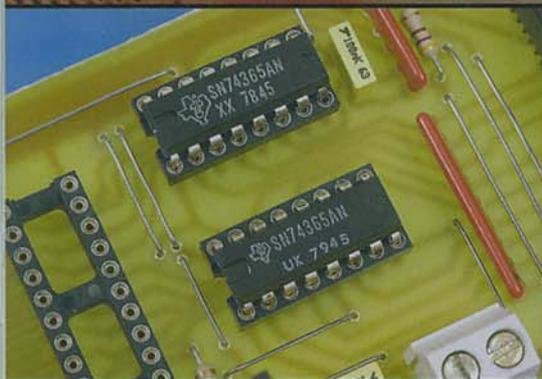


MODEM RADIO POUR PC

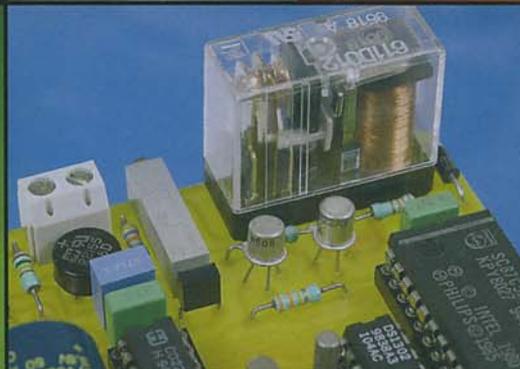
Communication
entre deux
ordinateurs



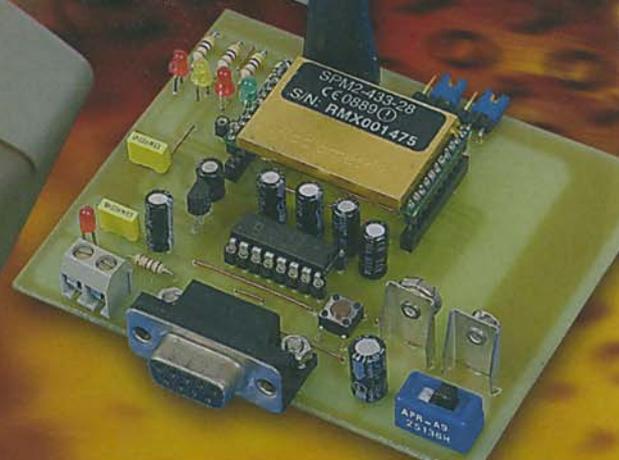
Programmateur de GAL



Lecteur/copieur de télécommande



Filter d'appels téléphoniques



RETROUVEZ AUSSI :

▷ Carte SIM
"minimum"

▷ Pratique
des interfaces

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

T 02437 - 288 - F: 4,50 €



Les procédés de transmission modernes font appels à des procédés de modulation de plus en plus complexes (les technologies 'xDSL' en sont le parfait exemple). Cependant, les procédés de modulation simples tels que la modulation d'amplitude ou la modulation de fréquence ne sont pas tombés en désuétude pour autant. La compréhension de ces procédés est même une base indispensable à assimiler avant d'aborder des techniques plus élaborées. Dans ce domaine également Internet peut s'avérer un terrain de recherches très intéressant. Pour vous en convaincre nous vous invitons à nous suivre dans la découverte de quelques sites en rapport avec la modulation d'amplitude.

internet PR@TIQUE

La première page Internet que nous vous invitons à charger dans votre navigateur se situe à l'adresse <http://www.ac-nice.fr/physique/modulation/modamfm.htm>. Cette page propose un rappel sur les notions mathématiques de base qui sont le fondement de la modulation d'amplitude. Les notions mathématiques en question sont peut être un peu avancées pour nos jeunes lecteurs (niveau de la terminale), mais que cela ne les empêche pas de visiter tout de même les sites proposés ici car les pages sélectionnées contiennent également de nombreuses petites applications Java qui permettent de manipuler les paramètres sans avoir besoin de maîtriser parfaitement les équations. En revanche, pour les plus anciens, la visite de ces quelques sites pourra être également l'objet d'un petit rappel des notions les plus élémentaires.

1 <http://www.ac-nice.fr/physique/modulation/modamfm.htm>

<http://www.univ-lemans.fr/enseignements/physique/02/electro/modula.html>

2

La deuxième page que nous vous invitons à visiter se situe à l'adresse <http://www.univ-lemans.fr/enseignements/physique/02/electro/modula.html>. Cette page contient une petite application Java qui permet d'illustrer les effets de la modulation sur la porteuse. L'intérêt de cet applet Java est double. Tout d'abord il permet de voir très facilement les effets néfastes de la surmodulation (lorsque le paramètre m est supérieur à 1). Ensuite il permet de visualiser la forme du signal que l'on obtiendrait en sortie d'un système démodulateur.

Le site suivant, accessible à l'adresse <http://perso.wanadoo.fr/arsene.perez-mas/transmission/modulation/modulation.htm>, reprend bien évidemment les mêmes notions mathématiques mais en illustrant un peu plus les notions de spectre des signaux modulés en amplitude ou en fréquence. La page située à l'adresse <http://membres.lycos.fr/bnathalieb/spcialit--terminale/physique/resumodula.lwp/odyframe.htm> illustre elle aussi les mêmes notions mais avec une présentation légèrement différente et des animations Java également.

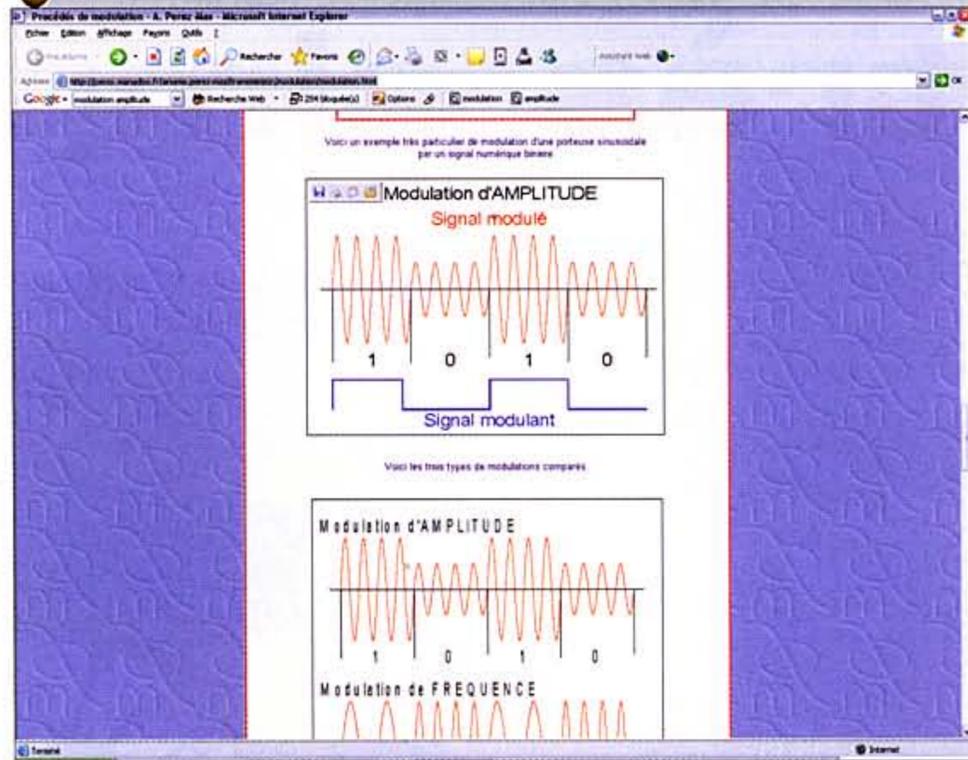
Enfin, le site <http://www.ac-creteil.fr/physique/DOCGRISP/modulation/modula.htm> propose une approche pédagogique intéressante puisqu'elle fait appel à un tableur Excel pour mettre en oeuvre et étudier les formules mathématiques qui régissent la modulation d'amplitude. Si vous disposez d'un tableur Excel n'hésitez pas à suivre pas à pas la méthode proposée car elle est vraiment accessible à tous.

Bien entendu, avec l'aide de votre moteur de recherche préféré il est très facile de trouver encore beaucoup de sites en rapport avec la modulation d'amplitude car ce sujet est vraiment une notion fondamentale de l'électronique.

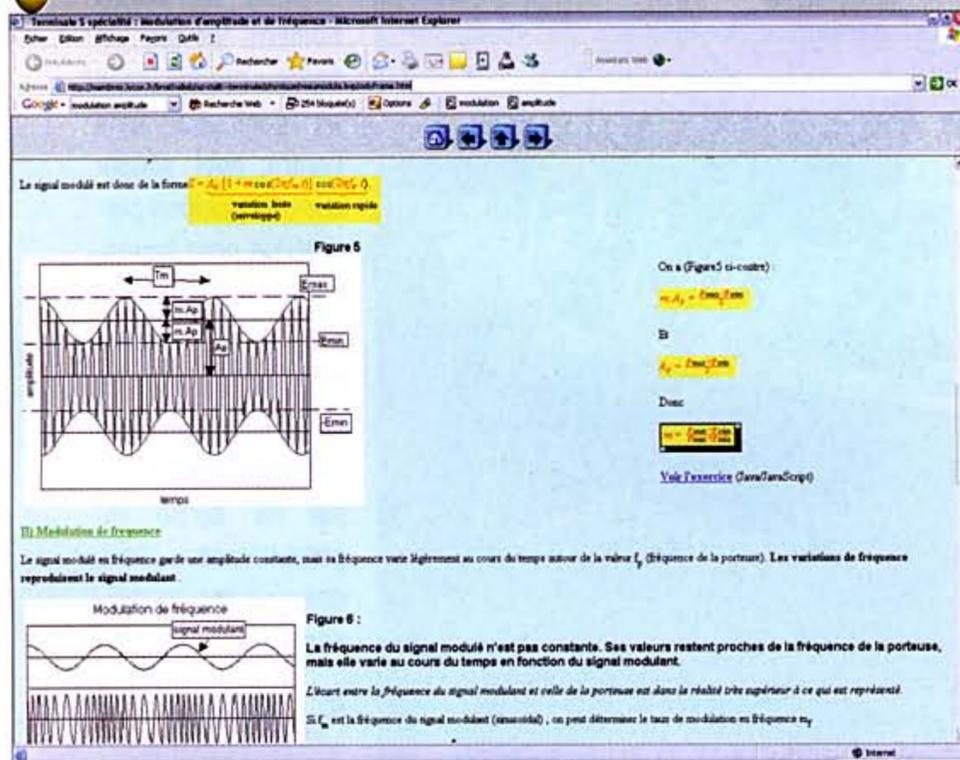
D'ailleurs, vous en trouverez quelques-uns dans la liste des liens proposés à la fin de cet article. Nous vous souhaitons donc une agréable découverte des sites proposés et nous vous donnons rendez-vous le mois prochain pour une autre découverte des trésors de l'internet.

P. MORIN

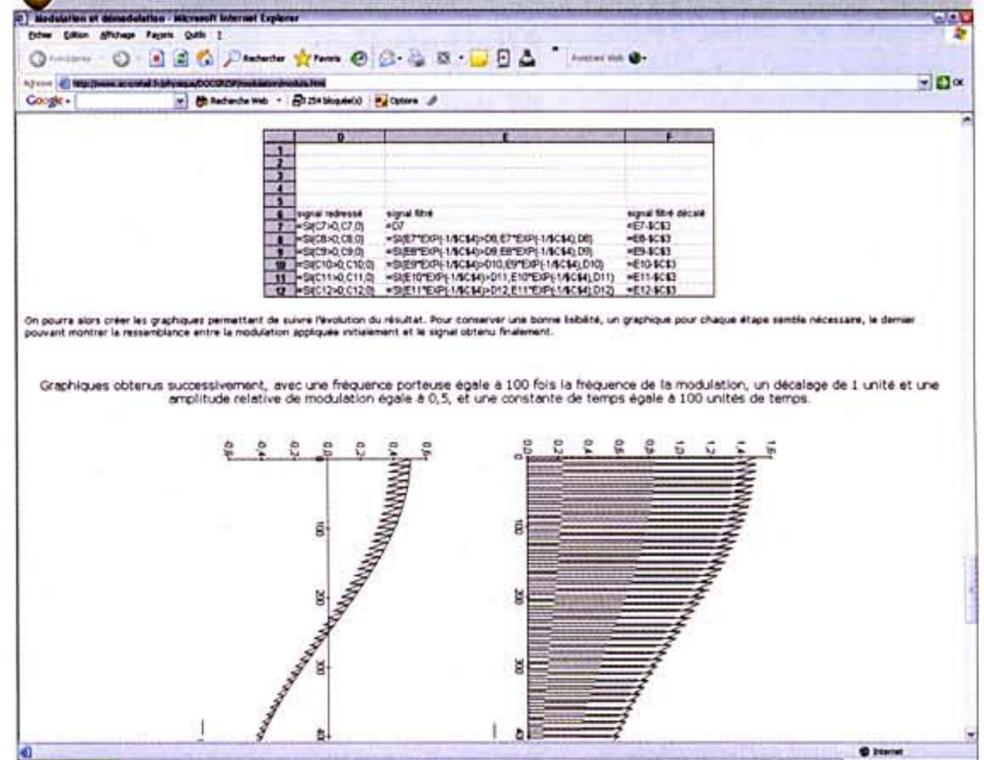
3 <http://perso.wanadoo.fr/arsene.perez-mas/transmission/modulation/modulation.htm>



4 <http://membres.lycos.fr/bnathalieb/sp-cialit--terminale/physique/resumodula.lwp/odyframe.htm>



5 <http://www.ac-creteil.fr/physique/DOCGRISP/modulation/modula.htm>



- <http://www.ac-nice.fr/physique/modulation/modamfm.htm>
- <http://www.univ-lemans.fr/enseignements/physique/02/electro/modula.html>
- <http://perso.wanadoo.fr/arsene.perez-mas/transmission/modulation/modulation.htm>
- <http://membres.lycos.fr/bnathalieb/sp-cialit--terminale/physique/resumodula.lwp/odyframe.htm>
- <http://www.ac-creteil.fr/physique/DOCGRISP/modulation/modula.htm>
- <http://www.epsic.ch/pagesperso/schneiderd/Telec/modulation.htm>
- <http://labo.ntic.org/electro/modula.html>
- <http://www.commentcamarche.net/transmission/transanalog.php3>
- <http://www.discip.crdp.ac-caen.fr/phch/lycee/terminale/modula.htm>
- <http://www.essi.fr/~leroux/transmission/node9.html>
- <http://www.lelectronique.com/ressource/voir.php?rub=cours&num=13>
- <http://www.discip.crdp.ac-caen.fr/physapp/bts/electronique/124/124.htm>
- <http://membres.lycos.fr/bnathalieb/sp-cialit--terminale/physique/ModulA/odyframe.htm>
- <http://www.chimix.com/T-fiches/spe3.htm>
- <http://www.ta-formation.com/cours-am/am/jav-am.htm>



Liste des liens

Un voltmètre très simple

Avec une dépense minime, vous pourrez réaliser ce voltmètre capable de mesurer des potentiels continus de 1 à 72 volts. Nous profiterons de la description de ce montage très pédagogique pour vous proposer la mise en œuvre d'une technique récente de réalisation du circuit imprimé qui vous évitera la rituelle et problématique exposition au rayonnement ultraviolet.

1. Le fonctionnement (figure 1)

Alimentation et potentiel de référence

L'énergie nécessaire au fonctionnement du montage est fournie par une pile de 9 volts qu'un interrupteur I met en service. Afin de disposer d'un référentiel de potentiel stabilisé, la diode zéner DZ, parcourue par un courant de l'ordre de 10 milliampères déterminé par la résistance R1, présente sur sa cathode un potentiel de 8,2 volts. Suivant la position angulaire du curseur du potentiomètre P, il est ainsi possible d'obtenir sur la sortie de ce dernier un potentiel réglable de 1 à 8 volts.

Comparateurs de potentiel

Le circuit intégré IC1 comprend deux ampli-op. Il s'agit d'un LM 358. Le potentiel à mesurer est présenté sur



cipe d'utilisation du voltmètre consiste à tourner le bouton-flèche du potentiomètre devant un secteur gradué, dans un sens ou dans l'autre, pour aboutir sur une position particulière pour laquelle une LED de signalisation verte s'allume.

Désignons par U_p le potentiel disponible sur la sortie du potentiomètre. Trois cas peuvent se présenter.

$$U_p < 5,52 \text{ V}$$

Pour les deux com-

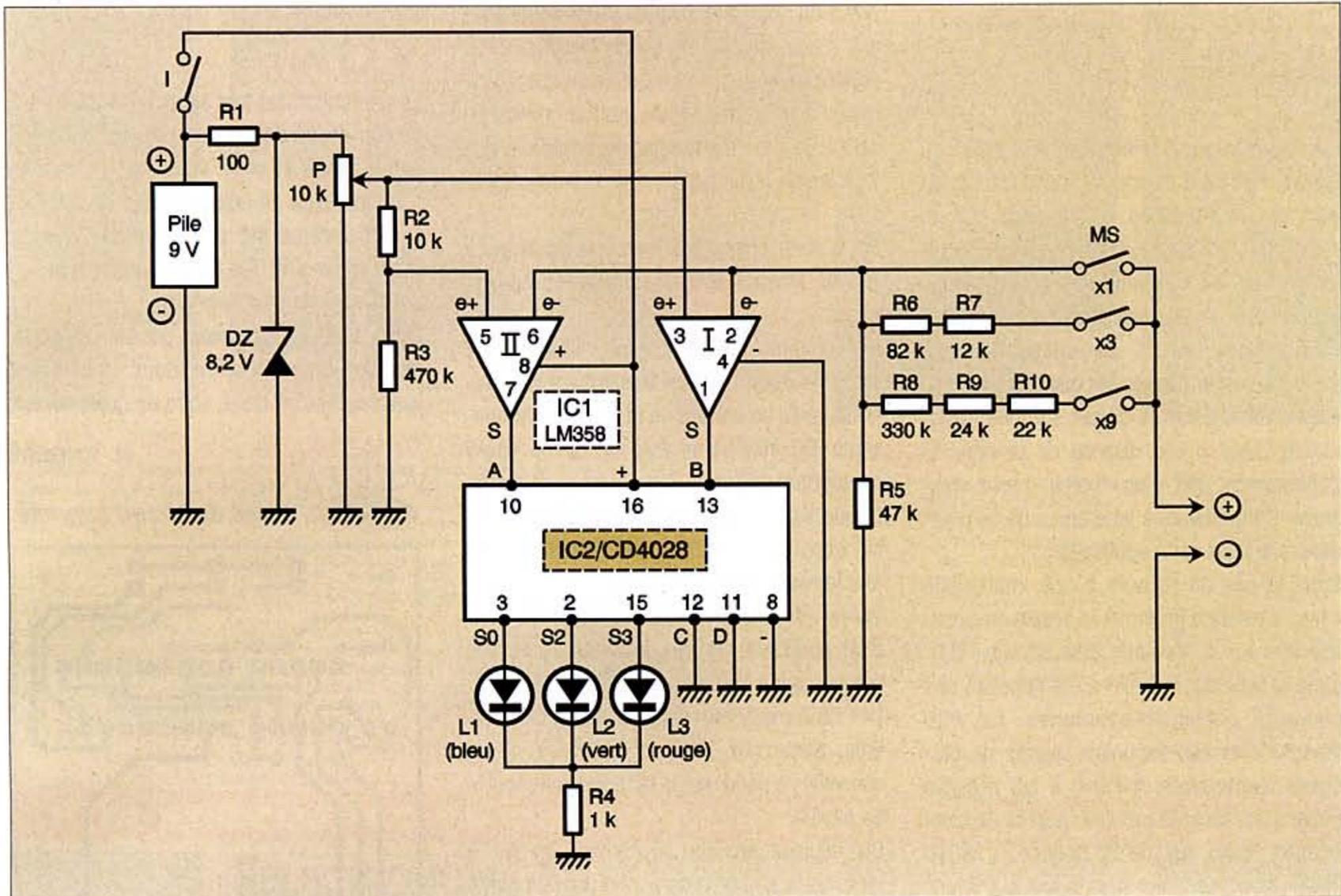
parateurs, le potentiel présenté sur l'entrée directe est inférieur à celui auquel est soumise l'entrée inverseuse. Les sorties des deux comparateurs présentent un état bas.

$$5,52 \text{ V} < U_p < 5,64 \text{ V}$$

La valeur de 5,64 volts est celle qui caractériserait la sortie-curseur du potentiomètre pour obtenir un potentiel de 5,52 volts sur l'entrée directe de l'ampli-op II. En effet :

$$5,64 \text{ V} \times \frac{47}{48} = 5,52 \text{ V}$$

les entrées inverseuses des deux ampli-op. L'entrée directe de l'ampli-op I est reliée à la sortie-curseur du potentiomètre. Quant à l'entrée directe de l'ampli-op II, elle est reliée à la même sortie du potentiomètre, mais par l'intermédiaire d'un pont diviseur dont les valeurs des résistances sont telles que la fraction du potentiel issu du potentiomètre est de 47/48. Pour expliquer le fonctionnement de ces deux comparateurs, il est plus simple de prendre un exemple numérique. Supposons que le potentiel à mesurer soit égal à 5,52 volts. Le prin-



Fonctionnement du voltmètre

Dans ce cas de figure :

- pour l'ampli-op I, le potentiel de l'entrée directe est supérieur à celui de l'entrée inverseuse : la sortie présente un état haut.
- à cause du pont diviseur R2/R3, l'entrée directe de l'ampli-op II reste inférieure à la valeur 5,52 V de l'entrée inverseuse : la sortie reste à l'état bas.

$U_p > 5,64$ V

Pour les deux ampli-op, le potentiel de l'entrée directe est supérieur à celui de l'entrée inverseuse : les sorties des deux comparateurs présentent un état haut.

Décodage

Le circuit intégré référencé IC2 est un décodeur BCD → décimal. Il comporte 4 entrées repérées D, C, B, A (sens de l'écriture d'un nombre binaire). Seules les entrées A et B sont activées. La sortie de l'ampli-op I est reliée à l'entrée B du décodeur et la sortie de l'ampli-op II à l'entrée A. Les trois cas mis en évidence au paragraphe précédent se traduisent donc respectivement par les configurations binaires suivantes (sens de lecture BA) :

$U_p < 5,52$ V	→ 00	État haut sur sortie S_0
$5,52$ V $< U_p < 5,64$ V	→ 10	État haut sur sortie S_2
$U_p > 5,64$ V	→ 11	État haut sur sortie S_3

Ces trois sorties sont respectivement reliées aux LED L1 (bleue), L2 (verte) et L3 (rouge) dont le courant est limité par R4.

Grâce à ces LED de couleurs différentes, l'utilisation du voltmètre est très aisée. Tant que la LED bleue est allumée, le potentiel U_p est trop faible : il convient de tourner le bouton de commande dans le sens horaire afin d'obtenir, pour une position bien précise du curseur, l'allumage de la LED verte. Si on dépasse cette position, la LED rouge s'allume et il est nécessaire de revenir en arrière.

Calibres de mesure

Grâce aux trois interrupteurs du micro-switch MS, il est possible de sélectionner l'un des trois calibres de mesure suivants :

Position x 1 : le potentiel à mesurer est directement présenté sur l'entrée du voltmètre.

