connectée à l'entrée horloge de  $U_2$  (74HC4040) duquel on utilise la sortie Q4. La fréquence de départ se trouve donc divisée par 4096, puis par 16, soit 65536. La fréquence qui en résulte est donc de  $32768 / 65536 = 0,5$  Hz, soit une période de deux secondes. Ce signal est acheminé, d'une part vers le trigger NAND  $U_{4c}$  (74HC132) et, d'autre part vers les entrées de validation de comptage des boîtiers  $U_6$  à  $U_8$  (74HC590) en broche 12.





2 Schéma de principe





**Le 74HC590 et le multitour 500Ω P1**

Ces circuits intégrés sont des compteurs 8 bits qui ont l'avantage de posséder un registre de sortie à 3 états. Les sorties de ces trois compteurs sont en parallèle et rejoignent les entrées 2 à 9 du port parallèle. Ces trois compteurs nous offrent donc une capacité de comptage de 24 bits, soit 16777216 coups d'horloge. Le signal de base de temps, disponible sur la sortie 11 de  $U_{4c}$ , est dirigé sur l'entrée 1 du trigger NAND  $U_{3a}$  ainsi que sur l'entrée 11 du port parallèle.

Lorsque ce signal est au niveau haut, durant une seconde, la porte  $U_{3a}$  délivre sur sa sortie 3 le signal présent sur son entrée en broche 2, dont on désire mesurer la fréquence. Durant la seconde suivante, le signal de base de temps, au niveau bas en sortie de  $U_{4c}$ , bloque la porte  $U_{3a}$ , tandis que les entrées de validation CCKEN des compteurs  $U_6$  à  $U_8$  sont invalidées. Le logiciel dispose de cette seconde pour récupérer le contenu des registres de sortie des compteurs, remettre ceux-ci à zéro par la ligne Strobe du port parallèle, puis afficher la valeur de la fréquence, ce qui n'est pas trop stressant pour le PC !

Lorsque le compteur  $U_6$  déborde, au-delà de 256 impulsions, sa sortie 9 passe au niveau bas.

Ce signal est transmis à l'entrée horloge du compteur  $U_7$ .

Si celui-ci déborde à son tour, sa sortie 9 incrémente le compteur suivant,  $U_8$ . Enfin, si ce dernier est saturé, il en

informe le logiciel par la ligne 12 du port parallèle (Paper error).

Le fréquencemètre possède deux entrées. Une entrée BF/HF permet de mesurer des fréquences de quelques hertz à 16,777216 MHz avec une résolution de 1 Hz et une entrée HF/UHF permettant la mesure de fréquences jusqu'à 1,0737418 GHz, avec une résolution de 64 Hz. La précision de la mesure dépend évidemment de celle du quartz. Notons à ce propos que la base de temps ne comporte aucun élément ajustable, mais que le calage précis du temps de mesure incombe au logiciel.

L'entrée de la voie BF/HF est protégée par les diodes  $D_1$  et  $D_2$ . Un premier étage est équipé d'un transistor à effet de champ BF245B, dont le rôle est moins d'amplifier le signal que de procurer une grande impédance d'entrée. L'étage suivant, bâti autour de  $Q_2$ , se charge par contre de porter le signal d'entrée à une amplitude suffisante pour déclencher le trigger  $U_{3c}$  par son entrée en broche 10, si tant est que la broche 9 de celui-ci se trouve au niveau haut. Le signal à mesurer est donc mis en forme successivement par les portes  $U_{3c}$ , puis  $U_{3b}$ , pour

aboutir sur l'entrée horloge du compteur  $U_6$  avec des fronts bien raides.

L'entrée de la voie UHF, isolée de toute composante continue par  $C_9$  et  $C_{10}$  et protégée par les diodes  $D_3$  et  $D_4$ , parvient, en broche 2, sur l'entrée de  $U_5$ . Ce circuit intégré est un diviseur rapide par 64 qui n'est pas de première jeunesse, mais intéressant par son coût relativement bas. Sa sortie, en broche 6, qui délivre un signal d'environ 0,7V d'amplitude, nécessite l'étage suivant, construit autour de  $Q_3$ , pour fournir un signal d'amplitude suffisante pour déclencher le trigger  $U_{3d}$  par son entrée en broche 12. Pour peu que l'entrée 13 de cette porte ainsi que l'entrée 5 de la porte  $U_{3b}$  soient au niveau haut, le signal à mesurer se retrouve bien sur l'entrée horloge du compteur  $U_6$ .

Plus précisément, le choix de l'entrée HF ou UHF est déterminé par l'état de la ligne "Autofeed", en broche 14 du port parallèle : si celle-ci est au niveau bas, l'entrée 13 de  $U_{3d}$  est au niveau haut tandis que la sortie 8 de  $U_{3c}$  est bloquée au niveau haut. On trouve donc, en sortie 6 de  $U_{3b}$ , le signal mis en forme présent sur le collecteur de  $Q_3$  de la voie UHF. Ce signal est acheminé sur l'entrée horloge de  $U_6$  via  $U_{3a}$  durant la fenêtre de comptage imposée par le signal de la base de temps présent en 1 de  $U_{3a}$ .

Inversement, si la ligne "Autofeed" est au niveau haut, c'est la porte  $U_{3d}$  qui est bloquée au niveau haut et c'est le signal présent sur le collecteur de  $Q_2$  de la voie BF/HF qui parvient au compteur  $U_6$ .

Les portes  $U_{4a}$  et  $U_{4b}$ , associées aux diodes  $D_5$  et  $D_6$ , permettent, selon l'état des lignes "Init" et "SelectIn" du port parallèle, de sélectionner exclusivement l'un des registres  $U_6$ ,  $U_7$  ou  $U_8$  ou de n'en sélectionner aucun.

Il est ainsi impossible que les sorties de deux compteurs soient sélectionnées simultanément.

A des fins de mise au point ou de dépannage, ces quatre possibilités sont résumées dans le **tableau** suivant :

Ligne Init (broche 16 J1)	Ligne SelectIn (broche 17 J1)	Sélection $U_6$ (broche 14)	Sélection $U_7$ (broche 14)	Sélection $U_8$ (broche 14)
0	0	1	1	0
0	1	1	1	1
1	0	1	0	1
1	1	0	1	1

## Nomenclature

R<sub>1</sub> : 5,6 MΩ (vert, bleu, vert)

R<sub>2</sub>, R<sub>3</sub>, R<sub>8</sub> : 220 Ω

(rouge, rouge, marron)

R<sub>4</sub> : 2,2 MΩ (rouge, rouge, vert)

R<sub>5</sub>, R<sub>7</sub> : 680 Ω (bleu, gris, marron)

R<sub>6</sub> : 100 kΩ (marron, noir, jaune)

R<sub>9</sub> : 56 kΩ (vert, bleu, orange)

R<sub>10</sub> : 220 kΩ (rouge, rouge, jaune)

R<sub>11</sub> : 33 Ω (orange, orange, marron)

R<sub>12</sub> : 1 kΩ (marron, noir, rouge)

R<sub>13</sub>, R<sub>15</sub>, R<sub>16</sub> : 10 kΩ

(marron, noir, orange)

R<sub>14</sub> : 10 Ω (marron, noir, noir)

C<sub>1</sub> : 680 pF céramique

C<sub>2</sub> : 470 pF céramique

C<sub>3</sub>, C<sub>4</sub> : 100 nF céramique

C<sub>5</sub> : 100 µF/25V chimique

C<sub>6</sub>, C<sub>7</sub> : 100 nF céramique

C<sub>8</sub>, C<sub>22</sub> : 10 µF/16V tantale

C<sub>9</sub>, C<sub>10</sub>, C<sub>11</sub> : 1 nF céramique

C<sub>12</sub> : 27 pF céramique

C<sub>15</sub> à C<sub>21</sub> : 100 nF céramique

Q<sub>z</sub> : Quartz 32768 Hz

D<sub>1</sub> à D<sub>6</sub> : 1N4148

Q<sub>1</sub> : BF245B

Q<sub>2</sub> : BC547C

Q<sub>3</sub> : BC557A

Reg : 7805

U<sub>1</sub> : 74HC4060

U<sub>2</sub> : 74HC4040

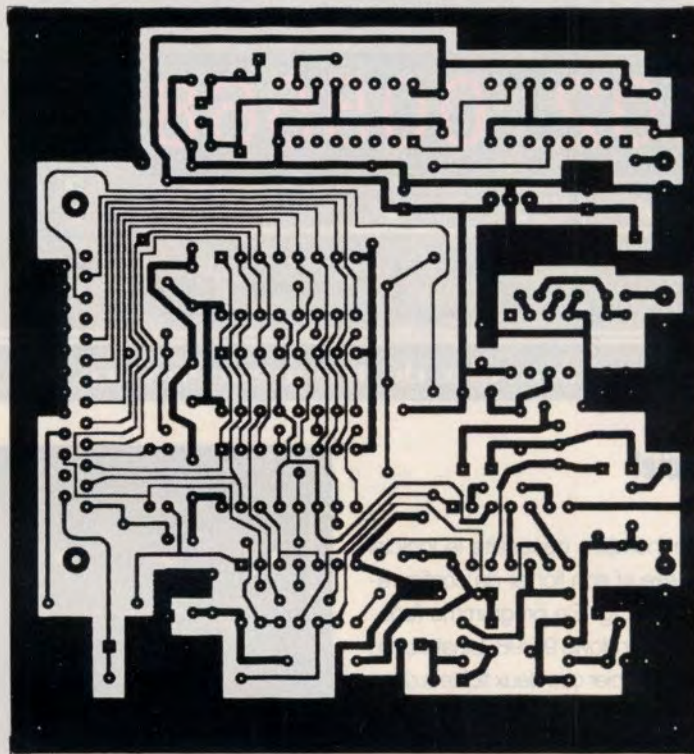
U<sub>3</sub>, U<sub>4</sub> : 74HC132

U<sub>5</sub> : SDA2101 ou U664B

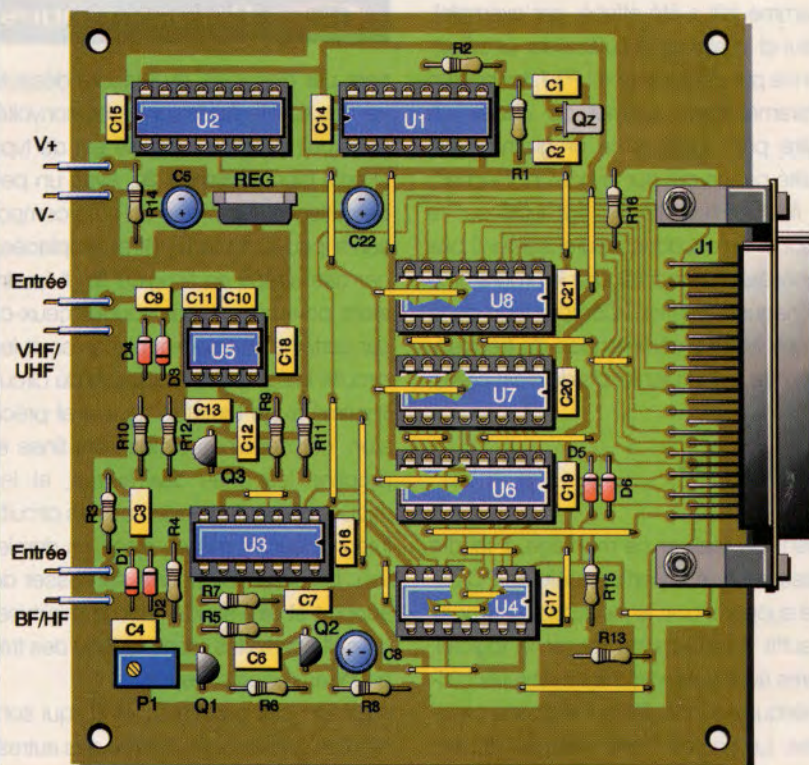
U<sub>6</sub> à U<sub>8</sub> : 74HC590

P<sub>1</sub> : 500 Ω multitours

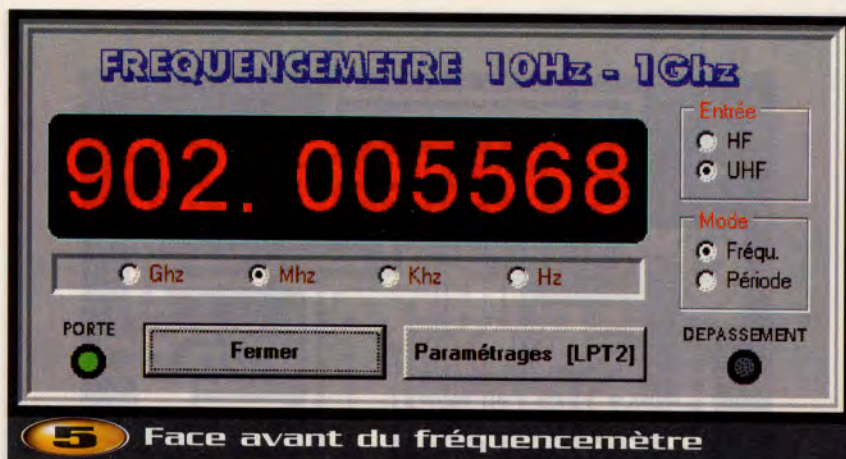
Connecteur SubD 25 mâle



### 3 Tracé du circuit imprimé



### 4 Implantation des éléments



## Le logiciel

Le fréquencesmètre nécessite le logiciel Freqmtr.exe et son fichier de configuration Freqmtr.cfg. Ce programme fonctionne sous Windows 95/98. L'installation se résume à copier ces deux fichiers dans le répertoire de son choix et la désinstallation... à les effacer simplement, puisque la base de registres n'est pas modifiée. Le fichier de configuration contient deux paramètres : le port parallèle utilisé et la correction de base de temps. Si ce fichier n'est pas trouvé au lancement du programme (s'il a été effacé, par exemple), celui-ci le signale à l'utilisateur et sélectionne par défaut le port LPT1. Le bouton "Paramétrages" permet de choisir un autre port. Lorsque le programme est quitté par un clic sur le bouton "Fermer", le fichier Freqmtr.cfg est écrit sur le disque. Il n'est donc (heureusement) pas nécessaire de reconfigurer le programme à chaque utilisation. Compte tenu de la simplicité des commandes du fréquencesmètre, aucun fichier d'aide n'est prévu.

## Réalisation

Avant de réaliser ce montage, il faudra s'assurer que le port parallèle du PC utilisé supporte bien le mode bidirectionnel. Il suffit pour cela de lancer le logiciel, après avoir débranché les éventuels périphériques connectés sur les ports parallèles. Le bouton "Paramétrages" du fréquencesmètre permet de choisir le port désiré. Le bouton "Tester LPT" permet de savoir si ce port est reconnu comme étant bidirectionnel (figure 6). Attention à certains pilotes de périphériques (scan-



ners par exemple), qu'il faudra désactiver s'ils sont installés sur le port convoité. Le circuit imprimé proposé est de type simple face (figures 3). Avec un peu de patience, les pistes du côté composants peuvent même être remplacées par des straps en fil rigide fin. Il faudra alors commencer par souder ceux-ci, car certains se trouvent placés sous les circuits intégrés. La réalisation du circuit imprimé exige toutefois rigueur et précision, car certaines pistes sont fines et proches les unes des autres, et les risques de pontage existent. Les circuits intégrés seront mis en place en dernier lieu. Le circuit  $U_5$  pourra se passer de support et être soudé, afin de minimiser les selfs parasites, compte tenu des fréquences mises en jeu. Attention aux circuits  $U_1$  et  $U_2$  qui sont montés dans le sens inverse des autres. Le montage pourra ensuite être installé dans un boîtier muni de 2 connecteurs BNC reliés par un petit tronçon de câble coaxial aux entrées HF et UHF, d'une découpe permettant l'accès au connec-

teur  $J_1$ , et d'une prise Jack destinée à l'alimentation.

## Essais

Après s'être assuré de l'absence de tout court-circuit, par un contrôle visuel et, en cas de doute, à l'ohmmètre, on pourra connecter l'alimentation. Vérifier la présence du +5V sur les broches adéquates des circuits intégrés. Vérifier à l'oscilloscope ou au multimètre que les broches 1 de  $U_{3a}$  et 11 de  $U_{4c}$  changent d'état toutes les secondes.

Une fois ces vérifications concluantes, on pourra connecter le fréquencesmètre au port du PC, au moyen d'un câble 25 points mâle/femelle standard. Lancer ensuite le logiciel Freqmtr.exe. La "face avant" du fréquencesmètre doit s'afficher ainsi que le montre la figure 5.

Le voyant "Porte" doit clignoter au rythme de la base de temps. Le voyant "Dépassement" s'allume si la fréquence du signal en entrée est trop élevée pour la capacité des compteurs. La valeur affichée correspond alors à la différence entre la fréquence réelle et 16,777216 MHz ou 1,0737418 GHz selon l'entrée concernée. Ceci n'est toutefois vrai que si la fréquence mesurée est inférieure à la fréquence maximum que peut transmettre l'étage d'entrée. Dans le cas contraire, le voyant de dépassement est éteint et l'affichage indique zéro. Lorsque les entrées sont en l'air, le fréquencesmètre peut indiquer des valeurs aléatoires.

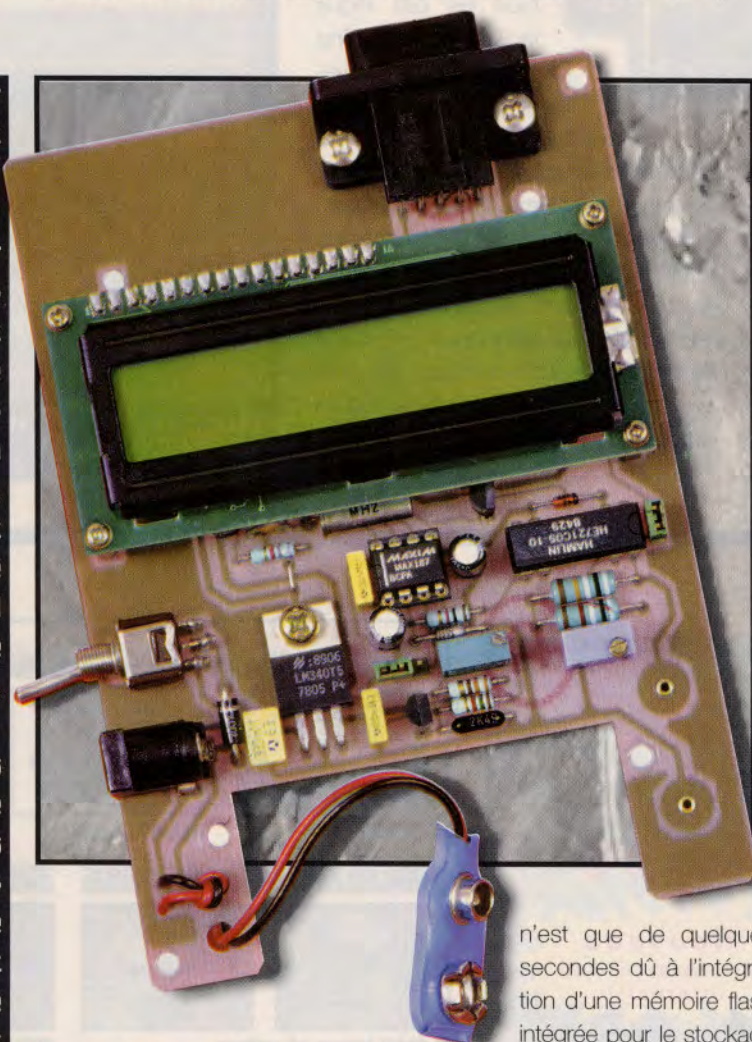
## Réglages

Sans signal à l'entrée, régler le potentiomètre  $P_1$  pour obtenir 2,3V sur le collecteur de  $Q_2$ . Commuter le fréquencesmètre sur l'entrée BF/HF. Appliquer un signal de fréquence connue, en haut de la gamme, par exemple 10 MHz. Cliquer sur le bouton "Paramétrages" et cliquer sur la barre de défilement (figure 6), en plus ou en moins, afin d'obtenir l'affichage exact de la fréquence. Ce petit fréquencesmètre ne prétend pas rivaliser avec les modèles du commerce, mais pourra se révéler fort utile malgré son prix de revient n'excédant pas deux billets de cent francs (hors PC !).

B. LEBRUN

# Précis-Volt

## Voltmètre numérique à calibre automatique et liaison avec un PC



Le cœur du montage repose sur l'emploi d'un convertisseur analogique/numérique 12 bits de résolution MAX1241 épaulé par un microcontrôleur ATMEL 89C2051 de la lignée 8051 INTEL comme le laisse supposer la dénomination.

Ce microcontrôleur présente un certain nombre d'avantages par rapport à un classique 80C31 accompagné d'une EPROM externe et d'un latch. Le prix n'en est pas supérieur, la facilité de routage des pistes du circuit imprimé est évidente et dernier avantage des plus décisifs, son cycle d'effacement programmation

n'est que de quelques secondes dû à l'intégration d'une mémoire flash intégrée pour le stockage du programme.

### Fonctionnement

La tension à mesurer est appliquée entre les bornes  $J_3$  et  $J_4$ . Le relais  $RL_1$  au repos sélectionne par défaut le calibre 40V par l'intermédiaire de l'atténuateur constitué de  $R_4$ ,  $R_3$  et  $RV_2$ . Le rapport de la tension d'entrée sur celle de sortie est égal à  $V_s/V_e = (R_3 + RV_2)/(R_3 + RV_2 + R_4)$  que l'on ajustera très précisément à 1/10ème en agissant sur la vis de réglage de  $RV_2$ . L'impédance caracté-

éristique de l'atténuateur est de  $(R_4 + R_3 + RV_2)$  environ égale à  $1 M\Omega$ . La position travail du relais sélectionne, quant à elle, la tension directement issue de l'ENTREE+ sans passer par l'atténuateur. Le transistor  $Q_1$ , de type MOS canal P, permet l'excitation de la bobine du relais à la demande du microcontrôleur. La diode  $D_1$  évite la destruction du transistor  $Q_1$  contre la f.e.m. induite par la bobine du relais lors de la désactivation de sa grille. De par sa technologie,  $Q_1$  ne nécessite pas de résistance en série avec la grille comme l'aurait exigé un transistor bipolaire PNP classique. A noter lors du reset du microcontrôleur, un état logique haut (1) est présent sur toutes les broches de port. Le transistor  $Q_1$  de type MOS se voit appliqué cet état à la mise sous tension du montage. Durant le temps du reset et jusqu'à l'exécution du programme adéquate, le transistor  $Q_1$  ne doit être rendu conducteur, même un court instant, afin que le contact du relais reste en position repos sélectionnant de ce fait le calibre le plus élevé. Le convertisseur analogique/numérique MAX1241 convertit la tension d'entrée en une valeur binaire codée sur 12 bits. Les valeurs représentées en décimal s'étaleront de 0 à 4095.

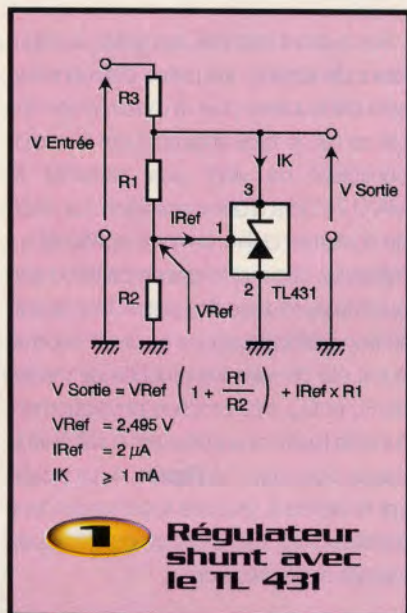
Pour une interprétation aisée du résultat, il suffit de choisir une tension de référence  $V_{ref}$  de telle sorte que 1 bit de poids représente 1mV de tension d'entrée.

Pour cela :

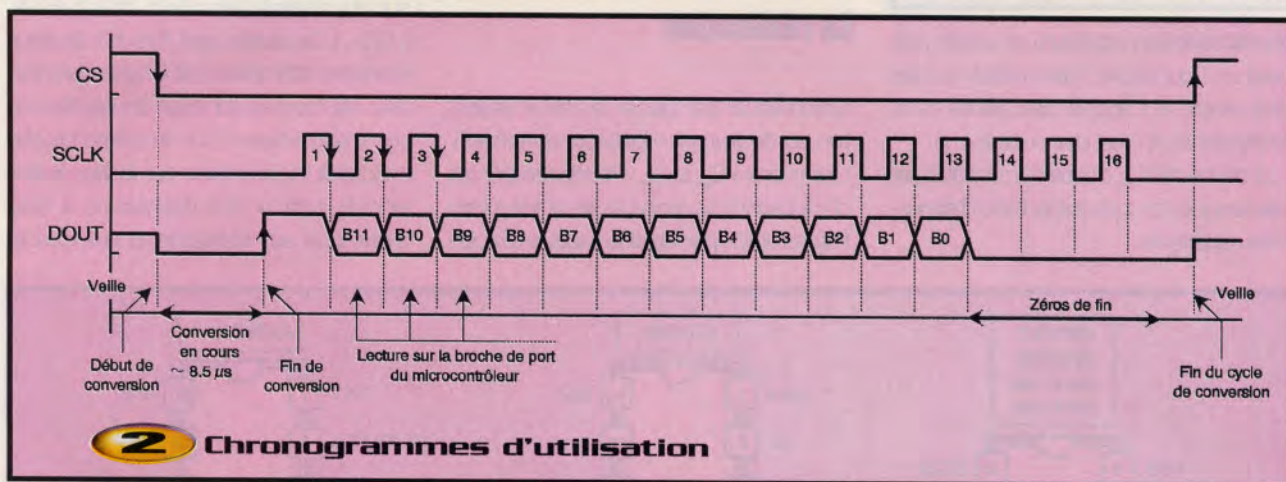
$V_{ref} = 1mV \times 2^{12} = 4096mV$  ou 4,096V tout simplement. Sur le calibre 40V, un bit code une tension de 10mV présente entre  $J_3$  et  $J_4$ . Pour la référence de tension, il est fait appel à un circuit TL431 (origine Texas In-

Le contrôleur numérique est l'outil incontournable possédé par chaque amateur du loisir électronique. Son prix et sa disponibilité ne sont plus des obstacles pour en faire l'acquisition, au point que cet instrument, rare chez le débutant il y a encore une dizaine d'années, est désormais présent dans beaucoup de rayons bricolages (bacs) des supermarchés pour une somme avoisinant les 50 francs. Le voltmètre, objet de cette description, dispose de deux calibres à commutation automatique 0/40V et 0/4V continus ainsi qu'une liaison série compatible PC pour l'exploitation intelligente des données recueillies.

truments), que l'on n'a pas l'habitude de rencontrer pour remplir cette fonction. Le TL431 est un régulateur de tension de type shunt présentant des caractéristiques remarquables vu son prix et son âge. Ce circuit est assimilable à une diode zéner programmable. La **figure 1** présente son utilisation typique et regroupe ses caractéristiques essentielles. Il ne faut pas hésiter à faire appel à ce composant pour générer des tensions stables et précises vu son prix et sa disponibilité. Le convertisseur A/N MAX1241, d'une résolution de 12 bits est du type série. Pour accéder au résultat d'une conversion, il faut extraire, au rythme de l'horloge SCLK que l'on impose, les données qui se présentent bit par bit sur la broche DOUT. Une conversion est lancée



**figure 3** regroupe, quant à lui, les différentes configurations de connexion en fonction de la référence choisie. La gestion des flux d'informations est confiée à un microcontrôleur AT89C2051. La liaison avec le PC est assurée par un MAX232 qui convertit les signaux TTL en signaux proches de la norme RS232. Le sens de communication va de PRECIS-VOLT vers le PC, mais la liaison a été prévue dans l'autre sens pour des aménagements éventuels du logiciel. La broche 9 de la prise DB9 peut recevoir de l'extérieur la tension à mesurer à condition de placer un cavalier en J<sub>2</sub>. Attention, la tension à mesurer est référencée par rapport à la masse numérique partagée avec le PC s'il est branché. L'afficheur LCD est un modèle à 1 ligne de



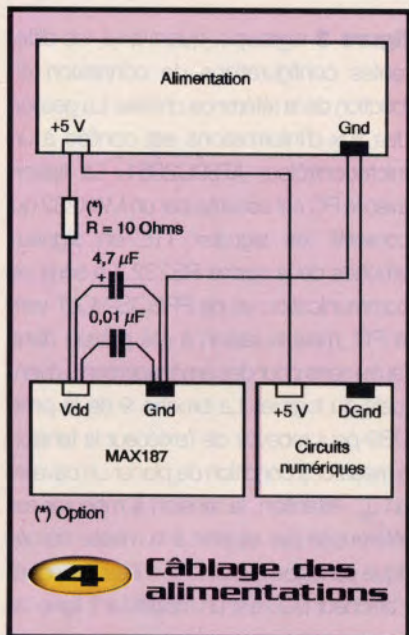
simplement en faisant passer à l'état bas la broche CS. Le résultat de la conversion est disponible 9 μs après cet événement. Le convertisseur ne sera pas poussé jusque dans ses derniers retranchements, nous lui demandons uniquement 3 conversions par seconde (maximum annoncé : 73 000 conversions/seconde). La broche REF reçoit la tension de référence et AIN, la tension à convertir. La broche SHDN a 3 états possibles. A "1", elle autorise la mise en fonction d'une référence interne uniquement disponible pour les références MAX1240 et MAX187. A "0", le convertisseur est placé en mode veille, sa consommation tombe à quelques μA. Laissée flottante, la référence de tension interne est désactivée. Cette broche est laissée impérativement dans cet état pour les MAX1241 et MAX189. Quatre références de convertisseurs sont citées. Chacun pourra utiliser le circuit qu'il trouvera disponible. Le fonctionnement est identique quel que soit le

convertisseur utilisé mais seul le MAX187 pourra se passer de référence de tension externe utilisant les composants annotés d'une étoile. La **figure 2** présente la chronologie à respecter avec les différents convertisseurs et le tableau présenté en

16 caractères connecté en mode 4 bits pour des raisons de simplification de câblage. Le strap J<sub>2</sub> est à placer lorsque l'on utilise un afficheur rétro-éclairé. La consommation de courant est sensiblement plus élevée avec ce type et l'utilisation

	Cavalier J2	Présence de R5-6-7, RV3 et U5	Type de Ref de tension	Remarques
MAX187	Présent	Non	Interne non ajustable	Ref de 4,096 V Précision de + ou - 20 mV sur calibre 4 V!
	Absent	Oui	Externe ajustable par RV3	Déconnexion de la Ref interne
MAX189	Absent	Oui	Externe ajustable par RV3	Pas de Ref interne
MAX1240				Ref interne de 2,5 V à déconnecter car montage calculé pour 4,096 V
MAX1241				Pas de Ref interne

**3** Tableau récapitulatif pour l'utilisation des références de tension



de l'alimentation extérieure se justifie pleinement. Pour sa part, l'alimentation est des plus simple et n'appelle pas grands commentaires si ce n'est deux détails :

- Le connecteur  $J_5$  coupe l'alimentation par pile lorsque l'on introduit la fiche d'alimentation extérieure.

- Sur le circuit imprimé, au niveau du régulateur de tension, les pistes d'alimentation sont dédoublées pour la masse et le +5V. De ce fait, le bruit engendré par le microcontrôleur ne vient pas perturber le MAX1241 lors d'une conversion. La stabilité du dernier chiffre, le moins significatif sur l'afficheur, dépend en grande partie du soin apporté au routage des pistes. Une simplification malheureuse sur le circuit imprimé aurait été de relier les broches de masse de  $R_1$  et  $C_{10}$  très proches physiquement l'une de l'autre et qui pourtant aurait évité la pause d'un strap. La **figure 4**, proposée par le fabricant, propose la connexion type à réaliser pour s'affranchir de problèmes de stabilité et de précision.

### La réalisation

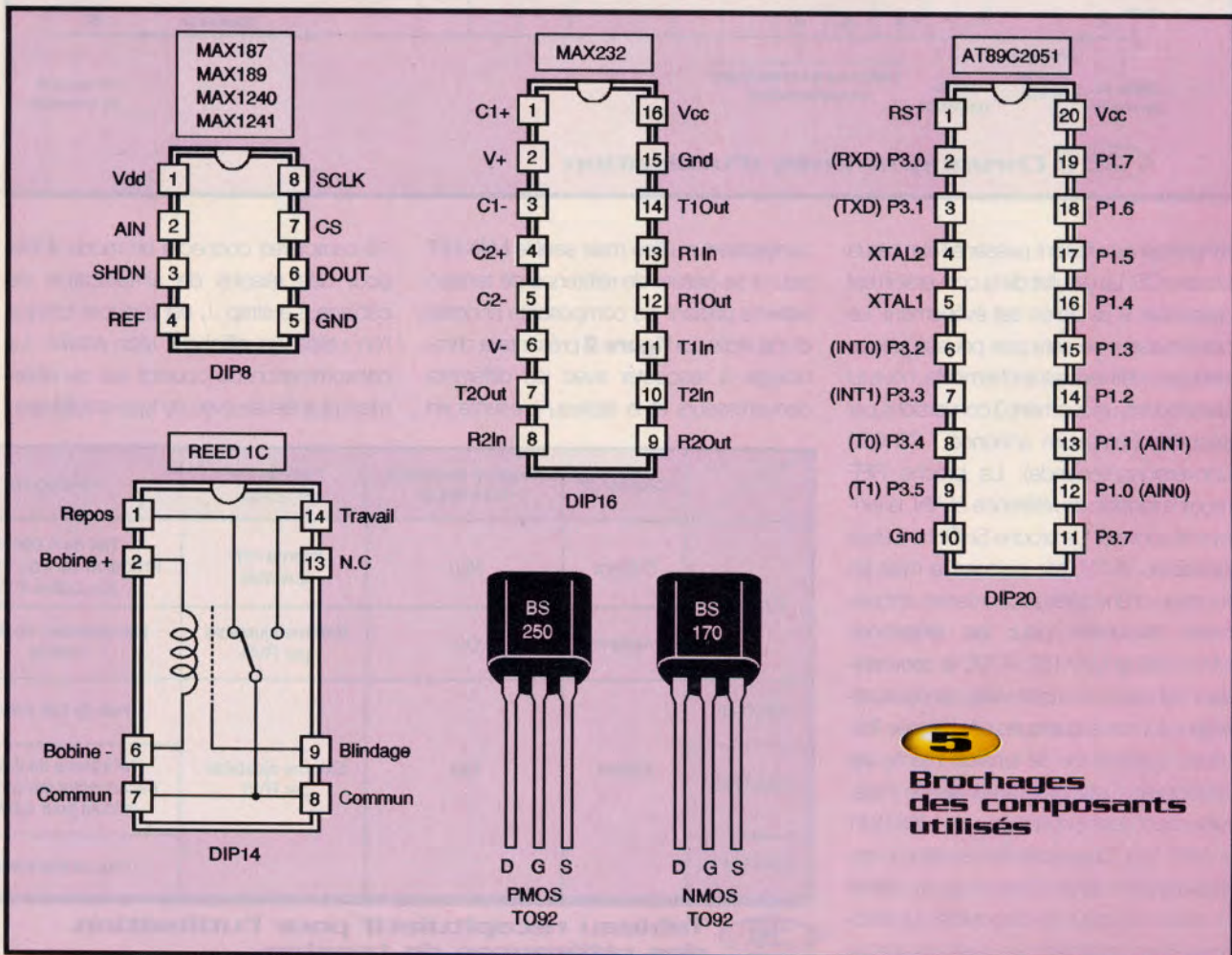
Commencer par placer les deux straps, l'un au-dessus du régulateur de tension, l'autre entre  $C_8$  et  $U_3$ . Poursuivre par les composants les moins épais et les moins fragiles et finir par les plus hauts et les plus

fragiles. En général, ceux dont il faut prendre soins sont faciles à repérer :

Ce sont ceux que l'on ne voudrait pas acheter à nouveau suite à une mauvaise manipulation...