Il est donc possible de mesurer des potentiels de 1 à 8 volts.

Position x 3 : le pont diviseur $\frac{R5}{R5 + R6 + R7}$ réalise une division par 3 du potentiel présenté

La plage de mesure s'étend donc de 1 à 24 volts.

Position x 9 : le pont diviseur $\frac{R5}{R5 + R8 + R9 + R10}$ effectue une division par 9, ce

qui permet de mesurer des potentiels de 1 à 72 volts.

La réalisation (figures 2 et 3)

Circuit imprimé

Le circuit imprimé a été réalisé à l'aide du logiciel PROTEUS disponible auprès de la plupart des fournisseurs. Il n'appelle aucune remarque particulière. Pour le reproduire sur la face cuivrée du circuit époxy, nous avons expérimenté la technique du " Press-n-Peel PCB Transfer film ". L'avantage de cette méthode réside dans le fait que l'on peut utiliser comme point de départ soit une imprimante Laser si l'on dispose de ce type de périphérique, soit d'un copieur Laser classique. L'imprimante à jet d'encre ne convient donc pas pour cette application.

Dans le cas du recours à une imprimante Laser, il suffira d'imprimer le dessin du circuit imprimé sur la face non lisse du film PCB : c'est la face sur laquelle a été déposée une émulsion chimique appropriée. Le PCB Transfer film est disponible auprès de plusieurs fournisseurs. A Paris, il est possible notamment de se le procurer auprès de Saint Quentin Radio, rue de St Quentin - 75010 Paris. Attention, il est indispensable d'imprimer le dessin du circuit imprimé tel que l'on peut le voir par transparence quand on l'ob-

Nomenclature

3 straps (2 horizontaux, 1 vertical)

R1 : 100 Ω (marron, noir, marron)

R2 : 10 kΩ (marron, noir, orange)

R3 : 470 kΩ (jaune, violet, jaune)

R4 : 1 kΩ (marron, noir, rouge)

R5 : 47 kΩ (Jaune, violet, orange)

R6 : 82 kΩ (gris, rouge, orange)

R7 : 12 kΩ (marron, rouge, orange)

R8 : 330 kΩ (orange, orange, jaune)

R9 : 24 kΩ (rouge, jaune, orange)

R10 : 22 kΩ (rouge, rouge, orange)

P : Potentiomètre 10 kΩ - linéaire + Bouton-fléché

DZ : Diode Zéner 8,2 V / 1,3 W

L1 : LED bleue Ø 3

L2 : LED verte Ø 3

L3 : LED rouge Ø 3

IC1 : LM 358 (2ampli-op)

IC2 : CD 4028 (décodeur BCD → décimal)

1 support 8 broches

1 support 16 broches

Bornier soudable 2 plots

I : Micro-switch (1 interrupteur)

MS : Micro-switch (3 interrupteurs)

Pile 9 volts

Coupleur pression

serve du côté composants. Si on ne dispose pas d'une imprimante Laser, il suffira de le reproduire sur papier, en utilisant une imprimante à jet d'encre mais dont on sélectionnera le mode d'impression " haute définition ", toujours en respectant le sens évoqué ci-dessus.

Par la suite, il conviendra de le reproduire sur le PCB Transfer film en utilisant un copieur Laser.

Bien entendu, il est tout à fait possible de se servir du dessin publié dans votre revue préférée, mais en partant de la vue côté composants. On utilisera ensuite un circuit époxy comportant une face cuivrée non pré-sensibilisée. Auparavant, il est nécessaire de bien la nettoyer à l'aide de laine métallique très fine (ou toile émeri très fine).

Après un lavage à l'eau savonneuse suivi d'un abondant rinçage, le module une fois sec est prêt à l'emploi.

Le PCB Transfer film est posé sur la face cuivrée, partie non lisse contre cuivre. Il est conseillé de poser sur la face lisse une feuille de papier.

On déplace ensuite doucement un fer à repasser posé sur le papier en prenant garde de ne pas décaler le film par rapport au cuivre. Le thermostat du fer est à régler sur la position " soie " ou " laine ". L'opération dure environ 3 à 4 minutes. Le circuit et le film qui y adhère seront ensuite refroidis sous un filet d'eau froide.

Le film se détache alors très facilement du cuivre. Seul le dessin du circuit imprimé reste sur la surface cuivrée.

La gravure s'effectue de façon classique dans un bain de perchlore de fer ou de persulfate d'ammonium.

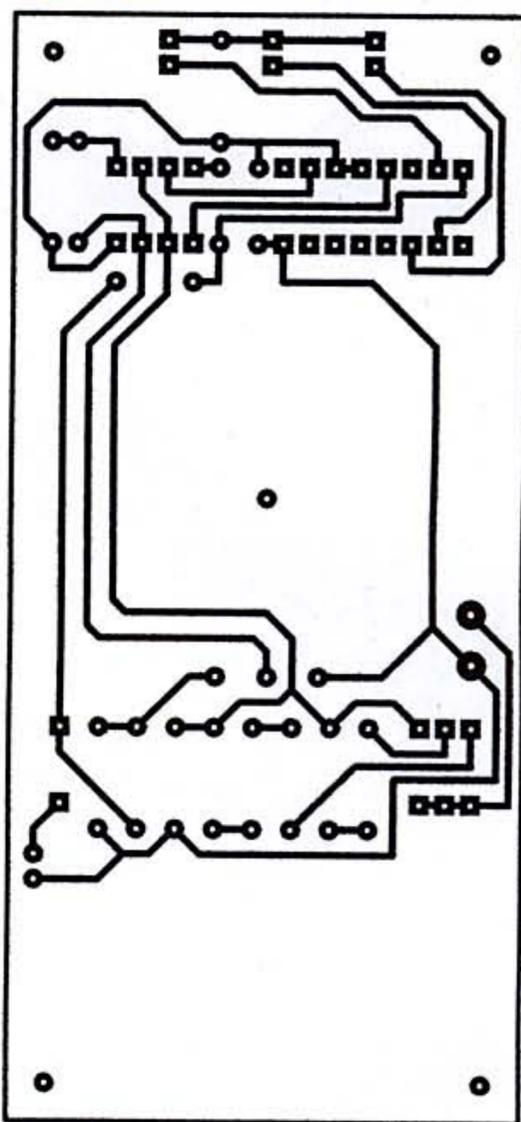
Graduation du voltmètre

La graduation est très simple. On tournera le bouton-fléché devant une plage circulaire en papier glacé et on y marquera les repères correspondant au nombre entier de volts (de 1 à 8) indiqué par un multimètre branché entre le " moins " de l'alimentation et la sortie-curseur du potentiomètre.

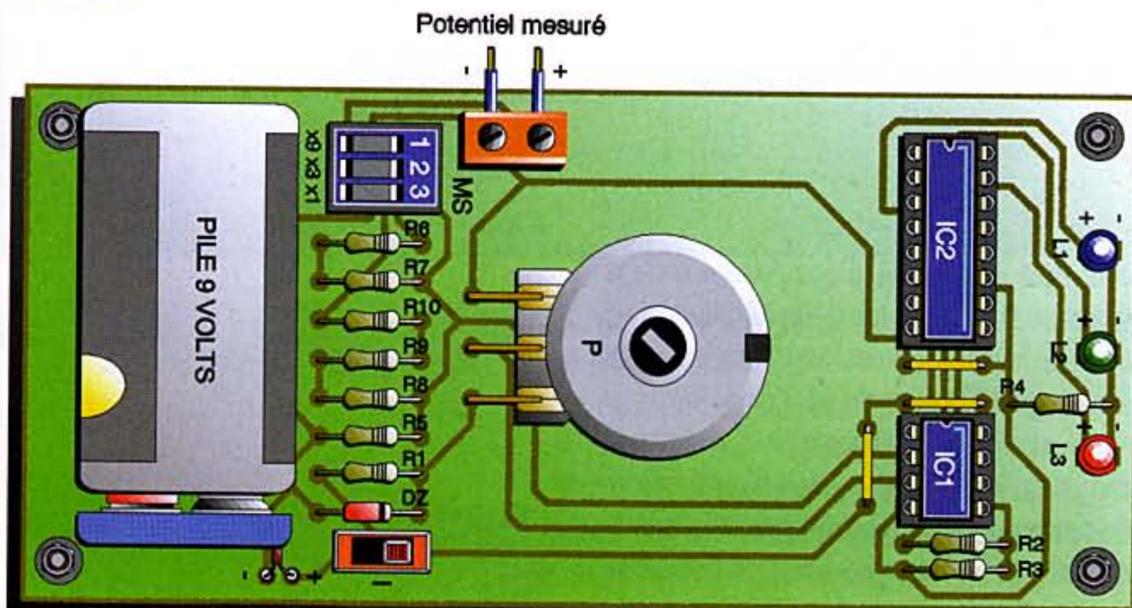
Par la suite, il sera encore possible de partager les espaces entre deux graduations consécutives en demi, voire en quart de volt.

R. KNOERR

2 Tracé du circuit imprimé

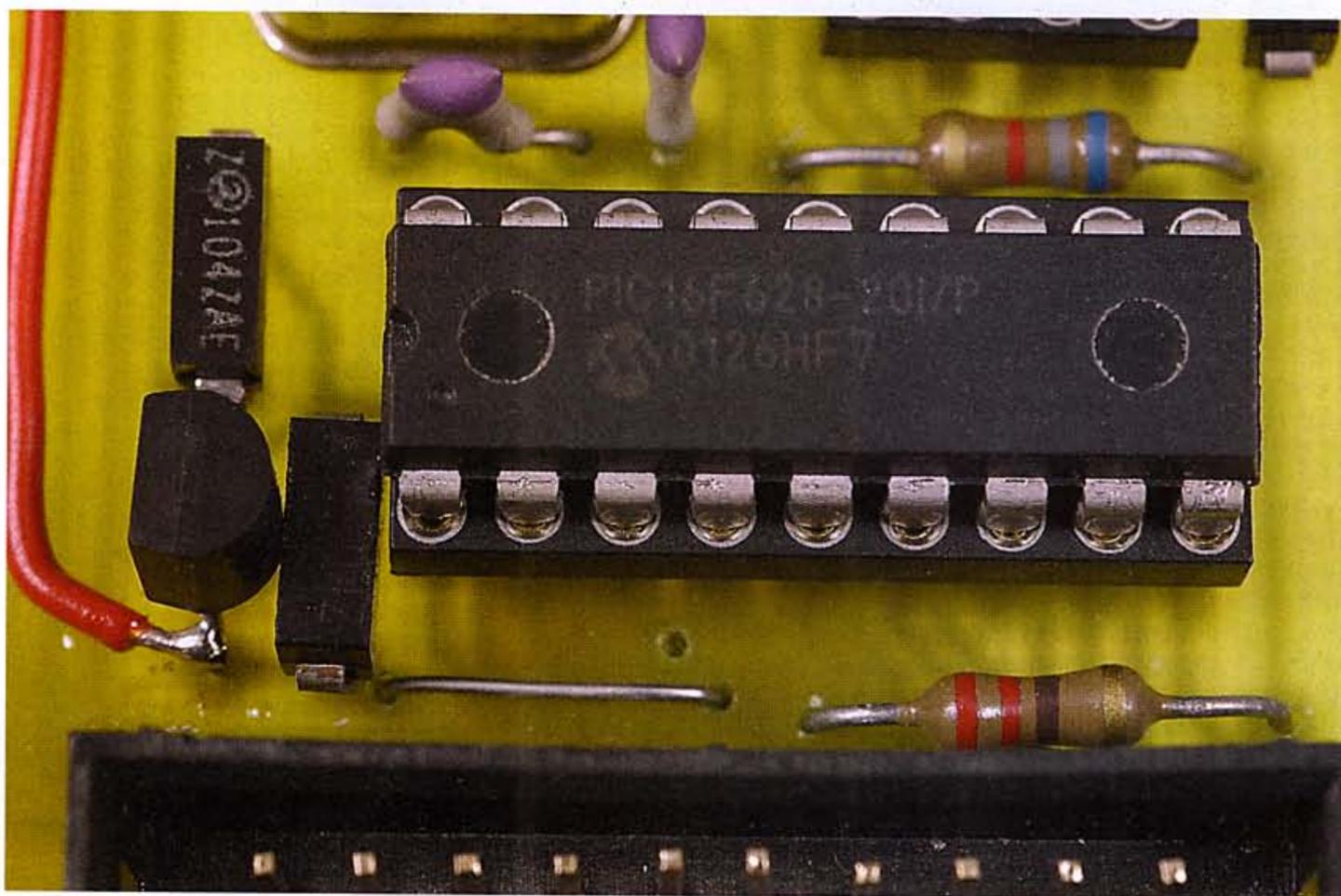


3 Implantation des éléments



À la découverte des microcontrôleurs PIC

(Sixième partie)



Dans ce numéro, nous allons mettre en œuvre les instructions que nous avons détaillées dans les leçons précédentes. Pour ce faire, nous utiliserons le logiciel MPLAB de MICROCHIP™. Celui-ci va nous permettre de réaliser un programme, de le compiler et de le simuler.

Le programme que nous allons analyser est très rudimentaire puisqu'il s'agit de faire tout simplement clignoter une led sur une des broches du pic déclarée en sortie. Ce programme sans prétention va nous permettre d'aborder de manière pédagogique la conception d'une application. Une fois ce programme détaillé, nous procéderons à l'étape suivante qui consistera à l'intégration sous MPLAB avec bien sûr des explications relatives à cet atelier logiciel.

Présentation du programme et analyse

Notre premier programme (**figure 1**) a pour but de faire clignoter une led. Avant de rentrer dans le source, nous allons aborder quelques directives nécessaires au compilateur ainsi que la façon d'écrire un programme. Notez

bien dans un premier temps que tout commentaire commence par un point virgule (;). Il est vivement conseillé de mettre des commentaires, pour vous permettre de comprendre et de reprendre le programme ultérieurement. Prenez donc le temps de réfléchir et choisissez des mots clés ainsi que des phrases explicites. Dans les commentaires, on retrouve souvent un titre avec une ou deux phrases sur ce qu'est censé faire notre programme ainsi que la date, la version du source, l'auteur et le PIC utilisé.

Quelques directives d'assemblage

On entend par directives d'assemblage des ordres qui seront exécutés par le compilateur lorsque vous demanderez de transformer votre programme source en un fichier binaire compré-

hensible par le microcontrôleur.

Une des premières directives indique au compilateur le type de PIC utilisé, cette information est représentée par la ligne : **list p=16f84A** (on a prévu un montage avec un pic 16f84A).

La deuxième directive présente dans notre programme est : **#include p16f84A.inc**, le #include signifie que le compilateur intégrera au programme le fichier qui suit (le fichier p16f84A.inc est fourni avec le logiciel MPLAB), ici dans notre cas, ce sera le fichier p16f84A (**figure 2**). Celui-ci définit les emplacements mémoire des registres internes ainsi que les valeurs des fusibles de configuration. C'est notamment grâce à ce fichier que l'on pourra utiliser dans notre programme des noms de registres (exemples: PORTA, TRISB, STATUS), ces noms figurent dans ce fichier de configuration.

La troisième directive est : **__config**

----- Exemple d'application avec un PIC : Un clignotant -----

```
; Titre : Clignotant
; Date : 01 OCTOBRE 2004
; Auteur : P.M
; PIC utilisé : PIC 16 F 84
; On réalise un clignotant sur la broche RB0 d'un PIC 16 F 84 le quartz est de 4
; Mhz , on effectue une temporisation environ égale à 0,2 seconde.
```



----- Directive d'assemblage pour MPLAB -----

```
list p=16f84A ; on indique ici le type de PIC utilisé
#include p16f84A.inc ;c'est le fichier qui contient la liste des adresses des registres internes
__config H'3FF9' ; définition des fusibles de configuration
```

----- Définition des constantes -----

```
#define inter 0 ; bouton marche
```

----- Définition des registres temporaires -----

```
retard1 EQU 0x0C ; le registre temporaire retard1 se trouve à l'adresse 0C
retard2 EQU 0x0F ; le registre temporaire retard2 se trouve à l'adresse 0F
```

----- Init des ports A et B -----

```
ORG 0 ; notre programme commencera à l'adresse 0 en mémoire

BSF STATUS,5 ; on met à 1 le 5ème bit du registre status pour accéder
; à la 2ème page mémoire ( pour trisa et trisb )

MOVLW 0x00 ; on met 00 dans le registre W
MOVWF TRISB ; on met 00 dans TRISB le port B est programmé en sortie

MOVLW 0x1F ; on met 1F dans le registre W
MOVWF TRISA ; on met 1F dans TRISA le port A est programmé en entrée

BCF STATUS,5 ; on remet à 0 le 5ème bit du registre status pour accéder
; à la 1ère page mémoire
```

----- raz de la led et du PORTB -----

```
CLRF PORTB ; RAZ du port B ( on éteint la led )
```

----- Programme principal -----

```
debut
btfss PORTA,inter ; interrupteur ( marche ) appuyé ? si oui, on continue sinon
goto debut ; on retourne à l'étiquette début

COMF PORTB,F ; On effectue le complément du PORTB
MOVLW 0xFF ; On charge le registre w avec la valeur FF (255)
MOVWF retard1 ; On copie le contenu de w (FF) dans le registre retard1
MOVWF retard2 ; On copie le contenu de w (FF) dans le registre retard2
CALL tempo ; on appelle la temporisation
GOTO debut ; retour au début du programme
```

----- Programme de temporisation -----

```
tempo
DECFSZ retard1,F ; on décrémente retard1 et on saute la prochaine instruction si
GOTO tempo ; le registre retard1 = 0 sinon retour à tempo

movlw 0xFF ; on recharge retard1
movwf retard1

DECFSZ retard2,F ; on décrémente retard2 et on saute la prochaine instruction si
GOTO tempo ; le registre retard2 = 0 sinon retour à tempo

RETURN ; retour au programme principal après l'instruction CALL
END
```

dans la variable `__config` donc il faut mettre 3FFF (hexadécimal) tel que précisé dans le fichier p16f84A.inc .

- Puis, si vous ne voulez pas utiliser la fonction power up timer (pwrtc est une temporisation de 72 ms activée au démarrage du PIC), il faut mettre `PWRTE_OFF` donc encore 3FFF dans la variable `__config`.

- Si vous ne voulez pas utiliser le watchdog (processus interne permettant de surveiller votre programme), il faut mettre `_WDT_OFF` donc 3FFB dans `__config`.

- Si vous utilisez un quartz de moyenne vitesse (4MHz), vous devez sélectionner l'oscillateur `_XT_OSC` donc on doit mettre 3FFD dans `__config`.

Quand nous faisons un "et" logique de tout cela, nous avons 3FFF & 3FFF & 3FFB & 3FFD et nous obtenons la valeur 3FF9

Au lieu de 3FF9, nous pouvons également écrire la ligne suivante :

```
__Config __CP_OFF & __WDT_OFF &
PWRTE_OFF & __XT_OSC
```

Ce qui est peut être un peu plus explicite...

Dans notre programme, nous utilisons également la ligne suivante : **#define inter,0** (cette ligne est optionnelle). Cela a pour intérêt de pouvoir utiliser (toujours dans un souci de clarté et de facilité de compréhension) le mot `inter` qui désigne comme son nom l'indique un interrupteur (bouton marche de notre application). Quand le compilateur, lors de la phase de compilation, rencontrera le mot `inter` dans le source, il le remplacera par 0, ainsi cela clarifie notre programme et le compilateur n'est pas perdu...

Si vous ne souhaitez pas utiliser de `#define`, il faudra alors travailler avec des valeurs dans votre programme. Ainsi, la ligne suivante :

BTFSS PORTA,inter

devra être remplacée par :

BTFSS PORTA, 0

et le fonctionnement sera identique.

Analyse du programme

Nous allons utiliser dans notre programme une subroutine (un sous-programme) de temporisation. Pour réaliser celle-ci, nous allons décrémente en cascade deux registres internes du PIC (registres 8 bits). Dans un premier temps, nous déclarons les deux registres que nous allons utiliser. Nous appellerons arbitrairement ces registres

1 Notre premier programme

H '3FF9'. Celle-ci va définir la position de certains fusibles qui gèrent les modes de fonctionnement souhaités pour l'application. Prenez le temps de regarder le fichier p16f84A.inc (**figure 2**) et notamment la

rubrique 'Configurations Bits' :

Si vous ne voulez pas utiliser le "code protect", ce bit qui interdit toute relecture de votre programme se trouvant dans la mémoire flash du pic, alors il faut placer `_CP_OFF`

```

W      EQU  H'0000'
F      EQU  H'0001'
----- Register Files -----
INDF   EQU  H'0000'
TMR0   EQU  H'0001'
PCL    EQU  H'0002'
STATUS EQU  H'0003'
FSR    EQU  H'0004'
PORTA  EQU  H'0005'
PORTB  EQU  H'0006'
EEDATA EQU  H'0008'
EEADR  EQU  H'0009'
PCLATH EQU  H'000A'
INTCON EQU  H'000B'
OPTION_REG EQU H'0081'
TRISA  EQU  H'0085'
TRISB  EQU  H'0086'
EECON1 EQU H'0088'
EECON2 EQU H'0089'

----- STATUS Bits -----
IRP    EQU  H'0007'
RP1    EQU  H'0006'
RP0    EQU  H'0005'
NOT_TO EQU  H'0004'
NOT_PD EQU  H'0003'
Z      EQU  H'0002'
DC     EQU  H'0001'
C      EQU  H'0000'

----- INTCON Bits -----
GIE    EQU  H'0007'
EEIE   EQU  H'0006'
T0IE   EQU  H'0005'
INTE   EQU  H'0004'
RBIE   EQU  H'0003'
T0IF   EQU  H'0002'
INTF   EQU  H'0001'
RBIF   EQU  H'0000'

----- OPTION_REG Bits -----
NOT_RBPU EQU H'0007'
INTEDG   EQU H'0006'
T0CS     EQU H'0005'
T0SE     EQU H'0004'
PSA      EQU H'0003'
PS2      EQU H'0002'
PS1      EQU H'0001'
PS0      EQU H'0000'

----- EECON1 Bits -----
EEIF    EQU  H'0004'
WRERR   EQU  H'0003'
WREN    EQU  H'0002'
WR      EQU  H'0001'
RD      EQU  H'0000'
    
```

à l'adresse 0, sachant que cette mémoire a une capacité de 1k (ou plus exactement de 1024 instructions, chaque instruction étant codée sur 14 bits). Dans notre exemple nous aurions pu loger notre programme à une autre adresse (adresse 100 par exemple), mais il y aura des cas (par exemple si l'on utilise les interruptions), où il faudra gérer correctement l'endroit où débute le programme. Sachant que pour les interruptions par exemple, l'adresse 4 est réservée ou plus

précisément à chaque fois qu'une interruption se produira, le compteur de programme (ou compteur ordinal) se chargera avec l'adresse 4, nous y reviendrons ultérieurement lorsque nous réaliserons un programme avec des interruptions (**figure 4** et **figure 5**).

Nous allons, dans la suite du programme, déclarer les broches du PIC (port) soit en entrée soit en sortie et ceci en plaçant une valeur dans les registres de direction TRISA (pour la configuration du port A) et TRISB (pour la configuration du port B). Si l'on regarde la **figure 3**, on s'aperçoit que les registres TRISA et TRISB sont dans la deuxième page mémoire.