Les circuits intégrés seront placés sur support pour faciliter les réglages ou leur éventuel échange. Utilisez de préférence des supports de type "tulipe" de bien meilleure qualité que leurs homologues à contacts lyre. L'afficheur LCD est posé sur 4 entretoises traversées par des vis M2,5 que l'on ne trouve pas aussi facilement que du M3. L'effort de recherche permet d'éviter de "charcuter" les trous métallisés de fixation d'origine.

La connexion électrique est assurée par une barrette sécable mâle/mâle s'adaptant sur du contact tulipe (voir chez MEGAMOS...). Le quartz peut être fixé de deux manières différentes, soit à l'aide d'un morceau de mousse adhésive double face ou par l'intermédiaire d'une troisième broche soudée à l'extrémité de son boîtier évitant ainsi la rupture des connexions à long terme due aux vibrations et aux chocs.



Pour le perçage du boîtier en plastique, mieux vaut présenter la carte de circuit à l'intérieur et aménager, les uns après les autres, les différentes ouvertures par repérage de proche en proche, que suivre à la lettre un plan mécanique coté qui risque de ne pas être adapté aux composants approvisionnés.

## Les réglages

Avant toute mise sous tension, prendre 5 mn de recul, retirer les circuits intégrés et l'afficheur, vérifier le sens des composants qui en ont un. Mettre sous tension sans replacer les circuits intégrés et vérifier, à l'aide du schéma électrique, la présence sur chaque support des tensions d'alimentations. Si tout est correct, couper l'alimentation et replacer (à l'endroit) les circuits intégrés et l'afficheur. Mettre sous tension et vérifier que quelque chose s'affiche, au besoin régler le contraste à l'aide de  $RV_1$ , qui doit présenter sur son curseur une tension proche de 0,8V pour un contraste moyen. A ce stade le montage fonctionne à 90%, passer à la phase suivante.

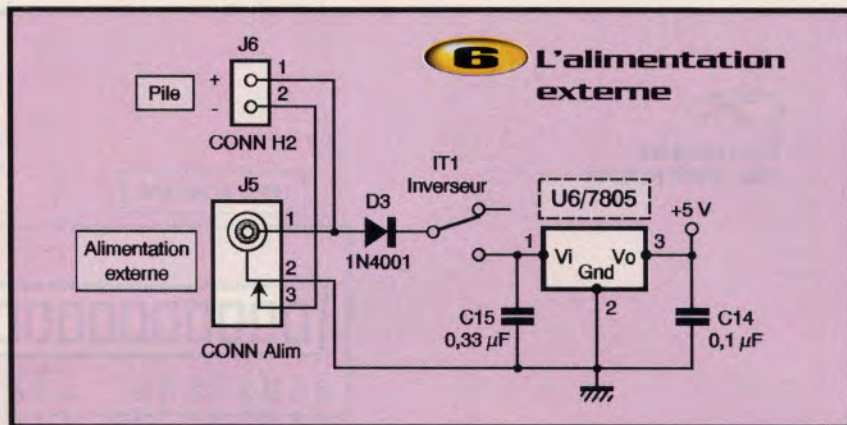
Comme tout appareil de mesure, Précis-Volt doit être étalonné. La précision ne se fabrique pas, elle se copie. Pour cela, il vous faudra trouver un voltmètre plus précis que 4000 points de mesure ou par défaut utiliser un multimètre classique. Procéder de la manière suivante :

- Appliquer sur les deux appareils de mesure branchés en parallèle une tension stable de 4V plus ou moins 50mV. La valeur n'a pas une importance primordiale pourvu que Précis-Volt soit ajusté sur la valeur affichée par le voltmètre de référence.

- A l'aide de l'ajustable  $RV_3$ , régler très précisément afin que l'indication de Précis-Volt soit identique à celle du voltmètre de référence. Le calibre 4V est réglé.

- Appliquer maintenant une tension de 30 à 40V sur l'entrée (les entrées). Le relais  $RL_1$  passe au travail (P3.5 passe à "0"). Procéder de même mais cette fois en agissant sur  $RV_2$ .

- Diminuer progressivement la tension à mesurer pour passer de 30 à 5V, puis doucement ensuite sous 5V. Vérifier qu'au voisinage de 4080mV (chaque pas affiché représente 10mV sur le calibre 4000mV) il n'y a pas de saut dans la page d'indica-



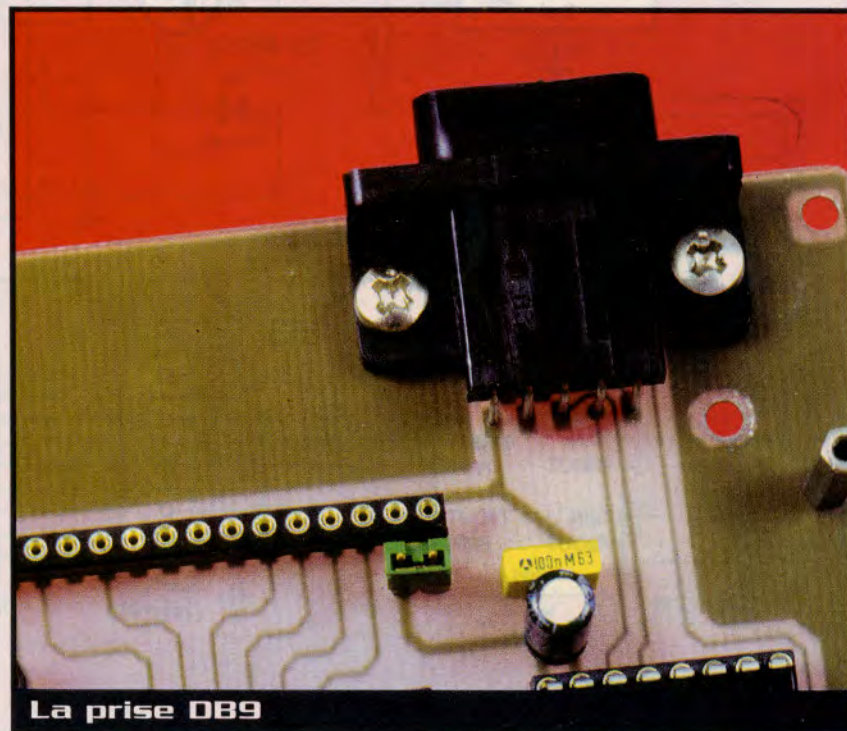
tion et que le calibre passe sur le calibre 4000mV (les pas affichés sont plus fins et égaux à 1mV). Si tout se passe bien, les réglages sont terminés, sinon reprendre à nouveau la phase de réglage des ajustables  $RV_2$  et  $RV_3$  pour affiner la mesure. Il est à noter que la précision de réglage du calibre 4V influe sur celle du calibre 40V. Tout ajustage du premier implique un nouveau réglage du second. En fait, le premier réglage est l'ajustage de la tension de référence alors que le second modifie le facteur de division par 10 du pont de résistances d'entrée.

## Liaison avec un PC

Pour relier Précis-Volt à un PC, utiliser un câble femelle/femelle (ou mâle, mais attention à la numérotation des contacts qui se

trouve inversée) de type prolongateur où RX et TX ne sont pas croisés. Pour l'exploitation des données, vous pouvez utiliser votre terminal préféré à condition de respecter le format des données émises. Les paramètres de communication sont à renseigner sous la forme 2400,N,8,1 COM1 ou COM2 en préférant un dialogue ne faisant pas intervenir les signaux de contrôle matériel du port.

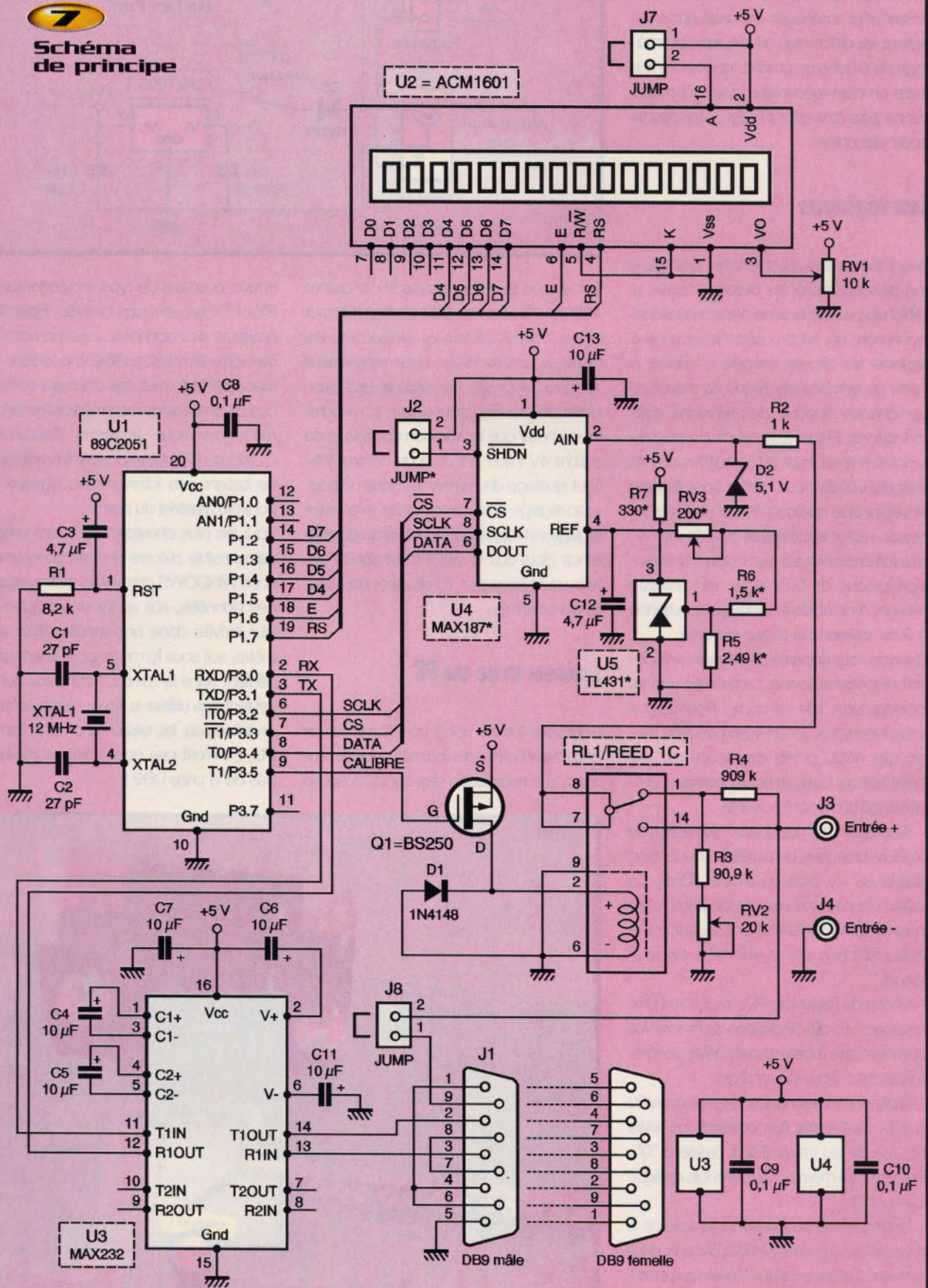
Pour les plus chevronnés d'entre vous, il est possible d'écrire un petit programme sous WINDOWS permettant la visualisation des données, soit au fur et à mesure de leur arrivée dans une fenêtre style voltmètre, soit sous forme graphique les représentant dans le temps. Pour ceux qui ne veulent pas utiliser la liaison série, ne vous en faites pas, les bits émis en direction du PC, s'il n'est pas connecté, ne tombent pas de la prise DB9 !



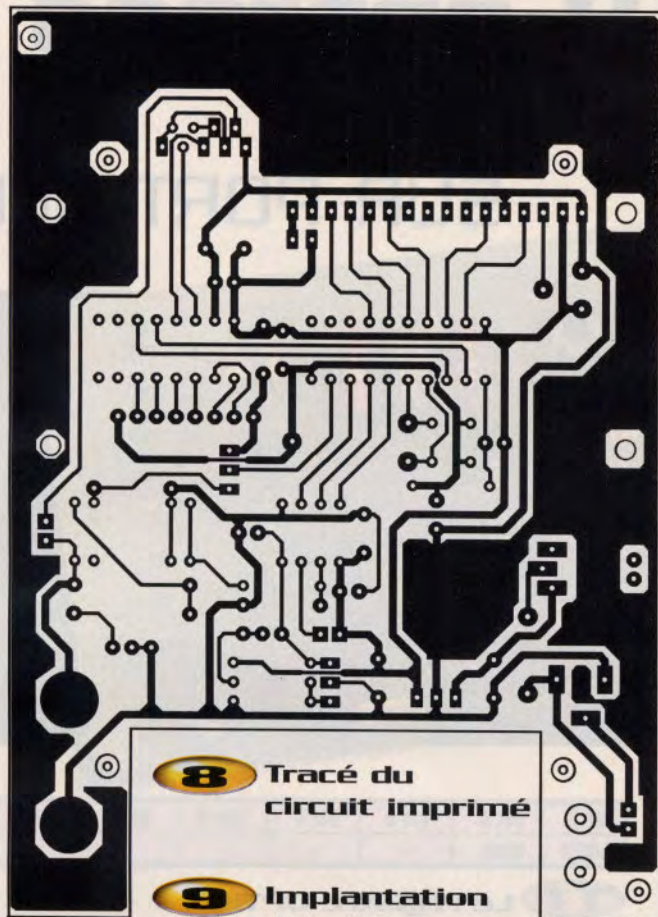
La prise DB9



# Schéma de principe

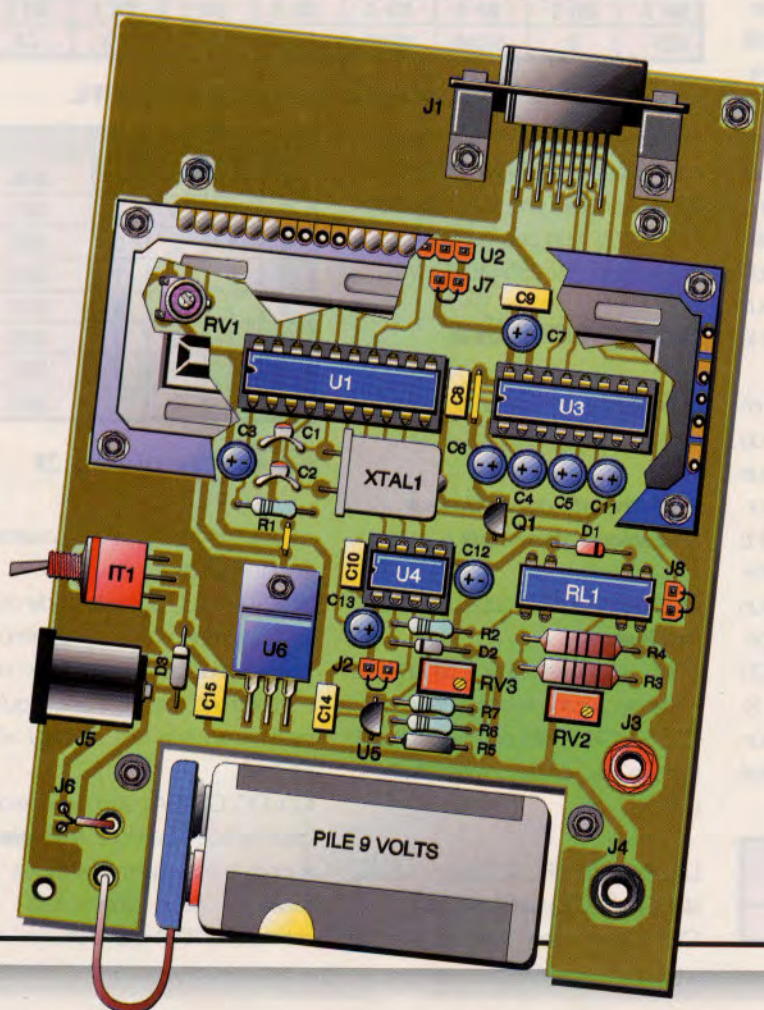






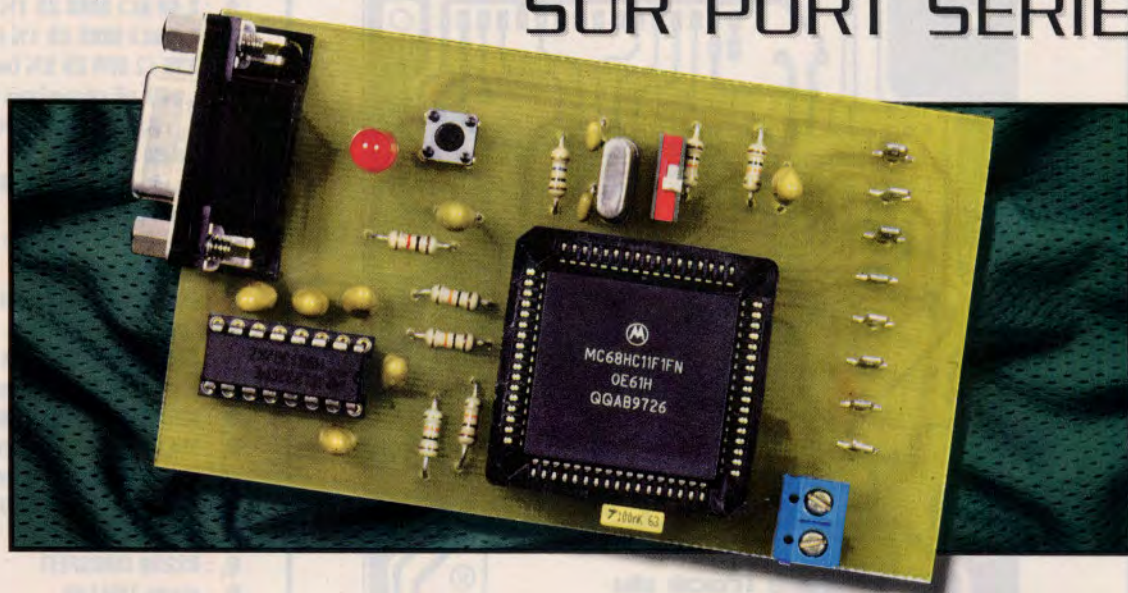
## Nomenclature

- R<sub>1</sub>** : 8,2 kΩ SFR 25 5%  
**R<sub>2</sub>** : 1 kΩ SFR 25 5%  
**R<sub>3</sub>** : 90,9 kΩ MRS 25 1%  
**R<sub>4</sub>** : 909 kΩ MRS 25 1%  
**R<sub>5</sub>** : 2,49 kΩ MRS 25 1% (voir texte)  
**R<sub>6</sub>** : 1,5 kΩ MRS 25 1% (voir texte)  
**R<sub>7</sub>** : 330 Ω SFR 25 5% (voir texte)  
**C<sub>1</sub>, C<sub>2</sub>** : 27 pF céramique type C681  
**C<sub>3</sub>, C<sub>12</sub>** : 4,7 µF/63V chimique radial au pas de 2,5mm  
**C<sub>4</sub> à C<sub>7</sub>, C<sub>11</sub>, C<sub>13</sub>** : 10 µF/63V chimique radial au pas de 2,5mm  
**C<sub>8</sub> à C<sub>10</sub>, C<sub>14</sub>** : 0,1 µF/63V LCC  
**C<sub>15</sub>** : 0,33 µF/63V LCC  
**U<sub>1</sub>** : AT89C2051-12 ou 24 ATMEL à 12 MHz  
**U<sub>2</sub>** : AFFICHEUR LCD 1x16 ACM1601 (avec ou sans rétro-éclairage)  
**U<sub>3</sub>** : MAX232, ICL232, AM232  
**U<sub>4</sub>** : MAX1241 ou MAX187 ou MAX1240 ou MAX189 (voir texte)  
**U<sub>5</sub>** : TL431CP (voir texte)  
**U<sub>6</sub>** : 7805 TO220.  
**Q<sub>1</sub>** : BS250 (MOSFET)  
**D<sub>1</sub>** : diode 1N4148  
**D<sub>2</sub>** : zéner BZX 55 C 5,1V  
**D<sub>3</sub>** : diode 1N4001  
**IT<sub>1</sub>** : inverseur simple miniature type MS-244  
**J<sub>1</sub>** : prise DB9 mâle (ou femelle suivant câble de liaison)  
**J<sub>2</sub>, J<sub>7</sub>, J<sub>8</sub>** : cavaliers strap + barrette sécable (voir texte)  
**J<sub>3</sub>** : douille 2mm (rouge) pour châssis  
**J<sub>4</sub>** : douille 2mm (noire) pour châssis  
**J<sub>5</sub>** : connecteur ALIME CI mâle 2,1mm  
**J<sub>6</sub>** : connecteur pression pour pile 9V, pile 9V 6F22  
**RL<sub>1</sub>** : relais REED DIP 1C/5V (1RT/5V)  
**RV<sub>1</sub>** : ajustable 10 kΩ horizontal petit modèle  
**RV<sub>2</sub>** : ajustable 20 kΩ 10-25T type 67W  
**RV<sub>3</sub>** : ajustable 200 Ω 10-25T type 67W (voir texte)  
**XTAL<sub>1</sub>** : quartz 12 MHz normal ou bas profil  
 Barrette sécable mâle/mâle 14 points (16 points si afficheur rétro-éclairé).  
 Barrette sécable femelle tulipe 14 points (16 points si afficheur rétro-éclairé).  
 1 support tulipe 8 broches  
 1 support tulipe 16 broches  
 1 support tulipe 20 broches  
 1 coffret série RX1000, ext : 130x95x25mm, int : 120x85x20mm vis M2,5x20mm, M3x10mm, entretoises  
 Circuit imprimé simple face 100x150mm



# Carte 8 entrées analogiques

## SUR PORT SERIE



Toutes les données que peut avoir à analyser un ordinateur ne sont pas numériques : il faut alors disposer d'un convertisseur analogique numérique pour permettre le traitement de telles informations par l'ordinateur. Le montage que nous proposons ici pourra assurer la conversion de 8 entrées analogiques. Le programme d'exploitation en basic est d'une extrême simplicité : pour lire la valeur d'une entrée, il suffira d'indiquer le numéro de cette entrée, la valeur s'affichant alors à l'écran.

### Schéma de principe

Une telle facilité est due à l'emploi d'un microcontrôleur, ici un 68HC11F1 qui renferme un convertisseur analogique numérique relié aux broches du port E et une interface de communication série reliée au port D. Un programme chargé dans l'EEPROM du microcontrôleur assure le transfert d'un port à l'autre. Il ne reste alors qu'à adapter la tension des signaux, les niveaux utilisés par l'ordinateur étant de + ou - 12 V et ceux du HC11 variant de 0 à 5 V : c'est le rôle du classique MAX 232.

Le schéma électronique représenté **figure 1** ne nécessite pas de grands commentaires : on remarque en plus du MAX232, le circuit d'horloge cadencé par un quartz à 8 MHz et le circuit de RESET constitué simplement d'un bouton poussoir, d'un condensateur et d'une résistance. L'interrupteur I relié à la broche MODB permet de passer du mode BOOTSTRAP utilisé pour la programmation au mode CIRCUIT SEUL utilisé ensuite.

	MODB
Programmation	Etat bas
Utilisation	Etat haut

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADPU	CSEL	-	-	-	-	-	-

### T1 Le registre OPTION

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCF	0	SCAN	MULT	CD	CC	CB	CA

### T2 Le registre de contrôle ADCTL

Les tensions de référence utiles au convertisseur analogique VREFH et VREFL sont tout simplement fixées ici à 5V et 0V : un pas de conversion vaudra donc  $5V/255=19,6$  mV.

### La programmation

Elle nécessite l'utilisation de quelques registres du 68HC11 :

#### Registres du convertisseur analogique numérique :

Le registre OPTION, adresse \$1039 (**tableau 1**)

ADPU (AD Power Up) : le bit ADPU à 1 met en service le convertisseur.  
CSEL (Clock SElect) : ce bit à 1 indique le choix d'une horloge externe.

Le registre de contrôle ADCTL, adresse \$1030 (**tableau 2**)

CCF (Conversion Complete Flag) :

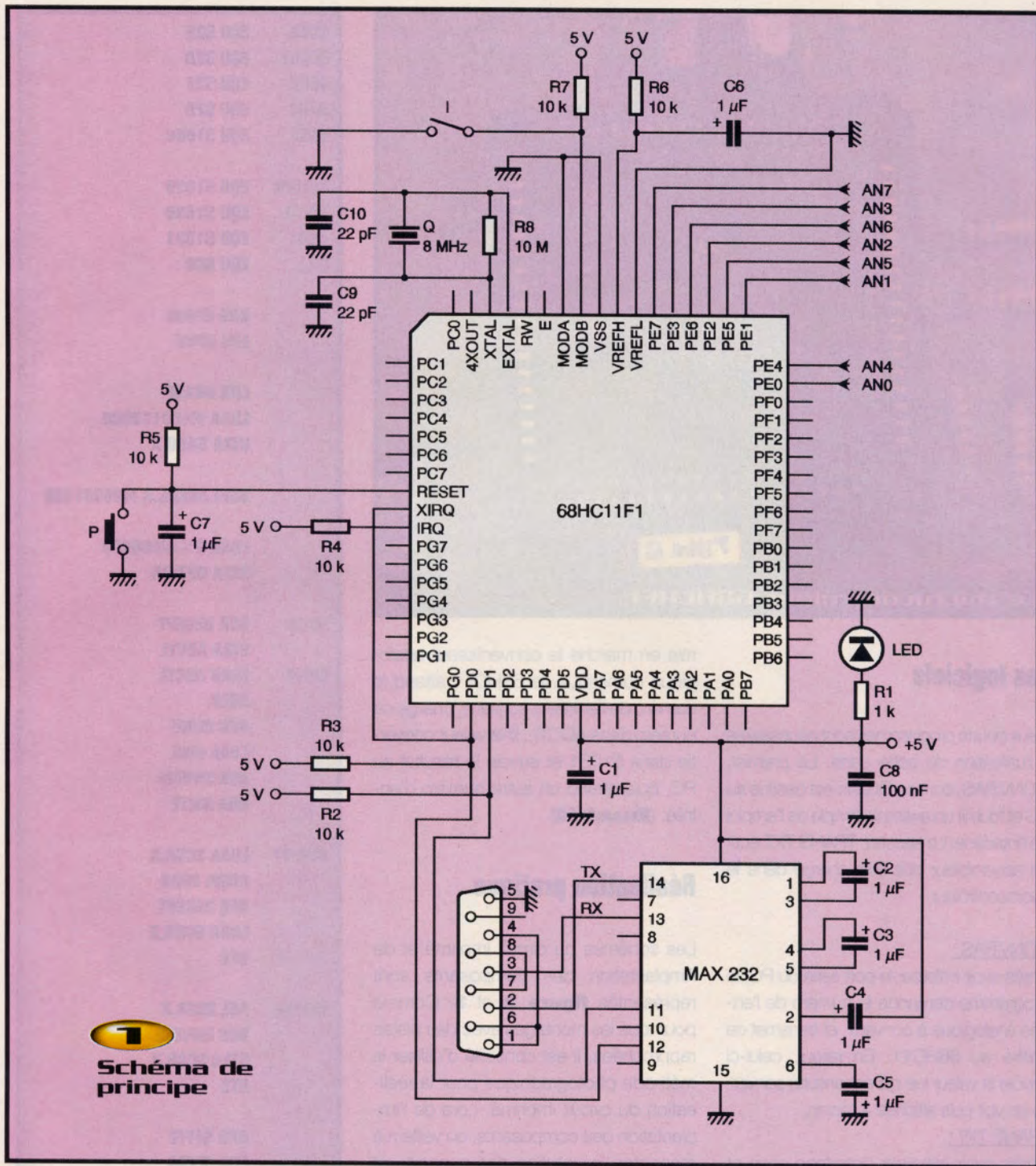
CD	CC	CB	CA	ENTREES
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7

### T3 Tableau 3

ce bit à 1 indique que les conversions sont terminées.

SCAN et MUL permettent de définir un choix de conversion continu ou non, sur 1 ou plusieurs canaux : nous ne traiterons dans cet article qu'une conversion à la fois : SCAN =0 et MUL=0.

CD, CC, CB, CA permettent de désigner l'entrée à convertir. (**tableau 3**) 4 conversions sont effectuées : les résultats sont mémorisés aux adresses \$1031 à \$1034.



**1**  
Schéma de principe

**Registres de l'interface série :**

Le registre SCCR2, adresse \$102D (tableau 4)

Seuls les bits TE et RE nous intéressent : à 1, ils mettent en service l'interface en émission et en réception.

Le registre d'état SCSR, adresse \$102E : (tableau 5)

Seuls les bits TDRE (transmit Data Register Empty) et RDRF (Receive Data Register Full) nous intéressent : ces bits à 1 indiquent

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

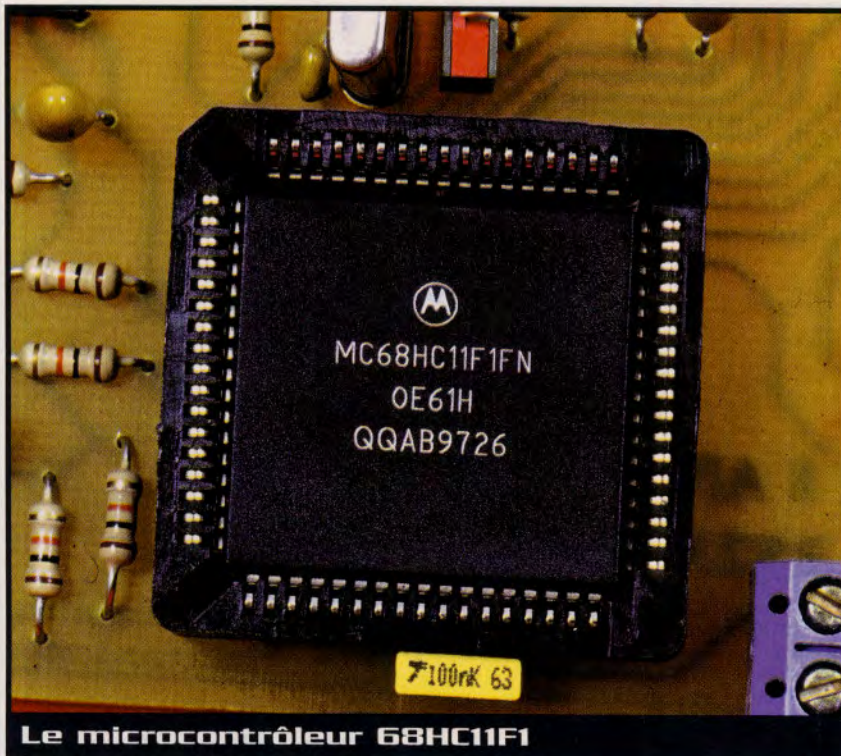
**T4** Le registre SCCR2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TDRE	TC	RDRF	IDLE	OR	NF	FE	0

**T5** Le registre d'état SCSR

l'interface va émettre un octet ou a reçu un octet.  
Le registre BAUD, adresse \$102B permet de définir la vitesse de transmission :

Ici on mettra dans ce registre %00110000 pour obtenir une vitesse de 9600 bauds. Les octets sont émis ou reçus dans le même registre d'adresse SCCR d'adresse \$102F.



Le microcontrôleur 68HC11F1

## Les logiciels

Deux courts programmes sont nécessaires à l'utilisation de cette carte. Le premier, CONV.BAS, écrit en Qbasic est destiné au PC et fournit un exemple simple de l'emploi de l'interface. Le second, TRANS.TXT, écrit en assembleur doit être chargé dans le microcontrôleur.

### CONV.BAS :

Après avoir initialiser le port série du PC, le programme demande le numéro de l'entrée analogique à convertir et transmet ce chiffre au 68HC11. En retour, celui-ci envoie la valeur lue qui est ensuite convertie en volt puis affichée à l'écran.

### TRANS.TXT :

Après avoir initialiser l'interface série et

mis en marche le convertisseur analogique numérique, le 68HC11 attend le numéro de l'entrée à convertir, charge ce numéro dans ADCTL, lit la valeur convertie dans \$1031 et envoie le résultat au PC, puis attend un autre numéro d'entrée. (**Encadré2**)

## Réalisation pratique

Les schémas du circuit imprimé et de l'implantation des composants sont représentés **figure 2** et **3**. Comme pour tous les montages avec des pistes rapprochées, il est conseillé d'utiliser la méthode photographique pour la réalisation du circuit imprimé. Lors de l'implantation des composants, on veillera à respecter l'orientation des supports et des éléments polarisés.

```

40 OPEN "COM1:9600,N,8,1,ASC"
   FOR RANDOM AS #1
50 DO WHILE N < 9
60 INPUT "NUMERO ENTREE", N
70 OUT &H3F8, N
80 FOR I = 1 TO 1000
90 NEXT I
100 X = INP(&H3F8)
110 Y = X * 5 / 255
120 PRINT Y
130 LOOP

```

**E1** Encadré 1

## Mise en œuvre

Mettez l'interrupteur I sur ON pour placer le HC11 en mode BOOTSTRAP et chargez le programme assemblé.

L'assemblage peut être effectué avec AS11.EXE et le chargement avec PCBUG11.EXE

Mettez l'interrupteur I sur OFF pour placer le HC11 en mode CIRCUIT SEUL. La carte est prête pour son utilisation prévue.

```

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

```

```

OPTION EQU $1039
ADCTL EQU $1030
ANO EQU $1031
PILE EQU $C0

```

```

ORG $FE00
LDS #PILE

```

```

LDX #BASE
LDAA #%00110000
STAA BAUD,X

```

```
BSET SCCR2,X %00001100
```

```
LDAA #%10000000
STAA OPTION
```

```

SUITE BSR RECEPT
      STAA ADCTL
CONV LDAB ADCTL
     ASLB
     BCC CONV
     LDAA ANO
     BSR ENVOIE
     BRA SUITE

```

```

RECEPT LDAA SCSR,X
        ANDA #$20
        BEQ RECEPT
        LDAA SCDR,X
        RTS

```

```

ENVOIE ASL SCSR,X
       BCC ENVOIE
       STAA SCDR,X
       RTS

```

```

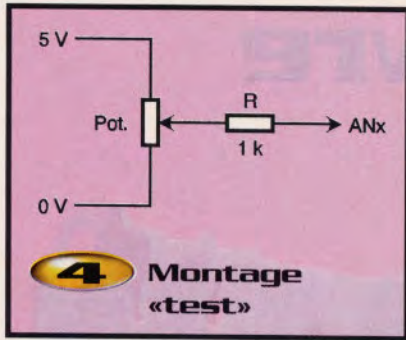
ORG $FFFE
FDB $FE00

```

**E2** Encadré 2

## Essai

La carte peut maintenant être testée avec un potentiomètre comme indiquée dans la **figure 4**. La résistance de 1 kΩ est obligatoire et protège le 68HC11. Lancez le programme CONV.BAS et entrez, à la demande, le numéro de l'entrée où est branché le point milieu du potentiomètre : la valeur de la tension correspondante s'affiche directement à l'écran.