L'instruction suivante **BSF STATUS,5** permet de passer à 1 le cinquième bit du registre STATUS (bit RP0). Ce bit, s'il est à 1, permet d'accéder à la deuxième page mémoire dans laquelle se situe justement les registres de direction des ports A et B, sur lesquels nous devons effectuer une opération de chargement.

Une fois la 2^{ème} page sélectionnée, la suite consiste à charger le registre de travail w avec la valeur indiquant la direction des ports (0 pour une sortie et 1 pour une entrée) puis à transférer le registre de travail w dans TRISA ou TRISB.

```

----- RAM Definition -----
_MAXRAM H'CF'
_BADRAM H'07', H'50'-H'7F', H'87'

----- Configuration Bits -----
_CP_ON    EQU  H'000F'
_CP_OFF   EQU  H'3FFF'
_PWRTE_ON EQU  H'3FF7'
_PWRTE_OFF EQU H'3FFF'
_WDT_ON   EQU  H'3FFF'
_WDT_OFF  EQU  H'3FFB'
_LP_OSC   EQU  H'3FFC'
_XT_OSC   EQU  H'3FFD'
_HS_OSC   EQU  H'3FFE'
_RC_OSC   EQU  H'3FFF'
    
```

2 Le fichier p16f84A.inc

"retard1" et "retard2" et nous indiquons l'adresse à laquelle chacun d'eux se situe dans le PIC. Sachant que les registres utilisateurs (ou variables mémoire) sont disponibles à partir de l'adresse 0C en hexadécimal (soit 12 en décimal) et jusqu'à l'adresse 4F en hexadécimal (soit 79 en décimal) ce qui représente 68 adresses donc 68 registres utilisables (**Figure 3**).

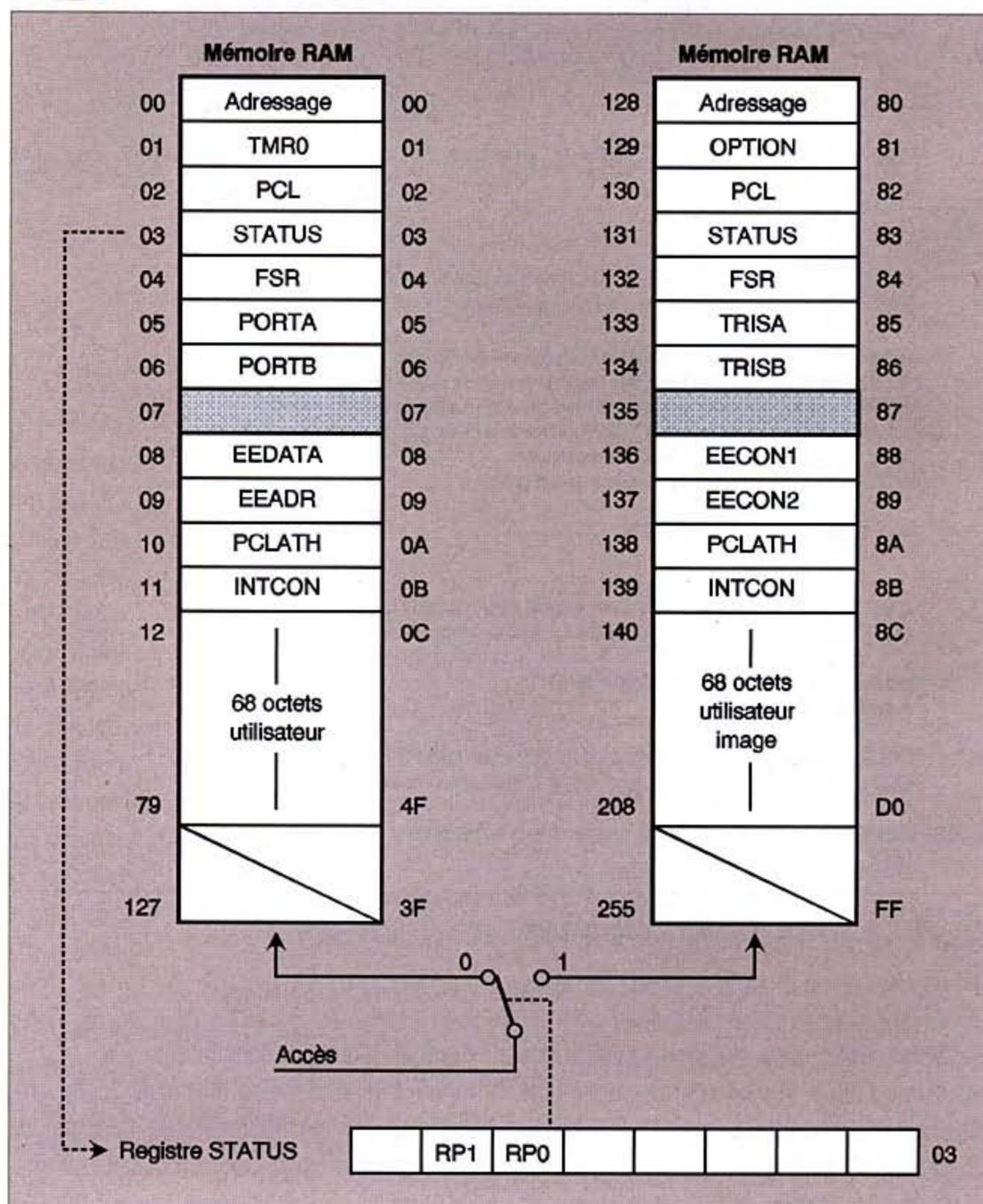
Les deux lignes suivantes permettent de déclarer ces deux registres, la directive EQU (comme équivalence) attribue une adresse mémoire au nom indiqué (retard1 sera à l'adresse 12 et retard2 sera à l'adresse 15). Vous pouvez choisir bien sûr d'autres adresses que celles choisies, en restant toutefois dans la plage autorisée (12 à 79). Dans l'écriture 0xC par exemple le "x" signifie que l'on est en hexadécimal.

```

Retard1 EQU 0xC
Retard2 EQU 0xF
    
```

Une fois les registres déclarés, nous avons une instruction du type ORG 0. Celle-ci indique au compilateur que le programme devra être logé dans la mémoire flash du PIC

3 Rappel du Mapping Mémoire



Address	Opcode	Hex	Machine	Symbolic
0000	1683	3000	0086	301F
0008	2807	0986	30FF	008C
0010	280F	30FF	008C	0B8F
0018	3FFF	3FFF	3FFF	3FFF
0020	3FFF	3FFF	3FFF	3FFF
0028	3FFF	3FFF	3FFF	3FFF
0030	3FFF	3FFF	3FFF	3FFF
0038	3FFF	3FFF	3FFF	3FFF
0040	3FFF	3FFF	3FFF	3FFF
0048	3FFF	3FFF	3FFF	3FFF
0050	3FFF	3FFF	3FFF	3FFF
0058	3FFF	3FFF	3FFF	3FFF
0060	3FFF	3FFF	3FFF	3FFF

4 Programme compilé à l'adresse 0 (org 0)

Address	Opcode	Hex	Machine	Symbolic
00C0	3FFF	3FFF	3FFF	3FFF
00C8	3FFF	3FFF	3FFF	3FFF
00D0	3FFF	3FFF	3FFF	3FFF
00D8	3FFF	3FFF	3FFF	3FFF
00E0	3FFF	3FFF	3FFF	3FFF
00E8	3FFF	3FFF	3FFF	3FFF
00F0	3FFF	3FFF	3FFF	3FFF
00F8	3FFF	3FFF	3FFF	3FFF
0100	1683	3000	0086	301F
0108	2907	0986	30FF	008C
0110	290F	30FF	008C	0B8F
0118	3FFF	3FFF	3FFF	3FFF
0120	3FFF	3FFF	3FFF	3FFF

5 Le même programme compilé à l'adresse 100 (org 100)

Pour le port B, on ne veut que des sorties donc TRISB vaudra 0 (en binaire 0000 0000 donc les huit broches RB0-RB7 sont en sortie)

```
MOVLW 0x00
MOVWF TRISB
```

Pour le port A, on ne veut que des entrées donc TRISA vaudra 1F (en binaire 0001 1111 donc les cinq broches RA0-RA4 sont en entrées)

```
MOVLW 0x1F
MOVWF TRISA
```

Puis on repasse le bit RPO à 0 pour revenir à la page mémoire 1 dans laquelle se trouvent notamment les port A et B sur lesquels nous devons travailler.

L'instruction suivante **BCF STATUS,5** permet de passer à 0 le cinquième bit du registre STATUS (bit RPO). Ce bit, s'il est à 0, permet d'accéder à la première page mémoire.

Ensuite vient la ligne **CLRF PORTB** qui a pour rôle de faire une RAZ (tous les bits à zéro) du port B. Cette instruction n'est pas obligatoire mais elle permet de commencer le programme de manière identique à chaque mise sous tension (on est sûr que la led connectée sur le port B sera éteinte).

Le début du programme

Nous avons choisi dans notre montage de mettre un bouton marche arrêt. Pour tester si ce bouton est en position "on" ou "off", nous allons utiliser les instructions suivantes :

```
debut
BTFSS PORTA,inter
Goto debut
```

Notez déjà dans un premier temps que les étiquettes (étiquette "début" dans notre exemple) sont toujours en début de ligne,

alors que les instructions sont décalées, il faudra toujours respecter cette règle.

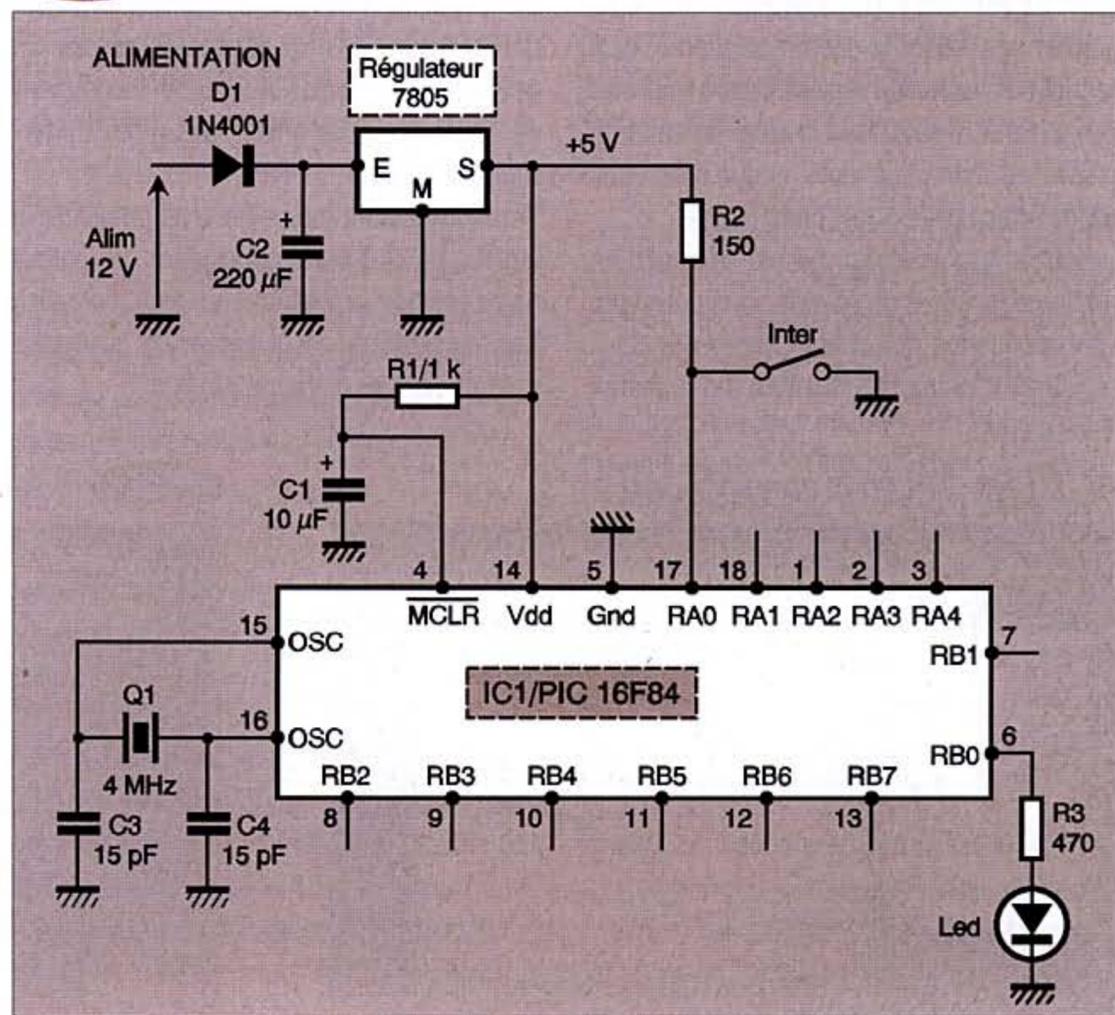
L'instruction **BTFSS PORTA,inter** va tester si le bit 0 (c'est la broche RA0 où est connecté l'interrupteur) du port A vaut 1 et dans ce cas sautera la prochaine instruction. Celle-ci, goto debut, fait un branchement inconditionnel à l'étiquette "début". En fait, on ne pourra commencer à exécuter le programme que si l'interrupteur est actionné (c'est exactement ce que l'on veut non?).

Si nous considérons que la led est éteinte, le but de notre programme étant de faire clignoter cette led, il va falloir l'allumer. Il y aurait plusieurs moyens d'allumer la led, (exemple **BSF PORTB,0** si la led est connec-

tée à RB0 **Figure 6**) l'instruction **COMF PORTB,F** fait un complément (on inverse la valeur de tous les bits 0=>1 et 1=>0) du PORTB. Sachant que celui-ci était dans notre initialisation à 0, il passe à FF (1111 1111) et la led s'allume (on peut donc câbler cette led sur n'importe quelle broche de sortie du port B RB0 à RB7 et cela fonctionnera). La prochaine fois que le programme passera par cette instruction, il fera l'inverse (tous les bits à zéro) et ainsi de suite.

Si nous en restions là et que l'on rebouclait le programme, cela fonctionnerait, mais il faudrait être "superman" pour voir la led clignoter car la fréquence serait trop élevée pour notre pauvre rétine...

6 Schéma de principe



Vous l'avez donc compris, il va falloir réaliser une temporisation.

Dans un premier temps, nous allons charger des valeurs dans deux registres (retard1 et retard2) puis nous les décrémenterons. Notez que si il y a deux registres, c'est que la fréquence serait encore trop élevée même si on utilisait un seul registre.

Pour charger les valeurs, nous passons dans un premier temps par le registre de travail w, celui-ci distribuera ensuite ces valeurs vers les registres retard1 et retard2.

L'instruction **MOVLW 0xFF** charge le contenu de w avec la valeur ff (255).

Puis la copie dans les registres se fait par les instructions **MOVWF retard1** et **MOVWF retard2**. Les deux registres sont maintenant chargés, nous pouvons faire appel à la temporisation : l'instruction **Call tempo** fait un branchement à l'étiquette tempo.

L'instruction qui suit le Call tempo est **Goto debut**, cette instruction permet de reboucler le programme à l'étiquette début, endroit où l'on teste l'état de l'interrupteur marche-arrêt.

Programme de temporisation

Nous commençons dans ce sous-programme par décrémenter le registre retard1 qui était chargé précédemment (avant l'appel de la temporisation) à 255 (FF). L'instruction **DECFSZ retard1,F (décrément f skip if 0)** permet de réaliser la décrémentation et ceci tant que la valeur du registre retard1 ne vaut pas 0. Si le registre vaut 0 alors on saute la prochaine instruction (goto tempo) donc nous allons effectuer 255 fois cette boucle.

Une fois cette première boucle effectuée on recharge la valeur de retard1, ceci pour faire une imbrication des décomptages des deux registres. L'instruction **MOVLW 0xFF** charge

le registre w avec 255 (FF) et l'instruction **MOVWF retard1** copie le contenu de w dans retard1.

Ensuite, nous allons décrémenter le registre retard2 avec l'instruction **DECFSZ retard2,F (décrément f skip if 0)**. Tout comme pour le registre retard1, nous allons effectuer 255 fois cette opération. Remarquez qu'à chaque fois que l'on fait retard2 - 1 on retourne à l'étiquette tempo donc on repasse par la décrémentation du registre retard1. Ceci implique qu'en fait on réalise 256 x 255 décomptages du registre retard1 soit 65280 ce qui au total nous donnera une temporisation de 0,2 seconde environ.

Détails simplifiés du calcul :

Toutes les instructions dans la boucle prennent 1 cycle d'horloge excepté l'instruction goto (2 cycles). Pour le premier registre que l'on décrémente 256 x 255, on a un temps machine de $1\mu s \times 65280 \times 3$ tops (1top pour DECFSZ et 2 tops pour goto) soit environ 196 ms (avec un quartz de 4 MHz sachant que l'horloge est divisée par 4 en interne, donc un top d'horloge = 1 μs). Pour le reste de la boucle, on passera 255 fois avec 5 tops d'horloges (2 pour goto et 3 pour MOVLW, MOVWF et DECFSZ) donc $5 \times 255 = 1,275$ ms. Au total, on aura donc 196 ms + 1,275 ms \approx 0,2 seconde

Vient ensuite la dernière instruction, l'instruction **Return**. Celle-ci permet de revenir au programme principal et le cycle recommence... tant que vous n'avez pas remis l'interrupteur sur arrêt.

Il est maintenant temps de vous présenter le logiciel MPLAB qui va permettre de réaliser notre premier programme et de le compiler. Les vues d'écran sont issues de la version 6.61 de MPLAB (octobre 2004).

La première chose à faire est bien sûr de télécharger cette version gratuite de MPLAB sur le site de Microchip (www.microchip.com).

Une fois le fichier MP661.zip (attention 38 Mo) correspondant à MPLAB version 6.61 téléchargé et installé, il vous reste à lancer l'exécutable.

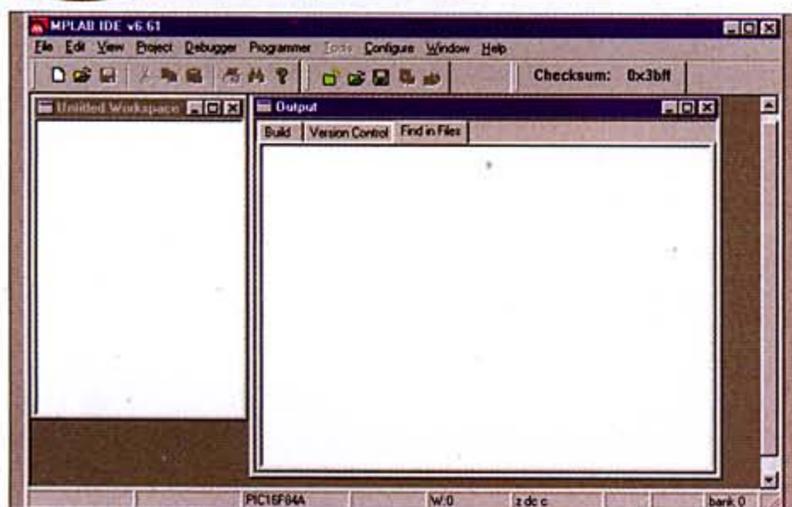
La **figure 7** donne le premier écran que vous devez obtenir.

La version 6.61 intègre un fichier d'aide à la mise en route (un WIZARD) qui est bien pratique. Il est à noter que l'on peut bien sûr créer un projet sans utiliser le Wizard. Pour lancer le wizard, cliquez dans le menu "Project" puis "Projet Wizard". Cliquez sur "suivant" puis ensuite sélectionnez le type de PIC utilisé, pour nous, ce sera le PIC 16F84A (**figure 8**), une fois sélectionné, cliquez sur "suivant". Sélectionnez le langage tel que représenté sur la **figure 9**. Ensuite, il vous reste à donner un nom à votre projet (par exemple "clignotant") et d'indiquer à quel endroit les fichiers doivent être stockés sur le disque dur de votre ordinateur (par exemple c:\clignotant - **Figure 10**).

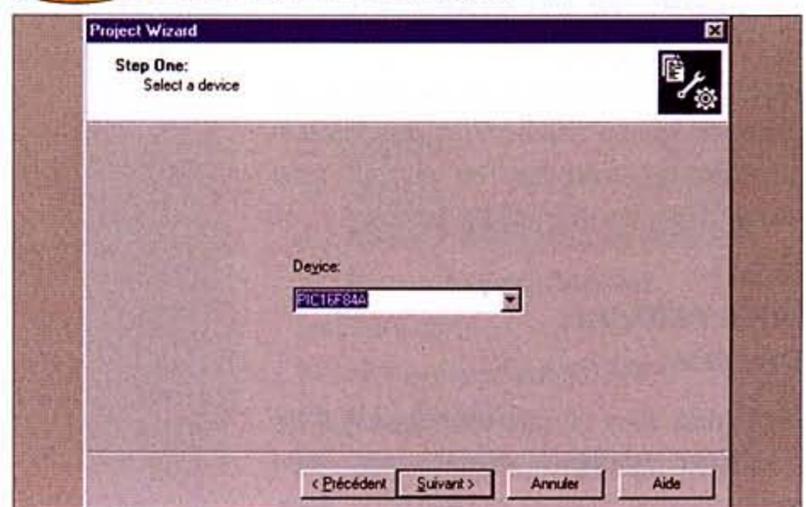
Ensuite, le Wizard vous propose d'ajouter un fichier assembleur à votre projet, si vous avez par exemple téléchargé le fichier "clignotant.asm" présent sur le site de la revue (www.electroniquepratique.com) ou sur le site de l'auteur (voir en fin d'article), vous pouvez alors le joindre *en faisant bien attention de valider la case à cocher présente sur la partie droite de la fenêtre* permettant de copier le fichier sélectionné dans le répertoire que vous avez défini (**figure 11**), sinon cliquez sur suivant (vous pouvez rajouter ou créer ce fichier assembleur plus tard)

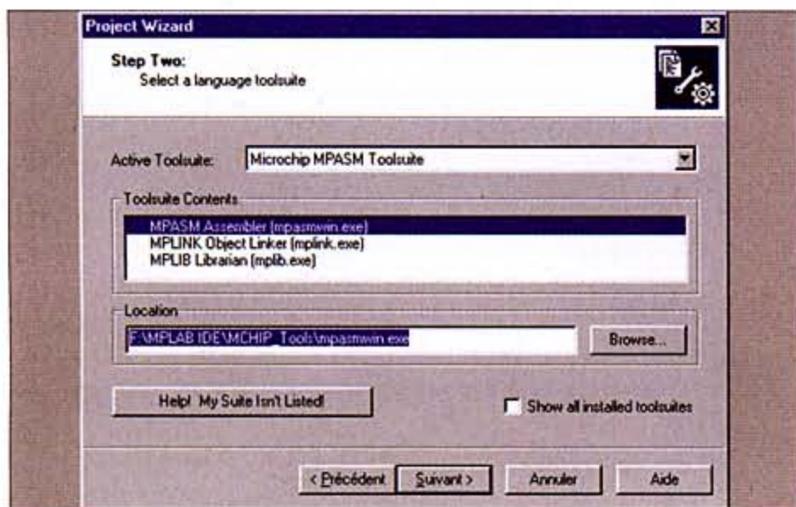
Ensuite ? appuyez sur le bouton "terminer". C'est la fin du Wizard qui peut paraître un peu long mais avec l'expérience cela ira beaucoup plus vite par la suite.