### Autre utilisation

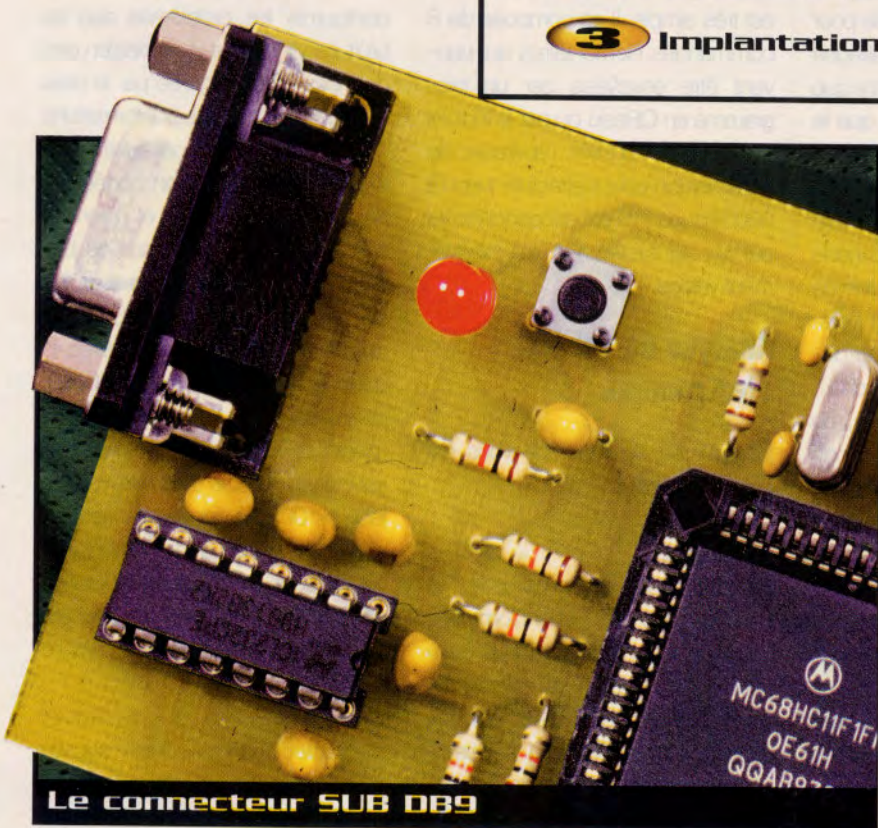
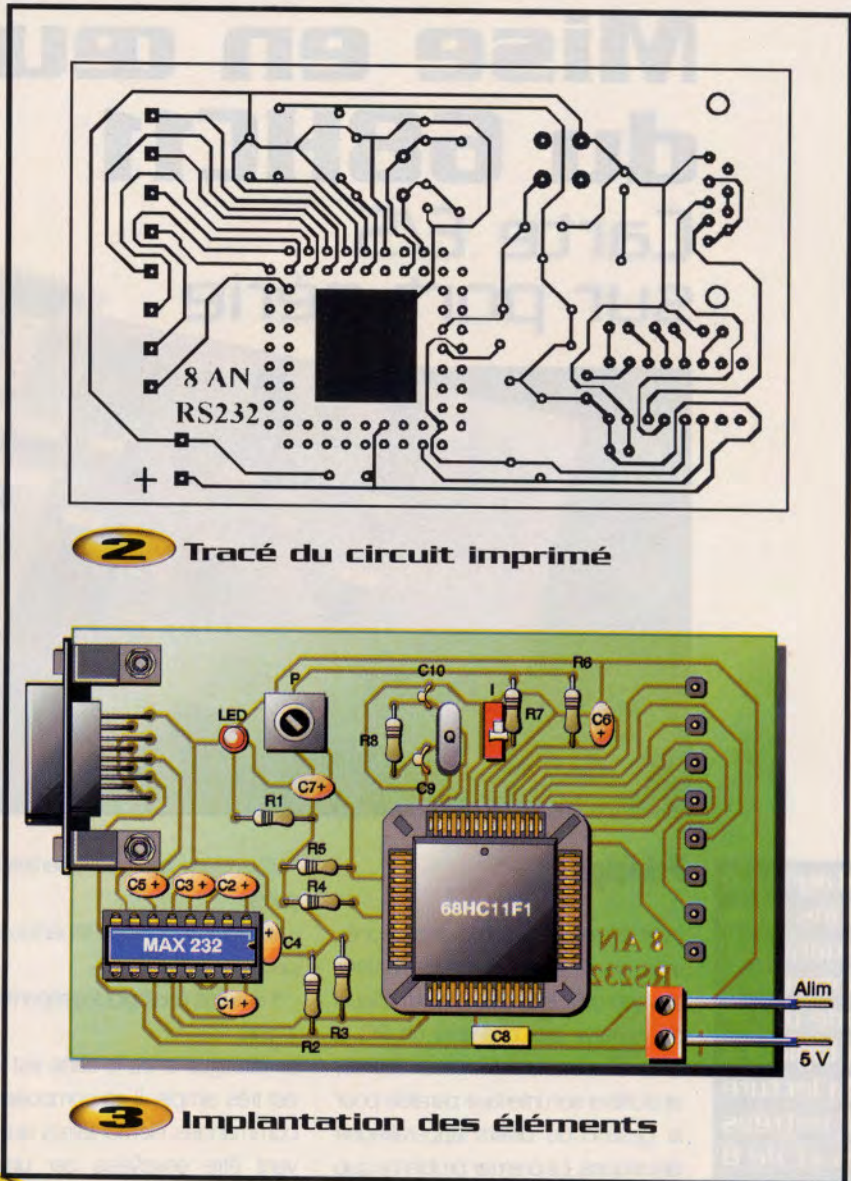
Cette carte peut aussi servir en tant qu'interface 8 entrées digitales sur port série si le convertisseur n'est mis activé. Il suffit dans le programme CONV.TXT de supprimer tout le texte après BSET SCCR2,X%00001100 et de remplacer par :

```
SUITE LDAA $100A
      STAA SCDR,X
      BRA SUITE
```

### E3 Encadré 3

La valeur du port E d'adresse \$100A est envoyée en permanence sur l'interface série.

A.REBOUX



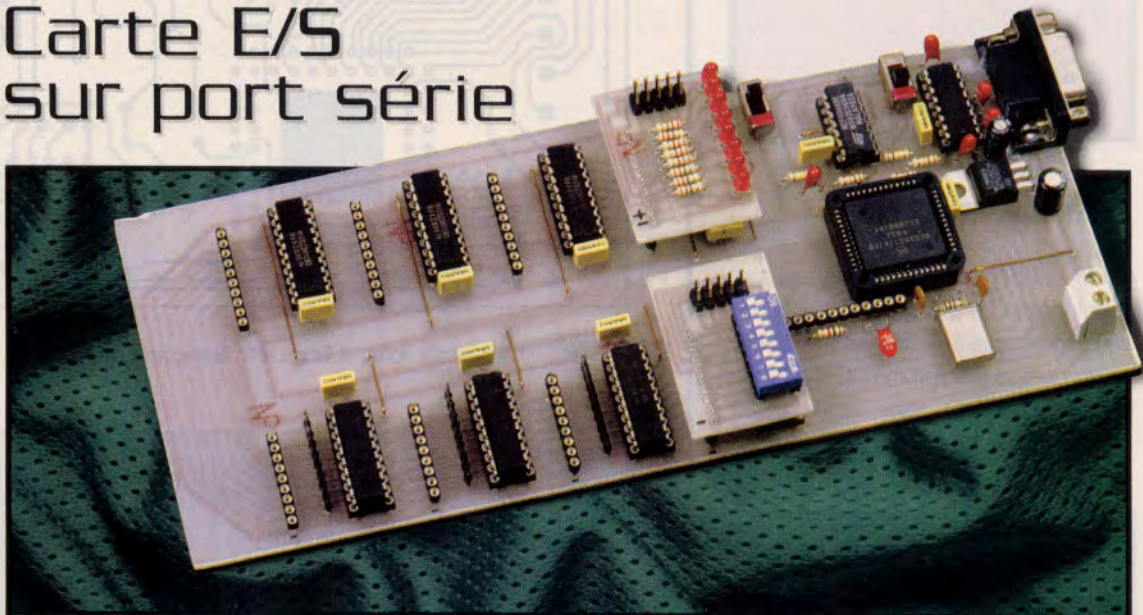
Le connecteur SUB DB9

### Nomenclature

- R1 : 1kΩ
- R2 à R7 : 10kΩ
- R8 : 10MΩ
- C1 à C7 : 1μF
- C8 : 100 nF
- C9 et C10 : 22 pF
- Q : quartz 8 MHz
- I : interrupteur DIP simple
- MAX 232 + support 16 broches
- 68HC11F1 + support 68 broches
- 8 picots
- 1 bornier à vis
- 1 connecteur subDB9 soudé

# Mise en œuvre du 68HC11

## Carte E/S sur port série



**Le montage que nous allons vous décrire dans cet article, permettra la commande de 32 sorties logiques ainsi que la lecture de 32 entrées logiques et de 8 entrées analogiques. Tout ceci étant réalisé au moyen de l'interface série que comporte n'importe quel PC ou compatible.**

### Principe

Il est désormais simple et peu onéreux de se procurer un ordinateur devenu obsolète pour l'informatique de bureau, par exemple un PC à base d'un microprocesseur 80286, et d'utiliser son interface parallèle pour la gestion de divers appareillages électriques. Le premier problème que l'on risque de rencontrer est que le port parallèle comporte un nombre limité d'entrées et de sorties. De plus, on ne peut utiliser une liaison parallèle que si la longueur du câble n'excède pas quelques mètres sous peine de voir le système soumis à des parasites, la commande d'organes distants devient alors peu commode... Si l'on utilise l'interface série, on peut alors envisager des câbles de plusieurs centaines de mètres, tout en utilisant que trois conducteurs (RxD/TxD/GND).

La carte que nous vous proposons de réaliser a une conception basée sur le microcontrôleur 68HC11 de MOTOROLA qui possède sa propre liaison série RS232 (appelée SCI). Cette carte permettra la gestion de 72 lignes d'entrées/sorties réparties comme suit :

- 32 entrées logiques réparties sur 4 ports,
- 32 sorties logiques réparties sur 4 ports,
- 8 entrées analogiques réparties sur 1 port.

Le dialogue entre la carte est le PC est très simple, il se compose de 8 commandes élémentaires qui peuvent être envoyées par un programme en QBasic ou par le logiciel terminal de Windows. La vitesse de transmission peut s'effectuer jusqu'à 9600 bauds (bits par seconde) ce qui permet au système d'être relativement vélocé.

### Analyse du schéma électronique

Le schéma du montage vous est donné **figure 1**.

#### Port série

Le  $\mu C$  dispose de son propre port série appelée SCI (Serial Communication Interface). Contrairement aux UART classiques qui se resynchronisent au début de chaque caractère reçu, la SCI du 68HC11 se resynchronise sur chaque bit reçu ce qui permet de gommer les écarts éven-

tuels de vitesse de transmission entre émetteur et récepteur. De plus la SCI réalise 3 fois de suite l'échantillonnage de chaque bit reçu, une logique à décision majoritaire est alors chargée de déterminer si le bit reçu est à 1 ou à 0. Ce processus permet de contourner les problèmes dus au bruit. La vitesse de transmission des données est configurée par le biais du registre BAUD. Deux informations sont contenues dans ce registre : l'information de prédivison codée par les bits SCP0 et SCP1 et l'information de vitesse codée par les bits SCRO et SCR1 (voir **tableaux 1 à 3**). Dans le programme du  $\mu C$ , le registre BAUD est chargé avec la donnée #NITBAUD = %00110000, ce qui configure la vitesse de transmission à 9600 bauds. Si vous désirez utiliser un câble série relativement long (>100m), il conviendra de descendre la vitesse de transmission à 1200 bauds en modifiant la valeur de #NITBAUD = %00110011, ceci afin d'éviter des problèmes de communication dus au bruit. Toutefois, la SCI ne peut être directement reliée au PC qui délivre sur son port sériel des signaux de type +/-10V. En effet, le  $\mu C$  n'accepte que

des signaux du type +/-5V. Le circuit chargé de rétablir la compatibilité est le 232 de chez MAXIM. Il s'agit d'un circuit adaptateur de niveau. Il est de ce fait câblé avec 4 condensateurs nécessaires au fonctionnement de sa pompe de charge interne, pour passer la tension de 5 à 10V.

## Mode de fonctionnement du 68HC11

Le  $\mu\text{C}$  fonctionne en mode spécial Bootstrap, il s'agit à l'origine d'un mode de test et de mise au point qui fonctionne de la façon suivante. Dans ce mode, le 68HC11 exécute un programme d'origine gravé dans sa ROM, ceci à condition qu'il reçoive auparavant l'octet \$FF à 1200 bauds. Dès lors, il configure sa SCI à 1200 bauds et les 256 octets suivants qu'il va recevoir seront stockés dans la RAM. Une fois la RAM remplie le compteur programme pointe sur l'adresse \$0000 de la RAM et le programme s'exécute. Ce mode de fonctionnement était prévu pour le test de petits programmes ne dépassant pas 256 octets. Pour pouvoir charger le programme souhaité en EEPROM, il existe un programme créé par MOTOROLA qui se charge tout d'abord en RAM, comme on a vu, et qui, par la suite, va charger les 512 octets que la SCI va recevoir, ces octets étant le programme que le  $\mu\text{C}$  devra par la suite exécuter.

Désormais, lors du prochain RESET, il faut que le  $\mu\text{C}$  démarre son programme non plus en RAM à l'adresse \$00 mais en EEPROM à l'adresse \$B600. Ceci est réalisé à condition que le  $\mu\text{C}$  reçoive sur sa SCI l'octet \$00.

Or, juste après un RESET, cet octet est disponible sur la ligne TxD, il ne reste alors plus qu'à relier la ligne TxD à la ligne RxD pour que le  $\mu\text{C}$  s'envoie l'octet \$00 à lui-même. L'inconvénient est que si l'on relie définitivement les deux lignes le port série est inutilisable par la suite ce qui serait fort dommage compte tenu de notre application. Il ne reste qu'à utiliser la solution parue dans le numéro 209 d'électronique pratique. Elle consiste à utiliser le CI 4066 qui comporte 3 Switch analogiques qui se chargent de court-circuiter les lignes RxD et TxD pendant quelques ms lors de chaque reset.

SCP1	SCP0	Taux de division	Fréquence quartz = 8MHz
0	0	1	125.000K
0	1	3	41.667K
1	0	4	31.250K
1	1	13	9600

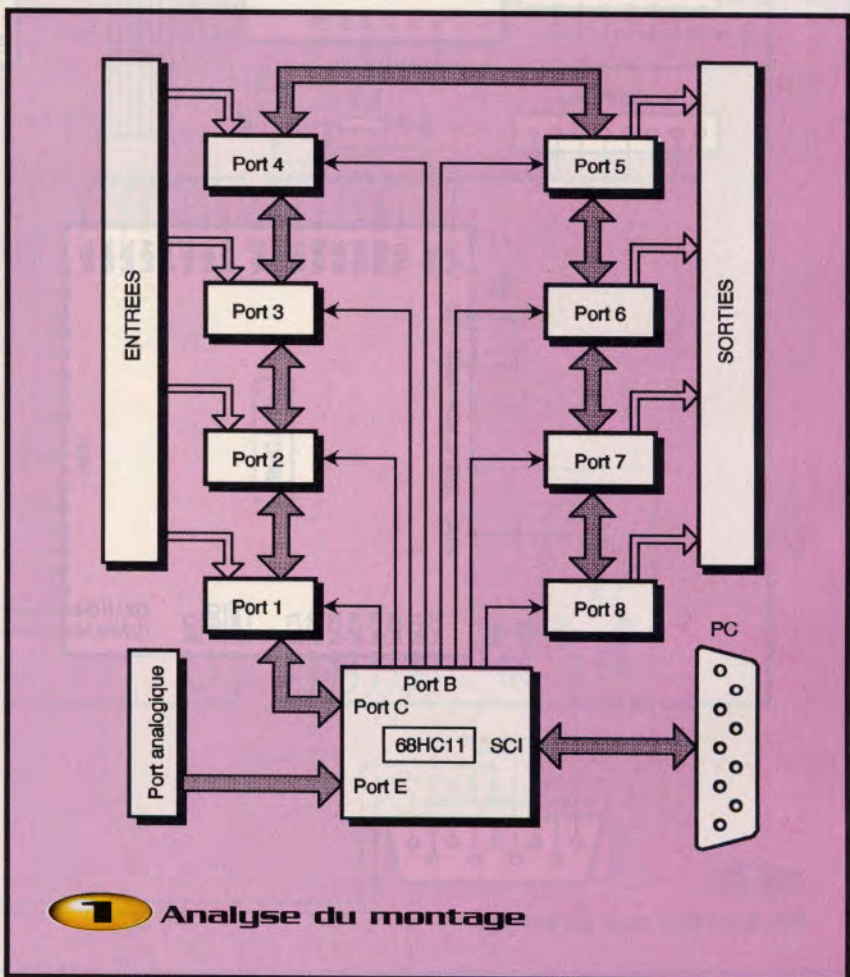
**T1** Tableau 1

SCR2	SCR1	SCR0	Taux de division	9600 bauds
0	0	0	1	9600
0	0	1	2	4800
0	1	0	4	2400
0	1	1	8	1200
1	0	0	16	600
1	0	1	32	300
1	1	0	64	150
1	1	1	128	75

**T2** Tableau 2

Registre BAUD								
bits :	7	6	5	4	3	2	1	0
	0	0	SCP1	SCP0	0	SCR2	SCR1	SCR0

**T3** Tableau 3



**T** Analyse du montage

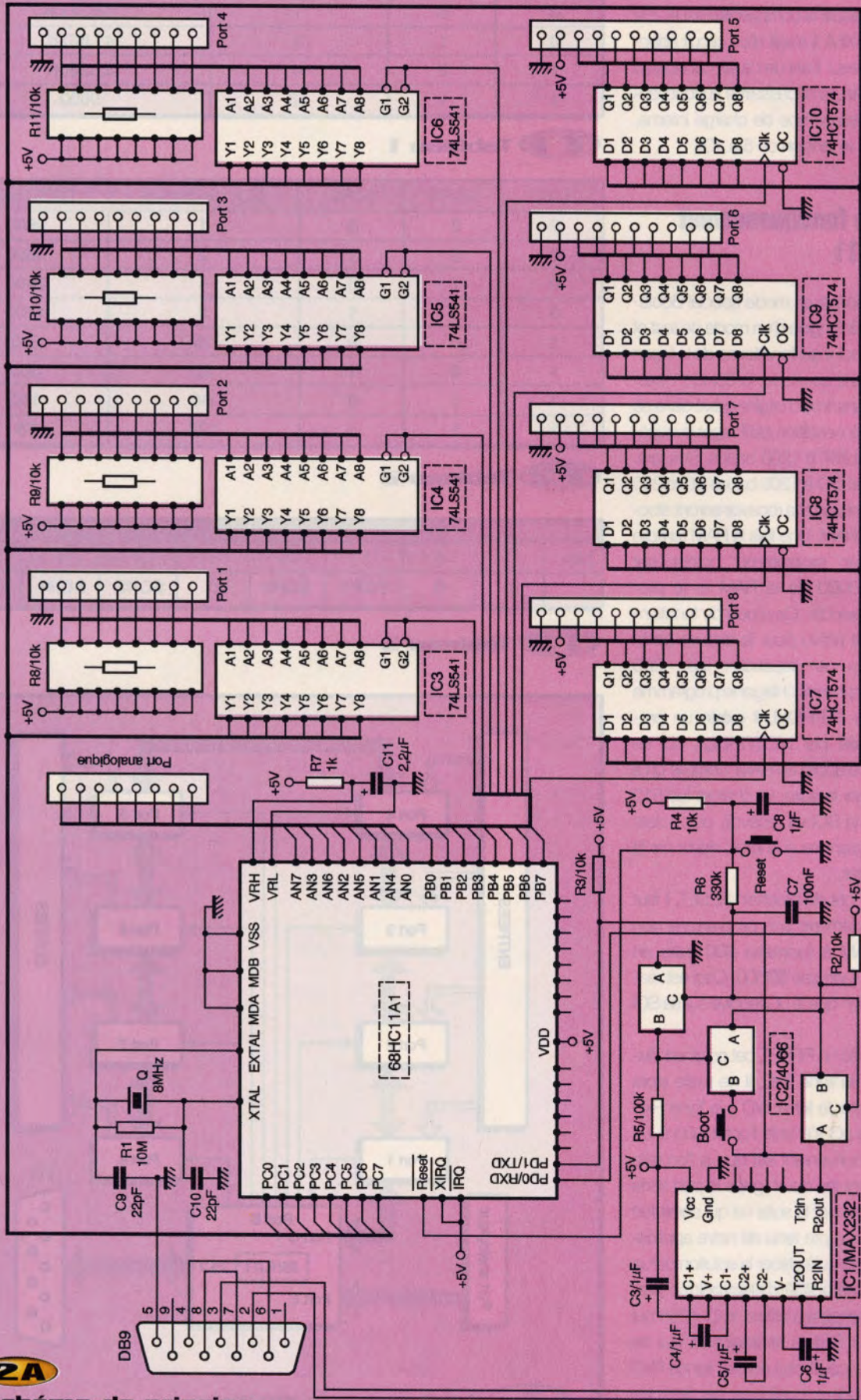


Schéma de principe



## Ports E/S

La carte possède 8 ports parallèles, 4 sont utilisables en entrées et sont numérotés de 1 à 4, 4 sont utilisables en sorties et sont numérotés de 5 à 8. Chacun des ports est relié au port C du  $\mu$ C, qui est utilisé comme un bus de donnée bidirectionnel.

Dans le cas d'une opération d'écriture, la donnée est d'abord lue sur la SCI puis recopiée sur le port C qui est alors configuré en sortie.

Dans le cas d'une opération de lecture, la donnée est d'abord lue par le port C qui est alors configuré en entrée, puis envoyée à l'ordinateur par la SCI.

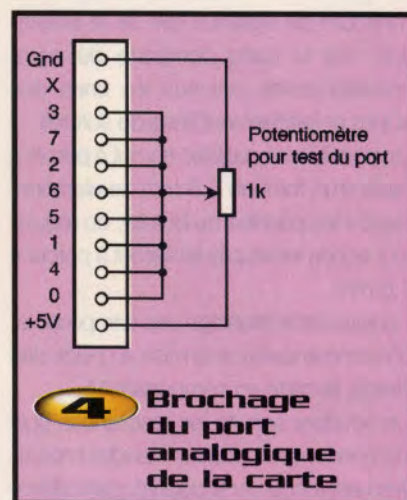
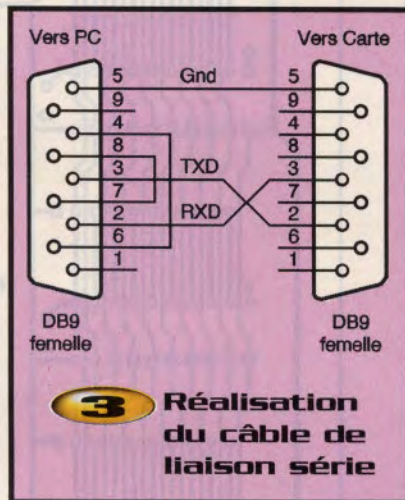
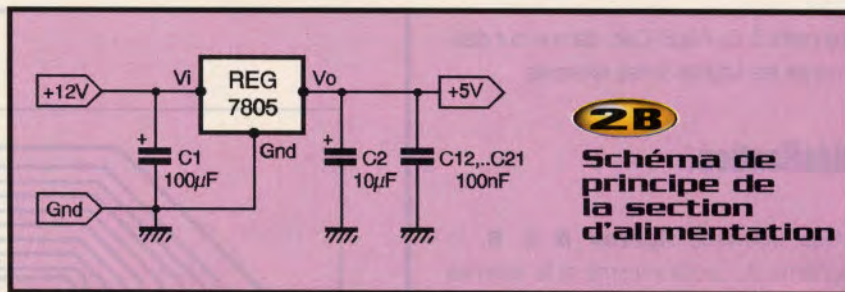
Il est évident qu'un seul port est utilisé à la fois, cette sélection s'effectuant par l'intermédiaire du port B qui est en quelque sorte utilisé comme bus d'adressage.

## Entrées

Chaque port d'entrée utilise un circuit 74LS541. La sélection des ports 1, 2, 3 et 4 s'effectue par les bits G1 et G2 qui sont reliés respectivement à PB0, PB1, PB2, PB3.

Si une de ces lignes est à l'état bas, l'octet présent sur l'entrée du 74LS541 correspondant est recopié sur sa sortie et envoyé de ce fait sur le port C qui est alors configuré en entrée. Si les bits G1 et G2 sont à l'état haut les sorties du 74LS541 sont à l'état de haute impédance, ce qui revient à dire que le circuit est déconnecté du port C.

Les entrées du 74LS541 sont reliées à des résistances de rappel. Si une entrée



n'est pas utilisée, elle est mise à l'état haut par une résistance de 10 k $\Omega$ .

## Sorties

Chaque port de sortie utilise un circuit 74LS574. La sélection des ports 5, 6, 7 et 8 s'effectue par les broches CLK (CLOCK) qui sont reliées respectivement à PB4, PB5, PB6, PB7. Chacune des 8 entrées de chaque 74LS541 est reliée au port C qui est alors configuré en sortie. Si une des entrées CLK est soumise à un front montant la donnée présente en entrée du

74LS574 correspondant est recopiée sur sa sortie. L'état de la sortie étant mémorisé jusqu'à la prochaine écriture.

## Port analogique

Le 68HC11 est équipé d'origine d'un port analogique comportant 8 voies. Il est ainsi possible d'effectuer la lecture de 8 tensions différentes sur le port série. La plage de tension est fixée entre 0 et +5V par les broches VRL et VRH. La résistance  $R_7$  et le condensateur  $C_{11}$  forment un filtre passe bas qui stabilise la tension de référence du convertisseur. Le brochage du port analogique de la carte vous est donné **figure 4**, pour la simplification du tracé du circuit les sorties ne sont pas dans un ordre logique.

## Alimentation

La régulation de l'alimentation est réalisée par un classique 7805. Les condensateurs électrolytiques  $C_1$  et  $C_2$  réalisent un filtrage supplémentaire de la tension primaire, tension qui pourra être délivrée par un bloc secteur du commerce et qui sera supérieure à 5V. Chaque circuit intégré du montage possède son propre condensateur de découplage de 100 nF (LCC jaune). On remarquera que la masse est ramenée sur chacun des 9 ports du montage et que le 5V est ramené sur le port analogique et sur



les ports 5, 6, 7 et 8. Ceci dans le but d'alimenter les futures cartes externes.

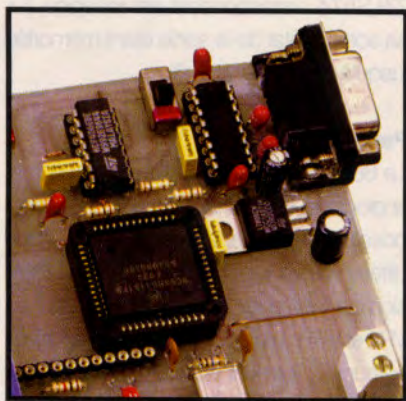
## Réalisation

Vous trouverez **figures 5 et 6**, le schéma du circuit imprimé et le schéma d'implantation de la carte. Il faudra un minimum de vigilance lors de la réalisation, car la carte comporte de nombreuses pistes proches les unes des autres et relativement fines de surcroît.

La majorité des pastilles seront à percer à l'aide d'un foret de 0,8 mm de diamètre. Seules les pastilles du bornier, du régulateur et des interrupteurs seront à percer à 1,5 mm.

Concernant le montage des composants, on commencera par la mise en place des straps, la carte en comportant 15.

On soudera ensuite les résistances puis les condensateurs dont il faudra impérativement respecter la polarité, particulièrement pour ceux qu'utilise le MAX232.

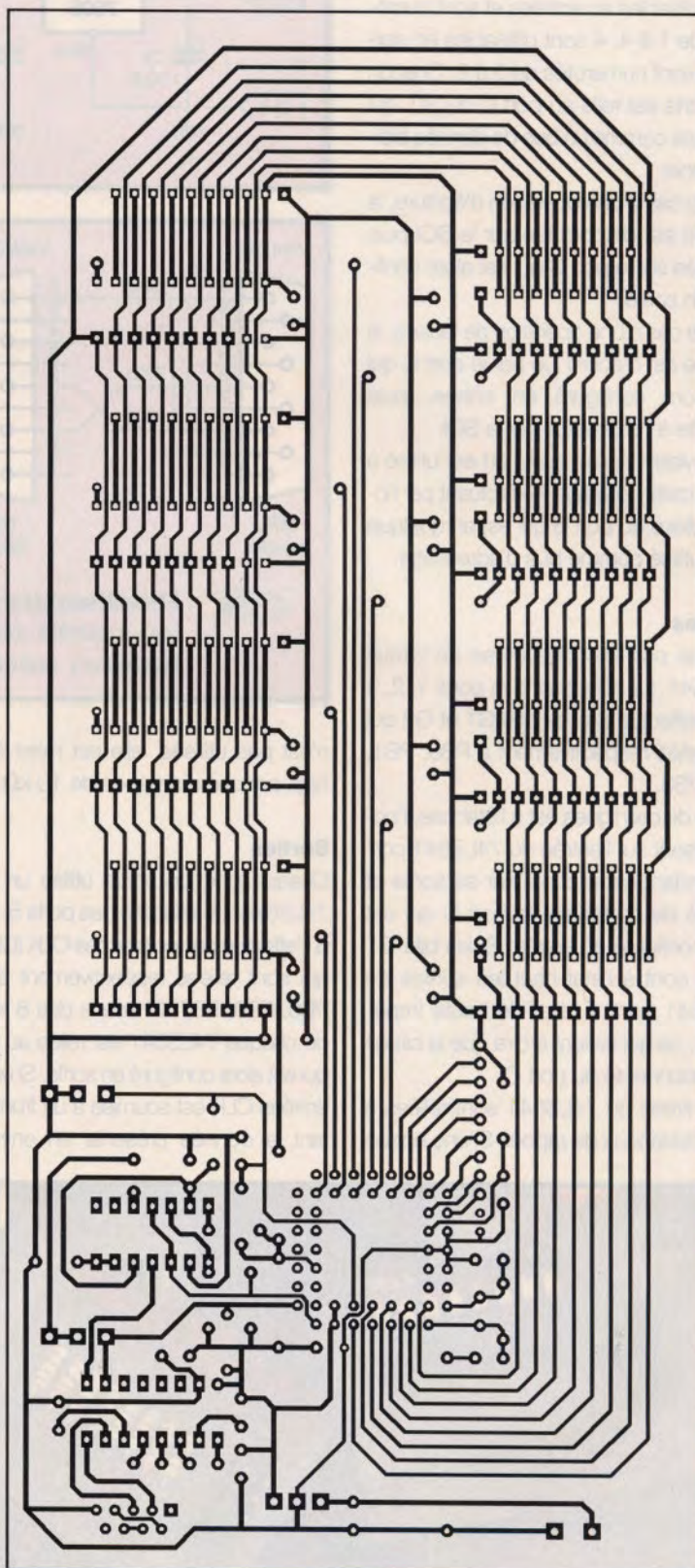


On terminera par les supports des CI, le 7805, les interrupteurs et le connecteur DB9. Un soin particulier sera pris lors de la mise en place du quartz, de la résistance  $R_1$  et des condensateurs  $C_9$  et  $C_{10}$  qui conditionnent l'horloge interne du 68HC11.

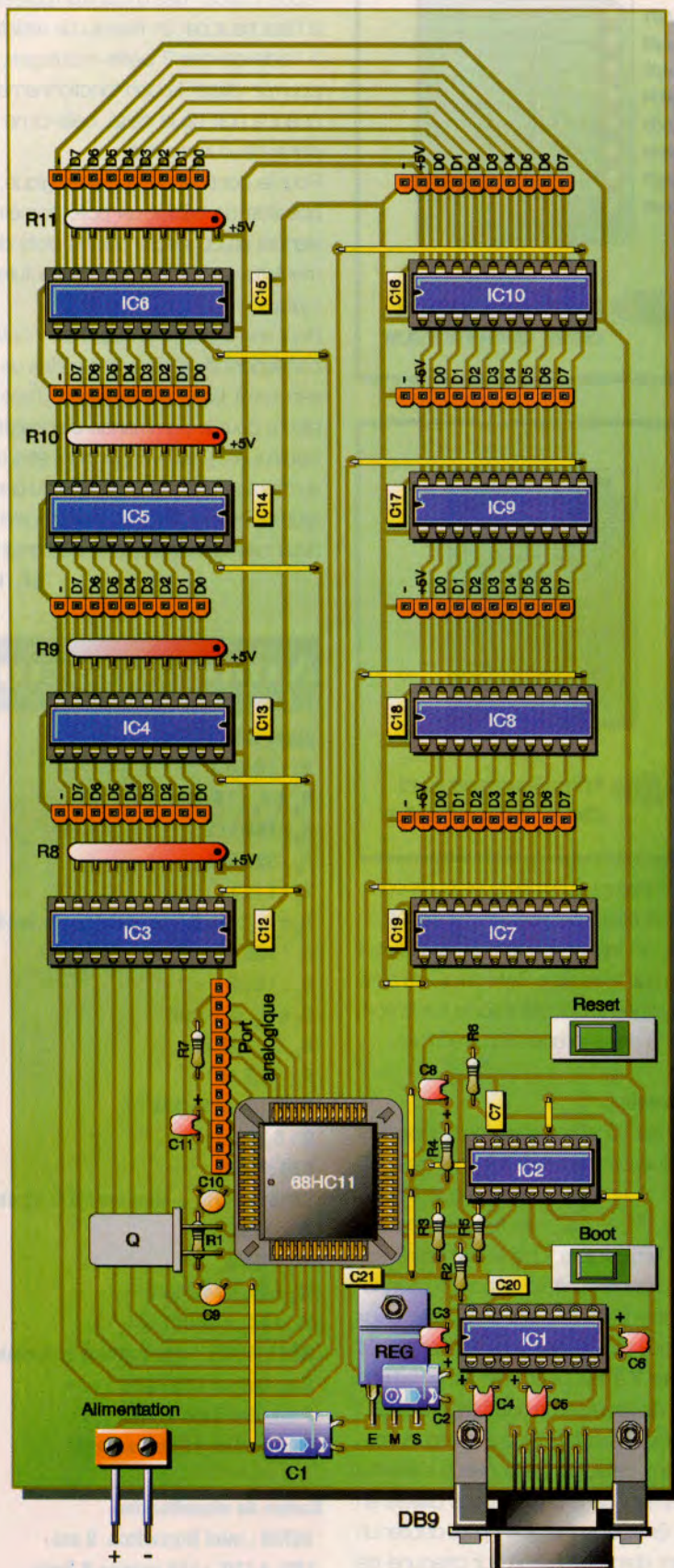
Avant la mise en place des CI sur leur support respectif, il est conseillé de mettre le montage sous tension et de vérifier la tension d'alimentation à l'aide d'un voltmètre qui doit être à +5V à +/- 5% près.

Si tout est correct, vous pourrez placer les différents circuits intégrés sur leur support respectif en ayant auparavant coupé l'alimentation.

Ensuite, il ne reste plus qu'à programmer le 68HC11.



**5** Tracé du circuit imprimé



## 6 Implantation des éléments

## Programmation du 68HC11

Pour la programmation de la carte, vous aurez besoin du programme EELoad.EXE qui va charger le programme PROG.REC en EEPROM. Vous pourrez vous procurer ces programmes auprès de la revue.

Protocole de transfert du fichier PROG.REC

- Mise sous tension de la carte  
RESET fermé,  
BOOT ouvert.
- Ouverture de RESET
- Lancement du programme EELoad.EXE
- Entrez le numéro du port de communication,
- Le programme de téléchargement est placé en RAM.
- A la question : DO YOU WANT ECHO TO SCREEN ?
- Si réponse Y on voit les octets téléchargés défilier à l'écran,
- Si réponse N on ne voit pas les octets défilier.
- A la question : SELECT INTERNAL, EXTERNAL OR VERIFY EEPROM OPTION (I/X/V) ?

Tapez : I

- A la question : ENTER FILENAME TO DOWNLOAD

Tapez : prog.rec

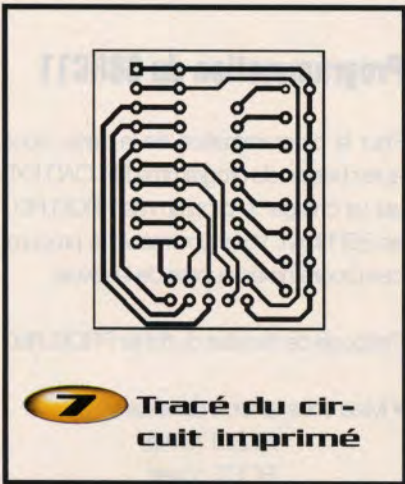
- A la fin du transfert, DOWNLOAD COMPLETE s'affiche
- fermez RESET  
fermez BOOT,  
ouvrez RESET.

La carte est alors prête à fonctionner.

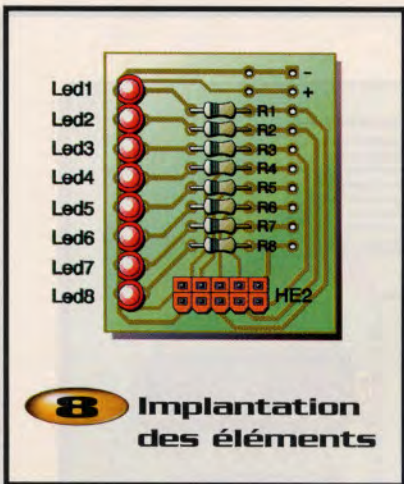
Remarque : Il est préférable de réaliser vous-même le câble de liaison série, surtout si sa longueur est importante. Il se compose de 3 fils et de deux connecteurs femelles DB9. Le schéma de câblage vous est fourni **figure 3**.

### Liste des commandes

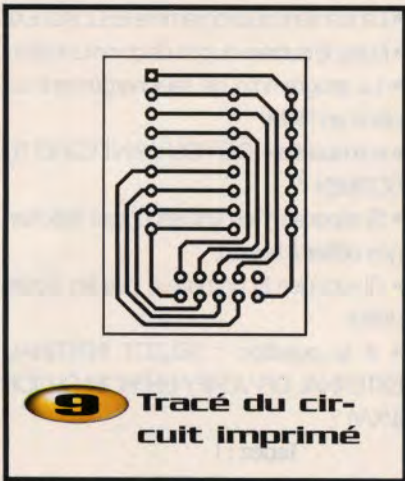
L'écriture ou la lecture de données sur chaque port s'effectue sur ordre du PC au  $\mu$ C par l'envoi d'une commande sur la ligne série. Pour cela, on peut utiliser Terminal de Windows 3.1, Hyper Terminal de Windows 95 ou le programme prog.bas écrit en QBasic que vous pourrez aussi vous procurer par l'intermédiaire de la revue. Dans tous les cas, le port COM du PC sera



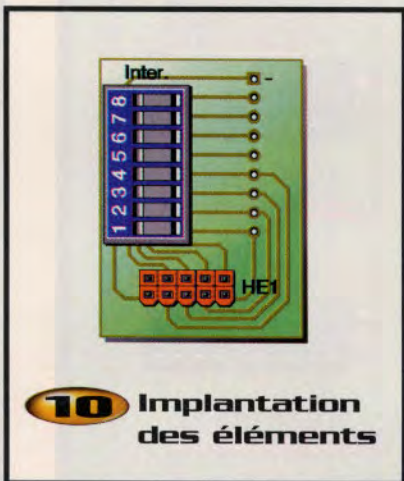
**7** Tracé du circuit imprimé



**8** Implantation des éléments



**9** Tracé du circuit imprimé



**10** Implantation des éléments

configuré à 9600 bauds (ou 1200 suivant utilisation) avec 8 bits de données et 1 bit d'arrêt.

**Liste des différentes commandes**

- P** : permet de tester la liaison série, en retour le  $\mu$ C envoie le caractère 'O',
  - T L** : lecture d'une tension, L étant le numéro de la ligne choisie ( $0 \leq L \leq 7$ ),
  - L port** : lecture d'un octet (port étant le numéro du port avec  $1 \leq \text{port} \leq 8$ ),
  - E port, donnée** : écriture d'un octet (donnée étant l'octet à écrire avec  $4 \leq \text{port} \leq 8$ ),
  - D port, n** : désactivation d'un bit ("n" étant le numéro du bit avec  $0 \leq n \leq 8$  et  $4 \leq \text{port} \leq 8$ ),
  - A port, n** : activation d'un bit ("n" étant le numéro du bit avec  $0 \leq n \leq 8$  et  $4 \leq \text{port} \leq 8$ ),
  - C port, n** : complémentation d'un bit ("n" étant le numéro du bit avec  $0 \leq n \leq 8$  et  $4 \leq \text{port} \leq 8$ ),
- Chaque commande d'écriture qui vient d'être effectuée par le  $\mu$ C est confirmée au PC par l'envoi du caractère 'O'. Si une commande transmise n'est pas reconnue par le  $\mu$ C, le caractère '?' est envoyé au PC.

On remarquera que même les ports configurés en sortie peuvent être accédés en lecture, en fait, la donnée lue ne provient pas du port mais de la RAM du  $\mu$ C où a été sauvegardée la donnée inscrite sur le port, lors de la dernière opération d'écriture.

**Les essais**

Le premier contrôle de bon fonctionnement à effectuer est celui de la liaison entre le PC et le  $\mu$ C. Pour cela, il suffit d'envoyer la commande 'P' au  $\mu$ C, en réponse celui-ci envoie le caractère 'O' si le test est réussi. Pour faciliter le contrôle de bon fonctionnement des différents ports de la carte, deux montages très simples vous sont proposés **figures 7 à 10**. Le premier est en fait une platine de visualisation qui comporte 8 LED rouges et 8 résistances qui limitent le courant dans celles-ci. Cette platine s'enfiche directement dans un des ports utilisés en sortie. En effet le 74LS541 peut débiter un courant d'environ 50mA sur chacune de ses sorties ce qui est amplement suffisant. Le deuxième montage permet, par l'intermédiaire d'un Dipswitch, de mettre indépendamment à la masse chacune des

lignes d'un port utilisé en entrée. A l'état de repos, chacun des 8 bits d'un port est mis à l'état haut par un réseau de résistance. A l'aide de ces 2 petits montages, vous pourrez valider le bon fonctionnement de chaque port de la carte, celle-ci ne comportant aucun réglage.

Pour le contrôle du port analogique, il est possible de câbler un potentiomètre qui viendra se connecter sur les plots de l'alimentation et sur les lignes de lecture analogique.

Pour une utilisation pratique du montage, il conviendra de réaliser des cartes de puissances à base de relais ou d'optocoupleurs pour la commande d'appareillages électriques. Celles-ci pourront être reliées au montage par l'intermédiaire du connecteur de type HE10 que comportent les deux platines décrites précédemment.

D. REY

**Nomenclature**

**Carte principale**

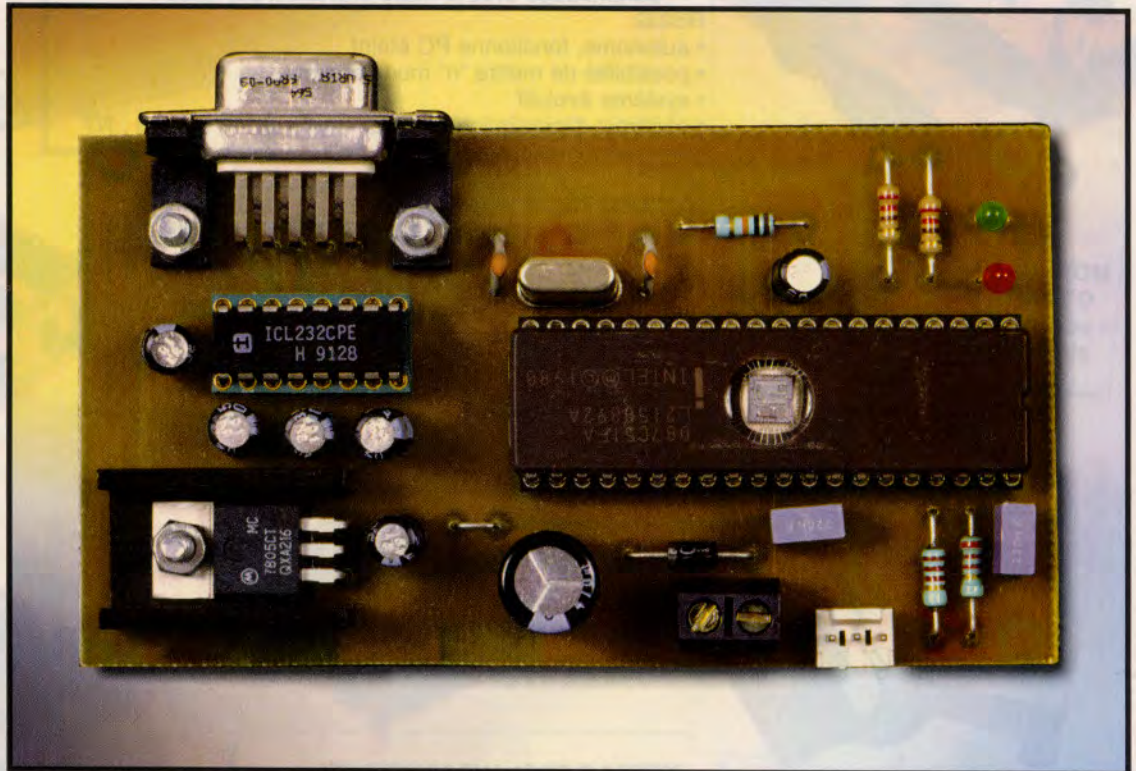
- R<sub>1</sub>** : 10 M $\Omega$
- R<sub>2</sub> à R<sub>4</sub>** : 10 k $\Omega$
- R<sub>5</sub>** : 100 k $\Omega$
- R<sub>6</sub>** : 330 k $\Omega$
- R<sub>7</sub>** : 1 k $\Omega$
- R<sub>8</sub> à R<sub>11</sub>** : réseau de résistances 8x10 k $\Omega$
- C<sub>1</sub>** : 100  $\mu$ F
- C<sub>2</sub>** : 10  $\mu$ F
- C<sub>3</sub> à C<sub>6</sub>, C<sub>8</sub>** : 1  $\mu$ F
- C<sub>9</sub>, C<sub>10</sub>** : 22  $\mu$ F
- C<sub>11</sub>** : 2,2  $\mu$ F
- C<sub>7</sub>, C<sub>12</sub> à C<sub>21</sub>** : 100 nF
- Q** : 8 MHz
- REG** : 7805
- 68HC11A1FN + support PLCC** 52 cts
- IC<sub>1</sub>** : MAX232
- IC<sub>2</sub>** : 4066
- IC<sub>3</sub> à IC<sub>6</sub>** : 74LS541
- IC<sub>7</sub> à IC<sub>10</sub>** : 74HCT574
- BOOT, RESET** : interrupteur unipolaire
- Connecteur DB9 mâle**
- 1 bornier à vis** 2 cts
- 3 barrettes** 32 cts/ sécable

**Cartes de visualisation**

- INTER** : mini Dipswitch, 8 cts
- LED<sub>1</sub> à LED<sub>8</sub>** : LED rouges  $\varnothing$  3mm
- R<sub>1</sub> à R<sub>8</sub>** : 220  $\Omega$ /1/4 W
- HE<sub>1</sub>, HE<sub>2</sub>** : 2 barrettes HE10 mâle, 5cts

# Convertisseur RS232 pour bus I2C

Le montage que nous vous proposons ici pourra vous être d'un grand secours pour la mise au point d'un système articulé autour du bus I2C. Il permet de prendre le contrôle d'un bus I2C avec un minimum d'effort, sans forcément avoir besoin de réaliser un programme spécifique (ce qui est généralement le cas des outils associés au bus I2C. Évidemment, vous pourrez aussi utiliser le montage pour réaliser une centrale domotique à l'aide de votre PC, mais dans ce cas de figure, il vous faudra réaliser le programme qui correspond à l'application que vous voulez créer. Cependant vous n'aurez qu'à programmer le dialogue avec l'interface RS232, ce qui est généralement bien plus simple que la programmation des composants qui contrôlent un bus I2C



## Schéma

Le schéma de notre montage est reproduit en **figure 1**. Comme vous pouvez le constater sur le schéma, un microcontrôleur 87C51 réalise toute la fonction de ce montage. Les signaux RS232 sont mis à niveaux par un circuit MAX232 ( $U_2$ ) auquel sont associés les 4 condensateurs habituels.

L'oscillateur du microcontrôleur est mis en œuvre de façon tout à fait classique au moyen d'un quartz et de deux capacités de "pied" associées. Quant à la remise à zéro du microcontrôleur, une simple cellule RC suffit pour notre application ( $R_1/C_1$ ).

Le microcontrôleur ne dispose pas de la logique nécessaire au décodage du protocole du bus I2C. C'est donc le logiciel qui se chargera de respecter ce protocole. Cela ne pose pas de problème dans le cas où notre montage est le

maître du bus. Dans le cas contraire, il pourrait y avoir des collisions lors des échanges sur le bus I2C. Du fait de la structure à collecteur ouvert des signaux SDA et SCL, cela ne pose pas de problème du côté matériel.

Simplement, quelques messages peuvent être perdus si le montage émet sur le bus en même temps qu'un autre maître.

En ce qui concerne les signaux SDA et SCL du bus I2C, les résistances de "Pull-Up" ( $R_4$  et  $R_5$ ) permettent de charger le bus pour garantir l'état haut et le temps de commutation. Si vous raccordez le montage à un bus déjà équipé de résistances de "Pull-Up", vous devrez augmenter la valeur de  $R_4$  et  $R_5$ . La résistance équivalente pour chacune des lignes SDA et SCL ne devra pas descendre en dessous de 1,5 k $\Omega$  environ.

Notez que les lignes SDA et SCL ne sont pas protégées contre les sur-

tensions (mise au +12V par exemple ou décharges électrostatiques). Par contre, elles ne craignent pas les courts-circuits à la masse (et pour cause !).

Pour visualiser le bon fonctionnement du montage, nous avons ajouté deux diodes LED. Étant donné le faible courant demandé pour faire fonctionner les diodes LED, nous avons jugé inutile d'ajouter des transistors en sorties des ports "P1.0" et "P1.1". Cela augmente la puissance dissipée par le microcontrôleur, ce qui ne pose aucun problème dans le cas de notre montage, puisque la plupart des ports du microcontrôleur sont inutilisés.

Le montage sera alimenté par une tension de 9VDC à 12VDC qui n'a pas besoin d'être stabilisée. Par exemple, vous pourrez utiliser un bloc d'alimentation d'appoint pour calculatrice capable de fournir 100mA sous 12VDC. La diode  $D_1$

permet de protéger le montage en cas d'inversion du connecteur d'alimentation.

### Réalisation

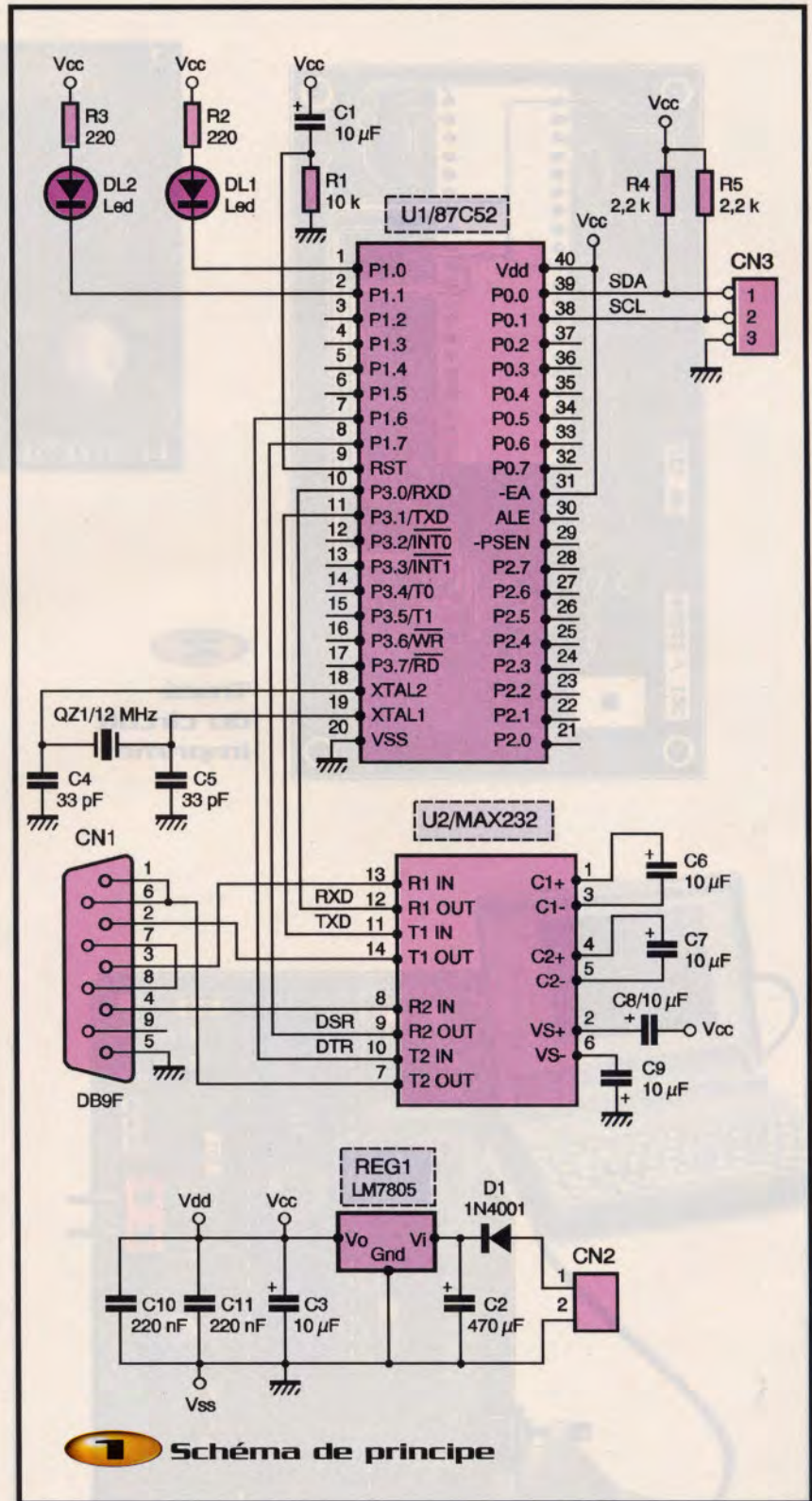
Le dessin du circuit imprimé est visible en **figure 2**. La vue d'implantation associée est reproduite en **figure 3**. Les pastilles seront percées à l'aide d'un foret de 0,8mm de diamètre, pour la plupart. En ce qui concerne REG<sub>1</sub>, CN<sub>2</sub>, CN<sub>3</sub> et D<sub>1</sub>, il faudra percer les pastilles avec un foret de 1mm de diamètre.

Il n'y a pas de difficulté particulière pour l'implantation. Soyez tout de même attentifs au sens des condensateurs et des circuits intégrés. Respectez scrupuleusement le découplage des lignes d'alimentations si vous voulez éviter les surprises. Vous noterez la présence d'un strap qu'il est préférable d'implanter en premier pour des raisons de commodité. Veillez bien à choisir un connecteur femelle pour CN<sub>1</sub>. Car un modèle mâle s'implante parfaitement, mais les points de connexions se retrouvent inversés par symétrie par rapport à l'axe vertical. Dans ce cas, il n'y a aucune chance pour que votre montage dialogue avec votre PC, à moins de fabriquer un câble spécial pour rétablir l'ordre voulu.

En ce qui concerne le câble nécessaire pour relier notre montage à un PC de type AT, il vous suffira de fabriquer un câble équipé d'un connecteur DB9 mâle d'un côté et d'un connecteur DB9 femelle de l'autre côté (liaison fil à fil de la broche 1 à la broche 9). L'utilisation de connecteurs à sertir est plus pratique, mais les liaisons nécessaires étant peu nombreuses, vous pourrez utiliser des connecteurs à souder. Enfin, ajoutons que le connecteur CN<sub>1</sub> sera immobilisé par deux boulons montés dans les passages prévus à cet effet.

Le régulateur REG<sub>1</sub> 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 plus d'importance.

Le microcontrôleur sera programmé avec le contenu d'un fichier que vous pourrez vous procurer par téléchargement d'après le CD joint à la revue ou sur notre site Internet. Le fichier

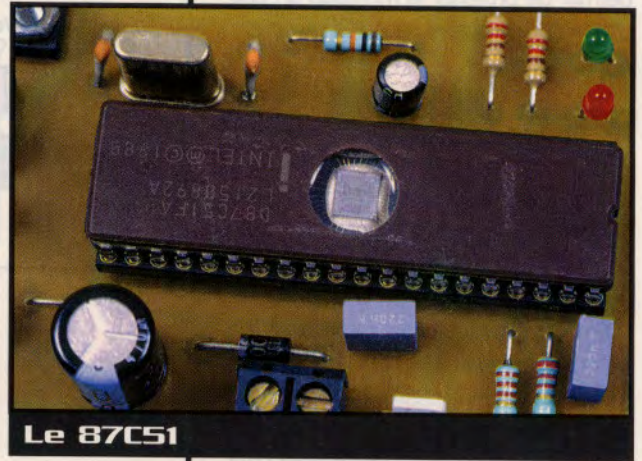
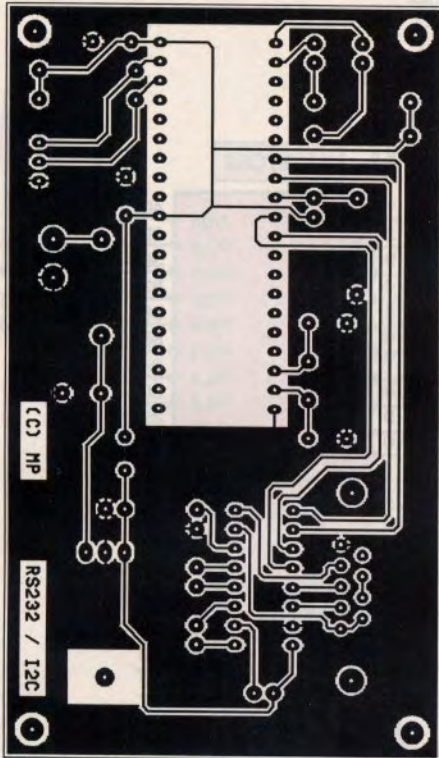


CVRSI2C.ROM qui est le reflet binaire du contenu de l'EPROM tandis que le fichier CVRSI2C.HEX qui correspond au format HEXA INTEL.

Le montage est relativement simple à utiliser. Les paramètres de la liaison RS232 sont habituels : 9600 bauds, 8 bits de données, 1 bit de stop et pas de parité. Dans un premier temps, pour tester le

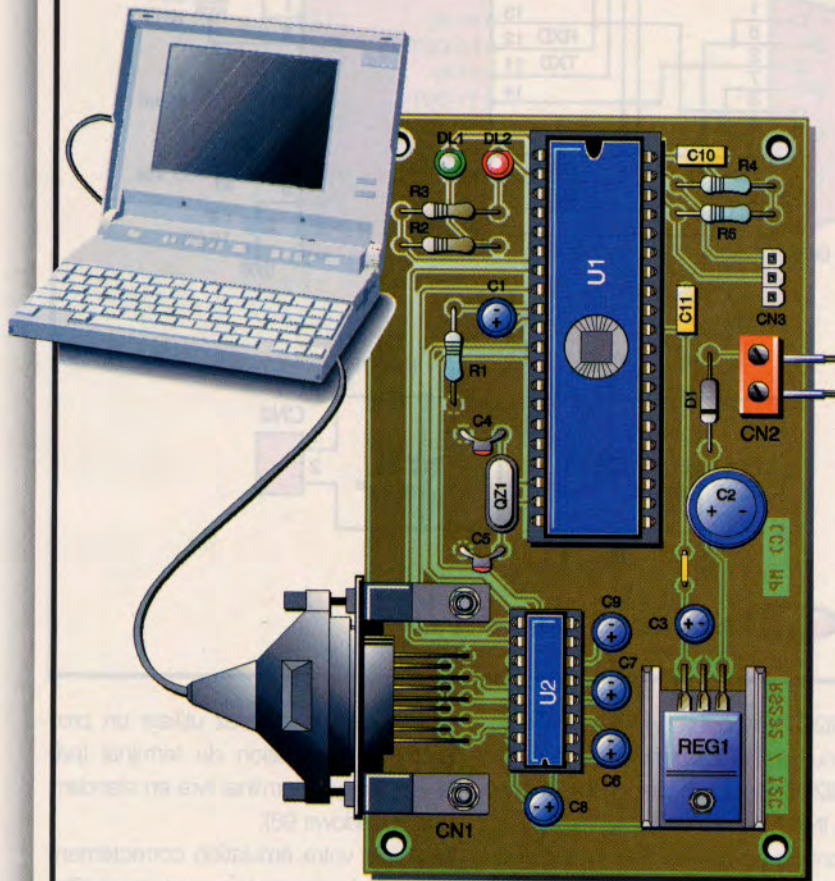
montage vous pourrez utiliser un programme d'émulation de terminal (par exemple Hyperterminal livré en standard avec Windows 98).

Une fois votre émulation correctement paramétrée, vous devez voir apparaître un message d'accueil lorsque vous mettez le montage sous tension. Si ce n'est pas le cas, vérifiez le câblage.



**2**

### Tracé du circuit imprimé



**3** Implantation des éléments

## Nomenclature

**CN<sub>1</sub>** : connecteur SubD 9 points femelle, sorties coudées, à souder sur circuit imprimé (par exemple référence HARTING 09 66 112 7601).

**CN<sub>2</sub>** : bornier de connexion à vis 2 plots, au pas de 5,08mm, à souder sur circuit imprimé, profil bas.

**CN<sub>3</sub>** : barrette mini-KK 3 contacts, sorties droites, à souder sur circuit imprimé, référence MOLEX 22-27-2031.

**C<sub>1</sub>, C<sub>3</sub>, C<sub>6</sub> à C<sub>9</sub>** : 10 µF/25V sorties radiales

**C<sub>2</sub>** : 470 µF/25V sorties radiales

**C<sub>4</sub>, C<sub>5</sub>** : 33pF céramique, au pas de 5,08mm

**C<sub>10</sub>, C<sub>11</sub>** : 220 nF

**DL<sub>1</sub>** : diode LED verte 3mm

**DL<sub>2</sub>** : diode LED rouge 3mm

**D<sub>1</sub>** : 1N4001

(diode de redressement 1A/100V)

**QZ<sub>1</sub>** : quartz 11,0592 MHz en boîtier HC49/U

**REG<sub>1</sub>** : régulateur LM7805 (5V)

en boîtier TO220 + Dissipateur thermique 18°C/W (par exemple Shaffner référence RAWA 400 9P)

**R<sub>1</sub>** : 10 kΩ 1/4W 5% (marron, noir, orange)

**R<sub>2</sub>, R<sub>3</sub>** : 220 Ω 1/4W 5%

(rouge, rouge, marron)

**R<sub>4</sub>, R<sub>5</sub>** : 2,2 kΩ 1/4W 5%

(rouge, rouge, rouge)

**U<sub>1</sub>** : 87C51 avec EPROM interne (12 MHz)

**U\*2** : MAX232

Si rien n'y fait, vérifiez la tension d'alimentation du montage et les tensions présentes en sortie du circuit MAX232 (+12V à +9V sur la broche 2 et -12V à -9V sur la broche 6).

Une fois la connexion établie avec l'interface, vous pourrez lui envoyer des ordres. L'interface reconnaît 3 commandes, comme l'indique la **figure 4**. Pour être prise en compte, les commandes doivent être suivies du caractère CR (code hexadécimal 0Dh) pour valider la saisie.

Les lettres en majuscules ou minuscules sont acceptées indifféremment par le montage. Néanmoins, la syntaxe doit être scrupuleusement respectée. Les deux points après le nombre de lecture demandée et l'espace entre chaque octet destiné à être transmis sur le bus I2C doivent être respectés précisément. Le passage dans le mode de réponse numérique (commande "M1") vous permettra d'alléger l'écriture de vos propres applications. Les codes d'erreurs sont alors indiqués par une valeur hexadécimale codée sur 2 chiffres ASCII, suivis des caractères CR, LF et du prompt ">". Il vous suffira donc simplement de filtrer les caractères CR, LF et ">", pour ne garder que le contenu utile du message. Dans ce mode de réponse, le code d'erreur est toujours transmis en premier. Sinon il ne serait pas évident de faire la différence entre le message d'erreur et les octets demandés.

Ajoutons que la diode LED verte permet de visualiser que le montage transmet sur le bus I2C, tandis que la diode LED rouge s'allume en cas d'erreur de communication (problème d'acknowledge lors des transferts sur le bus).

Pour tester facilement le montage, vous pourrez utiliser un circuit PCF8574 comme l'indique le schéma de la figure 6. Dans ce cas, le circuit PCF8574 répondra à l'adresse 40 (en hexadécimal). Notez que si vous utilisez un circuit PCF8574A, il répondra à l'adresse 70 (en hexadécimal). Les octets que vous envoyez aux PCF8574 apparaissent directement sur le port de sortie. Les bits qui sont à zéro imposeront l'allumage des diodes LED correspondantes. Si vous demandez une lecture au PCF8574, vous relirez l'état du port.

P. Morin

**M0**

La commande " M0 " passe les réponses en mode texte (mode actif par défaut).

**M1**

La commande " M1 " passe les réponses en mode numérique. Dans ce mode, l'interface remplace les messages par des chiffres, dont la signification est indiquée en figure 5.

**IxxWxxRxx:xx xx xx xx**

- 'Ixx' : I indique le début de la commande pour le bus I2C. Le paramètre xx correspond à l'adresse cible demandée sur le bus I2C (exprimé en hexadécimal).
- 'Wxx' : xx correspond au nombre d'octets que l'on souhaite écrire (les données suivent à la fin de la ligne de commande). Si xx=00 aucune écriture n'est demandée.
- 'Rxx' : xx correspond au nombre d'octets que l'on souhaite lire après l'opération d'écriture. Si xx=00 aucune lecture ne sera demandée sur le bus.
- ':' sépare la commande et les données éventuelles destinées au bus I2C. Suit la liste des octets à transmettre sur le bus I2C suivi un à un par le caractère espace.

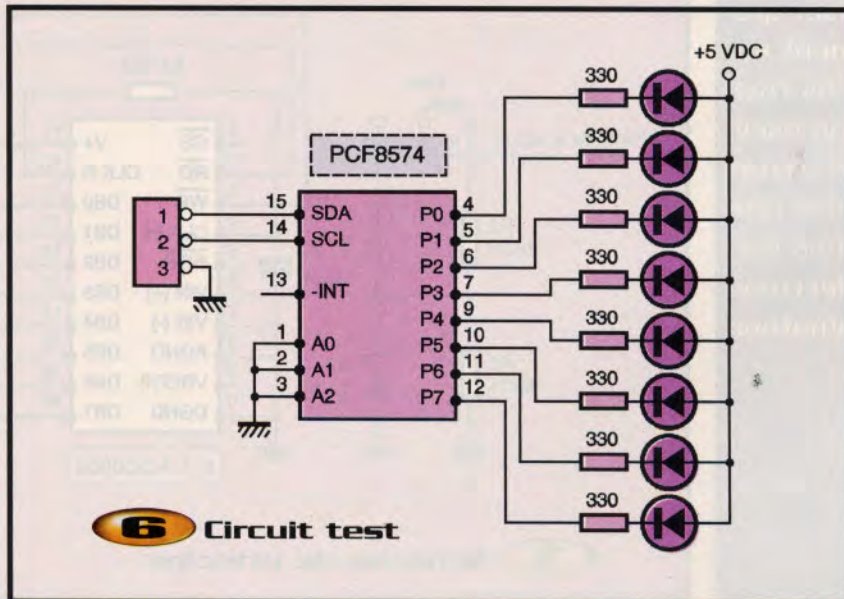
**Exemple de commande : I70W04R10:00 FF 55 AA**

Cette commande écrit successivement les octets 00 FF 55 AA à l'adresse 70, puis lit 16 octets (10 en hexadécimal) à la même adresse.