7 MPLAB le premier écran

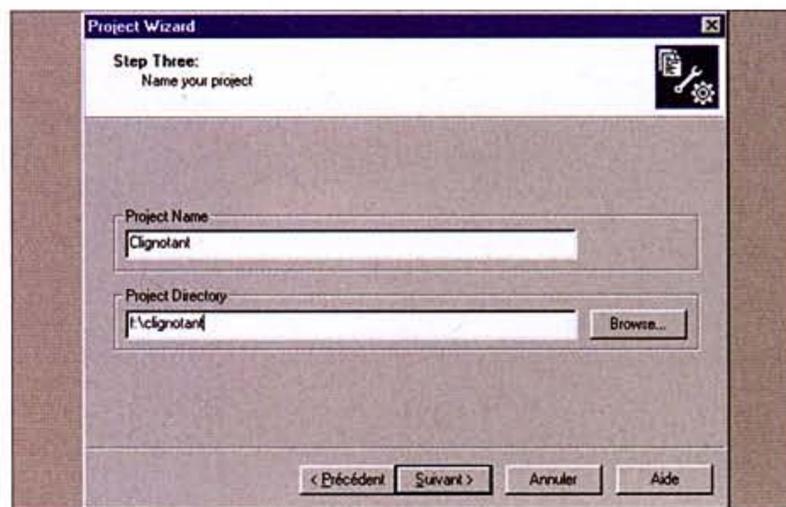


8 Sélection du PIC à utiliser

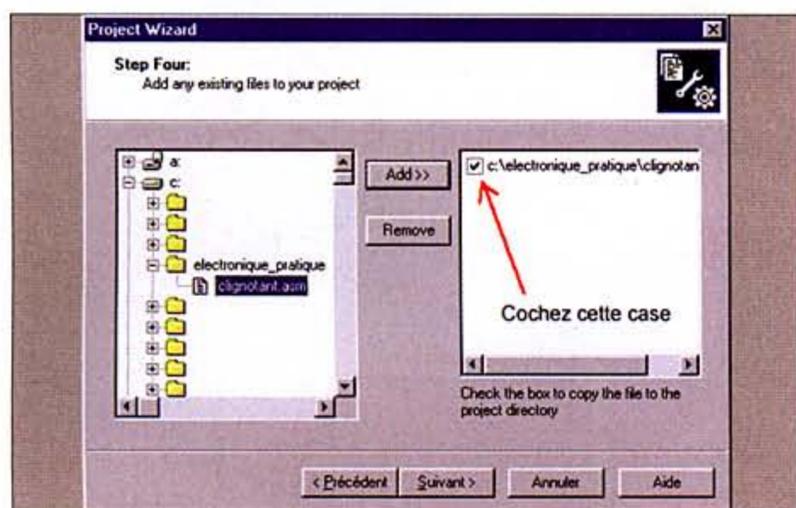




9 Sélection du langage



10 Un nom pour le projet

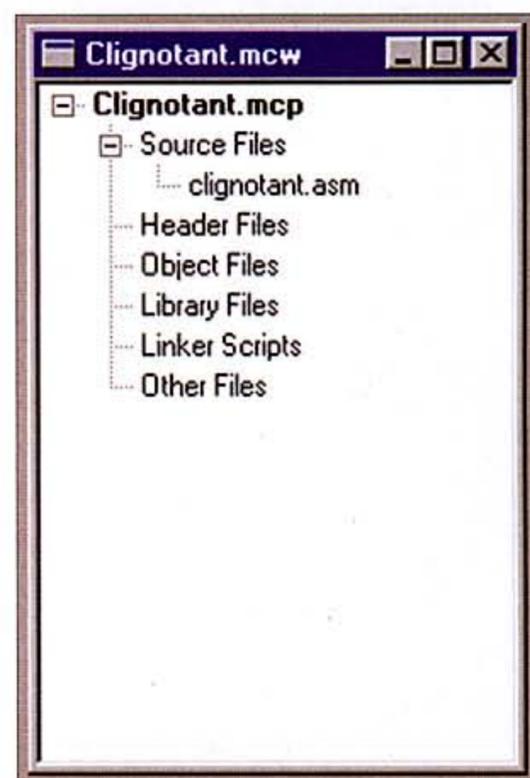
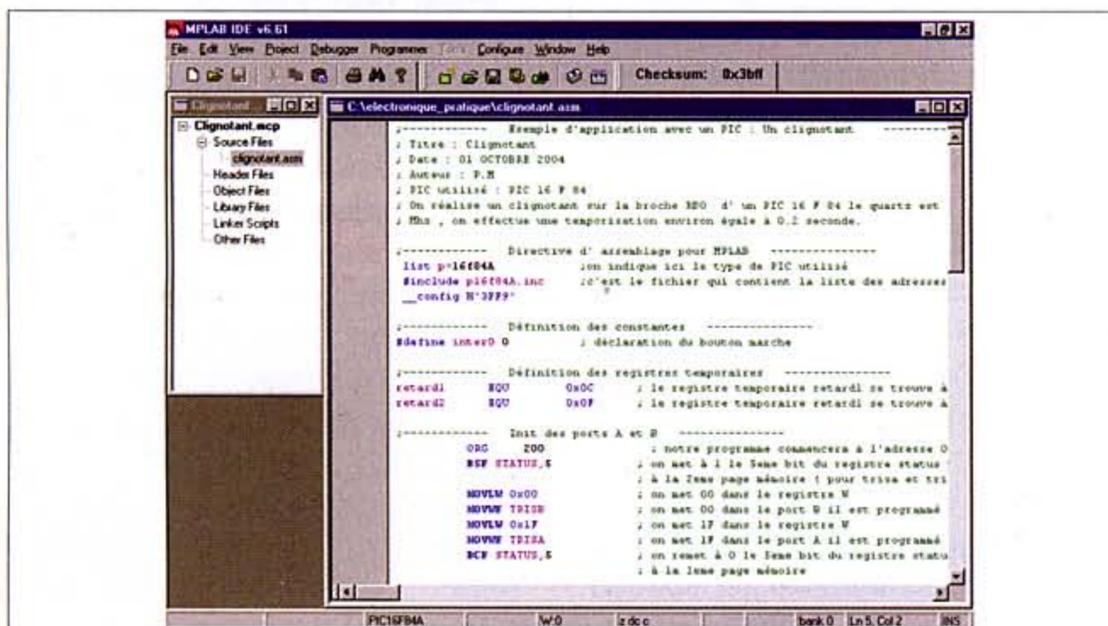


11 Insertion d'un fichier .ASM

Si vous avez respecté les vues précédentes, voici ce que vous devriez obtenir en **figure 12**. Dans le cas où vous avez téléchargé le fichier `clignotant.asm`, double-cliquez sur le nom de ce fichier dans la fenêtre représentée **figure 12**, vous devez alors voir la fenêtre de l'éditeur avec le source (**figure 13**). Si vous n'avez pas téléchargé le fichier `clignotant.asm`, il vous suffit d'aller dans le menu "File" puis "New", une nouvelle fenêtre apparaît alors à l'écran. Tapez votre source

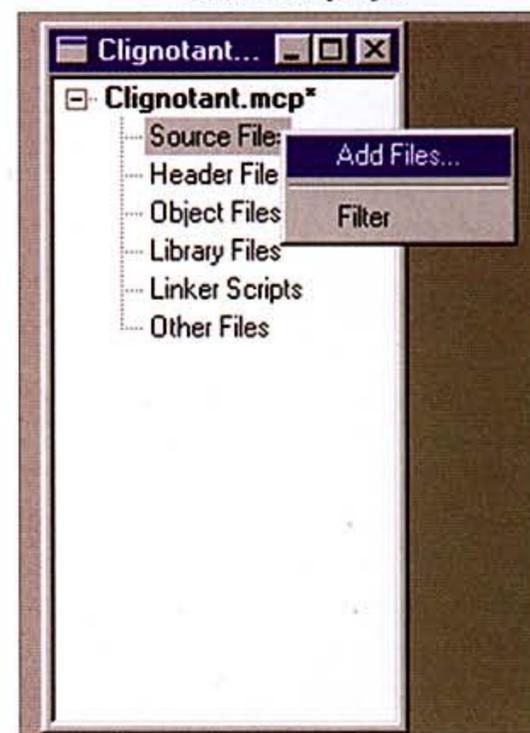
tel que représenté **figure 1**, puis, depuis le menu "File", cliquez "Save as...". Enregistrez ce fichier sous "clignotant.asm" dans la directory (répertoire) que vous avez déclarée dans le Wizard (exemple `c:\clignotant`). Une fois enregistré, faire un click droit sur le texte "Source Files" (**figure 14**) et sélectionnez "Add Files" puis le fichier "clignotant.asm" que vous venez de créer. Une fois ces actions réalisées, vous devez alors avoir la fenêtre représentée en **figure 12**

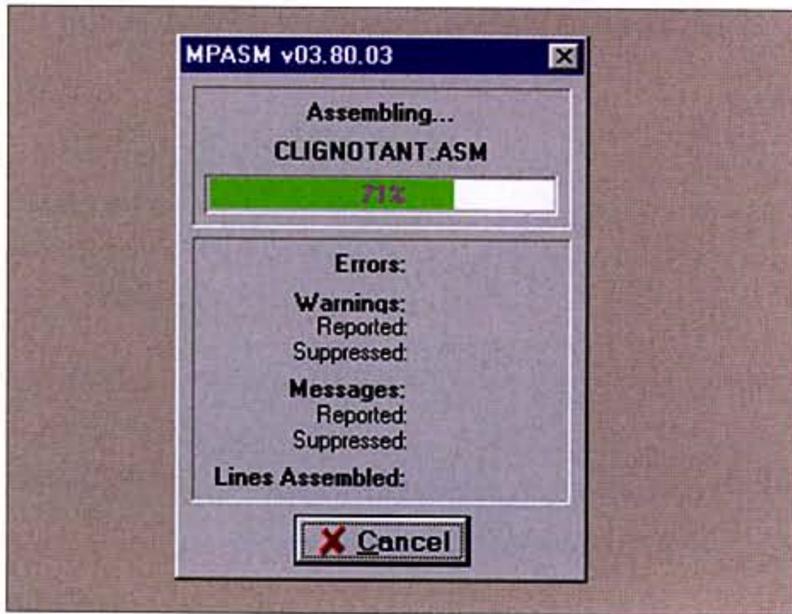
13 Le fichier source



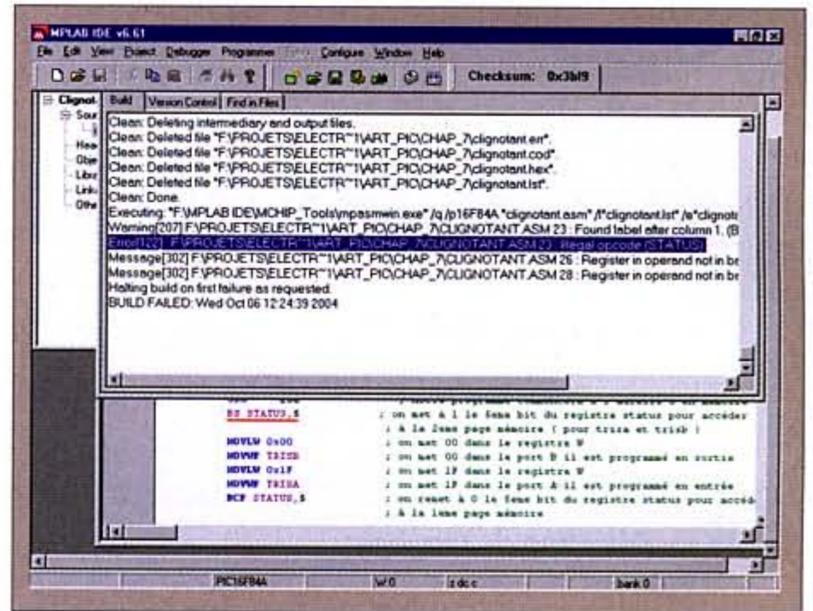
12 Le projet avec le fichier asm

14 Pour ajouter un fichier source au projet

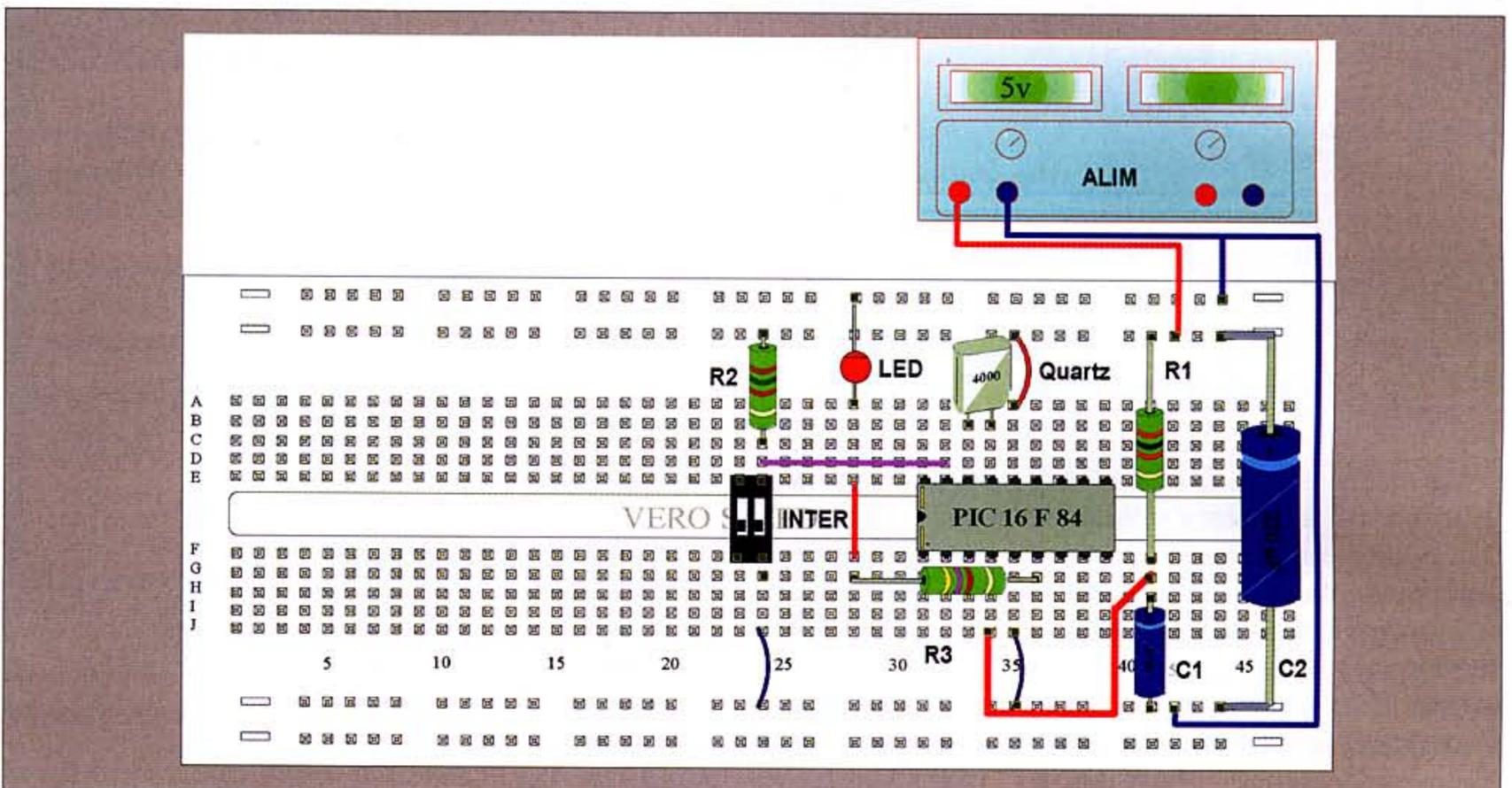




15 La fenêtre de compilation



16 L'erreur détectée par MPLAB



17 Le montage réalisé à l'aide d'une platine d'essai type LABDEC

puis, si vous double-cliquez sur le nom "clignotant.asm", être au même point que précisé sur la **figure 13**.

Pour compiler le source

Nous voici arrivés à la fin de cet article et nous allons conclure avec la compilation de notre programme.

Si cette compilation a bien fonctionné, un fichier nommé "clignotant.hex" sera créé dans le répertoire que vous avez indiqué dans le Wizard (c:\clignotant par exemple), ce fichier .hex correspond au fichier binaire qu'il faudra transférer vers la mémoire du PIC.

Pour ce faire, il faudra un programmeur et un logiciel de transfert. Pour les lecteurs possédant un programmeur reconnu par le logiciel (exemple le PICSTART PLUS de Microchip), le transfert se fera depuis MPLAB.

Pour la compilation, allez dans le menu "project" puis "Built All". Une fenêtre de compilation (**figure 15**) apparaît alors, celle-ci vous indique le pourcentage de compilation. Si des erreurs sont détectées, alors une fenêtre de debug vous indique le numéro de la ligne ou il y a une erreur afin que vous la corrigiez (exemple **figure 16** ou l'on a volontairement enlevé le "F" de l'instruction BSF STATUS,5 ligne 23).

Pour conclure ce chapitre

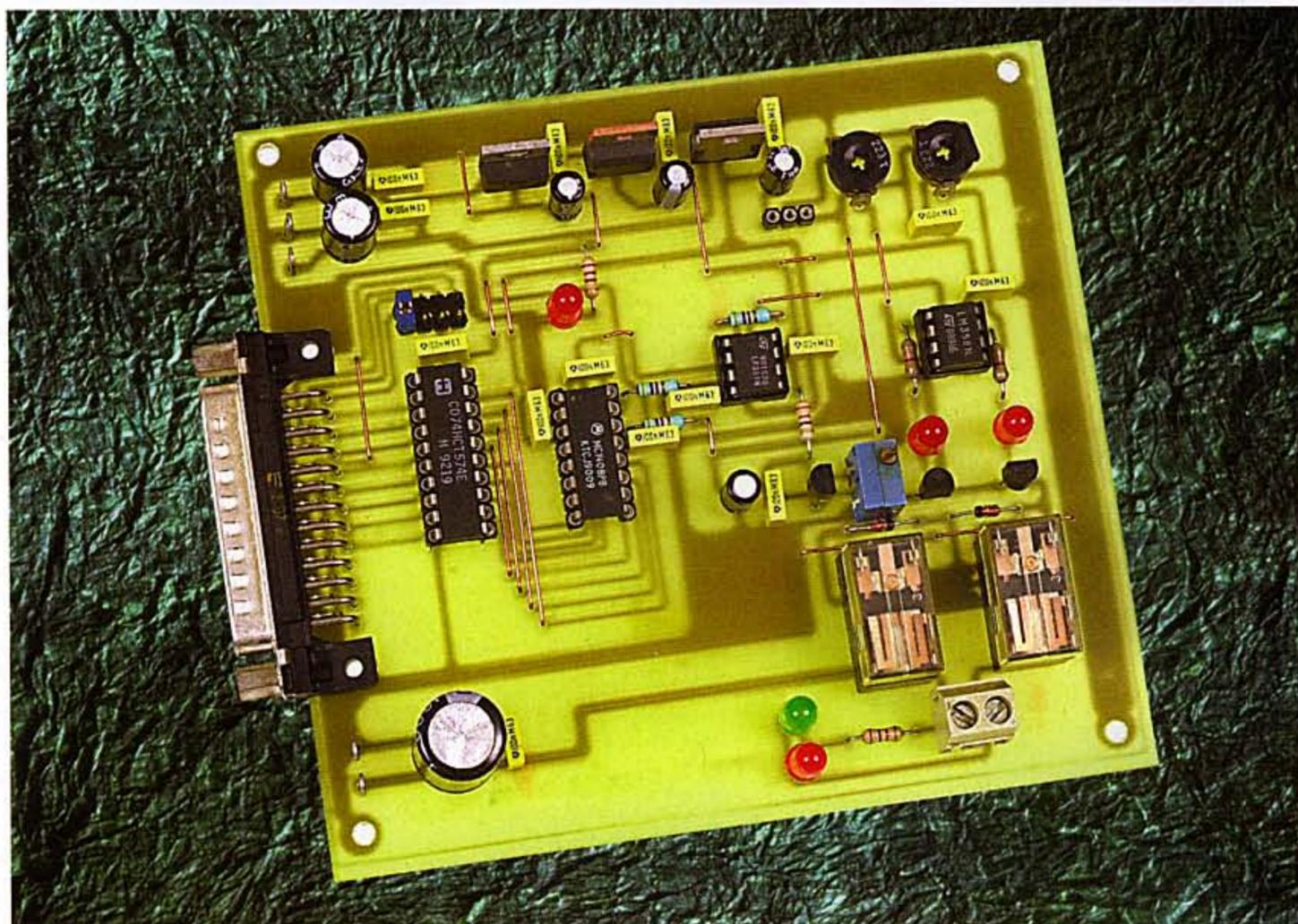
Avec cette sixième partie, nous avons analysé et compilé notre premier programme sous MPLAB. Dans le prochain numéro, nous aborderons les aspects "programmeur" afin de pouvoir transférer le fichier binaire vers le PIC ainsi que la partie simulation.

En attendant de réaliser une platine pour tester nos futurs programmes et pour les lecteurs désireux de réaliser le clignotant présenté dans cet article, voici **figure 17** le montage réalisé avec une platine de type LABDEC.

P. MAYEUX

<http://perso.libertysurf.fr/p.may>

Pratique des interfaces pour PC



Malgré la quasi disparition des connecteurs ISA dans les ordinateurs récents, de nombreux utilisateurs possèdent encore des machines d'avant dernière génération qui en sont équipées. Cela est très pratique lorsqu'il s'agit de concevoir des interfaces d'entrées/sorties permettant de nombreuses applications. C'est la théorie de ces interfaces que nous vous proposons d'aborder maintenant en se basant sur des schémas théoriques de cartes PC dont le fonctionnement a été testé de nombreuses fois.

Le décodage des adresses d'entrées/sorties

Tout comme les périphériques internes du PC qui possèdent une adresse fixe (disques durs, lecteurs de CD, lecteurs de disquettes, vidéo, etc.), les cartes additionnelles d'entrées/sorties possèdent également un domaine d'adresses qui permet de les utiliser.

Ces adresses sont situées entre 300h et 31Fh (le " h " signifie que ces adresses sont données en écriture hexadécimale).

Afin de pouvoir décoder ces adresses et envoyer des données ou en recevoir, le microprocesseur utilise plusieurs lignes :

- 16 lignes d'adresses, A0 à A15

- la ligne de lecture dans le domaine des entrées/sorties, IORD/

- la ligne d'écriture dans le domaine des entrées/sorties, IOWR/

- 8 lignes de données, D0 à D7

- la ligne RESET qui permet une remise à zéro de cartes utilisant certains circuits intégrés comme le PPI 8255 (port de 24 lignes d'entrées/sorties).

Les lignes IORD/, IOWR/ et RESET sont actives au niveau logique bas.