**4 L'interface reconnaît trois commandes**

Mode M1	Mode M0 (par défaut)	Signification du message
00	(rien, retour au prompt)	Commande correctement exécutée
01	ERR I2C ACK	Pas de réponse du bus I2C à l'adresse demandée
02	ERR I2C BUSY	Si un autre maître contrôle déjà le bus I2C
03	ERR SYNTAXE	Vous n'avez pas respecté scrupuleusement la syntaxe
04	ERR NB WRITE	Dépassement de la taille maximum du buffer (32)
05	ERR NB READ	Dépassement de la taille maximum du buffer (32)
06	ERR HEXA	Erreur dans la saisie d'un nombre hexadécimal

**5 Les modes**

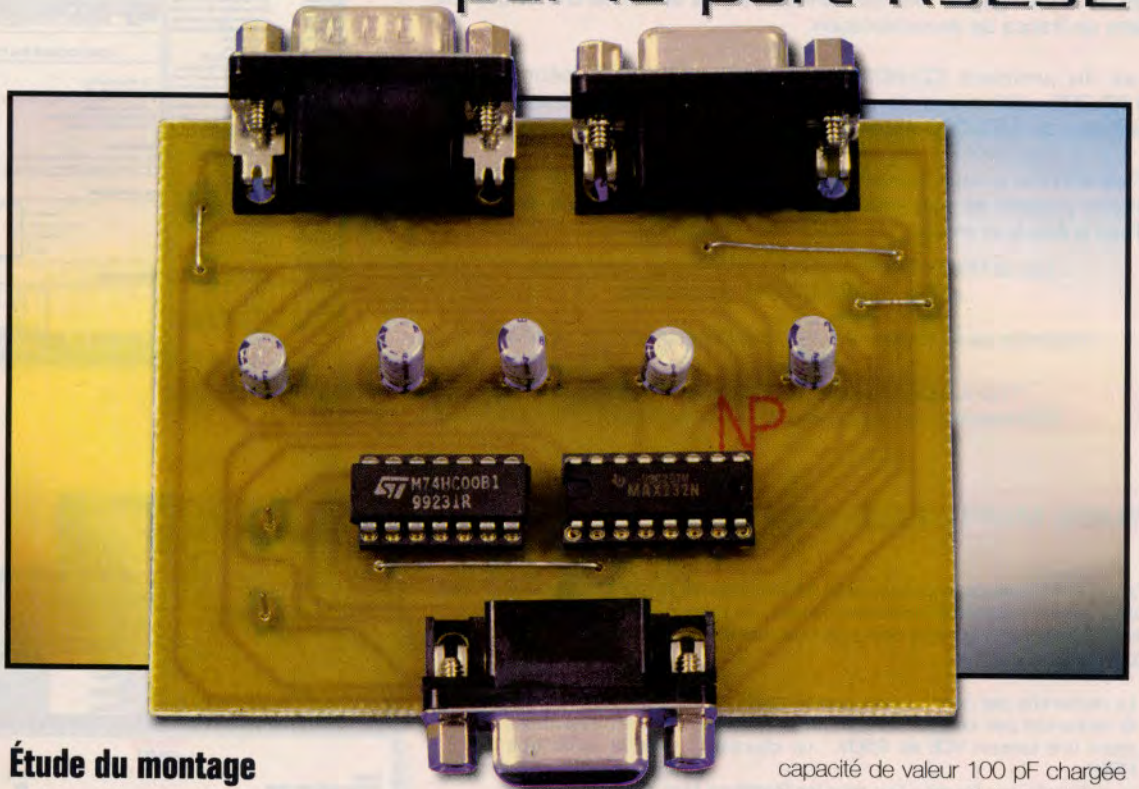


**6 Circuit test**



# Analyseur de protocole

par le port RS232



## Étude du montage

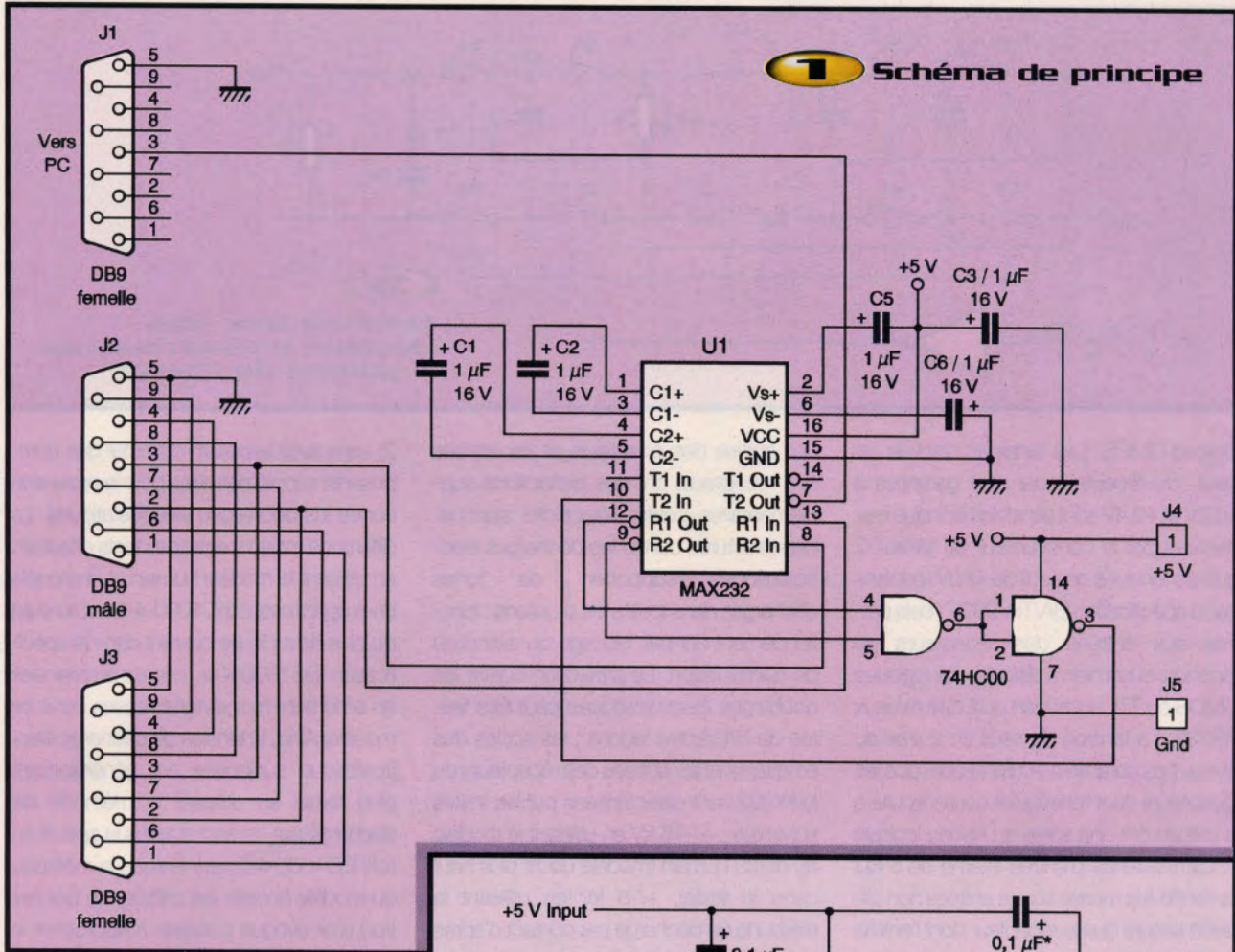
Notre circuit s'intercale entre une ligne RS232 par l'intermédiaire des connecteurs  $J_2$  (mâle) et  $J_3$  (femelle). Les deux signaux en émission et en réception (suivant de quel côté de la ligne on se place) issus de cette ligne sont prélevés pour être une première fois convertis en signaux CMOS. Ensuite, le signal combiné provenant des deux signaux CMOS est reconverti en un niveau RS232 pour être envoyé sur l'entrée de réception de l'ordinateur. Tout le montage est articulé autour du circuit MAX232 de chez le constructeur MAXIM représenté à la **figure 2** et dont nous allons expliquer le fonctionnement. Ce composant fait partie de la famille des émetteurs/récepteurs de ligne conçus pour les lignes qui respectent le protocole RS232 ou V.28 dans des environnements bruités. Toutes les sorties en émission et toutes les entrées en réception sont protégées contre des chocs de décharges électrostatiques pouvant atteindre +/-

15 kV sans blocage de la ligne. Les émetteurs et les récepteurs répondent aux spécifications EIA/TIA-232E et CCITT V.28 avec un taux de transfert des données pouvant aller jusqu'à 120 kbauds/seconde lorsqu'ils sont chargés conformément aux spécifications EIA/TIA-232E et CCITT V.28. Le MAX232 fonctionne avec cinq capacités qui sont toutes de valeur 1  $\mu$ F et qui remplissent la fonction de pompe de charge. De plus, ce composant possède un temps de balayage inférieur à 3V par  $\mu$ s, sous une tension unique d'alimentation égale à 5V. Le MAX232 comprend trois sections qui sont les suivantes : convertisseurs de pompe de charge en tension, émetteurs et récepteurs. Comme il a déjà été mentionné plus haut dans ce texte, ce composant survit à des décharges égales à +/- 15 kV au maximum sur les entrées et les sorties RS232, conditions testées en utilisant le modèle du corps humain (ce modèle consiste en une

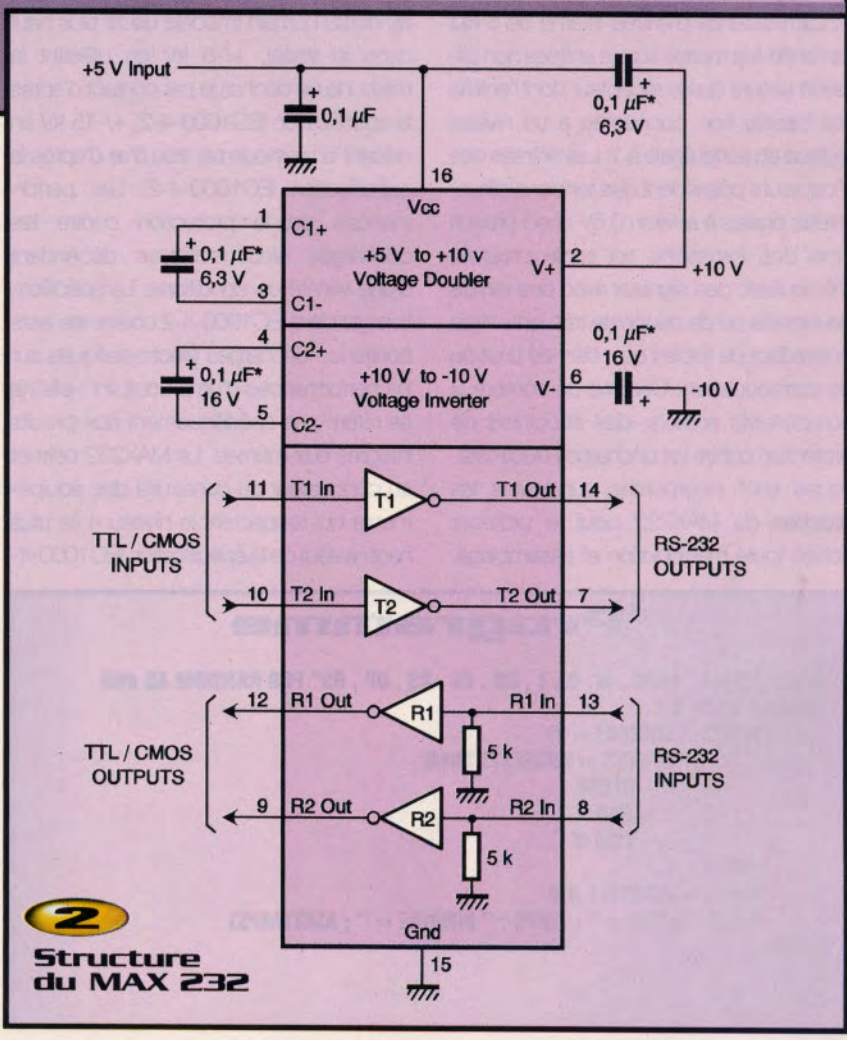
capacité de valeur 100 pF chargée par une tension de décharge électrostatique, puis déchargée à travers une résistance de valeur 1,5 k $\Omega$ ). Lorsque ce composant est testé selon la norme IEC-1000-4-2, il survit à des décharges par contact égales à +/-8 kV et à des décharges par trou d'air égales à +/-15 kV ; il est donc recommandé pour être utilisé dans des environnements rudes et dans des applications où les connexions RS232 sont changées fréquemment (comme dans les ordinateurs portables, par exemple, ou dans notre application dans laquelle on doit changer souvent les connexions de la ligne à espionner). Les conversions de +/-5V vers +/-10V (niveaux RS232) sont effectuées par des doubles convertisseurs à pompe de charge en tension dont la **figure 3** nous présente le schéma bloc. Le premier convertisseur à pompe de charge utilise la capacité  $C_1$  pour doubler la tension de +5V en +10V, stockant cette valeur de +10V dans le filtre de sortie composé de la capa-

Un analyseur de protocole est un des instruments très utilisés lorsqu'il s'agit d'espionner un dialogue entre deux systèmes, surtout lorsqu'une anomalie se produit. Mais cet appareil est très coûteux. Le schéma de notre montage représenté à la **figure 1** est une alternative peu onéreuse, facile à réaliser et qui s'interface au port série RS232 de n'importe quel ordinateur compatible PC.

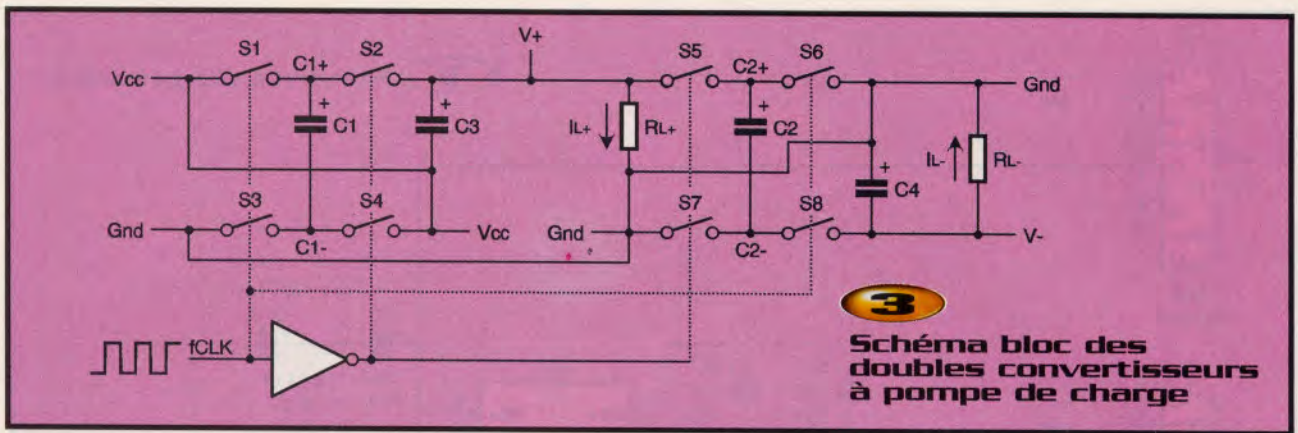
**1** Schéma de principe



cité  $C_3$ . Le second convertisseur à pompe de charge utilise la capacité  $C_2$  pour inverser la tension de +10V en -10V, stockant cette valeur de -10V dans le filtre de sortie composé de la capacité  $C_4$ . Avec une tension d'alimentation égale à +5V, la tension de sortie typique de l'émetteur oscille entre +/-8V lorsqu'une charge de 5 k $\Omega$  est placée sur le récepteur RS232. Les sorties sont garanties pour respecter les spécifications EIA/TIA-232E et CCITT V.28 qui impose des niveaux de sortie de +/-5V au minimum sous les pires conditions d'utilisation, c'est-à-dire incluant une charge de 3 k $\Omega$ , la tension d'alimentation minimale (-0,3V) et la température de fonctionnement maximale (+70°C). La tension de sortie en circuit ouvert oscille entre la tension d'alimentation -0,6V et V- (-10V nominal). Les tensions de seuil en entrée sont compatibles avec les niveaux CMOS et TTL. Ainsi, les émetteurs convertissent des signaux en entrée qui possèdent des niveaux CMOS ou TTL en signaux RS232. A l'opposé, les récepteurs convertissent les signaux en entrée compatibles avec la norme RS232 en niveaux de sortie compatibles avec la



**2** Structure du MAX 232



logique CMOS. Les tensions d'entrée de seuil du récepteur qui sont garanties à +0,8V et +2,4V sont sensiblement plus respectées par le constructeur du MAX232 que les tensions de seuil de +/-3V requises par la spécification EIA/TIA-232E ; cela permet aux entrées des récepteurs de répondre aussi bien à des niveaux logiques CMOS ou TTL aussi bien qu'à des niveaux RS232. La tension de seuil en entrée au niveau bas garantie à +0,8V assure que les récepteurs dont l'entrée est court-circuité à la masse ont une sortie au niveau logique 1. La résistance d'entrée interne de 5 k $\Omega$  ramenée à la masse sur les entrées non utilisées assure que le récepteur dont l'entrée est laissée non connectée a un niveau logique en sortie égale à 1. Les entrées des récepteurs possèdent des tensions d'hystérésis égales à environ 0,5V ; ceci produit ainsi des transitions en sortie propres, même avec des signaux avec des temps de montée ou de descente très lents mais possédant de faibles quantités de bruit ou de surmodulation. Comme de nombreux composants actuels, des structures de protection contre les décharges électrostatiques sont incorporées sur toutes les broches du MAX232 pour le protéger contre toute manipulation et assemblage.

Les sorties des émetteurs et les entrées des récepteurs ont des protections supplémentaires contre l'électricité statique. Les structures contre les décharges électrostatiques supportent de fortes décharges dans toutes les situations : fonctionnement normal, blocage ou extinction de l'alimentation. La protection contre les décharges électrostatiques peut être testée de différentes façons ; les sorties des émetteurs et les entrées des récepteurs du MAX232 sont caractérisées par les limites suivantes : +/-15 kV en utilisant le modèle du corps humain (modèle décrit plus haut dans le texte), +/-8 kV en utilisant la méthode de décharge par contact d'après la spécification IEC1000-4-2, +/-15 kV en utilisant la méthode par trou d'air d'après la spécification IEC1000-4-2. Les performances de la protection contre les décharges électrostatiques dépendent d'une variété de conditions. La spécification standard IEC1000-4-2 couvre les tests contre les décharges électrostatiques sur les performances d'un produit fini ; elle ne se réfère pas spécifiquement aux circuits intégrés eux-mêmes. Le MAX232 permet au concepteur de construire des équipements qui respectent le niveau 4 (le plus haut niveau) de la spécification IEC1000-4-

2, sans avoir le besoin d'ajouter des composants supplémentaires pour la protection contre les décharges électrostatiques. La différence majeure entre des tests effectués en utilisant le modèle humain et le modèle de la spécification IEC1000-4-2 est au sujet du plus haut pic de courant dans la spécification IEC1000-4-2, car les résistances en série sont moins nombreuses dans ce modèle. Ainsi, la tension de décharge électrostatique supportée est généralement plus faible en utilisant la méthode de décharge par contact d'après la spécification IEC1000-4-2 que lorsque la méthode du modèle humain est utilisée. Le test par trou d'air évoqué consiste à approcher le composant avec une sonde chargée. La méthode avec décharge par contact connecte la sonde au composant avant que la sonde ne soit stimulée par de l'énergie. Le modèle de la machine pour les tests de protection contre les décharges électrostatiques utilise une capacité de stockage de valeur 200 pF et une résistance de décharge de valeur égale à 0 ; son objectif est de simuler le stress causé par le contact produit lors de la manipulation ou de l'assemblage durant la fabrication. Bien sûr, toutes les broches nécessitent cette protection durant la fabrication, pas seulement les entrées et les sorties RS232. Par conséquent, après l'assemblage d'une carte d'ordinateur, le modèle de la machine est moins significatif pour les broches d'entrée/sortie. Le type de capacité utilisé pour C<sub>1</sub> à C<sub>5</sub> n'est pas critique pour un fonctionnement correct. Les valeurs nominales requises sont de 1  $\mu$ F, bien que ces valeurs peuvent aller jusqu'à 10  $\mu$ F sans problème. Des capacités de type céramique, électrolytique en aluminium ou en tantale peuvent convenir. Lorsque la plus faible valeur est utilisée pour ces capacités, il faut être certain que cette valeur ne se dégrade pas

## Programme

```
OPEN "COM1 : 9600 , N , 8 , 1 , CD , CS , DS , OP , RS" FOR RANDOM AS #99
WHILE [ 1 <> 0 ]
  WHILE ( LOC(99) = 0 )
    IF INKEYS = CHR$(27) THEN
      CLOSE
      END
    END IF
  WEND
  TMPS = INPUT$(1,99)
  PRINT " ASCII = " ; TMPS ; " BINAIRE = " ; ASC(TMPS)
WEND
CLOSE
END
```

excessivement avec des variations de la température de fonctionnement ; dans le doute, il vaut mieux utiliser des valeurs qui sont doubles de la valeur nominale. Les résistances effectives en série avec les capacités qui augmentent habituellement à basses températures, influencent l'ondulation sur les tensions V+ (+10V) et V- (-10V). Il faut filtrer la tension d'alimentation par une capacité de même valeur que celle des capacités utilisées pour les pompes de charge. Une faible augmentation de la puissance peut être dérivée de V+ et de V- bien que cela réduise à la fois la dynamique de tension en sortie des convertisseurs ainsi que la marge de bruit ; une augmentation de la valeur des capacités (jusqu'à 10  $\mu$ F) aide à maintenir les performances lorsque de la puissance est demandée à V+ et à V-. Chaque émetteur est conçu pour piloter un seul récepteur. Des émetteurs peuvent être mis en parallèle pour piloter plusieurs récepteurs.

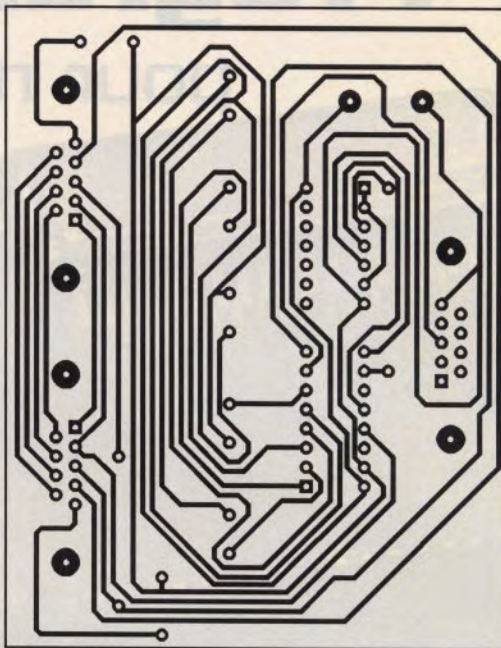
## Réalisation pratique

Le câblage de notre circuit ne pose aucune difficulté particulière. Ne pas oublier de câbler les 4 straps sur le circuit imprimé. Il est bien sûr recommandé de mettre le MAX232 sur un support dans le cas où l'utilisateur désirerait changer ce composant à la suite d'une mauvaise manipulation ou pour une autre raison. La **figure 4** représente le circuit côté pistes et la **figure 5** côté composants. Les données espionnées sur la ligne doivent être au format binaire pour que le programme en Quick Basic disponible sur le CDROM puisse rendre les caractères visualisés sur l'écran de l'ordinateur compréhensibles.

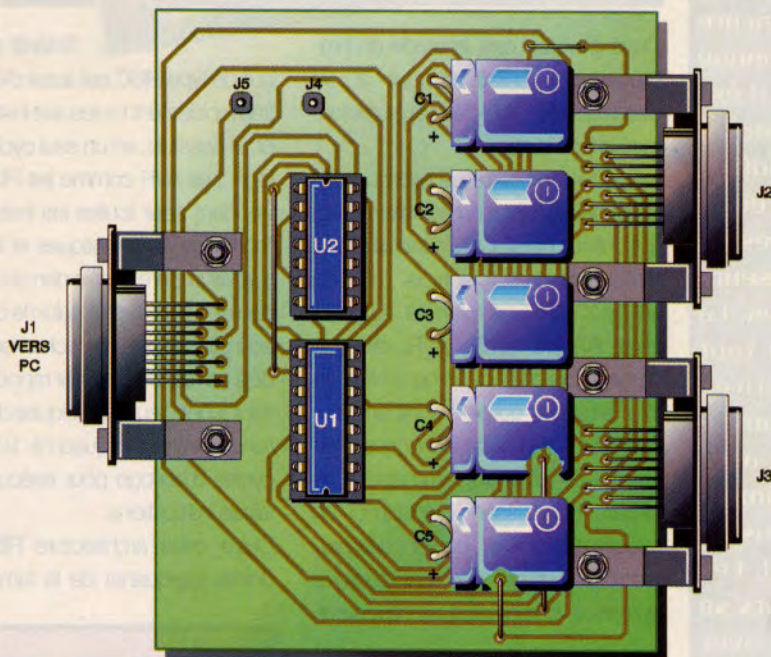
## Conclusion

Avec peu de composants, il est facile de réaliser un analyseur de protocole dont le résultat s'affiche sur l'écran de l'ordinateur. Le programme suppose un taux de transferts des données de 9600 bauds. Vous devez être sûr que le taux de transfert de l'ordinateur est identique à celui de la ligne à espionner. De plus, si la ligne à espionner comprend un nombre important de caractères, l'utilisateur peut envoyer le résultat dans un fichier au lieu de l'écran de l'ordinateur, fichier qui est ensuite relu par un éditeur.

M. LAURY



**4** Tracé du circuit imprimé



**5** Implantation des éléments

## Nomenclature

**U<sub>1</sub>** : MAX232 + support 16 broches

**U<sub>2</sub>** : 74HC00 + support 14 broches

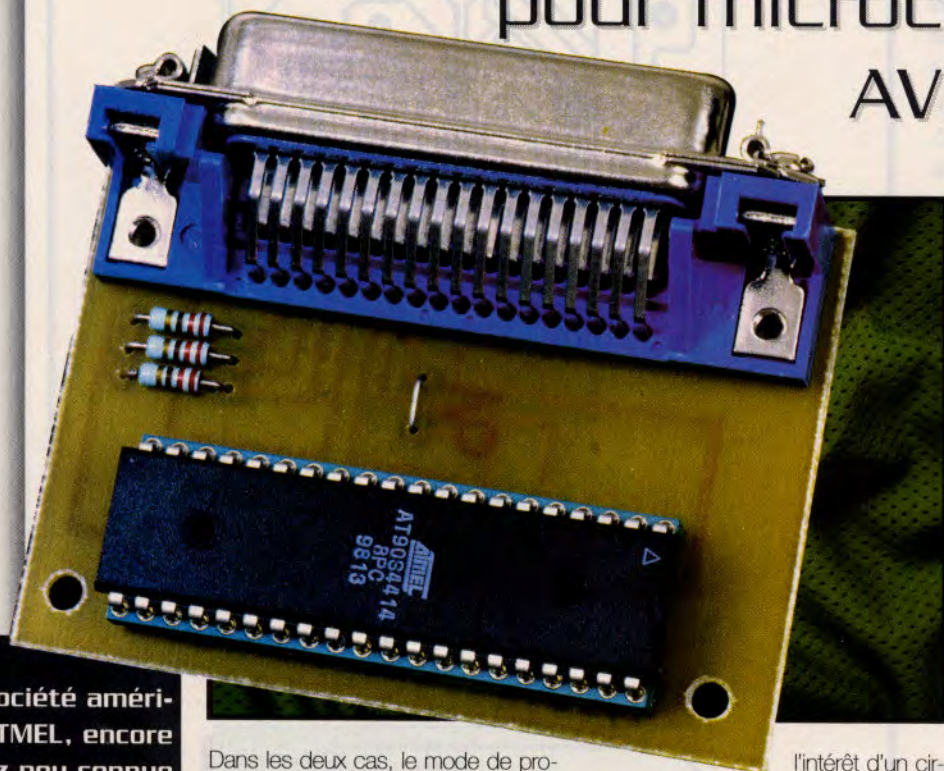
**J<sub>1</sub>, J<sub>3</sub>** : connecteurs DB9 femelle pour circuit imprimé

**J<sub>2</sub>** : connecteur DB9 mâle pour circuit imprimé

**J<sub>4</sub>, J<sub>5</sub>** : connecteurs 1 point

**C<sub>1</sub> à C<sub>5</sub>** : 1  $\mu$ F/16V

# Programmateurs pour microcontrôleurs AVR d'ATMEL



La société américaine ATMEL, encore assez peu connue chez nous il est vrai, fabrique, outre de nombreux autres semi-conducteurs, deux familles de microprocesseurs très différentes. La première est tout simplement dérivée de la célèbre famille 8051 d'INTEL avec des produits compatibles, au moins au plan logiciel. Ces microcontrôleurs se programment avec des programmeurs classiques, qu'ils soient du commerce ou de réalisation personnelle tel par exemple celui proposé dans le numéro de juillet/août 1999 d'E.P

Dans les deux cas, le mode de programmation utilisé conduit à un schéma assez complexe et à un montage assez lourd.

La deuxième famille de microcontrôleurs, beaucoup plus intéressante, est à architecture RISC, à la manière des désormais très célèbres PIC de MICROCHIP. Cette famille, appelée chez ATMEL famille AVR, dispose d'une possibilité de programmation dite "en circuit" qui permet de simplifier énormément le matériel nécessaire à la réalisation d'un programmeur. Ces circuits connaissant aujourd'hui un succès grandissant, ce sont des programmeurs qui leur sont destinés que nous vous proposons de réaliser.

## La famille AVR d'ATMEL

Ces circuits, qui sont nés à peu près à la même époque que les premiers PIC de MICROCHIP, adoptent une approche similaire. Ce sont en effet des microcontrôleurs à architecture RISC ce qui, contrairement à ce que l'on lit trop souvent, ne veut pas seulement dire circuits à jeu d'instructions réduit, même si c'est la signification de l'abréviation de cet acronyme. En effet,

l'intérêt d'un circuit de type RISC est aussi d'exécuter très rapidement toutes ses instructions et, si possible, en un seul cycle d'horloge. Les AVR comme les PIC y parviennent pour toutes les instructions "normales" (arithmétiques et logiques si vous préférez) mais demandent par contre 1 ou 2 cycles selon le cas pour les instructions de branchement. C'est déjà un réel progrès par rapport à des microcontrôleurs classiques dont certains demandent jusqu'à 10 ou 12 cycles d'horloge pour exécuter certaines instructions.

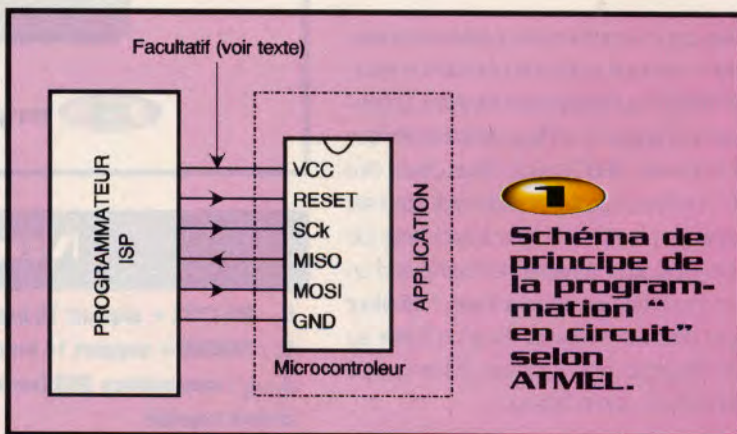
Outre cette architecture RISC, les points marquants de la famille AVR

peuvent être résumés de la façon suivante :

- la mémoire de programme interne est de type flash c'est à dire qu'elle est programmable et effaçable électriquement avec une endurance garantie d'au moins 1000 cycles. Ces circuits sont donc très bien adaptés à de la production en petite série et à de la réalisation d'amateur d'autant que les programmeurs nécessaires sont très simples et d'un prix de revient dérisoire comme vous allez pouvoir le constater dans un instant.

- Tous les circuits contiennent de l'EEPROM de données, de taille variable selon les boîtiers bien sûr. Cette mémoire se programme et s'efface, elle aussi, électriquement, avec une endurance garantie d'au moins 100 000 cycles. Elle peut être utilisée pour stocker les données non volatiles d'une application : chaînes dans un récepteur TV ou magnétoscope, points de consigne d'un thermostat, heures de changement d'état d'un programmeur domestique, etc.

- Ces deux mémoires peuvent être programmés en mode ISP (In System Programming) ce qui veut dire programmation "en circuit". Cela signifie qu'il ne faut qu'un minimum de liaisons pour programmer ces circuits et que cette programmation peut se faire



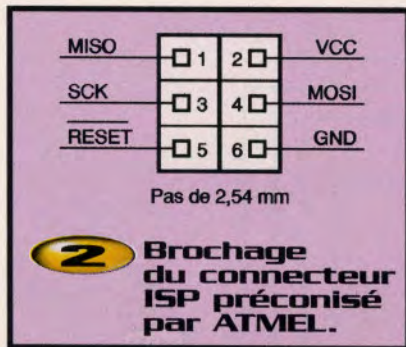
sans même enlever le microcontrôleur de l'application dans laquelle il se trouve monté. Nous exploiterons cela pour réaliser très simplement notre programmeur à partir du port parallèle de tout compatible PC.

- Tous les circuits contiennent un timer "chien de garde" avec son propre oscillateur d'horloge, ce qui contribue à sécuriser fortement le logiciel.

- Tous les circuits peuvent fonctionner en mode oscillateur à quartz ; il leur faut alors un quartz externe ; ou en mode oscillateur R-C dans lequel aucun composant externe n'est alors nécessaire.

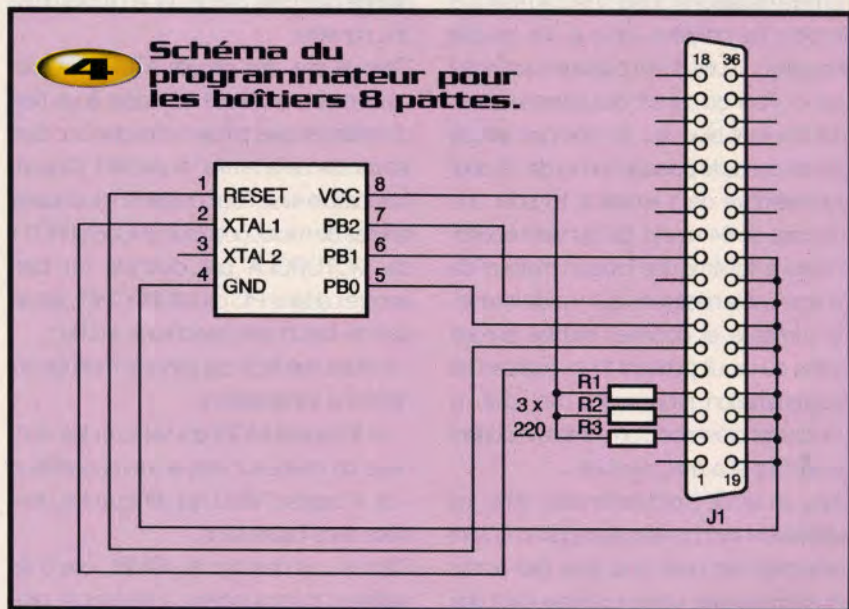
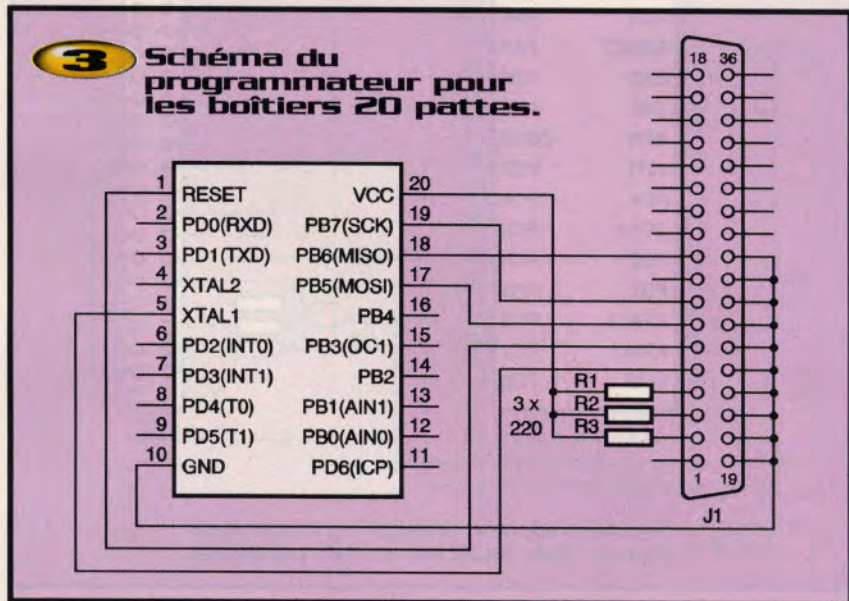
- Ils fonctionnent sous 2,7 à 6V, ou 4 à 6V selon les modèles, et sont réalisés en technologie HCMOS à très faible consommation. Un mode de repos permet même d'abaisser la consommation à moins de 1 µA tout en laissant le circuit réactif à certains stimuli.

- Les vitesses d'horloge s'étagent de 0 à 12 MHz selon les modèles et leurs différentes variantes, et les prix restent très abordables puisqu'ils oscillent entre 30 et 100 Francs environ à l'unité selon la référence choisie. Le dernier point à signaler, que nous avons volontairement ressorti à part de cette énumération, est l'existence d'outils de développement (assembleur mais aussi un excellent simulateur) totalement gratuits et libres de droits. C'est une véritable aubaine pour les amateurs que nous sommes et nous verrons dans un instant comment les installer sur votre PC car, bien évidemment, vous les trouverez sur le CD ROM ci-joint. Ceci étant précisé, la famille AVR d'ATMEL comporte aujourd'hui 6 membres différents, proposés chacun en plusieurs versions, pour ce qui est de la fréquence d'horloge et du type de boîtier. Le **tableau** ci-après vous permettra d'avoir une vue d'ensemble de cette famille dont le circuit le plus connu est le premier né, à savoir l'AT90S1200. Comme à l'accoutumée, vous pourrez obtenir plus d'informations et télécharger les fiches techniques de tous ces circuits sur le site Internet d'ATMEL dont l'adresse est [www.atmel.com](http://www.atmel.com).



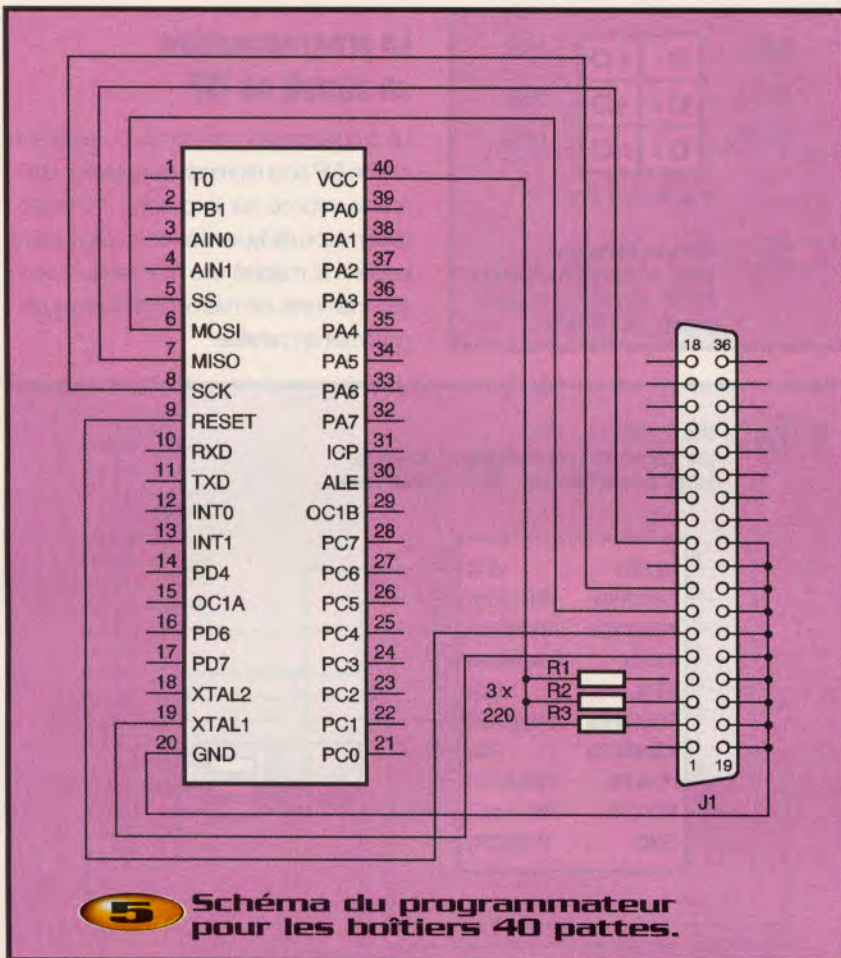
## La programmation en circuit ou ISP

La programmation en circuit, ou programmation ISP pour reprendre l'appellation américaine, repose sur le principe d'une programmation de type série alors que, jusqu'à présent, la majorité des mémoires et donc des microcontrôleurs se programmaient en parallèle.



Référence	Nb de pattes	Vitesse max.	Mémoire de programme	EEPROM interne	RAM
AT90S1200	20	4 et 12 MHz	1 K	64 octets	Néant
AT90S2313	20	4 et 10 MHz	2 K	128 octets	128 octets
AT90S2323	8	4 et 10 MHz	2 K	128 octets	128 octets
AT90S2343	8	4 et 10 MHz	2 K	128 octets	128 octets
AT90S4414	40	4 et 8 MHz	4 K	256 octets	256 octets
AT90S8515	40	4 et 8 MHz	8 K	512 octets	512 octets

Dans une mémoire ou un circuit à programmation parallèle, les choses se passent de la façon suivante. Après avoir mis le circuit en mode programmation, on applique l'adresse à programmer et la donnée correspondante sous forme parallèle, puis on envoie l'impulsion de programmation proprement dite. Cela nécessite un grand



nombre de liaisons puisqu'il faut au moins 8 fils pour les données lorsque l'on travaille avec des circuits 8 bits comme c'est notre cas ici. Pour ce qui est des adresses, il en faut d'autant plus que la mémoire est de grande capacité puisque l'on va de 10 pour une mémoire de 1 kmots à 16 pour une mémoire de 64 kmots. Ce qui vient alourdir encore le schéma des programmeurs de ce type est la nécessité qu'il y a de maintenir adresses et données stables sur les pattes du circuit pendant toute la phase de programmation proprement dite, d'où la nécessaire présence d'un nombre important de latches et autres bascules.

Dans un circuit à programmation série, les adresses et les données à programmer sont transmises les unes à la suite des autres sous forme série, un peu comme dans une banale liaison série classique. Le circuit s'occupe ensuite seul et en interne de leur gestion pour peu qu'on lui ait envoyé, au préalable et par le même chemin, les ordres adéquats. En pratique, et comme il faut tout de même pouvoir relire et contrôler le circuit que l'on programme, deux ou trois lignes supplémentaires sont nécessaires mais, dans tous les cas, la simplicité est bien plus

grande que celle offerte par la programmation parallèle.

Dans le cas des circuits ATMEL, la programmation en circuit fait appel à un port d'entrées/sorties présent d'origine sur tous les circuits de la famille : le port SPI. Ce port, compatible avec ceux présents sur d'autres familles de microcontrôleurs (HC05 et HC11 de MOTOROLA par exemple ou bien encore certains PIC de MICROCHIP), est un port de liaison série synchrone à 3 fils :

- un fil baptisé SCK qui véhicule l'horloge qui rythme la transmission,
- un fil baptisé MOSI qui véhicule les données de l'extérieur vers le microcontrôleur,
- un fil baptisé MISO qui véhicule les données dans l'autre sens.

Dans le cas des circuits ATMEL, ces 3 fils suffisent théoriquement à réaliser la programmation en circuit mais, en pratique, il est indispensable de disposer aussi d'une liaison de masse et il est souhaitable de pouvoir commander aussi la patte Reset du microcontrôleur. Ceci porte à 5 le nombre de fils nécessaires comme le montre le schéma synoptique de la **figure 1**.

Afin de prévoir toutes les situations possibles, il est d'usage de faire en sorte que le

programmeur génère aussi la tension d'alimentation nécessaire au circuit à programmer pour le cas où l'application sur laquelle il est installé ne fonctionnerait pas pendant la programmation. On arrive donc à une liaison à 6 fils, pour laquelle ATMEL préconise le connecteur dont le brochage est présenté

### figure 2.

Nous n'allons pas détailler ici le mode de fonctionnement de la programmation en circuit car cela occuperait la quasi-totalité de ce numéro. Si vous souhaitez en savoir plus, nous vous conseillons donc la lecture de la fiche technique AVR910, disponible en téléchargement sur le site Internet d'ATMEL ([www.atmel.com](http://www.atmel.com)), qui explique par le menu le principe de cette programmation, les différentes commandes disponibles, leur mode de codage et les comportements des circuits dans diverses situations.

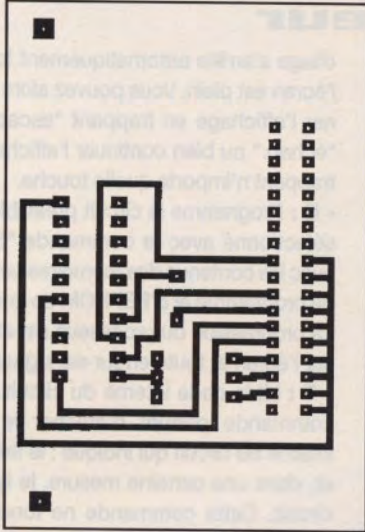
## La programmation en circuit à partir d'un PC

Contrairement à ce que l'on pourrait penser à première vue, c'est à partir du port parallèle d'un PC, et non du port série, que la programmation d'un circuit ATMEL est la plus facile. En effet les signaux disponibles sur le port série ne sont pas aux bons niveaux électriques (RS232 alors qu'il nous faut des niveaux TTL) et les chronogrammes ne sont pas non plus corrects. La liaison série d'un PC est en effet du type asynchrone alors que nous travaillons ici en mode série synchrone. Notre programmeur exploite donc le port parallèle qui, au moyen d'un logiciel adéquat fourni sur le CD ROM ci-joint bien sûr, est capable de générer les chronogrammes nécessaires.

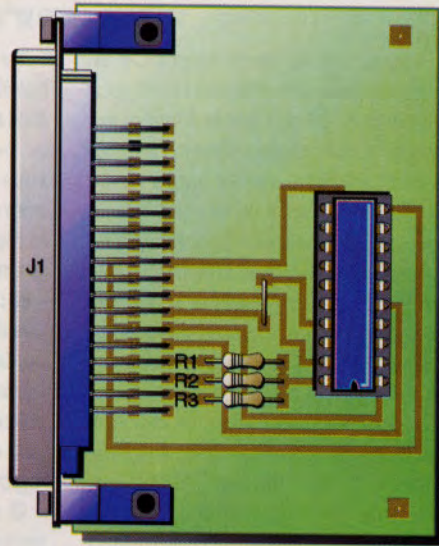
A l'heure actuelle, les circuits ATMEL les plus utilisés sont disponibles en boîtiers DIL 8 pattes, 20 pattes et 40 pattes. Nous avons donc prévu notre, ou plutôt nos programmeurs pour ces trois situations. Nous verrons aussi, bien sûr, comment réaliser le cordon de programmation en circuit pour ceux d'entre-vous qui souhaiteront faire appel à cette technique.

## Schémas de nos programmeurs

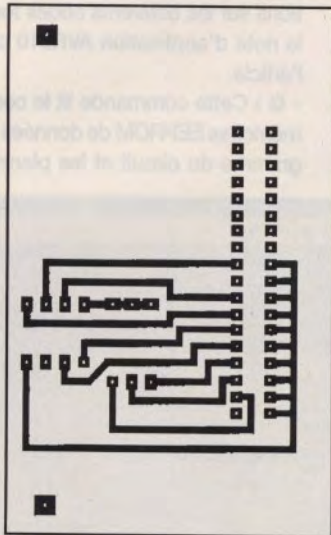
Le "s" de schémas et de programmeurs n'est pas du à une coquille, même s'il n'existe en fait qu'un schéma théorique pour programmer les microcontrôleurs ATMEL.



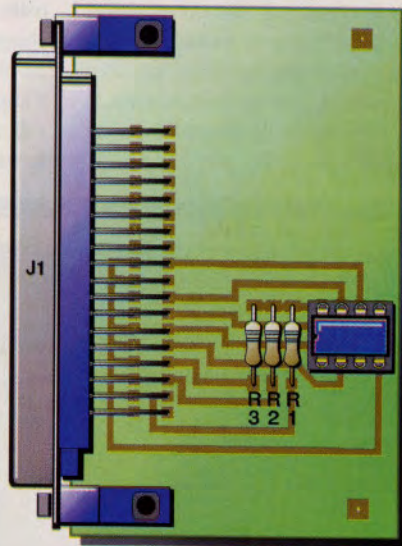
**6**  
Circuit imprimé du programmeur pour les boîtiers 20 pattes



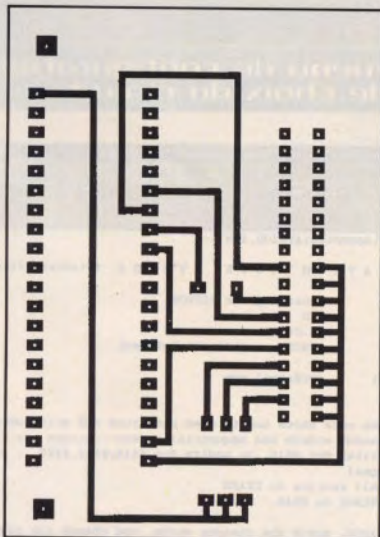
**9**  
Implantation des composants sur le programmeur pour les boîtiers 20 pattes.



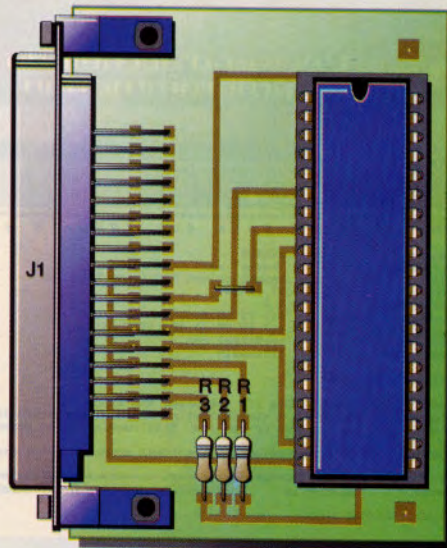
**7**  
Circuit imprimé du programmeur pour les boîtiers 8 pattes



**10**  
Implantation des composants sur le programmeur pour les boîtiers 8 pattes.



**8**  
Circuit imprimé du programmeur pour les boîtiers 40 pattes



**11**  
Implantation des composants sur le programmeur pour les boîtiers 40 pattes.

**Nomenclature pour un module**

J<sub>1</sub> : connecteur Centronics femelle coudé à 90° pour circuit imprimé

R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub> : 220 Ω 1/4W 5% (rouge, rouge, marron)

- Supports : 8, 20 ou 40 pattes selon le module



## Mode d'emploi du logiciel du programmeur

Le programme se lance sous DOS en frappant son nom (FBPRG16) et sous Windows 95 ou 98 en double cliquant sur le nom de fichier correspondant. Dans les deux cas, la fenêtre visible figure 12 (ou son seul contenu sous DOS) s'affiche.

La barre en vidéo inversée peut être déplacée avec les touches fléchées du curseur mais il est également possible d'accéder immédiatement à une commande en frappant sa lettre code rappelée entre parenthèse au début de chaque ligne.

Lorsqu'une commande propose plusieurs options (cas de la commande Setup par exemple), il faut frapper "enter" ou "entrée" pour les faire défiler. De même, pour valider une commande ou un choix il faut également frapper "entrée".

Bien que le descriptif des commandes affichées à l'écran soit assez explicite voici, pour ceux d'entre-vous qui sont réfractaires à la langue de Shakespeare, le rôle et les fonctions de chacune d'elles classées par lettre code.

- **A** : Charge un fichier au format normalisé INTEL HEX (produit par l'assembleur AVR notamment) dans la mémoire tampon ou buffer de programme. La commande vous demande alors, en bas d'écran, le nom du fichier concerné qui doit être dans le répertoire contenant le logiciel de programmation.

- **B** : Charge un fichier au format normalisé INTEL HEX dans la mémoire tampon ou buffer de l'EEPROM de données. La commande vous demande alors, en bas d'écran, le nom du fichier concerné qui doit être dans le répertoire contenant le logiciel de programmation.

- **C** : Affiche le contenu de la mémoire tampon de programme. L'affichage s'arrête automatiquement lorsque l'écran est plein. Vous pouvez alors terminer l'affichage en frappant "escape" ou "échap." ou bien continuer l'affichage en frappant n'importe quelle touche.

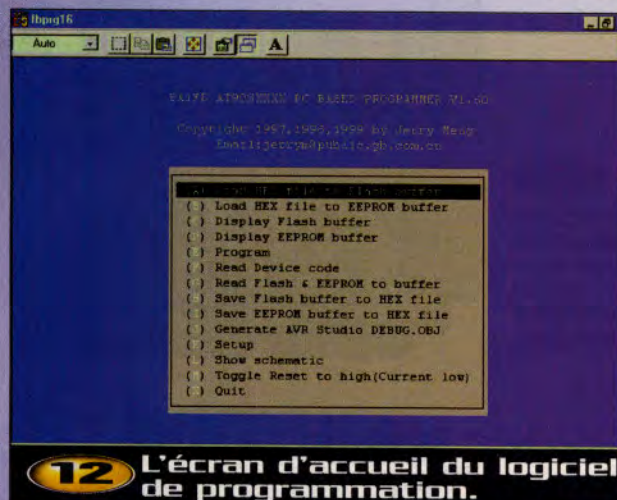
- **D** : Affiche le contenu de la mémoire tampon de l'EEPROM de données. L'affi-

chage s'arrête automatiquement lorsque l'écran est plein. Vous pouvez alors terminer l'affichage en frappant "escape" ou "échap." ou bien continuer l'affichage en frappant n'importe quelle touche.

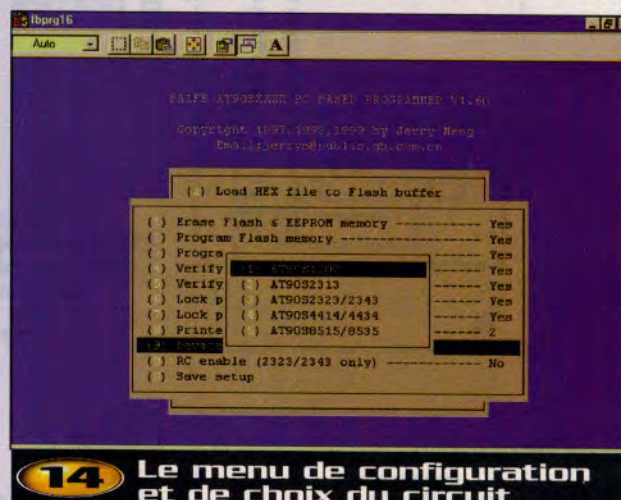
- **E** : Programme le circuit préalablement sélectionné avec la commande "Setup" avec les contenus des mémoires tampons de programme et d'EEPROM de données. La progression du processus est affichée sur l'écran et toute erreur est signalée.

- **F** : Lit le code interne du circuit. Cette commande permet d'afficher le code interne du circuit qui indique : le fabricant et, dans une certaine mesure, le type de circuit. Cette commande ne fonctionne pas si les bits de sécurité ont été programmés. Pour de plus amples informations sur les différents codes fournis, voir la note d'application AVR910 citée dans l'article.

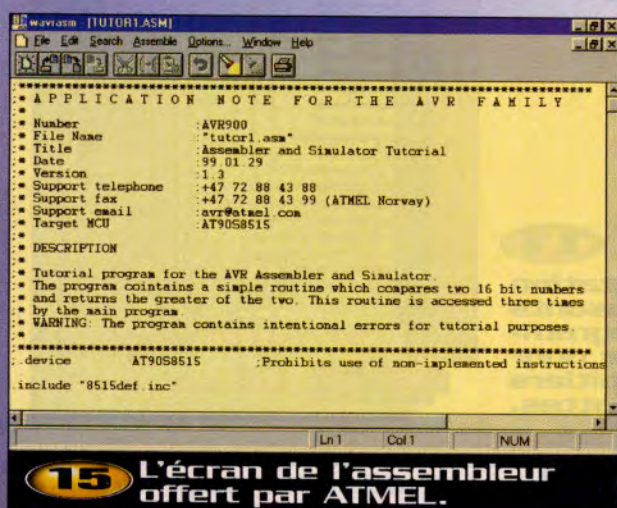
- **G** : Cette commande lit le contenu des mémoires EEPROM de données et de programme du circuit et les place dans les



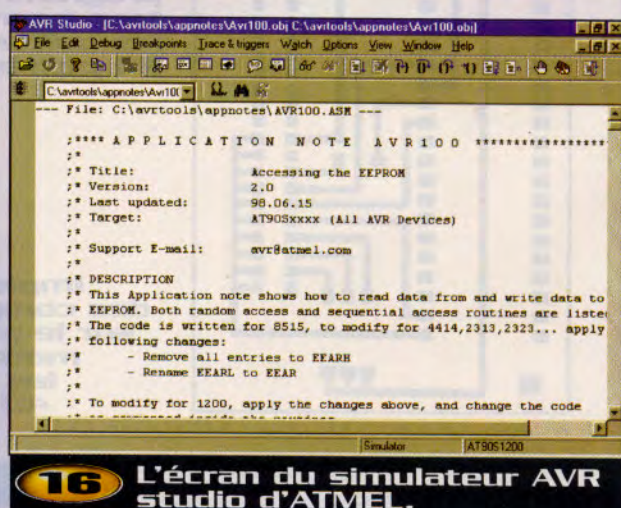
**12** L'écran d'accueil du logiciel de programmation.



**14** Le menu de configuration et de choix du circuit.



**15** L'écran de l'assembleur offert par ATMEL.



**16** L'écran du simulateur AVR studio d'ATMEL.

mémoires tampons correspondantes. Cette commande ne fonctionne pas si les bits de sécurité ont été programmés. On retrouve alors dans les mémoires tampons, à chaque adresse, une donnée égale à cette dernière (00 à l'adresse 00, 01 à l'adresse 01 et ainsi de suite).

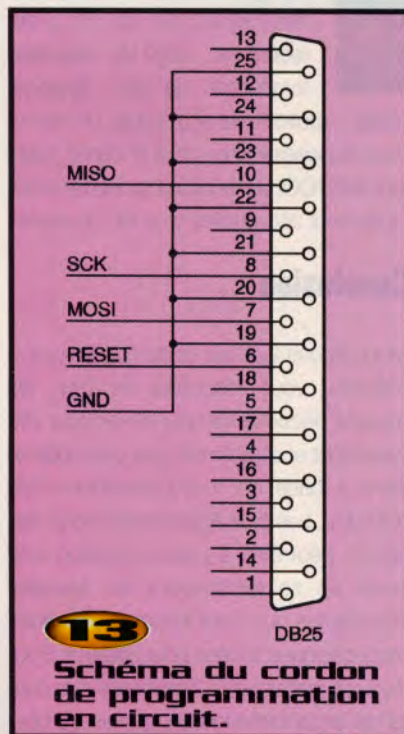
- **H** : Sauvegarde le contenu de la mémoire tampon de programme dans le fichier de votre choix qui sera au format normalisé INTEL HEX. La commande vous demande, en bas d'écran, le nom du fichier concerné qui sera placé dans le répertoire contenant le logiciel de programmation.

- **I** : Sauvegarde le contenu de la mémoire tampon d'EEPROM de données dans le fichier de votre choix qui sera au format normalisé INTEL HEX. La commande vous demande, en bas d'écran, le nom du fichier concerné qui sera placé dans le répertoire contenant le logiciel de programmation.

- **J** : Génère avec le contenu de la mémoire tampon de programme un fichier compatible du logiciel de simulation AVR Studio.

- **K** : Cette commande donne accès à une nouvelle fenêtre visible figure 14 qui permet un certain nombre de sélections décrites à la fin de cette notice. La sortie de cette commande se fait en validant le S pour "Save setup".

- **L** : Cette commande ne vous servira à rien car elle permet juste de faire afficher à l'écran un schéma synthétique du programmeur.



- **M** : Cette commande permet de changer alternativement l'état de la ligne RESET du circuit à programmer. Elle indique aussi l'état courant de cette ligne. Lorsqu'elle est à l'état haut (high) le circuit est en mode normal : lorsqu'elle est à l'état bas (low) le circuit est en mode programmation.

- **X** : Permet de quitter le logiciel et de revenir au DOS ou à Windows 95 ou 98 selon l'origine de l'appel du programme.

La commande K ou "Setup" donne accès à un certain nombre d'options repérées par des chiffres ; options que nous allons maintenant décrire rapidement.

- **1** : Permet d'effacer ou non les mémoires de programme et EEPROM de données avant programmation.

- **2** : Permet à la commande E de programmer ou non la mémoire de programme.

- **3** : Permet à la commande E de programmer ou non la mémoire EEPROM de données.

- **4** : Permet de vérifier ou non la mémoire de programme.

- **5** : Permet de vérifier ou non la mémoire EEPROM de données.

- **6** : Permet de verrouiller ou non le 1er bit de protection.

- **7** : Permet de verrouiller ou non le 2eme bit de protection.

- **8** : Permet de choisir le port parallèle utilisé par le programmeur (1 ou 2).

- **9** : Permet de choisir le type de circuit à programmer au moyen de la fenêtre qui s'ouvre alors. Il suffit d'amener la barre en vidéo inversée sur le circuit choisi ou de taper directement le numéro placé en face de sa référence.

- **0** : Permet de valider ou non l'oscillateur RC interne sur les circuits 2323 et 2343 seulement.

- **S** : Permet de sortir de la commande tout en validant et mémorisant les choix préalablement faits. Ces choix seront utilisés par défaut lors du prochain lancement du logiciel.

On le voit, même si son écran est un peu triste, ce logiciel dispose de toutes les fonctions souhaitables, d'autant que son rapport qualité/prix est imbattable. Nous l'avons longuement testé et n'avons pas trouvé de bug mais, si tel était votre cas, n'hésitez pas à en faire part à l'auteur du logiciel par e-mail (son adresse figure sur l'écran) ; il le corrigera au plus vite et vous en sera reconnaissant.

En effet, les trois tailles de boîtiers DIL que nous avons évoquées ont un brochage totalement incompatible et, plutôt que de dessiner un seul circuit imprimé complexe et comportant de nombreux straps, nous avons donc préféré réaliser trois circuits imprimés distincts. Ce choix a été rendu possible car le surcoût qui en résulte est dérisoire ; en effet, nos programmeurs ne comportent chacun qu'un connecteur, un support de circuit intégré et trois résistances ! Si vous arrivez à faire plus simple, faites-nous le savoir...

La **figure 3** présente donc le schéma du programmeur pour boîtiers 20 pattes c'est à dire pour les circuits : AT90S1200 et AT90S2313.

La **figure 4** présente le schéma du programmeur pour les boîtiers 8 pattes c'est à dire pour les circuits : AT90S2323 et AT90S2343.

La **figure 5** présente le schéma du programmeur pour les boîtiers 40 pattes c'est à dire pour les circuits : AT90S4414 et AT90S8515.

Comme vous pouvez le constater, ces trois schémas sont évidemment identiques, aux brochages des supports près.

Les lignes SCK, MISO, MOSI et RESET sont pilotées directement par des lignes de données du port parallèle du PC. La masse est évidemment reliée à la masse de la liaison parallèle tandis que l'alimentation nécessaire au circuit à programmer est prélevée, elle aussi, à partir de trois lignes de données du port parallèle via les résistances de 220  $\Omega$ . Il suffit ainsi de mettre ces trois lignes au niveau haut ou au niveau bas au moyen du logiciel de programmation pour commander l'alimentation du circuit à programmer.

Cette façon de faire n'est évidemment possible qu'en raison de la très faible consommation des circuits AVR en phase de programmation. Remarquez qu'une ligne supplémentaire est utilisée par rapport à ce que nous avons expliqué ci-dessus, puisque nous commandons aussi XTAL1 à partir du port parallèle. Ceci a été rendu nécessaire par le fait que le bit interne sélectionnant le fonctionnement avec un oscillateur RC ou une horloge à quartz de certains circuits n'est pas accessible en mode programmation série. Si donc le circuit que vous voulez programmer est configuré en mode horloge à quartz, il faut impérativement lui fournir cette horloge pour pouvoir le programmer et c'est le rôle dévolu à la liaison XTAL1.

## Réalisation

Compte tenu de nos trois schémas, nous avons évidemment dessiné trois circuits imprimés dont les tracés sont visibles **figures 6, 7 et 8** et les plans d'implantation correspondants **figures 9, 10 et 11**. Ces circuits sont munis de connecteurs Centronics femelles coudés à 90° permettant ainsi leur branchement immédiat à l'extrémité du câble qui aboutit normalement à votre imprimante. Pour ce qui est des supports destinés à recevoir les circuits à programmer, vous pourrez utiliser des supports à contacts tulipes si vous faites un usage modéré du programmeur. Dans le cas contraire, il vaudra mieux faire appel à des supports à force d'insertion nulle qui seront alors hélas les composants les plus coûteux de cette réalisation.

## Le logiciel de programmation

Avec un schéma aussi simple, il est bien évident que l'essentiel du travail est confié au logiciel de commande du programmeur ; logiciel que vous trouverez sur le CD ROM ci-joint. Celui-ci a été développé par Mr Jerry MENG, radioamateur chinois, qui le propose gracieusement en tant que freeware sur Internet. Vous pourrez donc l'utiliser sans avoir à payer les moindres droits, le copier et le redistribuer si nécessaire mais en prenant la précaution de toujours mentionner son origine et, bien sûr, sans demander la moindre rémunération en échange.

Ce logiciel est constitué du seul fichier baptisé FBPRG16.EXE que vous copierez dans le répertoire de votre choix qui deviendra alors son répertoire de travail. S'agissant d'un logiciel fonctionnant sous DOS, il est utilisable directement sous DOS ou dans une fenêtre DOS de Windows 95 ou 98 sans aucun problème.

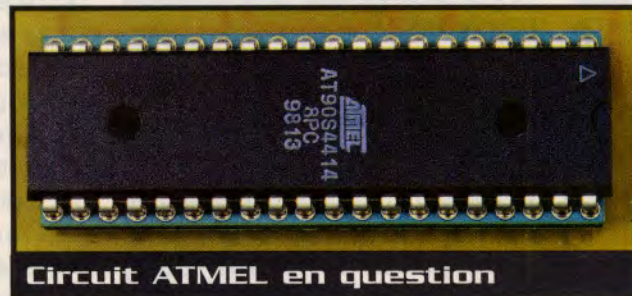
Pour le lancer, il suffit de taper son nom ou de double cliquer sur celui-ci si vous travaillez sous Windows 95 ou 98. Un écran analogue à celui visible figure 12 s'ouvre alors pour vous donner accès aux commandes du logiciel. Vous pouvez raccorder votre programmeur à tout instant au port parallèle de votre PC ; port parallèle qui peut être LPT1 ou LPT2 car le logiciel sait gérer l'un et l'autre. Par contre, veillez à ne pas pla-

cer de microcontrôleur sur le support tant que le logiciel n'a pas été lancé. En effet, dans le cas contraire, les lignes du port parallèle peuvent être dans n'importe quel état, dépendant de ce qui a été fait sur le PC au préalable, et le circuit pourrait donc se trouver alimenté lors de son insertion dans le support ce qui pourrait l'endommager irrémédiablement.

L'encadré ci-après présente la notice d'emploi du logiciel à laquelle vous voudrez bien vous reporter, au moins pour la première utilisation. Vous constaterez en effet très vite que les fonctions proposées sont classiques et que l'écran, bien que très sobre, affiche toutes les informations utiles.

## Le câble de programmation en circuit

Nous vous l'avions promis en introduction, la **figure 13** vient tenir notre promesse en vous en présentant le schéma qui n'a d'ailleurs plus rien d'original à ce stade de l'article ! Vous le réaliserez directement avec un connecteur DB25 mâle que vous connecterez au port parallèle de votre PC. Côté appli-



cation, vous le munirez évidemment d'un connecteur adapté à celui se trouvant sur cette dernière ou conforme aux préconisations d'ATMEL que nous avons rappelées en figure 2. Le logiciel utilisé est le même que celui qui pilote les programmeurs complets. Notez toutefois que le microcontrôleur de votre application devra être alimenté par cette dernière car notre câble ne prélève pas cette alimentation sur le PC.

## Les outils de développement

Comme nous vous l'avons indiqué en début d'article, ils sont mis gracieusement à notre disposition par ATMEL et il suffit pour cela de les télécharger sur le site Internet de ce fabricant. Pour vous faciliter les choses et, pour ceux d'entre-vous qui ne sont pas encore connectés au net, vous les trouverez aussi

sur le CD ROM ci-joint dans la dernière version présente sur le site d'ATMEL au 10 août 1999.

Deux fichiers auto-exécutables vous sont proposés :

- ASMPACK.EXE qui permet d'installer l'assembleur AT90S dans sa version 1.3 sur tout système fonctionnant sous Windows 95, 98 ou NT.

- ASTUDIO2.EXE qui permet d'installer le "studio" de développement ATMEL, capable de piloter les émulateurs en circuit ICE, mais qui dispose aussi d'un excellent simulateur vous permettant de faire de la mise au point de programmes avec un maximum de confort. Comme le précédent, ce logiciel fonctionne sous Windows 95, 98 ou NT.

Pour les installer, procédez un par un. Copiez le premier fichier dans un répertoire temporaire vide de votre choix et double-cliquez sur le nom du fichier pour en lancer l'exécution. Il se décompacte alors en une multitude de fichiers placés dans ce même répertoire temporaire. Lorsque c'est terminé, double-cliquez sur le fichier SETUP.EXE pour lancer l'exécution du logiciel proprement dit. Vous vous trouverez alors devant une suc-

cession de fenêtres "à la mode Windows" qui vous permettront d'installer le logiciel dans le répertoire de votre choix (conservez celui proposé par défaut si vous n'avez pas de raison valable d'en changer). Lorsque l'installation est terminée et que votre PC a été redémarré, videz le répertoire temporaire de son contenu,

copiez-y l'autre fichier et procédez de même. Vous disposerez alors, dans le même répertoire AVRTOOLS des deux logiciels de développement accessibles d'un clic de souris.

## Conclusion

Muni de ces logiciels et de nos programmeurs, vous disposez de tous les moyens nécessaires pour développer efficacement et à moindre coût des applications à base de microcontrôleurs AVR d'ATMEL. Les plus âgés d'entre-nous verseront peut-être à cette occasion une larme en se remémorant les sommes importantes qu'il fallait investir il y a seulement quelques années pour disposer d'outils aussi performants. Le progrès et la dure loi de la concurrence ont parfois du bon.

**C. TAVERNIER**

# Lecteur de cartes magnétiques pour PC



Bien que sévèrement concurrencées par les cartes à puce, les cartes à pistes magnétiques affichent encore une belle santé. Il n'est assurément pas sans intérêt de pouvoir lire leur contenu, pour s'attaquer peut-être même ensuite aux autres supports de données qui en dérivent, comme les billets de transport et autres tickets. Interfacer une simple tête de magnétophone au port de manettes de jeu d'un PC permet d'obtenir très vite des résultats plus que prometteurs...

## Un peu de normalisation

Massivement utilisées dans le monde entier (et pas seulement pour des applications bancaires !) les cartes magnétiques font l'objet d'une normalisation très précise et assez largement respectée.

La **figure 1** montre comment sont positionnées les trois pistes des cartes conformes à la norme ISO 7811, mais il faut savoir qu'on peut tout de même rencontrer des cartes "hors normes"...

En cas de doute, un examen visuel après application d'un produit "révélateur magnétique" pourra aider à déterminer précisément à quel type de cartes on a affaire.

La plus utilisée de toutes les pistes, et de loin, est la n°2 ou "ISO 2", dont le bord supérieur se situe à pratiquement 9 mm du haut de la carte. Encodée à raison de 75 bpi (bits par pouce), elle est relativement facile à lire si on la fait glisser le long d'une simple tête de magnétophone. Le positionnement relatif n'est pas trop critique car l'entrefer d'une tête monophonique pour cassette est large de 1,4 mm tandis que la largeur de la piste mesure exactement le double. Il importe toutefois que la tête soit en

bon contact avec la carte et bien "azimutée", c'est à dire que son entrefer soit parfaitement perpendiculaire à l'axe de la piste. Un quelconque système de guidage et de pressage est donc à prévoir, ce qui est automatiquement le cas si on arrive à se procurer une mécanique de lecteur de récupération (c'est un article assez courant dans les magasins de surplus). Pour peu que le lecteur soit complet, on disposera même de son électronique de lecture, accessible par au moins 5 fils (voir son schéma

de raccordement) :

- alimentation (souvent +5V),
- masse,
- horloge (CLOCK),
- données (DATA),
- présence carte (CARD PRESENT).

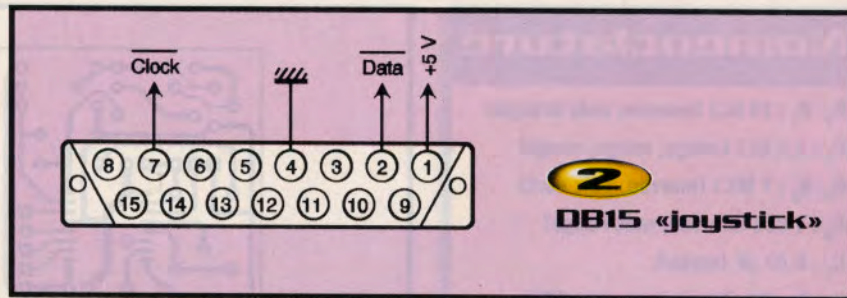
Des fils supplémentaires se rencontrent sur les lecteurs multipistes ou sur ceux qui sont prévus pour commander des voyants de signalisation externes. Les quatre premiers suffisent pour interfacer un tel lecteur avec le port "joystick" du premier PC venu, selon le schéma de la **figure 2**.



En effet, une alimentation 5V de bonne qualité est disponible ici, de même que 4 entrées "tout ou rien" dont 2 seulement seront exploitées par nos logiciels de décodage. Il faut savoir, en effet, que le train de bits disponible sur la ligne DATA au rythme de l'horloge CLOCK reproduit simplement les "1" et les "0" tels qu'ils se suivent sur la carte. Il reste encore à les interpréter pour reconstituer les caractères (utiles et "de service") qui ont été originellement encodés sur la piste, voire même pour opérer une détection d'erreur par bit de parité et clef de contrôle.

### L'électronique de lecture

Que l'on dispose d'une mécanique de récupération dépourvue de son électronique ou que l'on se soit construit un guide équipé d'une simple tête de magnétophone, le schéma de la **figure 3** résoudra aisément le problème du raccordement au PC. Il fait appel à un circuit intégré spécialement conçu pour cet usage, le U 4085 B. Fabriqué par TELEFUNKEN pour le compte du fabricant allemand de lecteurs Hopt+Schuler, il présente le gros avantage d'être disponible au détail chez SELECTRONIC.



Les composants qui l'entourent sont très précisément calculés pour obtenir, du premier coup, des performances irréprochables dans cette application beaucoup plus exigeante qu'on pourrait le penser de prime abord.

Le transistor, en particulier, doit impérativement être un BC 517 en raison du gain très important qui est nécessaire à ce niveau. Le circuit intégré n'étant fabriqué qu'en version CMS, il sera soudé côté cuivre du circuit imprimé de la **figure 4**, en veillant bien à son orientation.