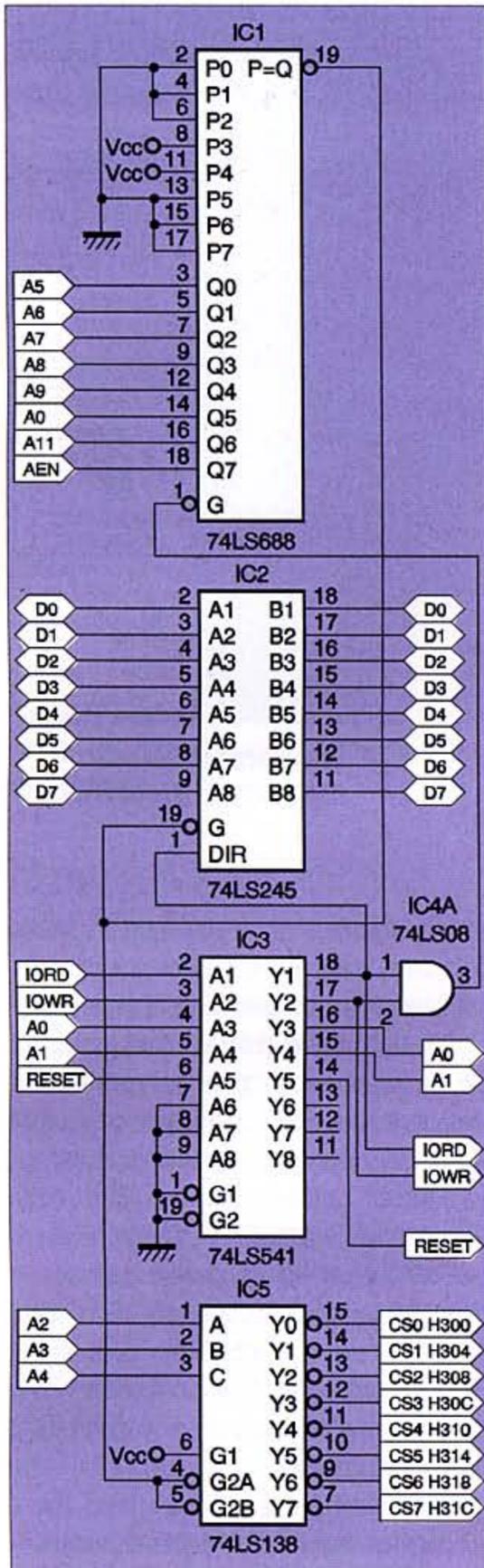
Abordons maintenant le premier schéma. Celui-ci est donné en **figure 1** et constitue le montage de base utilisé par de nombreuses cartes d'interfaces. Ce montage permet le contrôle des adresses ainsi que le sens de transfert des données (entrée ou sortie).

Ce montage utilise cinq circuits intégrés des plus classiques. Le décodage des adresses utilise un 74LS688 et un 74LS138.

Le premier est un comparateur qui possède deux ports de huit bits en entrées. Le port P0 à P7 voit ses lignes connectées à la masse sauf les entrées P3 et P4.

Le second port reçoit sur ses entrées Q0 à Q7 les signaux véhiculés par les lignes d'adresses A5 à A11, ainsi que la ligne AEN (Adress ENable ou validation d'adresse).

Le fonctionnement est alors le suivant : lorsque l'adresse envoyée par le microprocesseur sur les entrées Q correspond à la configuration des broches P, soit toutes les lignes d'adresses au niveau logique bas sauf



1 Schéma de la carte de base de décodage des adresses

A8 et A9 au niveau logique haut, alors la broche P=Q passe au niveau logique bas et valide ainsi le circuit décodeur suivant. Il faut signaler que le 74LS688 est validé lui-même (broche G) soit par un signal d'écriture, soit par un signal de lecture. On voit donc que le circuit réagit à l'adresse 300h. Mais on constate également qu'il réagira de la même manière à d'autres adresses comme le montre le tableau de la **figure 2**. Un second décodeur est donc nécessaire, le 74LS138 qui détermine les huit groupes d'adresses

A11	A10	A9	A8	A7	A6	A5	-	A4	A3	A2	A1	A0	Adresses
0	0	1	1	0	0	0		0	0	0	0	0	300h, sortie Y0
0	0	1	1	0	0	0		0	0	1	0	0	304h, sortie Y1
0	0	1	1	0	0	0		0	1	0	0	0	308h, sortie Y2
0	0	1	1	0	0	0		0	1	1	0	0	30Ch, sortie Y3
0	0	1	1	0	0	0		1	0	0	0	0	310h, sortie T4
0	0	1	1	0	0	0		1	0	1	0	0	314h, sortie Y5
0	0	1	1	0	0	0		1	1	0	0	0	318h, sortie Y6
0	0	1	1	0	0	0		1	1	1	0	0	31Ch, sortie Y7

2 Adresses disponibles dans le domaine E/S

principaux. Pour cela, le circuit, un décodeur 3 vers 8, reçoit sur ses entrées les signaux des lignes d'adresses A2, A3 et A4.

On peut voir dans le tableau que le décodeur 74LS138 détermine huit domaines d'adresses. En utilisant les lignes d'adresses A0 et A1, on dispose dans chacun de ces huit domaines de quatre adresses, ce qui nous donne au total 32 adresses simples.

Trois circuits intégrés supplémentaires sont utilisés :

- un octuple buffer, le 74LS541, qui permet de tamponner les lignes A0, A1, IORD/, IOWR/ et RESET/

- une porte AND contenue dans un 74LS08 qui permet, par combinaison des lignes IORD/ et IOWR/, d'obtenir le signal de validation du comparateur 74LS688

- un octuple buffer bidirectionnel, le 74LS245 qui, par validation de sa broche DIR au moyen du signal IORD/, permet de déterminer le sens de passage des données (entrée ou sortie)

En résumé, la carte fonctionne alors ainsi :

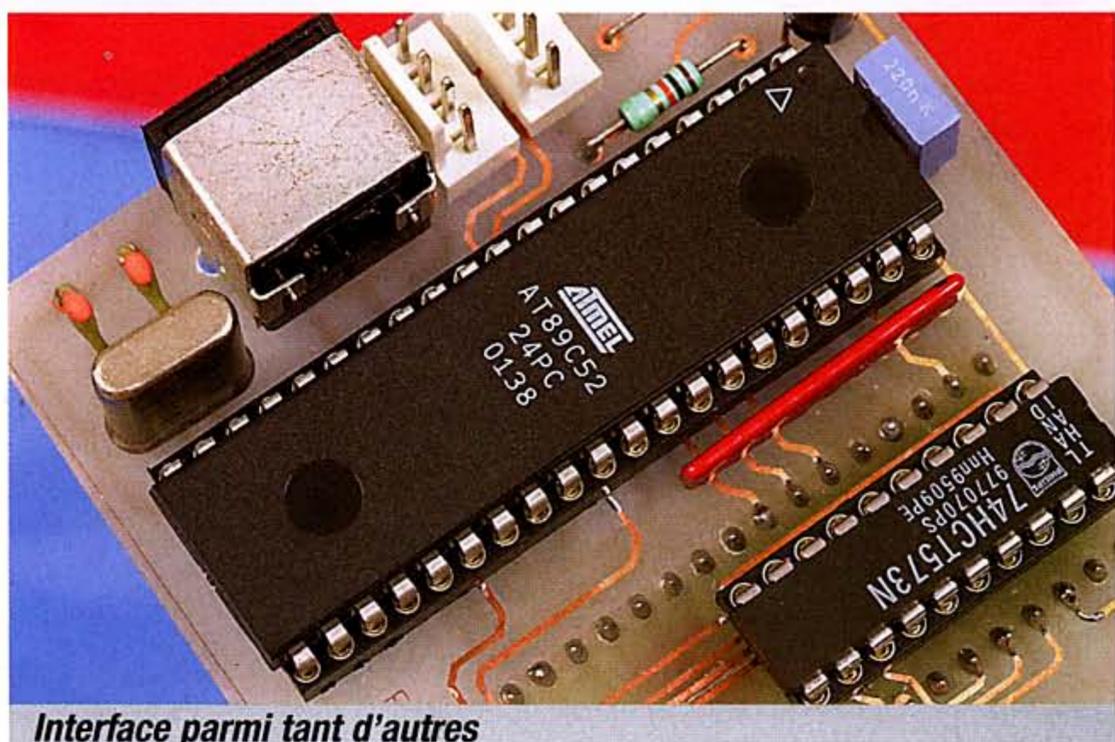
- une adresse est envoyée sur le bus par le microprocesseur qui valide également sa ligne AEN

- la donnée est envoyée ou reçue et est validée par les lignes IORD/ ou IOWR/. Le programme de gestion de la carte peut alors traiter la donnée

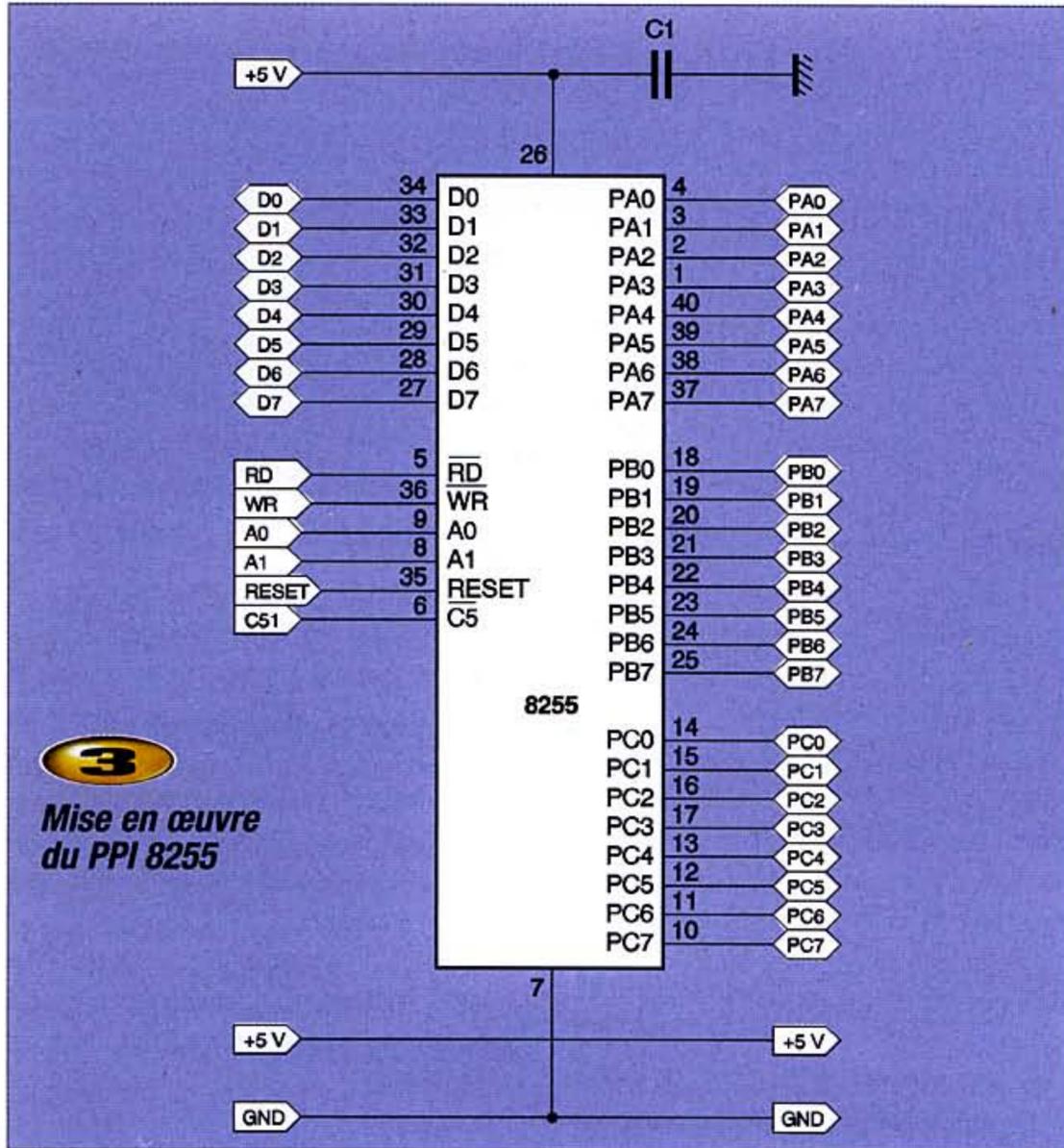
Le procédé de décodage des adresses étant élucidé, nous pouvons maintenant passer à l'étude des circuits permettant diverses applications.

Système d'entrées/sorties numériques

Les dispositifs d'entrées/sorties numériques (ou tout ou rien) sont parmi les plus prisés par l'amateur. Ils permettent toutes sortes de commandes dans de nombreux domaines : domotique, alarmes, modélisme ferroviaire, etc. Le schéma théorique que nous propo-

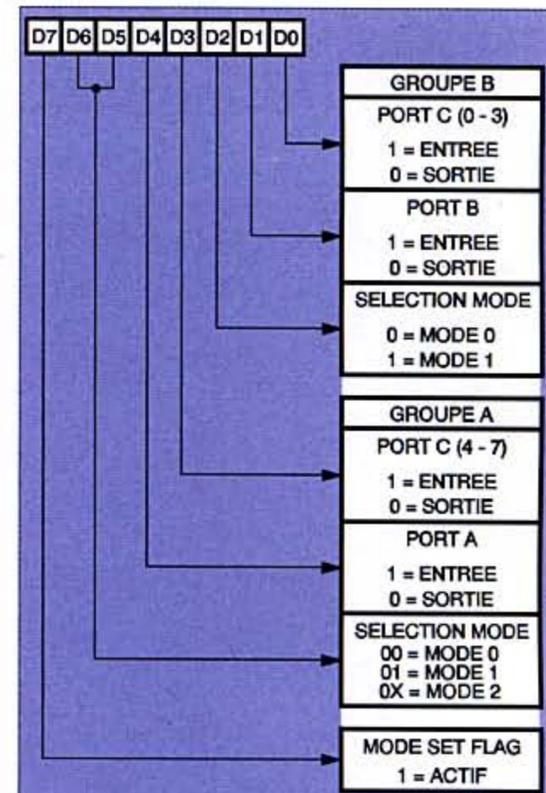
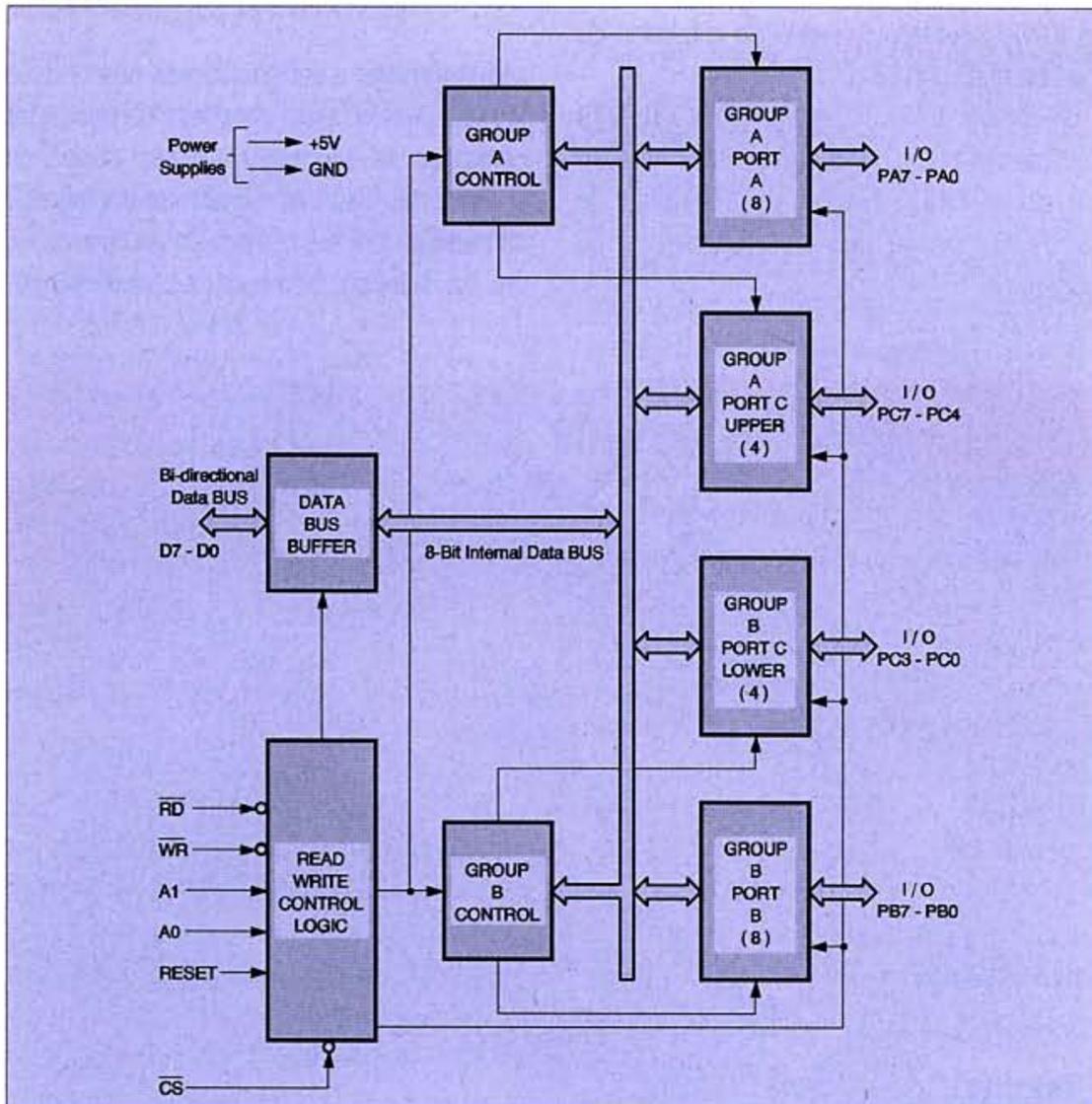


Interface parmi tant d'autres



3 Mise en œuvre du PPI 8255

4 Structure interne du PPI 8255



5 Programmation du registre de contrôle

sons en **figure 3** est un classique et l'un des plus simples qui soit. Il n'utilise que deux circuits intégrés identiques et permet de disposer de 48 lignes en entrées ou en sorties. Ce circuit, comme tous ceux que nous décrivons, doit être, bien évidemment, précédé de la carte de décodage étudiée auparavant.

Le circuit utilisé est le PPI 8255 (Programmable Peripheral Interface) conçu par Intel. C'est un composant déjà ancien mais toujours commercialisé car aucun autre circuit ne l'a encore supplanté.

Ses caractéristiques principales sont :

- 24 lignes d'entrées/sorties (3 ports de 8 bits) programmables
- compatible TTL
- il dispose de plusieurs modes de programmation :

mode 0 : programmation de deux groupes de 12 entrées/sorties par groupe de quatre

mode 1 : programmation de deux groupes de 8 entrées/sorties et quatre lignes pour les signaux d'échanges.

mode 2 : programmation d'un bus bidirectionnel de huit lignes et cinq lignes utilisées pour les signaux d'échanges.

C'est dans le mode 0 que nous utiliserons la carte, mode permettant de disposer de 24 lignes d'entrées/sorties. Le dessin donné en **figure 4** montre la structure interne du 8255. Outre les 24 lignes d'entrées/sorties et les deux entrées d'alimentation, le 8255 dispose de broches permettant sa programmation :

- un bus de huit lignes de données (D0 à D7)

- une ligne WR/ validant l'écriture d'une donnée, active au niveau logique 0.
- une ligne RD/ validant la lecture d'une donnée, active au niveau logique 0.
- une ligne CS/ validant le fonctionnement du circuit, active au niveau logique 0.
- deux lignes A0 et A1 permettant la sélection des trois ports et le registre de contrôle.

adresse 0 → port A
 adresse 1 → port B
 adresse 2 → port C
 adresse 3 → port de contrôle

- une ligne RESET qui, lorsqu'un niveau haut lui est appliqué, réinitialise le circuit et positionne les trois ports en entrées.

Afin de configurer le sens de transfert des données dans les trois ports, il convient d'écrire une donnée dans le port de contrôle. Comme on le voit sur le schéma de l'interface, les lignes RD/ et WR/ sont respectivement reliées aux mêmes lignes du bus PC, ainsi que les lignes d'adresses A0 et A1.

Ces dernières permettent d'adresser l'un des quatre ports des PPI8255. Les lignes CS/ sont, quant à elles, connectées à deux des sorties du décodeur 74LS138.

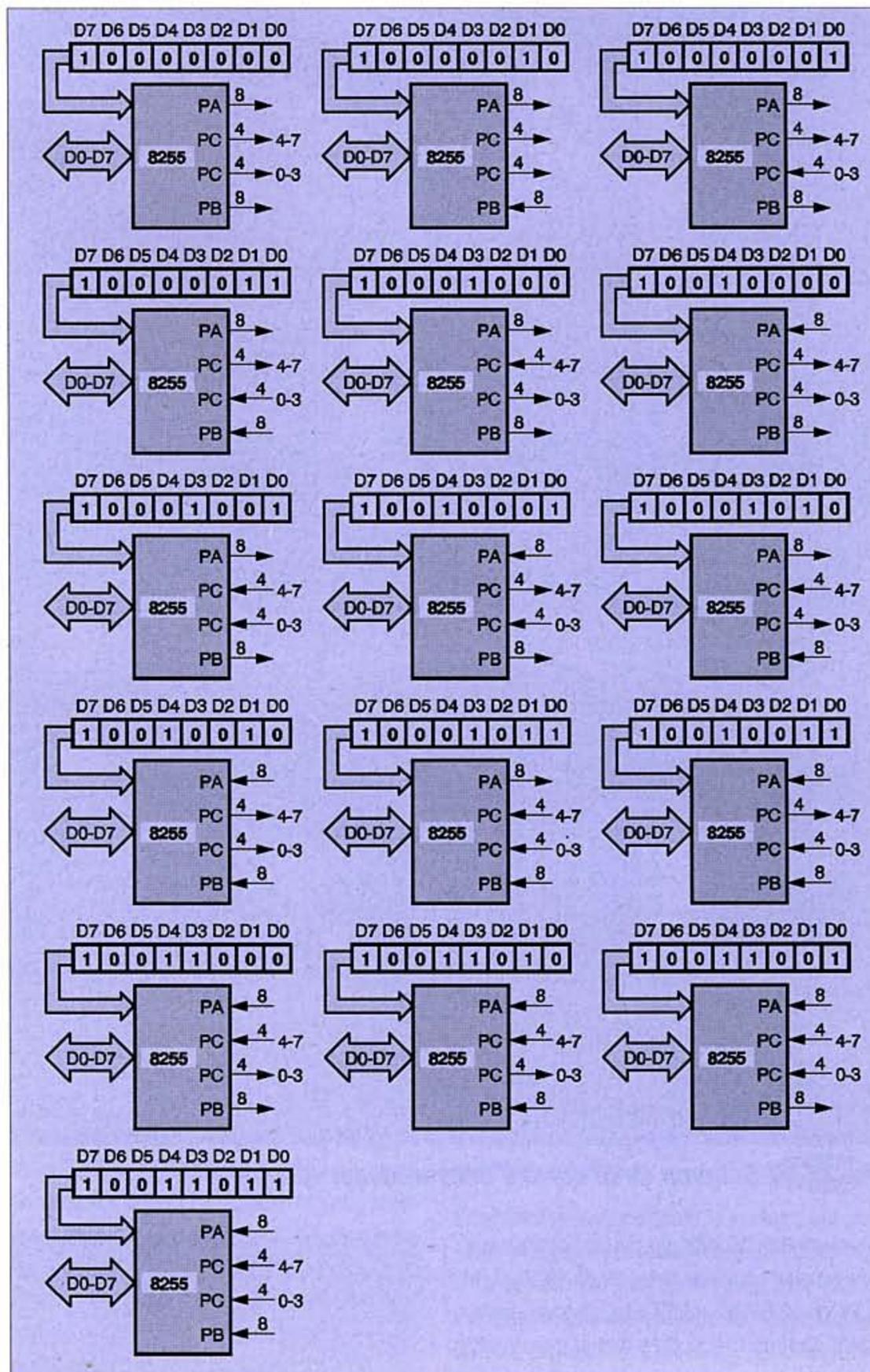
On peut ainsi choisir à quelles adresses seront situés les deux circuits 8255. Le port de contrôle nécessite, pour la configuration des 8255, une donnée codée sur huit bits.

Le dessin de la **figure 5** donne la fonction de chacun des bits. Ainsi, comme représenté en **figure 6**, chacun des trois ports A, B et C peut dans le mode 0, être configuré en entrées ou en sorties.

Par exemple, si nous souhaitons positionner les trois ports en sorties, le bit 7 (drapeau) est actif (niveau logique 1), le bit D2 est à 0 (sélection du mode 0) et les bits D0, D1, D3 et D4 sont à 0 (positionnement des ports en sorties).

En langage Quick Basic (le plus simple), cela se traduit par les instructions suivantes, en admettant que les deux PPI 8255 soient situés respectivement aux adresses 300h et 304h (lignes CS/) :

- OUT &H303,128 : REM Ecriture dans le registre de contrôle du premier 8255**
- OUT &H300,0 : Mise à 0 des huit lignes du port A**
- OUT &H301,0 : Mise à 0 des huit lignes du port B**
- OUT &H302,0 : Mise à 0 des huit lignes du port C**
- OUT &H307,128 : REM Ecriture dans le registre de contrôle du second 8255**



6 Configurations des 3 ports dans le mode 0

OUT &H304,0 : Mise à 0 des huit lignes du port A

OUT &H305,0 : Mise à 0 des huit lignes du port B

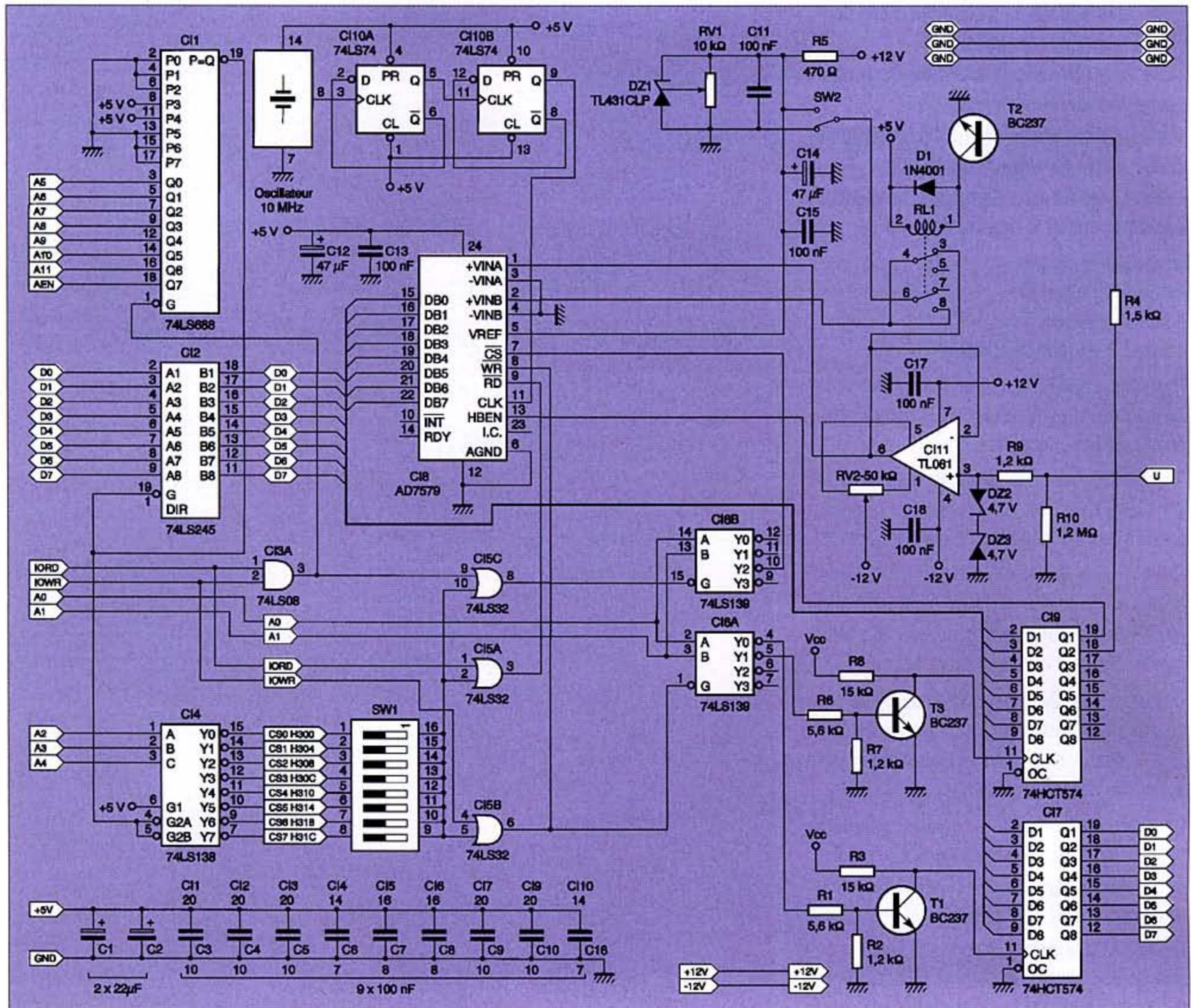
OUT &H306,0 : Mise à 0 des huit lignes du port C

Signalons qu'il est également possible de piloter un PPI 8255 au moyen de l'interface parallèle du PC (port imprimante) si l'on ne désire pas placer une carte dans son PC ou si l'on ne possède pas de port ISA. Dans ce cas, les quatre lignes de contrôle du port parallèle génèrent les signaux appliqués aux broches RD/, WR/, A0 et A1. La ligne CS/ est,

dans ce cas précis, reliée perpétuellement à la masse. La programmation du port ayant été expliquée dans un numéro précédent d'Electronique Pratique, nous ne reviendrons pas sur le mode à employer.

La mesure de grandeurs physiques

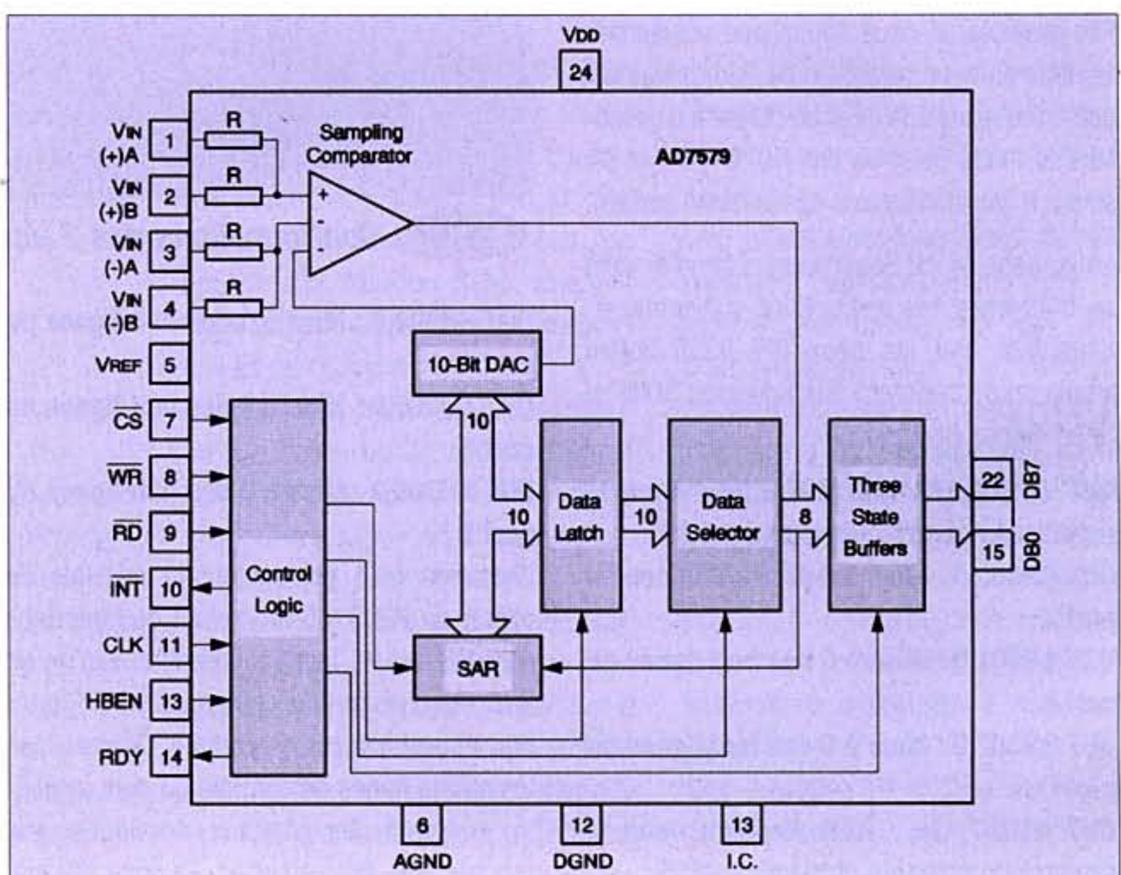
Il est intéressant de disposer d'une carte permettant de lire une tension continue avec une bonne résolution. Le schéma proposé en **figure 7** est celui d'une carte à convertisseur



7 Schéma de la carte à convertisseur A/D

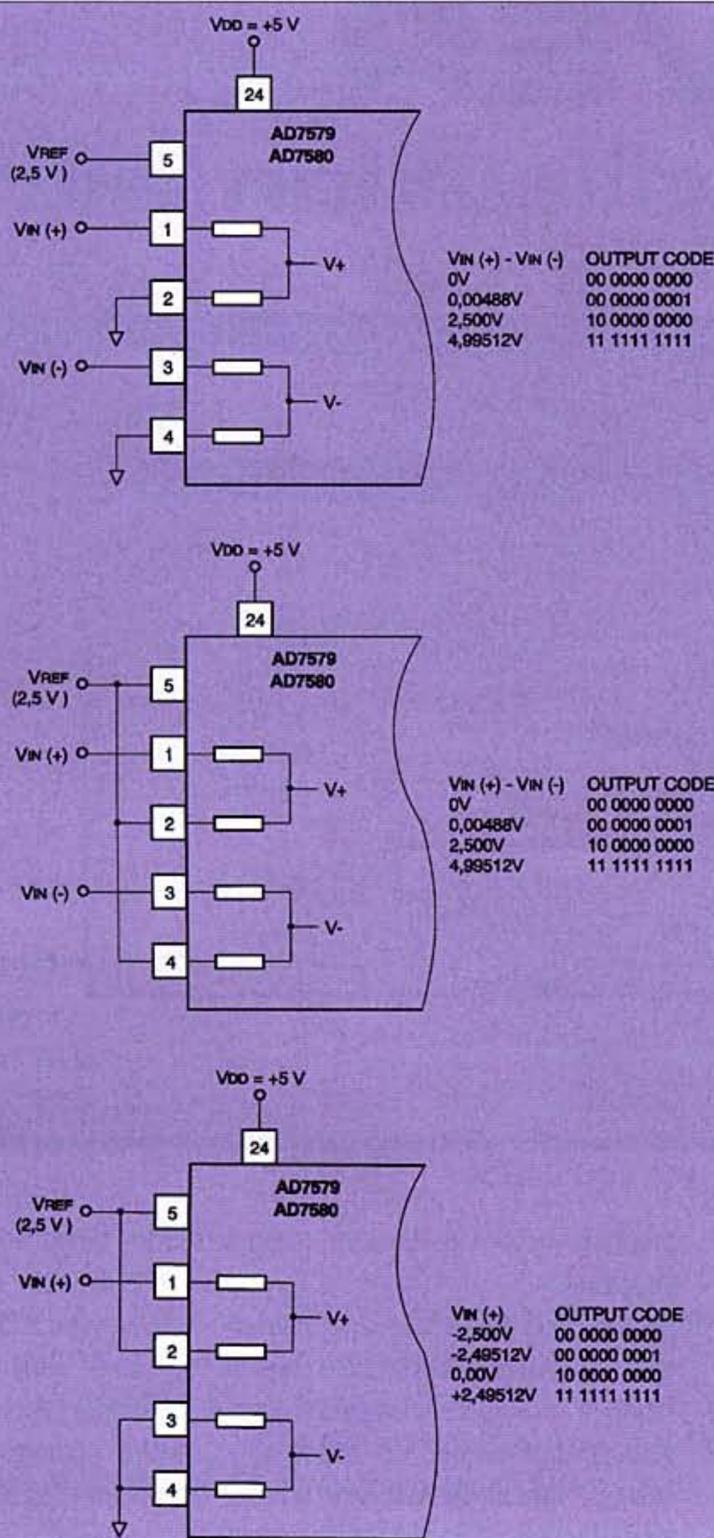
de résolution 10 bits qui permet de mesurer une tension comprise entre 0V et +2,5V, 0V et +5V ou -2,5V et +2,5V. Elle dispose également d'un port de sorties numériques 8 bits permettant de sélectionner la source à mesurer ou la commutation de systèmes divers en fonction des résultats des mesures. Le convertisseur utilisé est un AD7579. Son schéma interne est donné en **figure 8**. C'est un convertisseur analogique/numérique 10 bits. Il fonctionne selon le procédé d'approximations successives et 20µs lui sont nécessaires pour la réalisation d'une mesure. Son horloge fonctionne à une fréquence de 2,5MHz et sa vitesse d'échantillonnage est de 50kHz ce qui permet la mesure d'un signal de fréquence maximum de 25kHz.

Structure interne du convertisseur AD7579 **8**



9

Configurations des entrées de l'AD7579



Une particularité doit être signalée : le circuit AD7579 ne disposant que d'un bus de données de huit bits alors qu'il donne un résultat sur dix bits, une fonction supplémentaire lui a été adjointe.

La broche HBEN (High Bits ENable), lorsqu'elle est soumise à un niveau logique bas, permet de lire les huit bits de poids faible. Lorsqu'elle est soumise à un niveau haut, les deux bits de poids fort peuvent être lus. Une autre particularité du circuit est qu'il possède des entrées différentielles qui peuvent être utilisées si on le désire.

Les dessins donnés en **figure 9** montrent les différentes configurations. Les résistances internes forment des atténuateurs.

Notre carte n'utilise pas l'entrée différentielle mais les trois possibilités de mesures de tensions unipolaires ou bipolaires. C'est le relais incorporé sur la carte et le commutateur SW2 qui permettent de choisir la gamme de mesures :

- lorsque le relais est au repos, les deux entrées non inverseuses sont mises en parallèle et la gamme est de 0V à +2,5V.

- lorsque le relais est en position travail, l'entrée +Vin B est mise à la masse et l'atténuateur est alors en fonction.

La gamme s'étend de 0V à +5V (position du commutateur SW2).

- lorsque le relais est en état de travail et le commutateur SW2 positionné vers la tension de référence, la gamme s'étend de -2,5V à +2,5V.

On distingue sur le schéma un second décodeur double dont les entrées sont reliées aux lignes d'adresses A0 et A1. Cette façon de procéder permet de disposer de trois adresses secondaires dans l'un des domaines d'adresses principaux comme nous l'avons vu plus haut.

Comme on le voit, trois adresses sont nécessaires pour le fonctionnement de la carte : 300h, 301h et 302h. Les deux premières permettent de piloter les deux octuples bascules 74LS574 et la troisième de lire le résultat de la conversion.

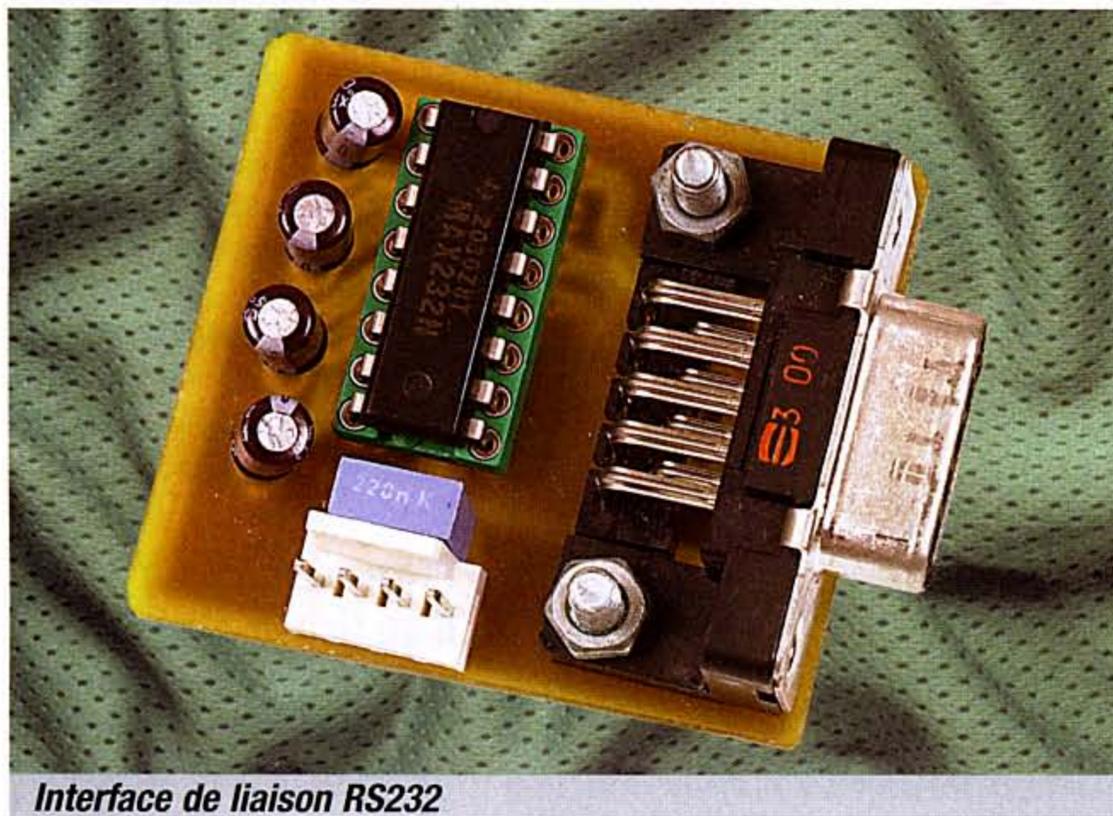
Nous espérons que cette approche des interfaces PC répondra aux questions que se posent nos lecteurs. Nous aurons l'occasion d'y revenir en proposant, cette fois, des réalisations pratiques qui ne manqueront pas de susciter l'intérêt des passionnés d'informatique.

P. OGUIC

patrice.oguic@tiscali.fr

Bibliographie : Interfaces PC (ETSF, Dunod)

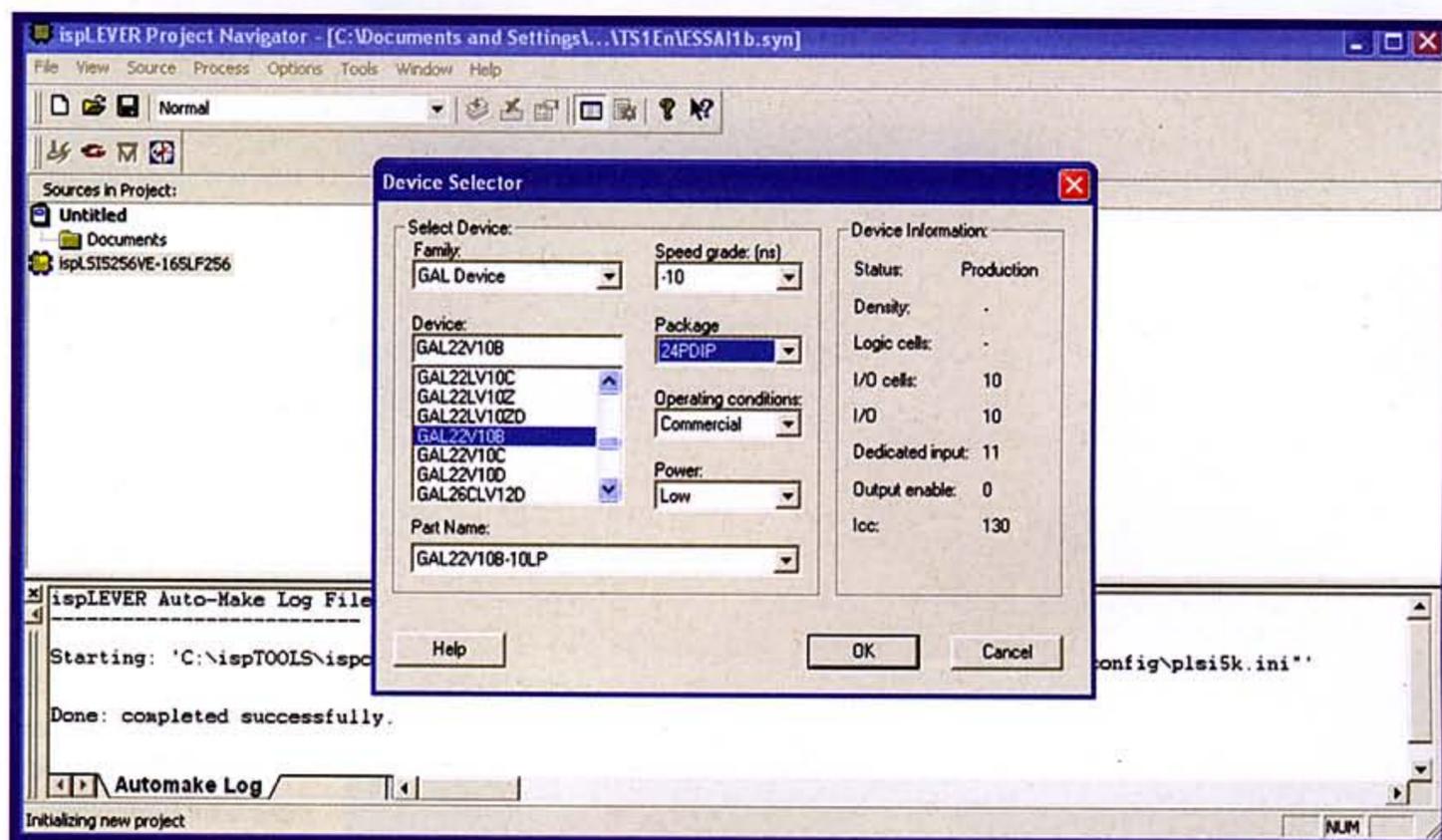
Mesures et PC (ETSF, Dunod)



Interface de liaison RS232

IspLEVER

Manuel d'utilisation simplifié



Nous vous proposons, dans cet article, de télécharger et de tester un logiciel qui vous permettra de développer des applications autour de circuits logiques programmables.