A condition d'utiliser un bon fer à panne fine, l'opération ne présente aucune difficulté, à condition de commencer par souder seulement deux broches diagonalement opposées et de ne passer aux suivantes qu'après avoir vérifié que toutes tombent parfaitement en face de leurs pastilles respectives. Bien entendu, il convient d'em-

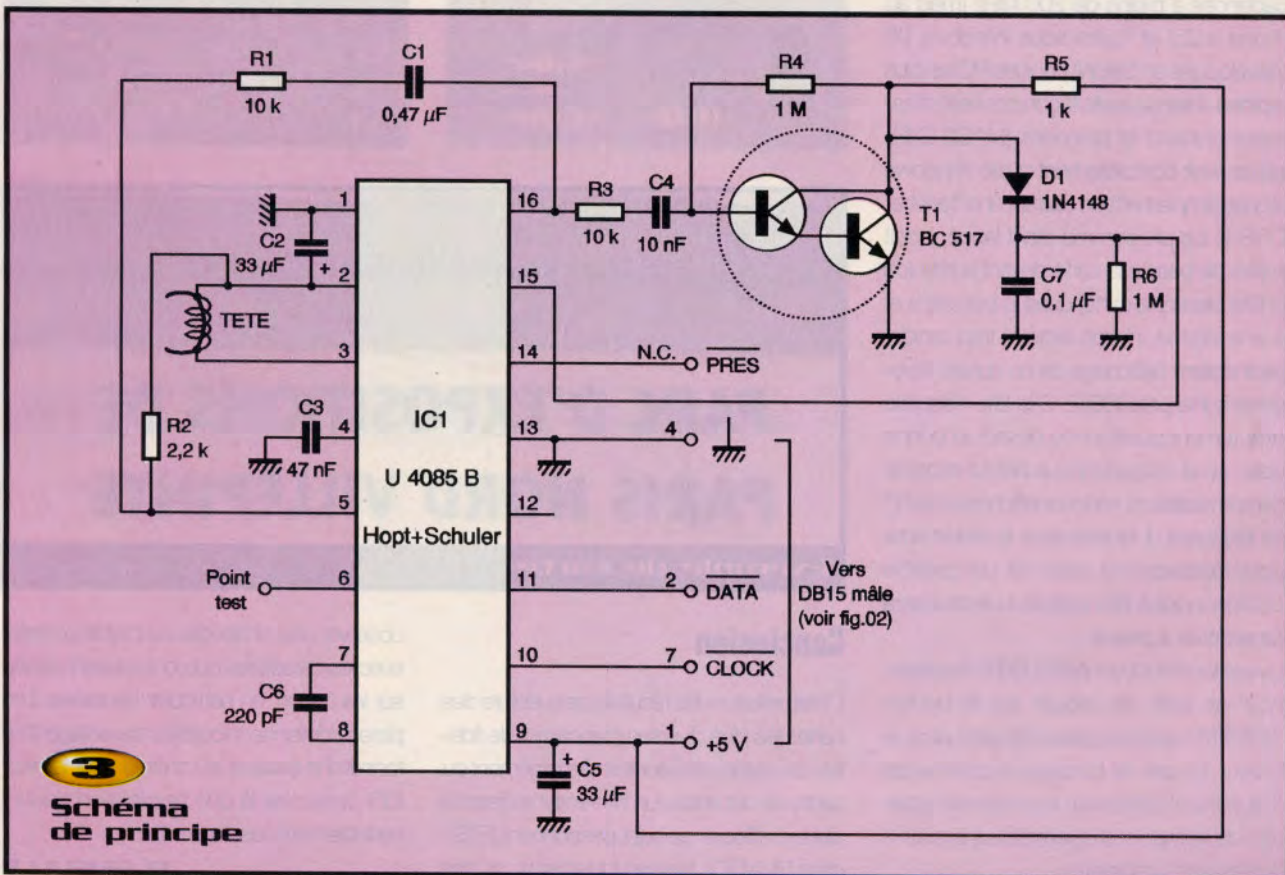
ployer du fil de soudure aussi fin que possible, de préférence 0,5 à 0,7 mm et à 60% d'étain.

On câblera ensuite le reste des composants, de façon tout à fait classique, selon le plan de la **figure 5**.

La tête sera raccordée par deux fins fils souples torsadés entre eux et d'une longueur n'excédant pas 10 cm. Dans ces conditions, aucun blindage n'est nécessaire, mais si la tête est métallique, on pourra avantageusement relier son corps à la masse par un troisième fil torsadé avec les deux premiers.

Le câble à quatre conducteurs rejoignant la fiche DB15 mâle pourra, de son côté, mesurer jusqu'à un mètre de long sans précaution particulière ni blindage.

Avant tout raccordement au PC, on vérifiera scrupuleusement le câblage, et surtout l'absence de court-circuit sur l'alimentation 5V.



## Nomenclature

$R_1, R_3$  : 10 k $\Omega$  (marron, noir, orange)

$R_2$  : 2,2 k $\Omega$  (rouge, rouge, rouge)

$R_4, R_6$  : 1 M $\Omega$  (marron, noir, vert)

$R_5$  : 1 k $\Omega$  (marron, noir, rouge)

$C_1$  : 0,47  $\mu$ F (mylar)

$C_2, C_5$  : 33  $\mu$ F tantale goutte 10V

$C_3$  : 47 nF (mylar)

$C_4$  : 10 nF (mylar)

$C_6$  : 220 pF (céramique)

$C_7$  : 0,1  $\mu$ F (mylar)

IC<sub>1</sub> : U 4085 B

T<sub>1</sub> : BC 517 (impératif)

D<sub>1</sub> : 1N 4148

Tête de lecture K7 monophonique

Fiche DB15 mâle avec capot

fil de câblage souple

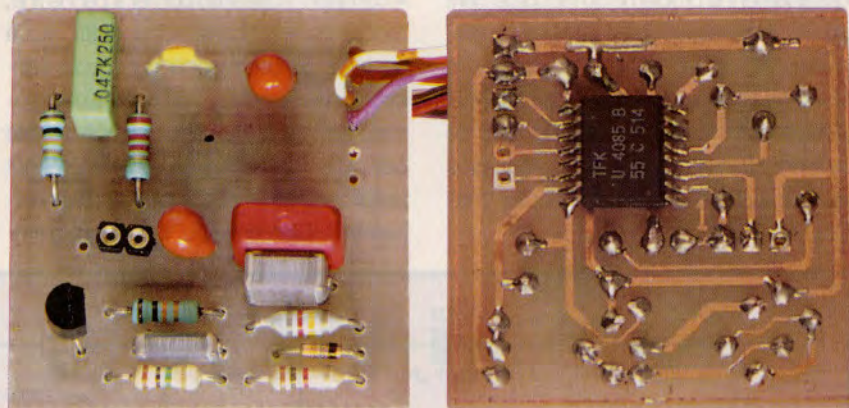
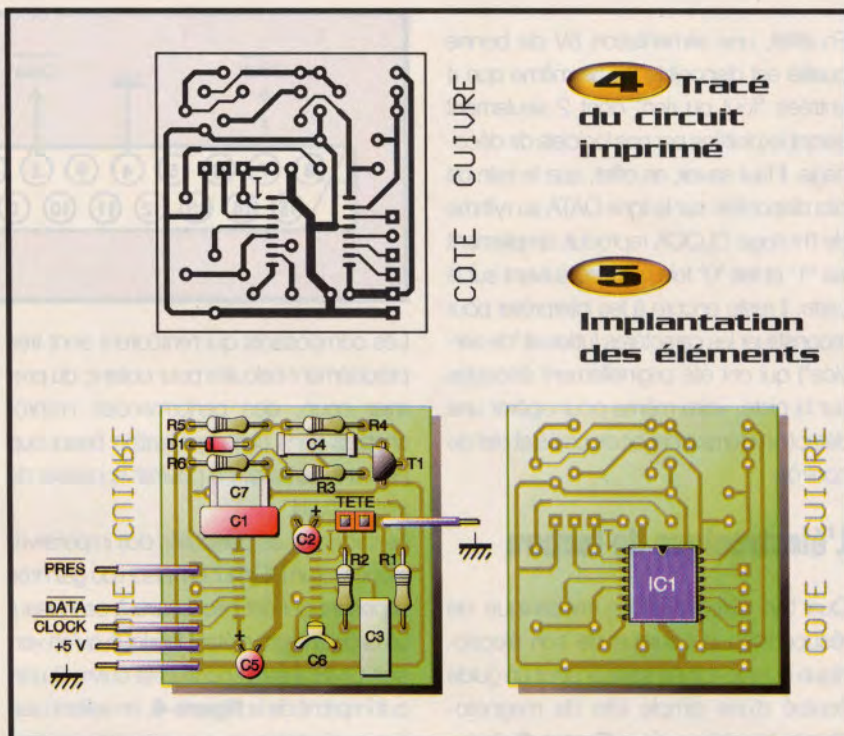
câble 4 conducteurs

Ce peut être une question de vie ou de mort pour la carte mère de l'ordinateur...

## Les logiciels

Deux versions du logiciel de décodage sont offertes : l'une sous DOS pour les PC cadencés à moins de 200 MHz (mais au moins à 25) et l'autre sous Windows 95 (développée en Delphi) pour les PC les plus rapides. Il est de toute façon conseillé d'essayer d'abord la première (MAG2.EXE), après avoir complètement quitté Windows (et pas simplement en ouvrant une "fenêtre" DOS !). Le programme étant lancé, il doit suffire de passer la carte devant la tête (ou la tête devant la carte) dans le bon sens et à une vitesse, ni trop lente ni trop rapide, pour obtenir l'affichage de ce qui est enregistré sur la piste ISO 2. Si la tête n'est pas exactement positionnée devant la bonne piste, ou si un quelconque défaut subsiste dans l'installation, rien ne s'affichera et le PC se bloquera. Il faudra alors le redémarrer pour reprendre "la main" et un contrôle oscilloscopique des sorties du lecteur sera sans doute à prévoir...

La version Windows (MAG.EXE) nécessite, pour sa part, de cliquer sur le bouton "DÉPART" avant de passer la carte dans le lecteur. En cas de blocage, la commande "Ctrl-Alt-Del" (effectuée une seule fois !) permet de fermer le programme sans avoir à redémarrer l'ordinateur.



## Conclusion

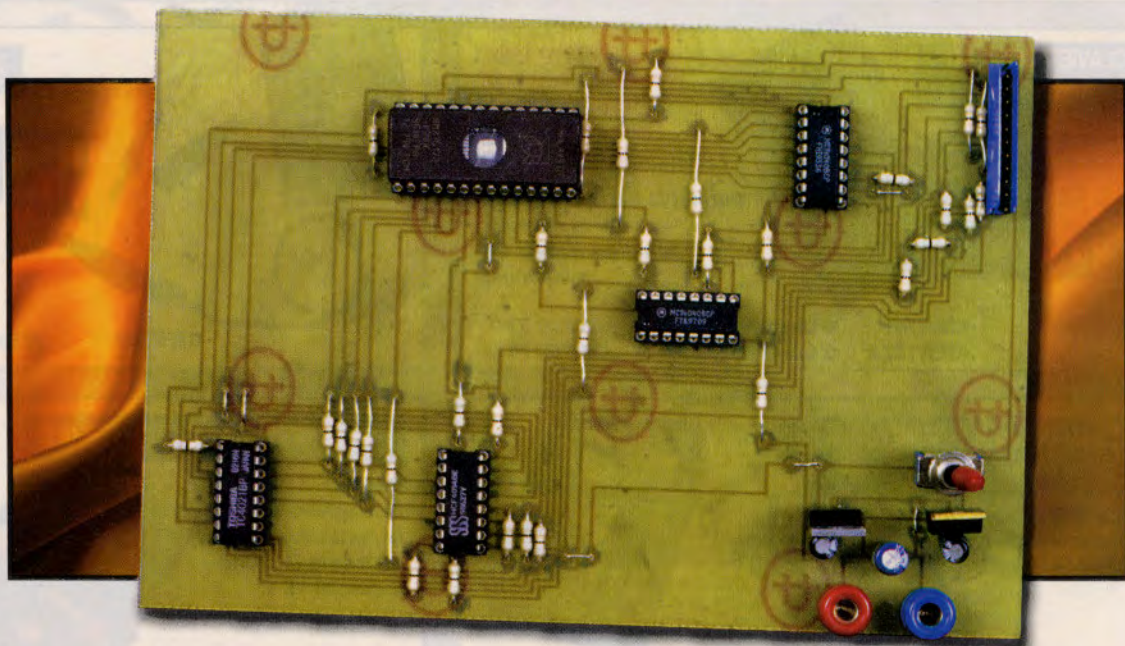
L'interprétation des résultats de la lecture des cartes les plus diverses (bancaires, de fidélité, de péage, etc.) sortirait très largement du cadre de cet article, un livre entier suffisant à peine à effleurer ce sujet passionnant (ETSF chez DUNOD). Notons toutefois qu'on doit

observer une similitude, au moins partielle, avec les caractères qui sont souvent gravés sur les cartes, en particulier bancaires. Les pistes contiennent toutefois davantage d'informations (jusqu'à 40 chiffres pour la piste ISO 2) et c'est là que l'exploration devient véritablement passionnante !

P. GUEILLE

# Programmateur de 27(C)64/27(C)128

par le port parallèle du PC

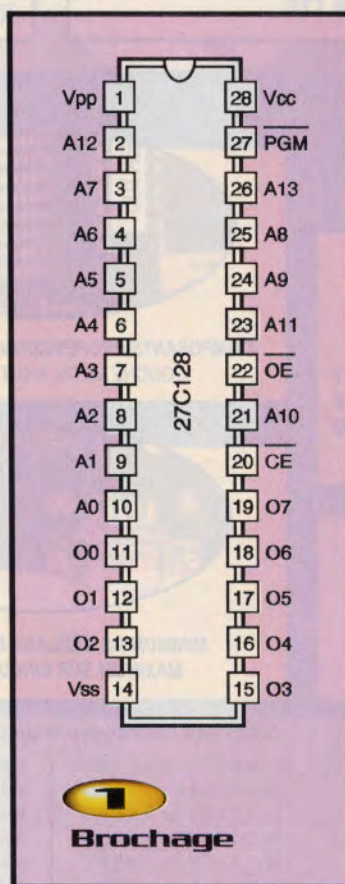


Les montages utilisant les EPROM sont de plus en plus fréquents. En effet, il est rare que l'on ne trouve un "Électronique Pratique" sans que l'on ait affaire à une EPROM. Les articles "Mettez un microcontrôleur dans vos montages" en possèdent généralement tous une, couplée aux microcontrôleurs. L'électronicien, même amateur, doit posséder un programmeur d'EPROM qui fera l'objet de cet article.

## Brochage du 27C64/27C128 (figure 1)

Le 27CXX est une UVPROM qui s'alimente en 5V. Nous remarquons que PGM, OE et CE sont activés quand ils sont à la masse. La lecture se fait selon la succession des signaux donnée **figure 2**. Lorsque CE, OE sont à la masse et PGM au niveau logique haut (5V), il faudra alors appliquer une tension de 5V à Vpp, présenter les bits d'adresse sur [A0-A13] et lire le contenu de l'EPROM à cet emplacement, sur les bits [Q0-Q7]. Lorsque l'EPROM est vierge, tous les bits sont au niveau logique haut. La programmation consiste alors à en placer au niveau bas. L'écriture se fait selon la succession des signaux donnée **figure 3**.

Tout d'abord VPP est à 12V ; PGM et OE sont au niveau logique haut ; l'adresse est présentée sur [A0-A13]. Viennent ensuite les données à placer à l'adresse indiquée. Puis PGM passe un court moment au niveau logique bas. On vérifie enfin si l'octet



**1**  
Brochage

a été correctement inscrit. Si ce n'est pas le cas, on réécrit l'octet par-dessus.

## Le schéma de principe (figure 4)

L'alimentation se fait en 15V. Elle est ramenée à 5V avec le 7805 afin d'alimenter les CI et l'EPROM et à 12V avec le 7812 pour le Vpp de l'EPROM. Le mode écriture ou programmation se fait grâce à un commutateur qui amènera soit 5V à Vpp soit 12V.

Lors d'une programmation ou d'une écriture, il faudra faire défiler les adresses une à une. C'est pour cela que l'on utilise deux 4040 (compteur binaire à 12 étages), montés en cascade (CLK du second est placé sur Q11 du premier). Ainsi chaque front descendant sur CLK du premier 4040 incrémentera le bus des adresses de l'EPROM. La lecture se fait, bit par bit, grâce à un registre à décalage qui est le 4021. De ce fait,

même les utilisateurs des anciens PC ne possédant pas un port bidirectionnel (ECP/EPP) pourront bénéficier du programmeur. L'écriture se fait quand OE de l'EPROM est placé au niveau haut, quand la lecture est inhibée. Elle se fait aussi grâce à un registre à décalage qui est le 4094. Vous l'aurez donc compris, le 4094 et le 4021 sont tous les deux sur le bus des données de l'EEPROM.

Afin de simplifier le montage, les bus de données et d'adresses sont croisés. Mais pas d'inquiétude, le programme se charge de tout remettre en ordre. Le principe de ces registres à décalage est plus amplement détaillé dans le n°220 d'E.P.

Pour économiser une ligne de données du port d'imprimante, le CLK du 4021 et du 4094 sont reliés. Le port d'imprimante est donc connecté au montage de la façon suivante :

- 2 : CLK 4040
- 3 : RST 4040
- 4 : CLK 4021/4094
- 5 : D 4094
- 6 : STR 4094
- 7 : STR 4021
- 8 : PGM EPROM
- 9 : OE EPROM
- 10 : Q 4021
- 25 : GND

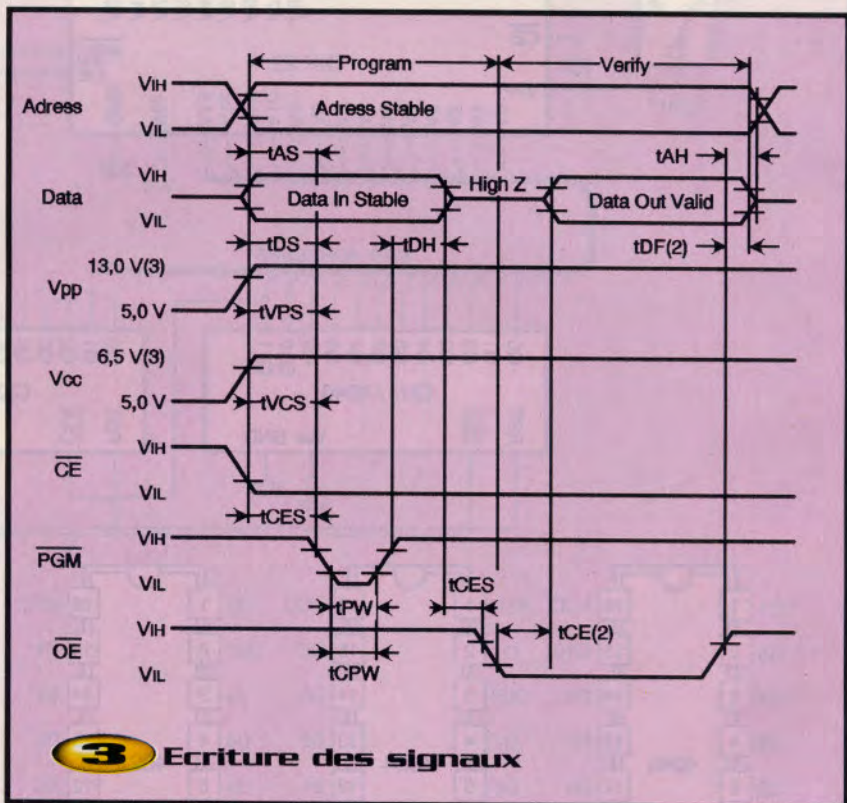
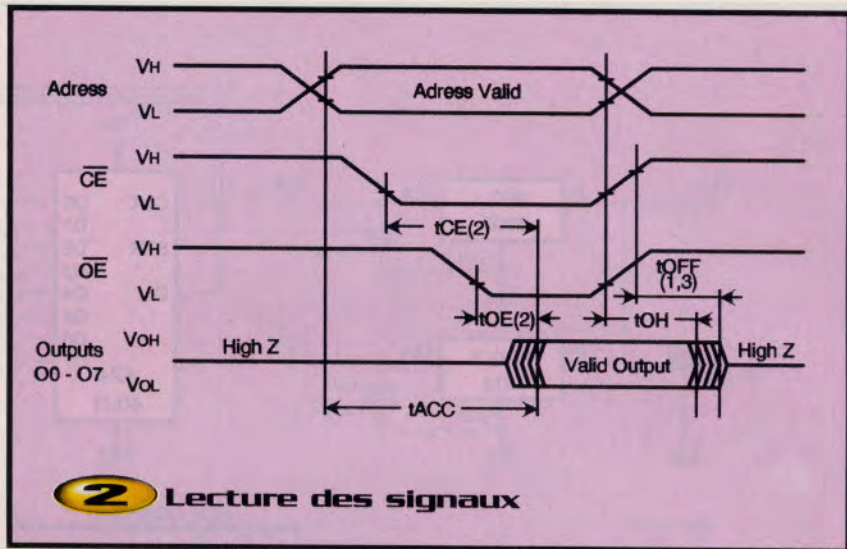
Le reste n'est qu'une histoire de logiciel.

## La réalisation (figures 5 et 6)

La réalisation du circuit imprimé n'est pas difficile étant donné que les pistes sont assez larges. Ceci explique pourquoi le circuit imprimé est aussi grand pour les 10 composants qui y sont implantés.

Le perçage se fera avec un foret de 0,8 mm sauf pour les régulateurs. Il est assez important de souder d'abord les straps, qui sont presque au nombre de quarante. Puis on implantera les supports des circuits intégrés et, finalement, les 3 condensateurs, les 2 régulateurs et le commutateur. Il convient d'utiliser un câble de 10 fils qui relira le montage à l'ordinateur. Il sera relié au montage de la façon suivante :

Montage	DB 25	Montage	DB 25
1	2	6	8
2	3	7	7
3	4	8	9
4	5	9	10
5	6	10	25



## Le logiciel

Vous pourrez le télécharger sur notre site Internet. Il a été réalisé en C et les sources sont comprises dans le fichier afin que vous puissiez les modifier ou les recompiler si nécessaire.

Il fonctionne sous DOS et, donc, ne nécessite pas forcément Windows.

Il se compose de 4 petites fenêtres (**écran 1**) : une contenant les adresses, un buffer avec le contenu de l'EPROM en hexadécimal, un buffer avec le contenu de l'EPROM en caractères ASCII et une petite fenêtre pour vous guider.

Vous avez la possibilité de charger un fichier dans le buffer, lire le contenu de l'EPROM, programmer l'EPROM avec le contenu du buffer, vérifier que le contenu de l'EPROM est le même que celui du buffer et enregistrer le buffer dans un fichier.

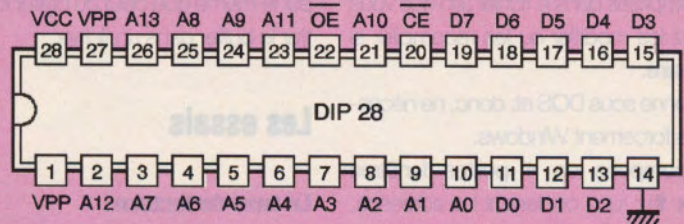
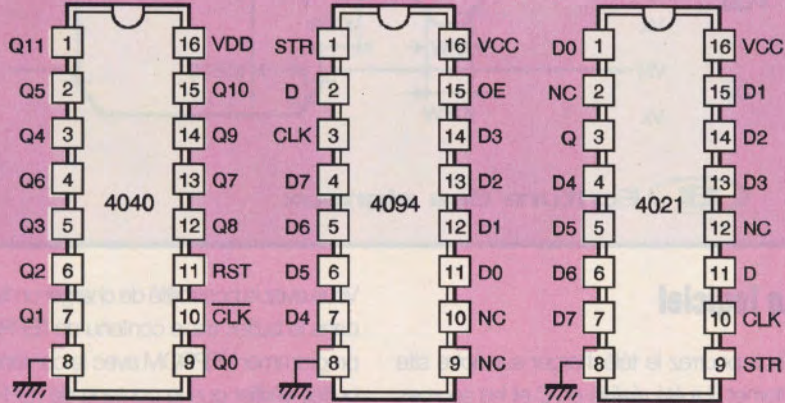
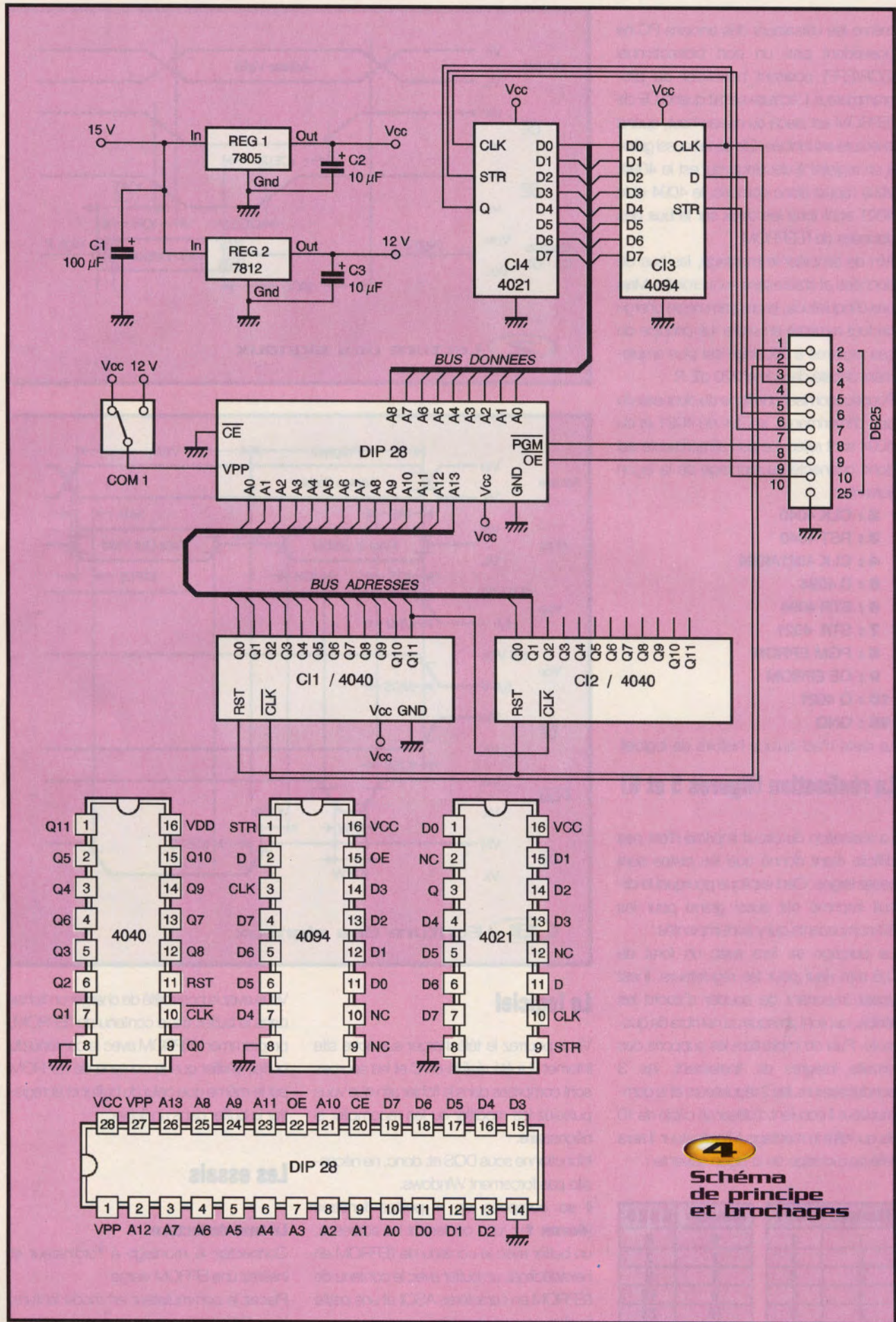
## Les essais

### Le test de lecture

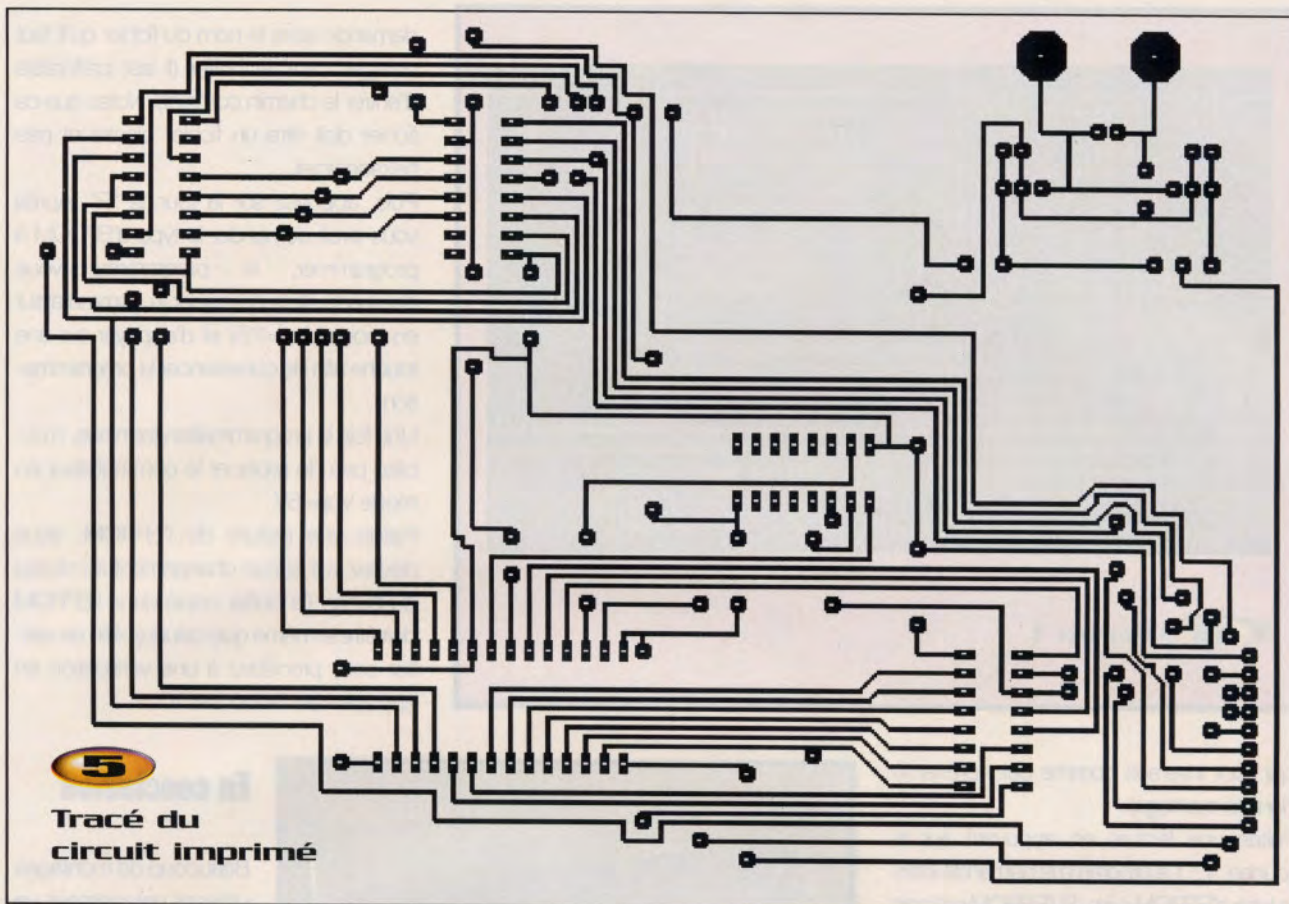
Connectez le montage à l'ordinateur et insérez une EPROM vierge.

Placez le commutateur en mode lecture,  $V_{pp}=5V$  (attention à certains commutateurs



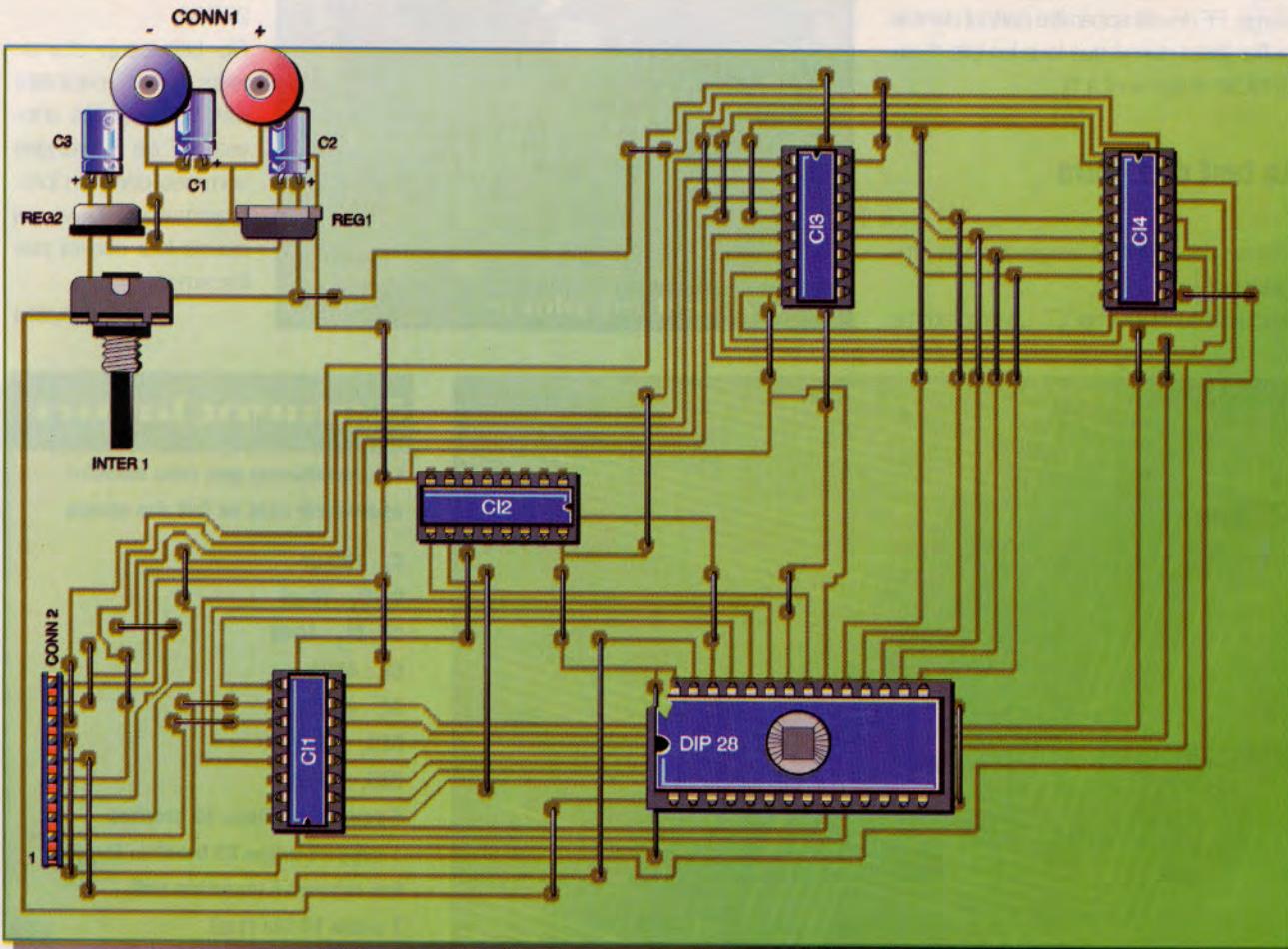


**4**  
Schéma de principe et brochages



**5**

Tracé du circuit imprimé



**6** Implantation des éléments

ADRESSE	HEX	OUTET
0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
16	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
32	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
48	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
64	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
80	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
96	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
112	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
128	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
144	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
160	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
176	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
192	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
208	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
224	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
240	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	

Programmeur de 27C64 / 27C128 par le port parallèle du PC  
Assurez vous que le commutateur est placé sur Vpp = 5V\_

**8** Ecran 1

qui sont inversés comme celui présenté dans le montage)

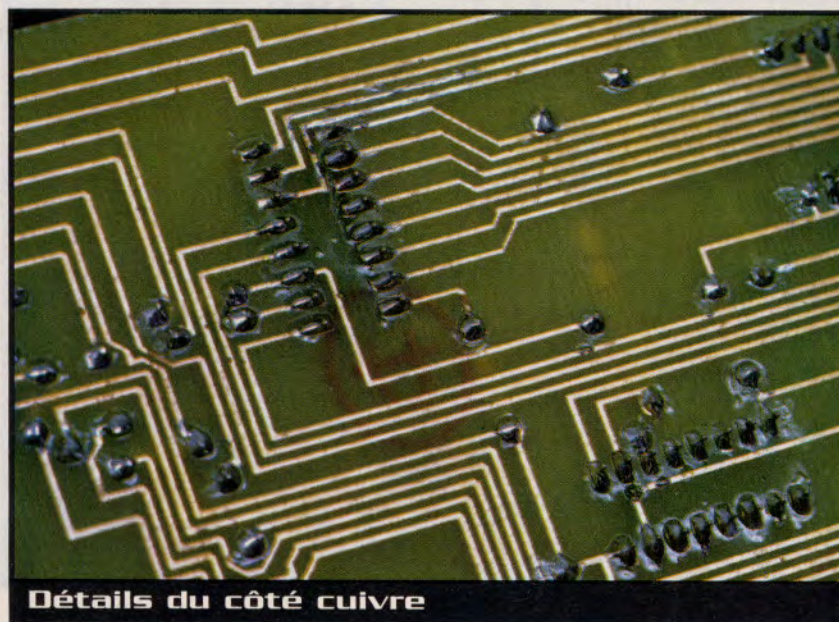
Faites une lecture en appuyant sur la touche "L". Le programme demande alors le type d'EPROM à lire. Si l'EPROM est bien vierge, FF devrait apparaître partout dans le buffer (étant donné que tous les bits d'une EPROM vierge sont à 1).

### Le test d'écriture

Placez une EPROM vierge dans le programmeur.  
Appuyez sur la touche "C". Le programme



Le 27C128 en place



Détails du côté cuivre

demande alors le nom du fichier qu'il faut charger dans le buffer (il est préférable d'entrer le chemin complet). Notez que ce fichier doit être un fichier binaire et pas hexadécimal.

Puis, appuyez sur la touche "P". Après vous avoir demandé le type d'EPROM à programmer, le programme vous demande alors de placer le commutateur en mode Vpp=12V et d'appuyer sur une touche afin de commencer la programmation.

Une fois la programmation terminée, n'oubliez pas de replacer le commutateur en mode Vpp=5V.

Faites une lecture de l'EPROM. Vous devriez voir aucun changement au niveau du buffer (le buffer copié dans l'EPROM doit être le même que celui lu). Afin de vérifier ceci, procédez à une vérification en appuyant sur la touche "V"

### En conclusion

Beaucoup de montages utilisent uniquement un 27C64.

Or, beaucoup d'amateurs n'ont peut-être pas la possibilité d'investir de grandes sommes dans un programmeur universel qui ne leur servira pas forcément !

R. AMIN

### Nomenclature

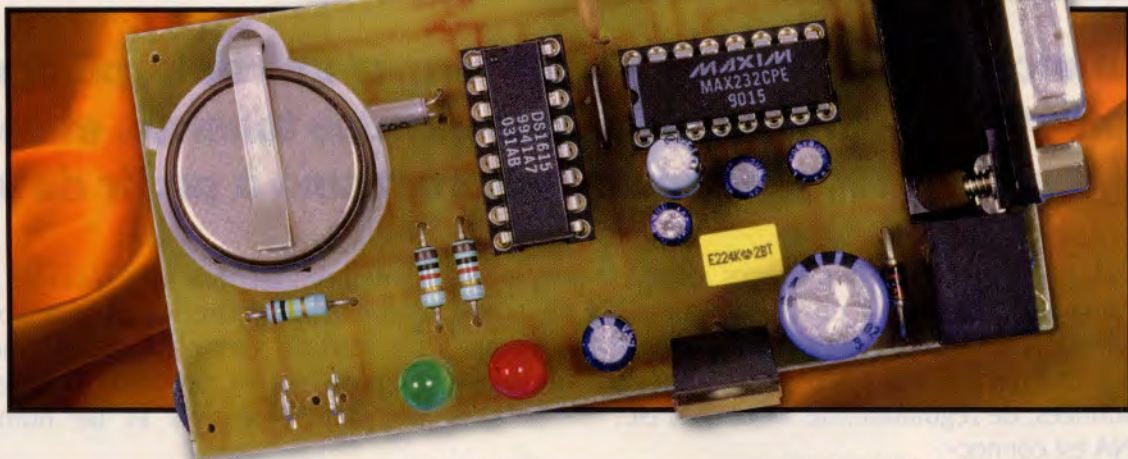
Les résistances que vous semblez apercevoir sont en fait des straps

- C<sub>1</sub> : 100 µF
- C<sub>2</sub>, C<sub>3</sub> : 10 µF
- CI<sub>1</sub>, CI<sub>2</sub> : 4040
- CI<sub>3</sub> : 4094
- CI<sub>4</sub> : 4021
- REG<sub>1</sub> : 7805
- REG<sub>2</sub> : 7812

- 4 supports tulipes 16 broches
- 1 support tulipe 28 broches larges (ou support à insertion nul)
- 1 câble 10 fils (1m)
- 1 connecteur DB25 mâle

# Enregistreur de température autonome

Sous ce titre relativement banal, se cache un montage dont vous ne pourrez très vite plus vous passer pour peu que vous ayez, un jour ou l'autre, besoin de faire de la mesure de température. En effet, après une phase de configuration de ses conditions de mesure qui a lieu par connexion à un PC, notre enregistreur de température devient totalement autonome et peut être placé où bon vous semble : intérieur de votre congélateur, de votre ampli hi-fi ou bien encore des surgelés que vous ramenez du supermarché. Il assure alors seul la mesure de la température à des intervalles programmables de 1 à 255 minutes et peut mémoriser de quelques dizaines d'heures de mesures à plus d'un an en horodatant chaque mesure à la seconde près.



## Le DS 1615

Lorsque vous estimez que votre session de mesures est terminée, la simple connexion du montage à un PC permet, à ce dernier, de récupérer les données enregistrées pour les afficher sur divers graphes ou pour tout autre traitement que vous voudrez leur faire subir.

Jusqu'à ces derniers temps, la réalisation d'un tel montage, même si elle ne présentait pas de difficultés particulières, nécessitait la mise en œuvre d'un microcontrôleur, d'un capteur de température, d'un convertisseur analogique/digital et d'une mémoire EEPROM pour stocker les résultats des mesures. C'était assez encombrant et, surtout, peu compatible de l'autonomie que nous avons évoquée ci-dessus ; autonomie bien réelle puisque notre montage peut fonctionner en continu pendant cinq ans avec sa propre pile.

C'est donc une autre solution que nous avons utilisée, grâce à la commercialisation par Dallas Semiconductors d'un circuit intégré bien distribué en France et peu coûteux : le DS 1615. Nous allons voir sans plus tarder quelles sont ses possibilités car, comme il constitue le cœur de notre montage, ce sont donc les mêmes que celles dont nous pourrions bénéficier.

Ce circuit a été conçu spécifiquement pour réaliser les fonctions que nous souhaitons réaliser, c'est à dire mesurer une température à un instant donné de manière répétitive et enregistrer et dater cette mesure. Bien qu'il soit présenté dans un boîtier DIL 16 pattes (une version CMS est même disponible pour les situations où un faible encombrement est primordial), le DS 1615 renferme un grand nombre de sous-ensembles puisque l'on y trouve :

- une horloge temps réel avec ses registres associés. Cette horloge indique heures, minutes et secondes mais aussi jour, mois et années et sait tenir compte des années bissextiles

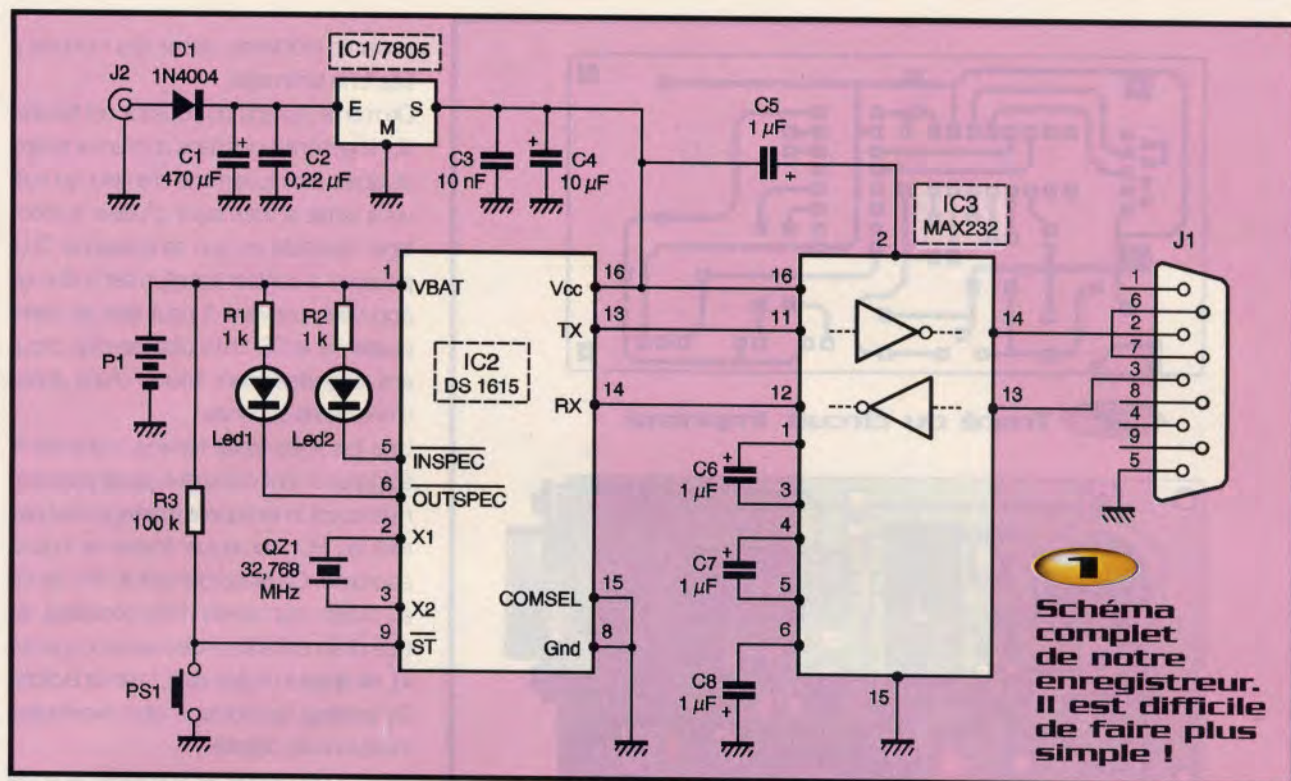
(et elle a échappé au bug de l'an 2000 !),

- une mémoire vive non volatile de 32 octets pour le stockage des données de l'utilisateur du circuit,
- une mémoire de 96 octets destinée au stockage des conditions d'alarme sur date et heure ou sur des températures hors limites que l'on peut librement définir,
- une mémoire d'histogramme de 128 octets,
- une mémoire de stockage des résultats de mesure de 2048 octets,
- deux interfaces séries : une synchrone pour microcontrôleurs et une asynchrone que nous utiliserons avec notre PC,
- un capteur de température, puisque celui-ci est intégré dans le

Paramètre	Spécifications
Gamme de température	-40°C à +85°C
Résolution de mesure	+/-0,5°C
Précision de mesure	+/-2°C
Mémoire de température	2048 échantillons
Mémoire d'histogramme	65 535 échantillons
Intervalle de mesure	1 à 255 minutes
Alarmes	Températures haute et basse, heure et date
Horloge temps réel	Seconde, minute, heure, jour, mois, année, Compatible an 2000
Alimentation	Pile lithium 2,9V à 90 % de Vcc
Consommation	200 nA au repos
sur pile lithium	600 µA en phase de mesure



Principales caractéristiques du DS 1615.



**1**  
Schéma complet de notre enregistreur. Il est difficile de faire plus simple !

DS 1615 lui-même, et son convertisseur analogique/digital associé,

- et bien sûr, un oscillateur d'horloge et une logique très évoluée pour gérer tout cela. Compte tenu de cette intégration très poussée, la mise en œuvre du DS 1615 ne peut qu'être très simple comme nous allons le découvrir dans un instant. Mais, avant cela, faites un petit détour par le **tableau 1** afin de découvrir les principales caractéristiques du circuit.

On y remarque qu'il réalise systématiquement deux types de mesures :

- une mesure de température à des intervalles réguliers programmables de 1 à 255 minutes. Ce mode de fonctionnement est limité à 2048 mesures distinctes ce qui donne tout de même une capacité pouvant dépasser l'année pour des mesures espacées de 255 minutes !

- une mesure de type histogramme qui indique, pendant la période de fonctionnement, le nombre de températures enregistrées pour chaque pas de mesure de 2°C. La mémoire d'histogramme est indépendante de la mémoire de mesures à intervalles réguliers et peut contenir, quant à elle, jusqu'à 65 535 résultats.

Les résultats de ces deux types de mesures peuvent évidemment être ensuite récupérés sur le PC, auquel le montage est connecté, pour affichage graphique et/ou traitement avec un tableur ou tout autre logiciel de votre choix.

## Schéma de notre enregistreur

Le paradoxe de notre montage est que l'essentiel de son schéma ne sert que pendant la phase de connexion au PC comme vous pouvez le constater à l'examen de la **figure 1**. Ce schéma peut, en effet, être scindé en trois parties distinctes :

- celle organisée autour de IC<sub>2</sub>, qui n'est autre que le DS 1615, qui réalise tout le travail de mesure et d'enregistrement de température,
- celle organisée autour de IC<sub>3</sub> qui réalise la conversion des niveaux logiques TTL du DS 1615 en niveaux RS232 de l'interface série du PC,
- et enfin, la partie entourant IC<sub>1</sub>, qui ne sert qu'à alimenter IC<sub>3</sub> et IC<sub>2</sub> pendant la phase de connexion avec le PC.

Examinons, tout d'abord, la mise en œuvre du DS 1615 dont l'essentiel des composants se trouve placé sur sa gauche. On y reconnaît le quartz de pilotage de son horloge interne, deux LED et un poussoir (dont nous verrons le rôle lors de la description du logiciel) et une pile au lithium standard P<sub>1</sub> capable d'alimenter le circuit seul pendant une durée minimum de cinq ans !

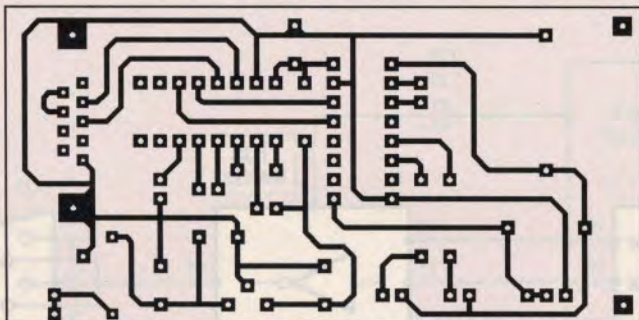
Côté droit du DS 1615, nous remarquons la patte COMSEL reliée à la masse, ce qui valide l'interface série asynchrone. Cette interface est disponible sur les pattes Tx et Rx, sous réserve que le circuit reçoive une tension de 5V sur sa patte Vcc. En d'autres

termes, l'ensemble du DS 1615 fonctionne à partir de la pile P<sub>1</sub> et l'alimentation externe 5V ne sert que lors des liaisons avec le PC. Comme les niveaux fournis par le DS 1615 sont aux normes TTL, nous réalisons leur conversion en vrais niveaux RS232 avec IC<sub>3</sub> qui n'est autre qu'un MAX232 ou une de ses secondes sources. Cette solution est préférable à tous les montages plus ou moins "bricolés" que l'on rencontre encore trop souvent de nos jours. Elle permet, en effet, de garantir une véritable compatibilité RS232, même avec les moins tolérantes des interfaces séries de PC.

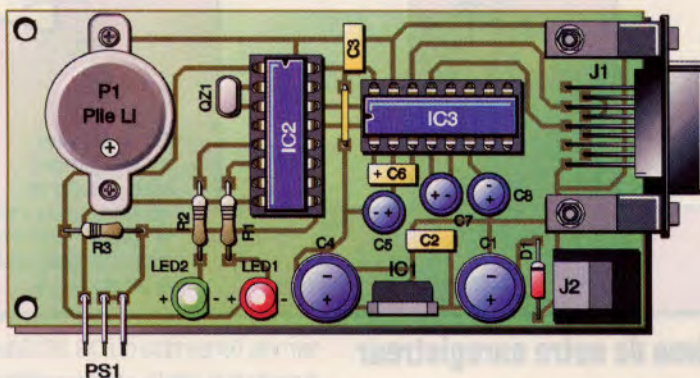
L'alimentation de la patte Vcc du DS 1615 et du MAX232 est assurée par le régulateur intégré IC<sub>1</sub>, monté de façon très classique. N'importe quel bloc secteur style "prise de courant", délivrant environ 9V sous quelques dizaines de mA, convient en entrée, en rappelant bien que cette alimentation n'est nécessaire que tant que le montage est connecté au PC, c'est à dire lors de la programmation initiale des conditions des mesures ou lors de la récupération de ces dernières.

## La réalisation

L'approvisionnement des composants ne pose aucun problème particulier puisque le DS 1615 est tenu en stock par divers revendeurs et que tous les autres composants utilisés sont classiques ; pile au lithium com-



**2** Tracé du circuit imprimé



**3** Implantation des éléments

prise puisque nous avons choisi le modèle qui alimente l'horloge de nombreux PC ainsi que des organisateurs et autres agendas de poche.

Nous n'avons pas cherché à miniaturiser le montage à l'extrême comme vous pouvez le constater avec le tracé du circuit imprimé présenté **figure 2**. Rien ne vous interdit de faire plus petit si vous avez des applications qui le nécessitent et, même, de recourir à la version CMS du DS 1615. Dans ce dernier cas, vous aurez alors intérêt à choisir une pile de plus petit diamètre car elle sera alors, à elle seule, plus grosse que le montage qu'elle alimente.

Revenons à nos moutons en implantant les composants comme indiqué **figure 3**. Ici aussi tout est classique, mais il nous faut vous donner quelques précisions relatives aux LED et au poussoir PS<sub>1</sub>.

L'indication donnée par les LED est d'utilisation facultative comme vous le verrez ci-dessous. Il n'est donc pas indispensable qu'elles soient visibles de l'extérieur du boîtier qui recevra peut-être le montage. Notez que, afin de minimiser la consommation du DS 1615 sur sa pile, ces LED sont parcourues par un courant très faible et sont donc très peu visibles. Si cela risque de vous

poser un problème, utilisez des modèles à très forte luminosité.

De même, l'usage du poussoir est facultatif. Ne le montez pas dans un premier temps et, après avoir lu le mode d'emploi qui suit, vous verrez si votre façon d'utiliser le montage nécessite ou non sa présence. Tout poussoir à contact travail, c'est à dire en appuyant, convient. Il peut être de piètre qualité car le DS 1615 dispose d'un circuit anti-rebondissement interne d'une durée d'une demie seconde.

Une fois votre travail terminé, contrôlez-le soigneusement car aucun essai préalable n'est possible tant que le montage n'est pas relié au PC, sur lequel tourne le logiciel approprié. Ce raccordement au PC se fait au moyen d'un cordon DB9 normalisé, de type droit ou fil à fil. Le bloc secteur, quant à lui, fait appel à un jack de 2,1 mm et la diode D<sub>1</sub> protège le montage des éventuelles inversions de polarité.

Veillez à bien respecter la polarité de la pile au lithium lors de sa mise en place dans son support. Le positif de la pile, repéré de façon bien visible sur son boîtier, est situé côté circuit imprimé du support.

## Nomenclature

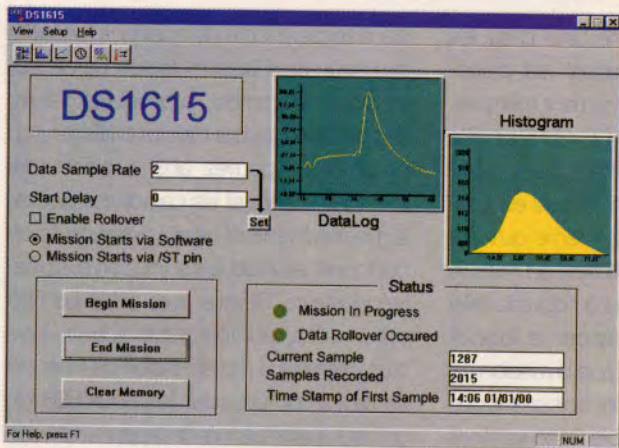
- IC<sub>1</sub> : 7805 (régulateur +5V/1 A, boîtier T0220)**
- IC<sub>2</sub> : DS 1615**
- IC<sub>3</sub> : MAX232, ICL232 ou équivalent**
- D<sub>1</sub> : 1N4004**
- LED<sub>1</sub> : LED rouge**  
**(haute luminosité éventuellement)**
- LED<sub>2</sub> : LED verte**  
**(haute luminosité éventuellement)**
- R<sub>1</sub>, R<sub>2</sub> : 1 kΩ 1/4W 5%**  
**(marron, noir, rouge)**
- R<sub>3</sub> : 100 kΩ 1/4W 5%**  
**(marron, noir, jaune)**
- C<sub>1</sub> : 470 µF/25V chimique radial**
- C<sub>2</sub> : 0,22 µF mylar**
- C<sub>3</sub> : 10 nF céramique**
- C<sub>4</sub> : 10 µF/25V chimique radial**
- C<sub>5</sub> à C<sub>8</sub> : 1 µF/25V chimique radial**
- QZ<sub>1</sub> : quartz 32,768 kHz, boîtier horloger**
- P<sub>1</sub> : pile lithium CR 2032 et son support à souder sur circuit imprimé**
- PS<sub>1</sub> : poussoir, contact en appuyant (facultatif)**
- J<sub>1</sub> : connecteur DB9 femelle coudé à souder sur circuit imprimé**
- J<sub>2</sub> : jack mâle 2,1 mm à souder sur circuit imprimé**
- 2 supports de C.I. 16 pattes**

## Le logiciel d'exploitation

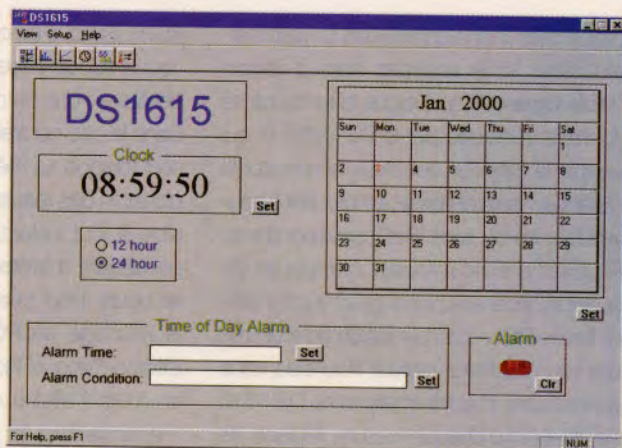
Le logiciel que nous vous proposons sur le CD Rom joint est celui fourni par Dallas avec son kit d'évaluation du DS 1615. Il présente l'avantage d'être complet et de permettre ainsi une mise en œuvre rapide de l'enregistreur, mais aussi celui d'être disponible intégralement sous forme de source afin de vous permettre de le modifier ou de l'adapter à vos besoins.

La version exécutable de ce programme, qui fonctionne sous Windows 95 ou 98, est contenue dans les deux fichiers compressés DS1615K1.ZIP et DS1615K2.ZIP. Le source complet du programme, en Visual C++, est, quant à lui, contenu dans le fichier compressé 1615\_src.ZIP. La décompression de ce fichier donne accès aux différents modules du source et ne présente pas de difficulté particulière si vous avez l'habitude de la programmation en Visual C++. Pour ce qui est du programme exécutable, par contre, il faut suivre à la lettre la procédure suivante :

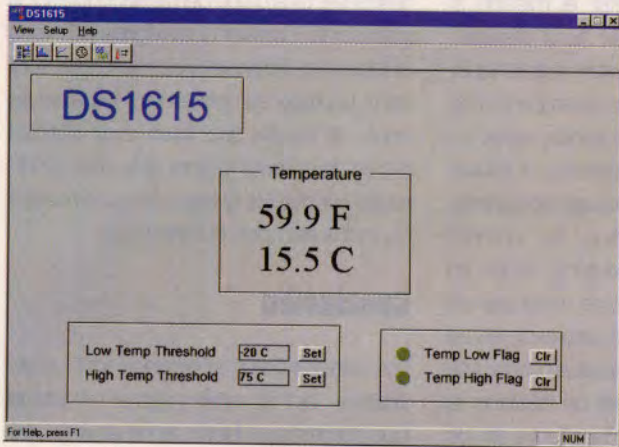
- copier les deux fichiers K1 et K2 dans le répertoire de votre choix sur votre disque dur : répertoire qui ne servira plus à rien



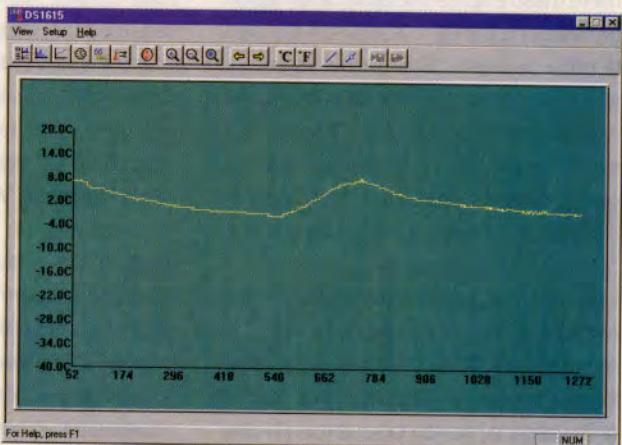
**4** L'écran de paramétrage des conditions d'enregistrement



**5** L'écran de mise à l'heure de l'horloge temps réel et de définition d'alarme



**6** Mesure de la température courante et définition des alarmes de température



**7** Affichage des données enregistrées en fonction du temps

ensuite (utilisez le TEMP de Windows par exemple),

- décompressez les deux fichiers dans le répertoire de votre choix avec un décompresseur ZIP tel que WinZip par exemple,
- la décompression donne naissance à un répertoire appelé DS1615 contenant deux sous-répertoires DISK1 et DISK2. Copiez les deux fichiers contenus dans le sous-répertoire DISK2 dans le sous-répertoire DISK1,

- depuis le sous-répertoire DISK1 exécutez le fichier SETUP non sans avoir, au préalable, fermé toutes les applications ouvertes sur votre machine,

- suivez les indications fort simples de l'écran d'installation et, sauf contrainte majeure, acceptez les répertoires d'installation proposés par le logiciel car ils n'interfèrent en aucun cas avec ceux utilisés par Windows.

Une fois cette installation terminée, le logiciel est prêt à être utilisé ce qui, comme nous allons le voir sans plus tarder, est fort simple. Lancez le logiciel d'exploitation du

DS 1615. Une première fenêtre s'affiche vous demandant d'indiquer le port série utilisé par votre montage. Dès que vous avez répondu à cette question, le programme tente d'entrer en communication avec le DS 1615 et affiche l'écran d'accueil visible **figure 4**, permettant de paramétrer les missions de mesures. Si la communication s'établit correctement, vous pouvez alors programmer le DS 1615. Dans le cas contraire, une fenêtre s'ouvre vous indiquant que la communication n'a pu être établie et vous propose d'essayer à nouveau. Si cela vous arrive, vérifiez, en premier lieu, si le câble que vous avez utilisé est un câble droit, c'est à dire encore un câble où les liaisons sont établies fil à fil (patte 2 sur patte 2, patte 3 sur patte 3 et ainsi de suite). Vérifiez ensuite votre montage, ce qui ne doit pas présenter de grande difficulté vu le faible nombre de composants utilisés.

Pour programmer une mission de mesure, il faut commencer par mettre à l'heure le DS 1615. Pour cela, cliquez sur l'icône représentant une petite horloge ce qui fait afficher

l'écran visible **figure 5**. Pour mettre le circuit à l'heure et à la date, cliquez sur le bouton "Set" placé à droite de l'affichage de l'heure. Une fenêtre s'ouvre et vous propose de saisir heures, minutes, secondes et date. Cliquez dans chaque case avant d'y saisir les données numériques et notez que la date doit être saisie sous la forme de ses deux derniers chiffres (mais oui le DS 1615 passe l'an 2000 comme le montre la figure 5 !). Le jour, quant à lui, est saisi sous forme numérique de 1 à 7 ou en faisant ouvrir la liste déroulante proposée.

Une fois cette mise à l'heure effectuée, vous pouvez également saisir une heure d'alarme et une condition d'alarme si nécessaire (toutes les secondes, toutes les minutes, lorsqu'il y a égalité complète de l'heure, etc.) en cliquant à chaque fois sur les boutons "Set" placés à côté des rubriques correspondantes. L'alarme est matérialisée par l'allumage du voyant rouge que vous pouvez éteindre ensuite en appuyant sur le bouton "Clr". Vous pouvez également définir des températures d'alarme, haute et basse, en

cliquant sur l'icône comportant un petit thermomètre. Vous accédez alors à l'écran visible **figure 6** qui indique la température courante mesurée par le DS 1615 et qui permet la définition des deux températures d'alarme, haute et basse, en cliquant à nouveau sur les boutons "Set" correspondants. Revenez ensuite à l'écran principal en cliquant sur l'icône la plus à gauche pour définir les conditions de la mission de mesures que vous souhaitez réaliser. Pour cela, vous devez fournir trois informations au DS 1615 : le délai entre chaque mesure, réglable de 1 à 255 minutes dans la boîte de saisie "Data Sample Rate", le délai avant la première mesure après validation de la mission, réglable lui aussi de 1 à 255 minutes dans la boîte de saisie "Start Delay" et enfin le mode de déclenchement de la mission en cochant une des deux cases "Mission Start via Software" pour un démarrage par logiciel ou "Mission Start via S/T pin" pour un démarrage par action sur le poussoir optionnel prévu sur le montage.

Une fois ces paramètres définis, il faut remettre à zéro la mémoire du DS 1615 en appuyant sur "Clear Memory" puis valider les paramètres en appuyant sur "Set" et enfin valider la mission lorsque vous le désirez en appuyant sur "Begin Mission". Le voyant vert "Mission in Progress" s'allume alors et l'enregistreur est prêt à redevenir autonome. Vous pouvez alors quitter le logiciel, débrancher le montage de son alimentation et du port série du PC et le placer où bon vous semble pour enregistrer les températures dans les conditions que vous avez définies.

Notez que l'écran de programmation disposait d'une case à cocher "Enable Rollover" autorisant l'enregistrement continu des températures, même en cas de dépassement de la capacité de la mémoire du DS

1615. Si vous avez coché cette case, les nouvelles données écrasent les précédentes en cas de débordement mémoire. Dans le cas contraire, le DS 1615 s'arrête seul lorsque sa mémoire est pleine. L'exploitation des résultats de mesure est possible à tout instant, sans même qu'il soit nécessaire d'arrêter la mission de mesure en cours. Pour cela, il faut à nouveau relier le montage au PC et lancer le logiciel. L'écran d'accueil indique que la mission est en cours et affiche, dans les trois cases correspondantes, le nombre d'échantillons mesurés, l'échantillon en cours de mesure et la date et l'heure du premier échantillon. Vous pouvez alors arrêter la mission si nécessaire en cliquant sur "End Mission". Pour voir les résultats sous forme graphique, cliquez sur la troisième icône en partant de la gauche, ce qui donne accès, après un délai d'une ou deux secondes, à l'écran visible **figure 7**. Vous pouvez alors lire les températures en fonction du numéro d'échantillon et divers boutons, situés en haut d'écran, permettent de manipuler ce graphique dont l'affichage peut avoir lieu en degrés Fahrenheit ou Celsius au choix. Les flèches jaunes permettent de déplacer la courbe dans un sens ou dans l'autre afin de la lire dans son intégralité, tandis que les loupes permettent de faire des zoom dans un sens ou dans l'autre pour mieux lire certaines zones. L'absence de quadrillage de l'écran est compensée par le fait qu'une petite fenêtre s'ouvre sous la flèche de la souris pour indiquer précisément la température mesurée sur tel ou tel point de la courbe. En cliquant sur la deuxième icône en partant de la gauche, vous pouvez également accéder à un écran d'affichage des températures sous forme d'histogramme : écran disposant des mêmes fonctionnalités que celles que nous venons de voir. Ce logi-

ciel dispose encore de quelques fonctions que nous vous laissons le soin de découvrir avec, notamment, la possibilité d'écrire des données dans la mémoire utilisateur du circuit. Vous pouvez ainsi y placer, par exemple, le nom et les caractéristiques de la mission à réaliser, ce qui évite ensuite de confondre les modules si vous en avez réalisé plusieurs. Si vous avez câblé les LED optionnelles, leur rôle est le suivant. Lors d'un démarrage d'une mission de mesures au moyen du poussoir, les deux LED clignotent quatre fois lors de l'appui sur ce dernier pour lancer la mission. Par ailleurs, pendant que la mission est en cours, le fait d'appuyer plus d'une demie seconde sur le poussoir fait clignoter quatre fois la LED verte si toutes les températures mesurées sont dans la plage comprise entre les alarmes haute et basse que vous avez définies (écran de la figure 6) alors que c'est la LED rouge qui clignote quatre fois si une mesure au moins est hors de cette plage.

## Conclusion

Les utilisations d'un tel montage sont innombrables, tant en environnement industriel que domestique. Nous avons ainsi pu vérifier avec lui le bon (et aussi le mauvais, hélas) comportement de certains réfrigérateurs ou congélateurs et nous avons pris peur à la lecture des résultats enregistrés sur les radiateurs des transistors de puissance dans certains amplificateurs haute fidélité, pourtant de grande marque. Les transistors de puissance modernes sont plus solides qu'on ne le croit... Sa simplicité d'emploi et sa grande capacité de mémorisation permettent de mettre au musée les classiques thermomètres à minima et maxima bien connus.

C. TAVERNIER

- |  |   |   |   |
|--|---|---|---|
| <input checked="" type="checkbox"/> ACCESSOIRES DJ | <input checked="" type="checkbox"/> CONNECTEURS   | <input checked="" type="checkbox"/> JEUX LUMIERES | <input checked="" type="checkbox"/> OUTILLAGE   |
| <input checked="" type="checkbox"/> ALIMENTATIONS  | <input checked="" type="checkbox"/> COMPOSANTS    | <input checked="" type="checkbox"/> LAMPES-TUBES  | <input checked="" type="checkbox"/> PILES-ACCUS |
| <input checked="" type="checkbox"/> AMPLIFICATEURS | <input checked="" type="checkbox"/> ENCEINTES     | <input checked="" type="checkbox"/> MIXAGES       | <input checked="" type="checkbox"/> PLATINES CD |
| <input checked="" type="checkbox"/> CABLE-CORDONS  | <input checked="" type="checkbox"/> HAUT-PARLEURS | <input checked="" type="checkbox"/> MULTIMETRES   | <input checked="" type="checkbox"/> etc ...     |

**E44**  
ELECTRONIQUE  
www.e44.com

Plus de 800 pages WEB  
Plus de 80Mo de données  
Documents fabricants  
Catalogue E44 intégral  
classé par catégories  
Les sélections de E44



Des promos chaque semaine  
Les liens vers les marques  
Des conseils pratiques  
Le téléchargement tarif  
Des fiches "contact"  
... à visiter absolument !