IspLever est un logiciel développé par Lattice qui permet la création du fichier jedec nécessaire à la programmation de PLD, CPLD et FPGA du même constructeur. Plusieurs possibilités nous sont offertes, entre autres :

- programmation à partir d'un fichier Abel,
- programmation à partir d'un fichier VHDL,

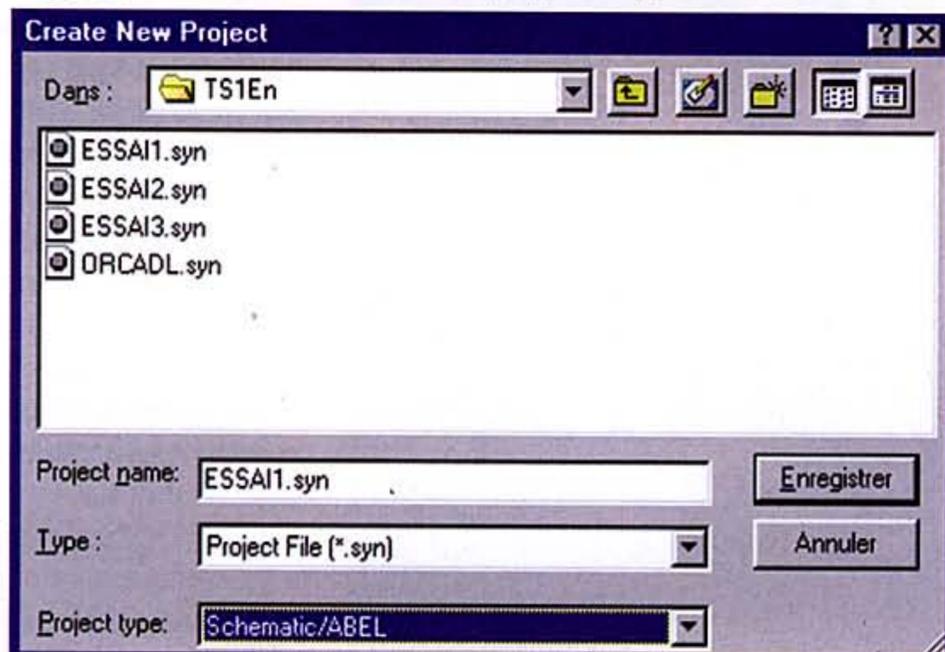
programmation à partir d'un schéma structurel.

Rendez-vous sur le site du constructeur <http://www.latticemi.com> dans la rubrique download software où vous pourrez télécharger un exécutable appelé " **ispLEVER starter** " qui fait approximativement 107 Mbits. Il est fortement conseillé de posséder une connexion rapide !

Vous devez aussi obtenir une licence renouvelable tous les six mois et nécessaire à l'utilisation du starter kit. Il suffit pour cela d'aller dans la rubrique licensing puis d'indiquer votre numéro de carte réseau ainsi que votre adresse Email.

L'obtention de la licence est pratiquement instantanée. Une fois le téléchargement terminé, exécutez le programme et suivez la procédure d'installation sans oublier de copier votre fichier licence dans le répertoire approprié. Si vous devez utiliser la programmation à partir d'un fichier VHDL, vous devez également télécharger un module nommé " **ispLEVER simplify** " qui occupe, lui, 18 Mbits et qui ne demande pas de licence supplémentaire.

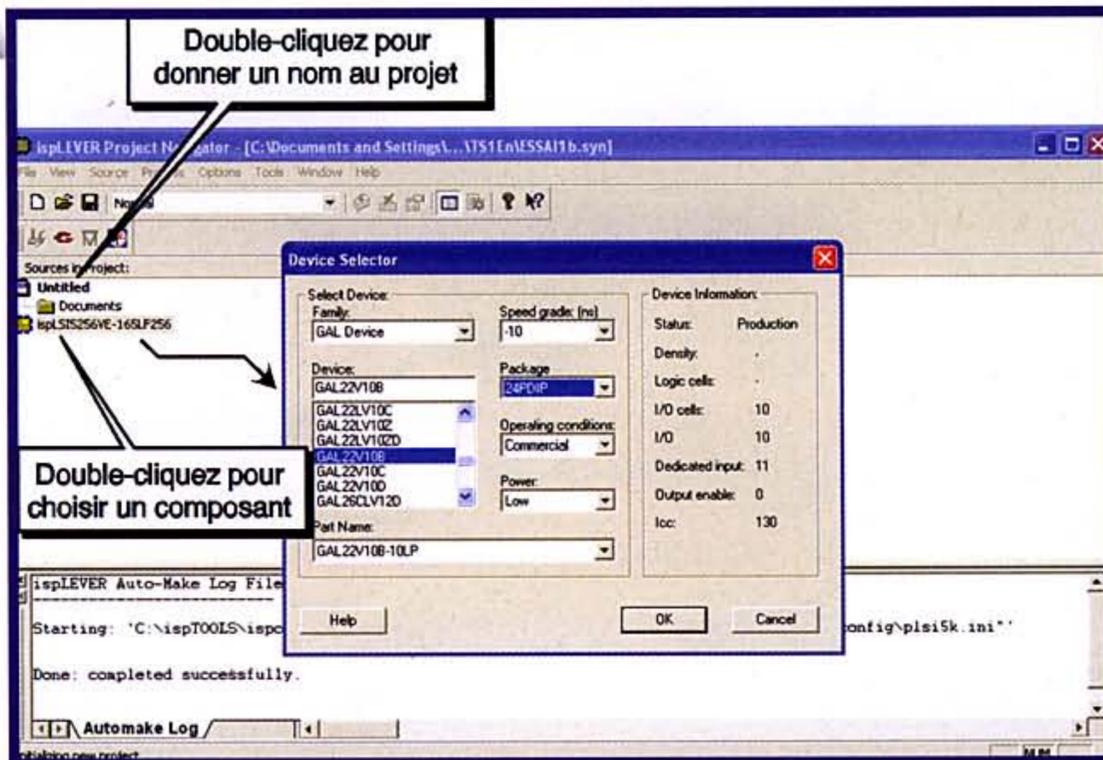
Création d'un nouveau projet de type Abel



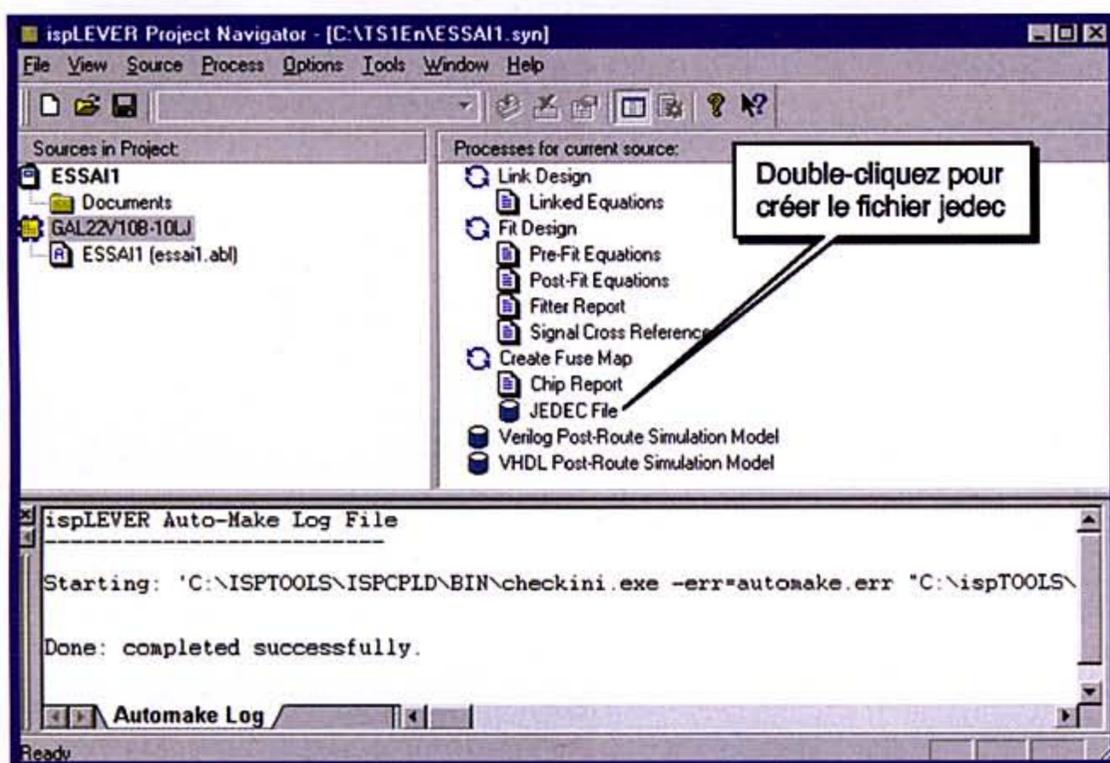
Programmation à partir d'un fichier Abel

Ouvrez ispLEVER puis créez un nouveau projet de type Abel (fig. 1 et fig. 2).

Une fois le composant cible choisi, il faut lui associer le fichier Abel (**Source puis Import**) ou le créer (**Source puis New**) en utilisant l'extension ".abl". Il ne reste plus qu'à créer le fichier jedec nécessaire à l'implantation du programme dans le composant cible (**fig. 3**).



2
Création du projet



Pour cela, vous devez ouvrir l'éditeur de stimulus (**Source puis New et Waveform stimulus**).

Choisissez l'échelle la plus adaptée (**Options puis Timing options**) à n'importe quel moment de la saisie de vos chronogrammes (**fig. 4**).

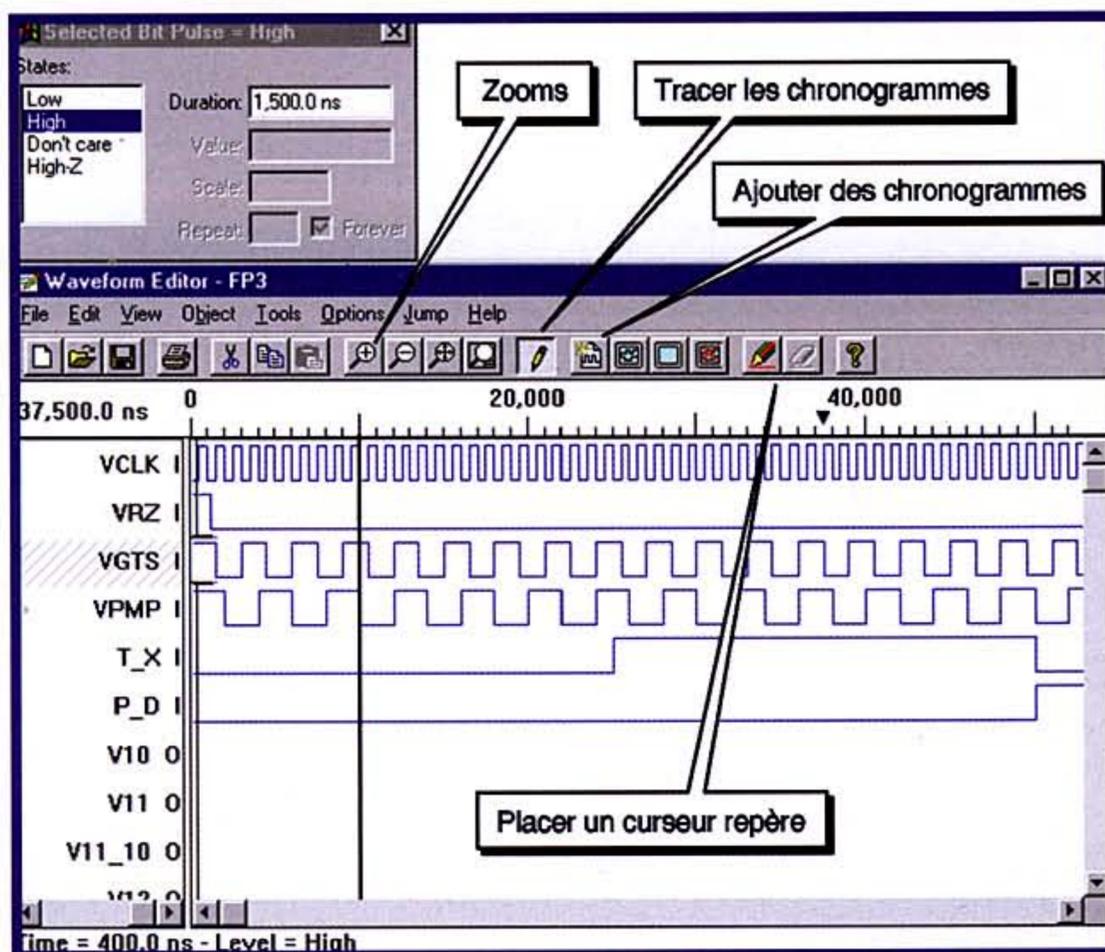
3
Création du fichier jedec

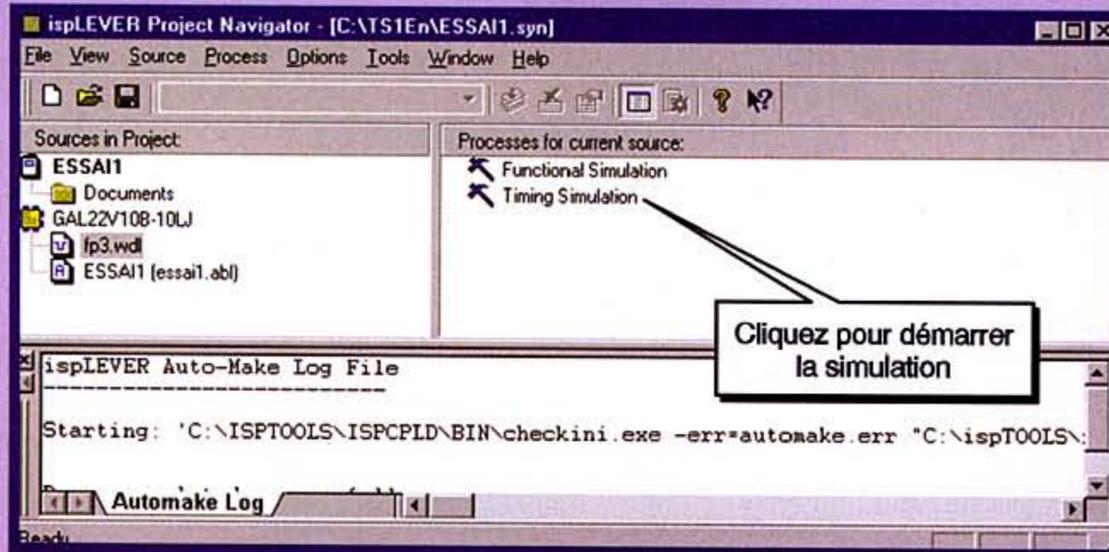
Vous pouvez nettoyer le projet ainsi que le disque (**File puis Clean up All**) mais attention, il faut d'abord copier le fichier jedec dans un autre répertoire ou sur une disquette.

Simulation du fonctionnement

Afin de simuler le fonctionnement de notre composant programmé, nous allons créer les chronogrammes relatifs à toutes les entrées.

4
Saisie des chromogrammes





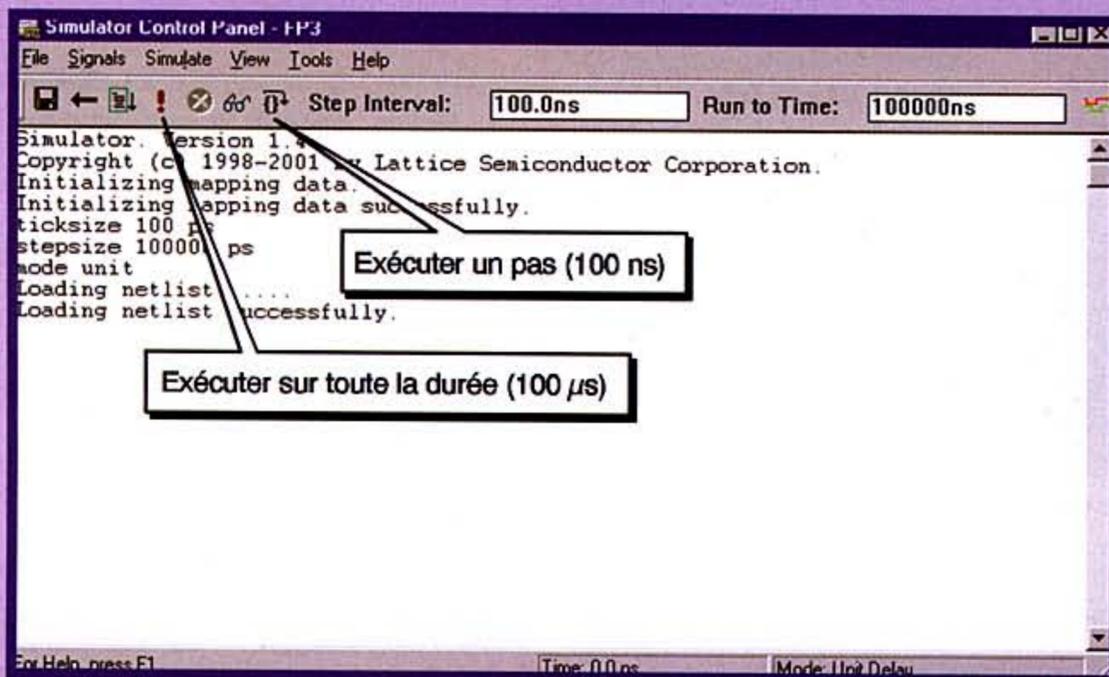
Une fois que toutes les entrées ont été saisies, sauvegardez les chronogrammes et quittez. Vous devez maintenant associer les chronogrammes au composant.

Lancez le processus de simulation puis réglez le pas de simulation ainsi que la durée totale dans la fenêtre suivante (fig. 6).

Vous pouvez exécuter la simulation pas à pas ou totalement.

5

Lancement du processus de simulation



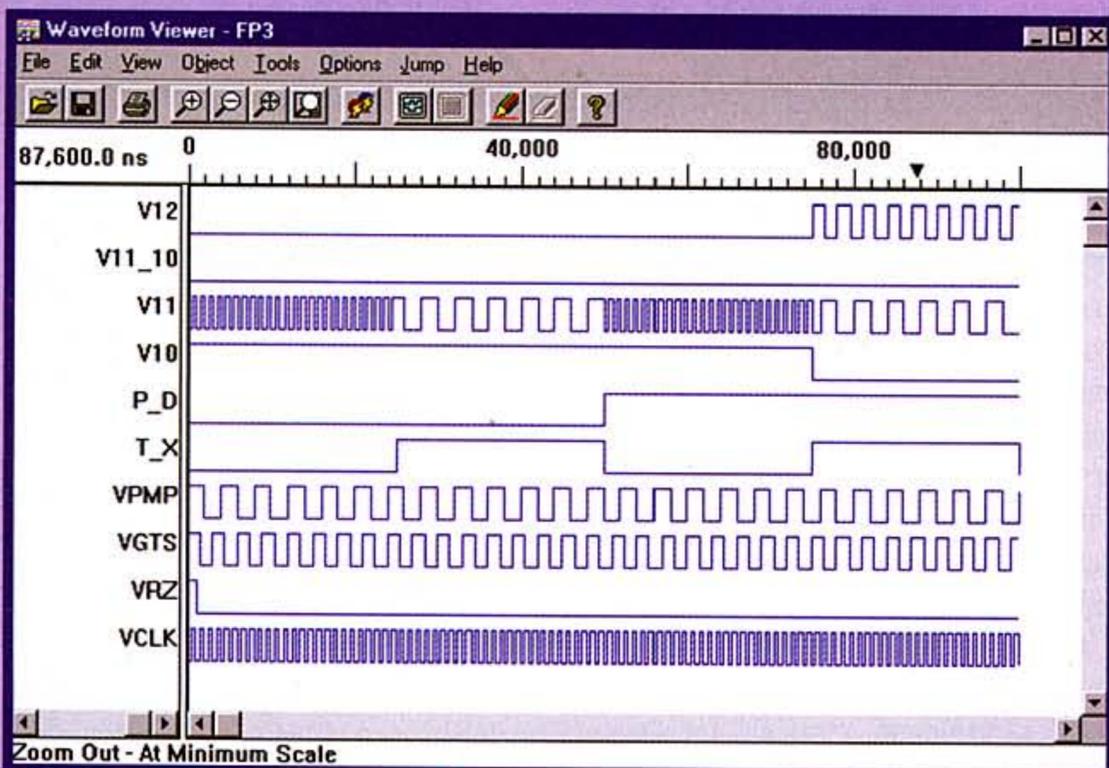
Les chronogrammes définis précédemment apparaissent complets (fig. 7).

Programmation à partir d'un fichier VHDL

Pour compiler le document VHDL, vous devez installer préalablement le module de synthèse "Simplify".

6

Réglage du pas de simulation ainsi que la durée totale



Ouvrez ispLEVER puis créez un nouveau projet de type VHDL (fig. 8)

Choisissez un composant cible (fig. 2) et importez ou ouvrez un nouveau document source (**Source puis New et VHDL module**). Pour créer le fichier Jedec, reportez-vous à la fig. 3.

7

Les chronogrammes au complet

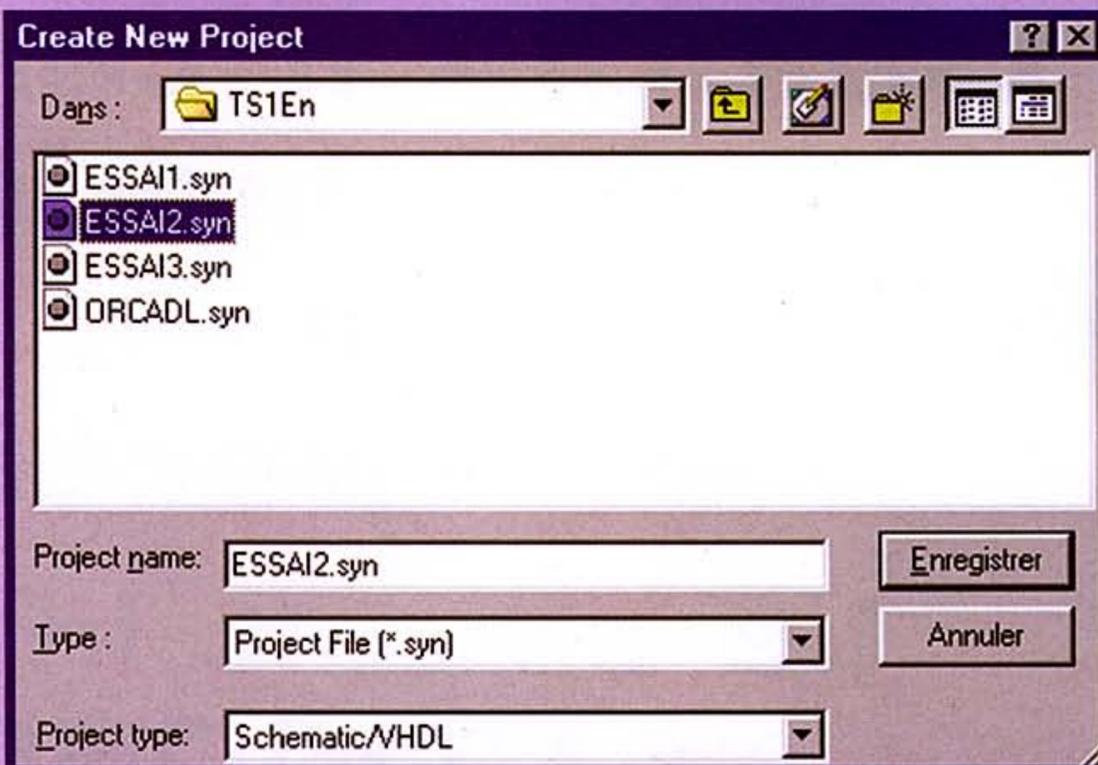
Programmation à partir d'un schéma

Ouvrez un nouveau projet (fig. 1) puis choisissez le composant cible (fig. 2).

Importez ou ouvrez un nouveau document (Source puis New et Schematic) puis donnez un nom à votre schéma (fig. 9)



Projet de type VHDL

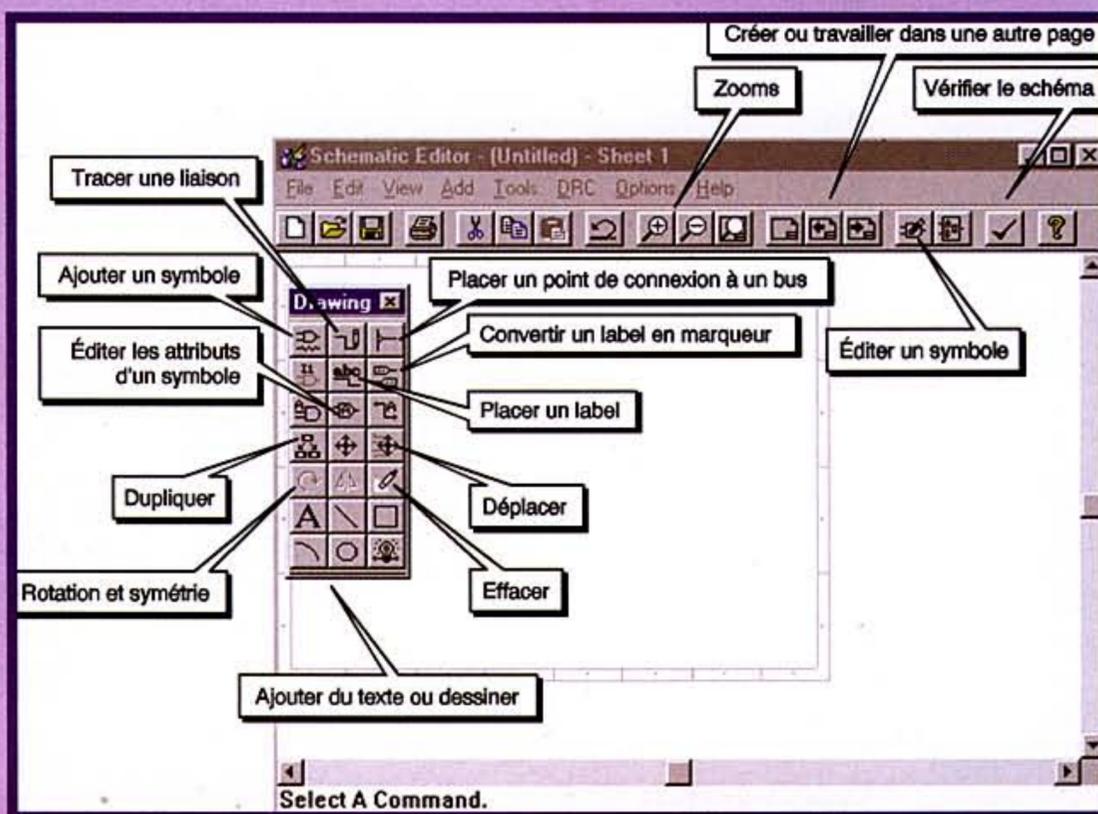


Pour créer un schéma, vous devez :

- Ajouter les symboles de composants (ils ont pu être créés préalablement à partir d'un fichier Abel ou VHDL).
- Ajouter les symboles des entrées et des sorties (bibliothèque IOPADS.lib).
- Placer les labels des entrées et des sorties à l'extérieur du schéma, puis les convertir en marqueurs après avoir choisi le type d'entrée/sortie.
- Placer les liaisons.
- Éditer les attributs des symboles d'entrées et de sorties en cliquant sur le symbole afin de fixer les numéros de broches (pinNumber).



Le schéma doit avoir un nom



- Effectuer une vérification du schéma puis sauvegarder et quitter.

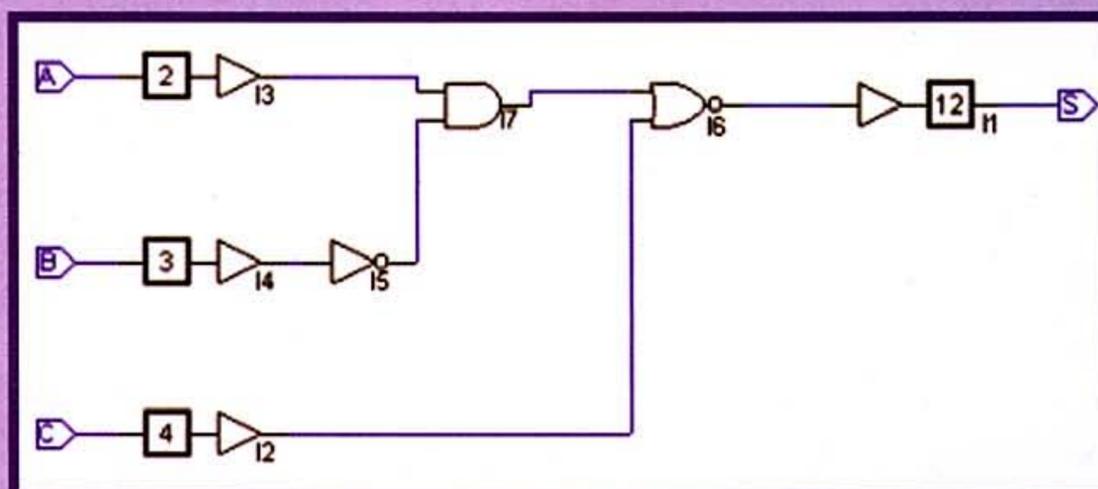
Le schéma, une fois terminé, doit ressembler à l'exemple de la figure 10.

Pour créer le fichier jedec correspondant, reportez-vous à la fig. 3.

L. JOSSE



Le schéma terminé

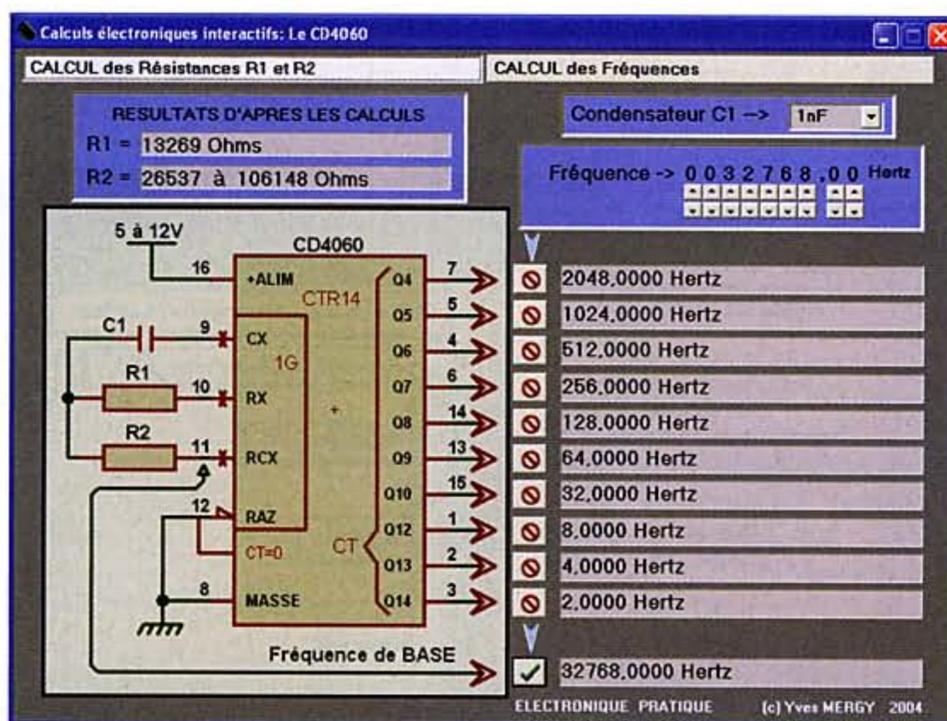


Calculs interactifs de circuits électroniques sur PC

4^{ème} partie :

Le circuit intégré « CD4060 »

Vous êtes maintenant habitués à cette série d'articles destinés à vous offrir des logiciels de calculs inédits et développés par nos soins afin d'élaborer vos projets électroniques. En effet, certains circuits intégrés nécessitent de fastidieux calculs pour déterminer les caractéristiques d'un signal, ou des valeurs précises de composants. Rien ne vaut l'ordinateur pour y parvenir de manière fiable, conviviale et en toute simplicité pour l'électronicien occasionnel ou professionnel.



Présentation

Parmi le vaste choix de circuits intégrés, certains se retrouvent régulièrement au fil des pages de votre magazine « Électronique Pratique ». Notre quatrième article se consacre au célèbre CD4060. Ce circuit logique renferme dans son boîtier un oscillateur et une bonne cascade de diviseurs ; ce qui le prédispose à de nombreux montages dès qu'il s'agit de travailler sur une, ou plusieurs fréquences simultanément. A partir d'une fréquence de base, vous obtenez dix fréquences très précises. Ne tâtonnez plus sur vos réalisations !

Familiarisez-vous avec notre logiciel sans plus tarder.

Principales caractéristiques

Tension: 4,5 à 15 volts.

Sorties adaptées aux charges CMOS et TTL.

Fréquence maxi : plus de 10 MHz.

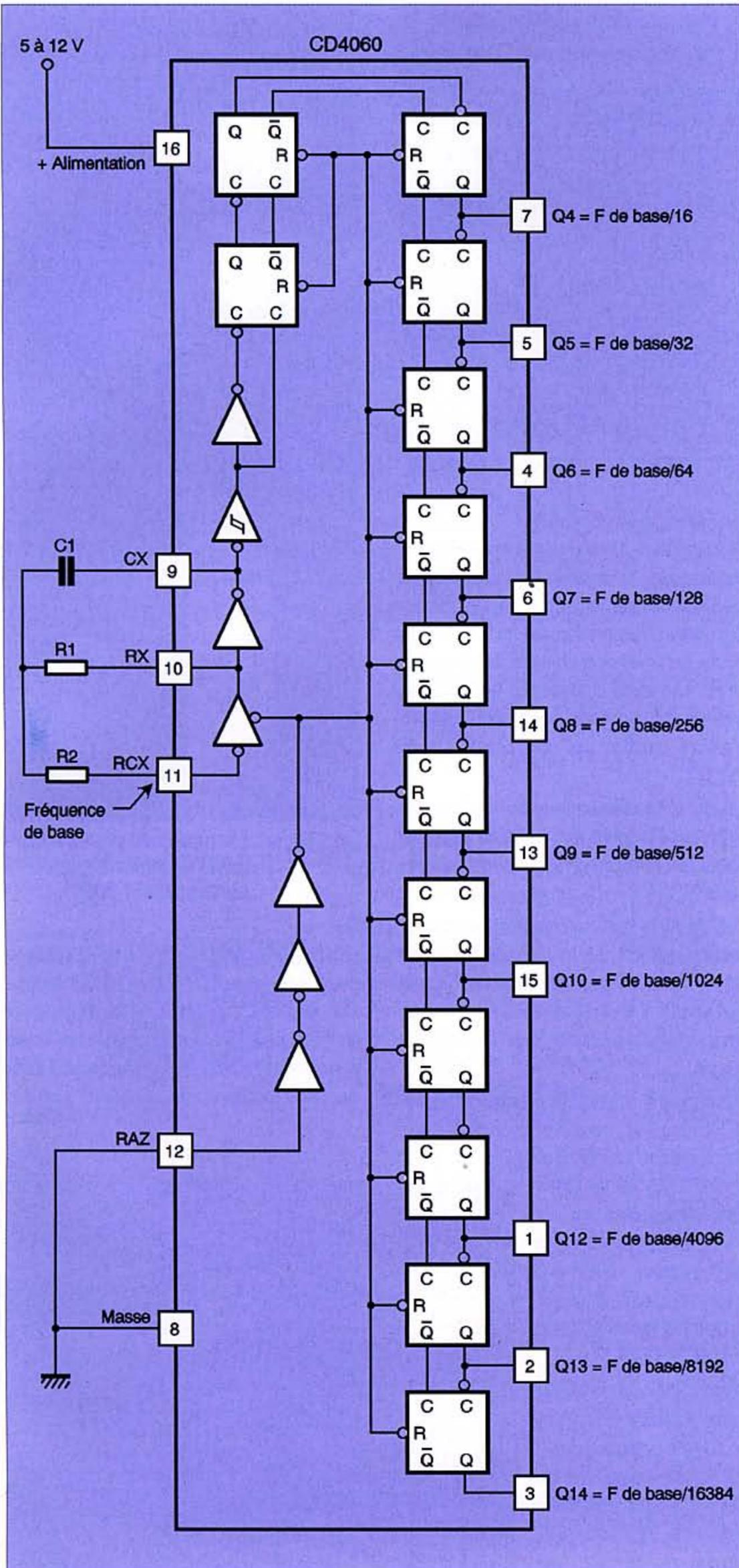
10 étages de divisions binaires (4 à 10 et 12 à 14).

Ligne de remise à zéro commune à toute les sorties.

Protection des entrées par diodes.

Schémas de principe

La **figure 1** montre le schéma d'utilisation de base employé par notre logiciel. Une particularité cependant, la structure interne simplifiée du CD4060 apparaît sur le même plan, ce qui offre une meilleure compréhension. Le couple formé par la résistance R1 et le condensateur C1 détermine la fréquence de base de l'oscillateur. La valeur de la résistance R2 doit être comprise entre 3 et 8 fois la valeur de R1. Vous l'aurez compris, la qualité de ces trois composants influe directement sur la précision et la stabilité du montage. La broche d'initialisation générale (12) doit être raccordée à la masse. Si cette dernière se trouve au niveau haut, toutes les sorties sont à « 0 » (niveau bas). La **figure 2** donne la table de vérité du circuit CD4060. Il est également possible de ne pas utiliser les portes logiques internes formant l'oscillateur et d'entrer un signal d'horloge, à la fréquence voulue, sur la broche 11 (RCX) du circuit. Un dernier mot sur l'oscillateur ; afin d'obtenir une stabilité parfaite, il suffit de remplacer R1, R2 et C1 par un quartz taillé pour la fréquence désirée et accompagné de ses quelques composants. Cette modification est donnée en **figure 3**. Le condensateur « Ct » peut être fixe ou variable selon le degré de précision



1 Schéma d'utilisation de base

RCX Clock (11)	RAZ (12)	Sorties Q4 à Q14 (7, 5, 4, 6, 14, 13, 15, 1, 2, 3)
	L	Pas de changement
	L	Sortie suivante
X	H	Toutes sorties à l'état bas

2 Table de vérité du CD 4060

requis. Notre logiciel convient également pour le choix du quartz et des fréquences qui en découlent.

A titre d'indication, le quartz de 32768 Hz est l'un des plus employés avec ce circuit intégré. Il fournit, après la dernière division (Q14), un signal précis de 2 Hz très utilisé en horlogerie électronique. Par un comptage simple à l'aide d'une bascule ou d'un microcontrôleur, vous obtenez la seconde précise !

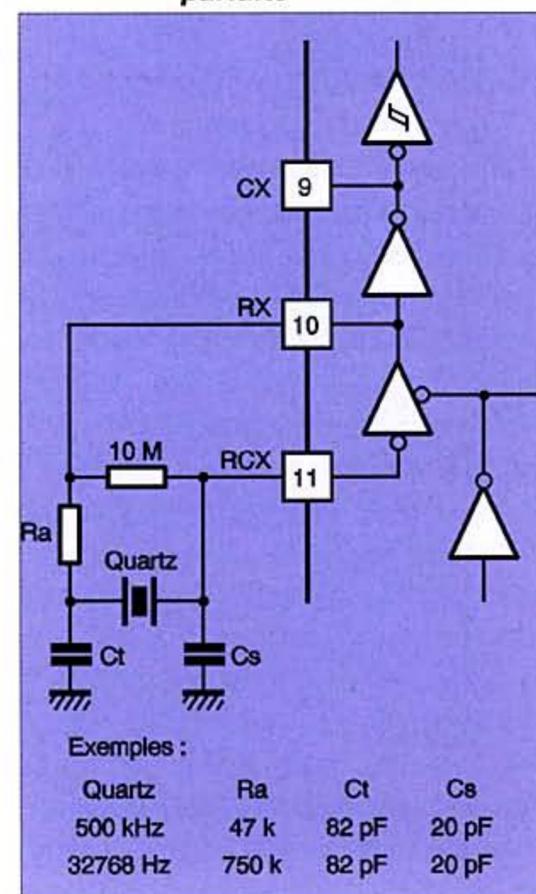
La formule de base permettant de trouver la fréquence d'horloge en fonction de R1 et de C1 est la suivante :

$$F_{(hertz)} = \frac{1}{2.3 \times R1_{(\Omega)} \times C1_{(Farad)}}$$

avec $3 \times R1 \leq R2 \leq 8 \times R1$

Comme nous l'avons vu précédemment, la valeur de R2 n'est pas critique dès lors qu'elle se situe dans la fourchette de 3 à 8 fois R1. Il convient ensuite, éventuellement, de transposer cette formule pour déterminer la valeur de la résistance R1 ou du condensateur C1,

3 Utilisation d'un quartz pour obtenir une stabilité parfaite



N° de broche	Nom de la sortie	Facteur de division	Exemple
11	RCX ou clock	0 (fréquence de base)	32768 Hz
7	Q4	(F(base)) / 16	2028 Hz
5	Q5	(F(base)) / 32	1024 Hz
4	Q6	(F(base)) / 64	512 Hz
6	Q7	(F(base)) / 128	256 Hz
14	Q8	(F(base)) / 256	128 Hz
13	Q9	(F(base)) / 512	64 Hz
15	Q10	(F(base)) / 1024	32 Hz
sortie Q11 (division par 2048) inexistante			
1	Q12	(F(base)) / 4096	8 Hz
2	Q13	(F(base)) / 8192	4 Hz
3	Q14	(F(base)) / 16384	2 Hz

Facteur de division pour chaque sortie du CD 4060

puis de rechercher la fréquence sur chaque sortie, d'où la nécessité de notre logiciel. Nous en parlons ci-dessus, le CD4060 fournit 10 signaux. La fréquence de chacun de ces signaux est issue d'une division de celle de base (F_{base}).

Le tableau ci-dessus donne le facteur de division pour chaque sortie.

Ces sorties ne peuvent en aucun cas piloter des charges de puissance. Il convient de les tamponner par un circuit spécifique ou à transistors dès qu'un fort courant est requis. La **figure 4** donne plusieurs exemples de circuits pouvant se raccorder aux sorties du CD4060.

A l'évidence, si la fréquence est trop élevée, n'utilisez pas un relais électromécanique, mais plutôt un relais statique.

Le programme

Le programme "CD4060.EXE" est prévu pour tourner sous WINDOWS® à partir des versions 9x et supérieures, bien sûr. Comme toujours, nous vous rappelons qu'avec notre magazine, le logiciel vous est offert en téléchargement libre sur notre site Internet: <http://www.electroniquepratique.com>.

Les rares lecteurs n'ayant pas l'opportunité de se connecter à Internet, pourront se procurer le logiciel auprès de la rédaction en fournissant un CD vierge et une enveloppe auto adressée suffisamment affranchie.

La convivialité et la simplicité d'utilisation sont toujours de mise avec notre programme. Dès l'ouverture, deux onglets en haut de page vous proposent de choisir le type de valeur que vous souhaitez calculer. Après un clic gauche sur l'un d'eux, vous vous trouvez

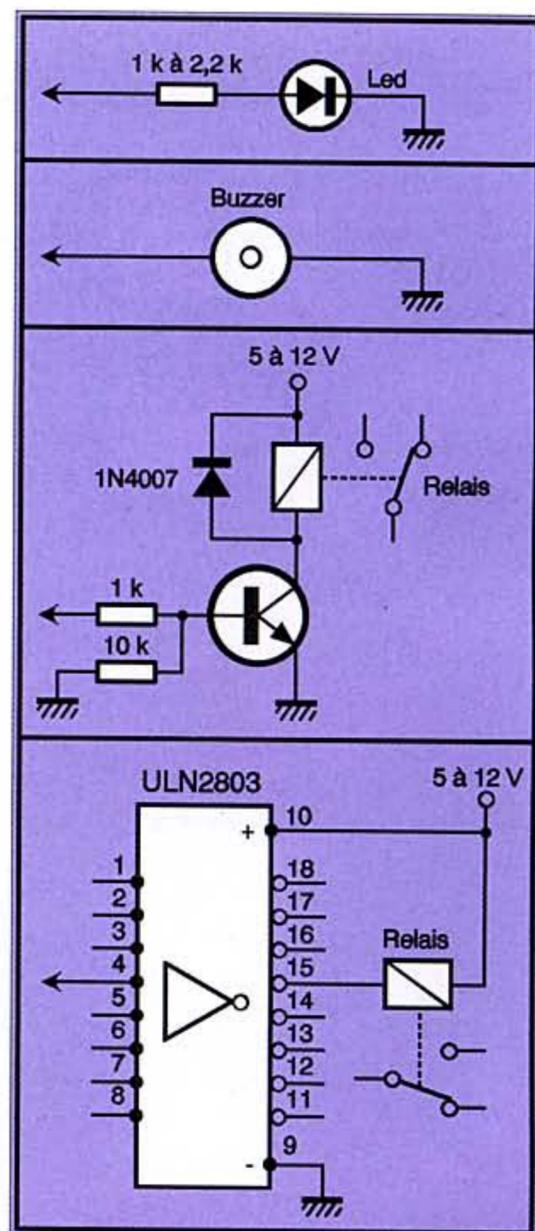
sur un écran vous rappelant le schéma de principe et vous invitant de la même manière à préciser la ou les valeurs que vous désirez faire calculer au logiciel. Il peut s'agir, soit des valeurs des résistances en fonction des autres paramètres (y compris de la fréquence sur une sortie quelconque), soit de la fréquence accompagnée des rapports de divisions en fonction des composants (R1, R2 et C1).

Les paramètres se rentrent de manière interactive avec la souris (pas de saisie au clavier) selon les valeurs normalisées des composants.

Les résultats des calculs sont instantanément mis à jour, en temps réel. Nous vous rappelons que le résultat donne un nombre non arrondi. Il faut choisir le composant électronique dont la valeur s'en approche au mieux.

Sur la page du calcul des résistances, à droite du schéma, vous voyez onze boutons représentant un symbole d'interdiction ou de validation. En cliquant sur l'un d'eux, le symbole change et la fréquence que vous réglez en haut de page correspond à la sortie sélectionnée.

Comme toujours, la modification des valeurs est prise en compte instantanément. En cas d'un choix inadéquat des paramètres, au lieu et place des résultats, vous obtenez un message vous indiquant clairement le type d'erreur rencontré (fréquen-

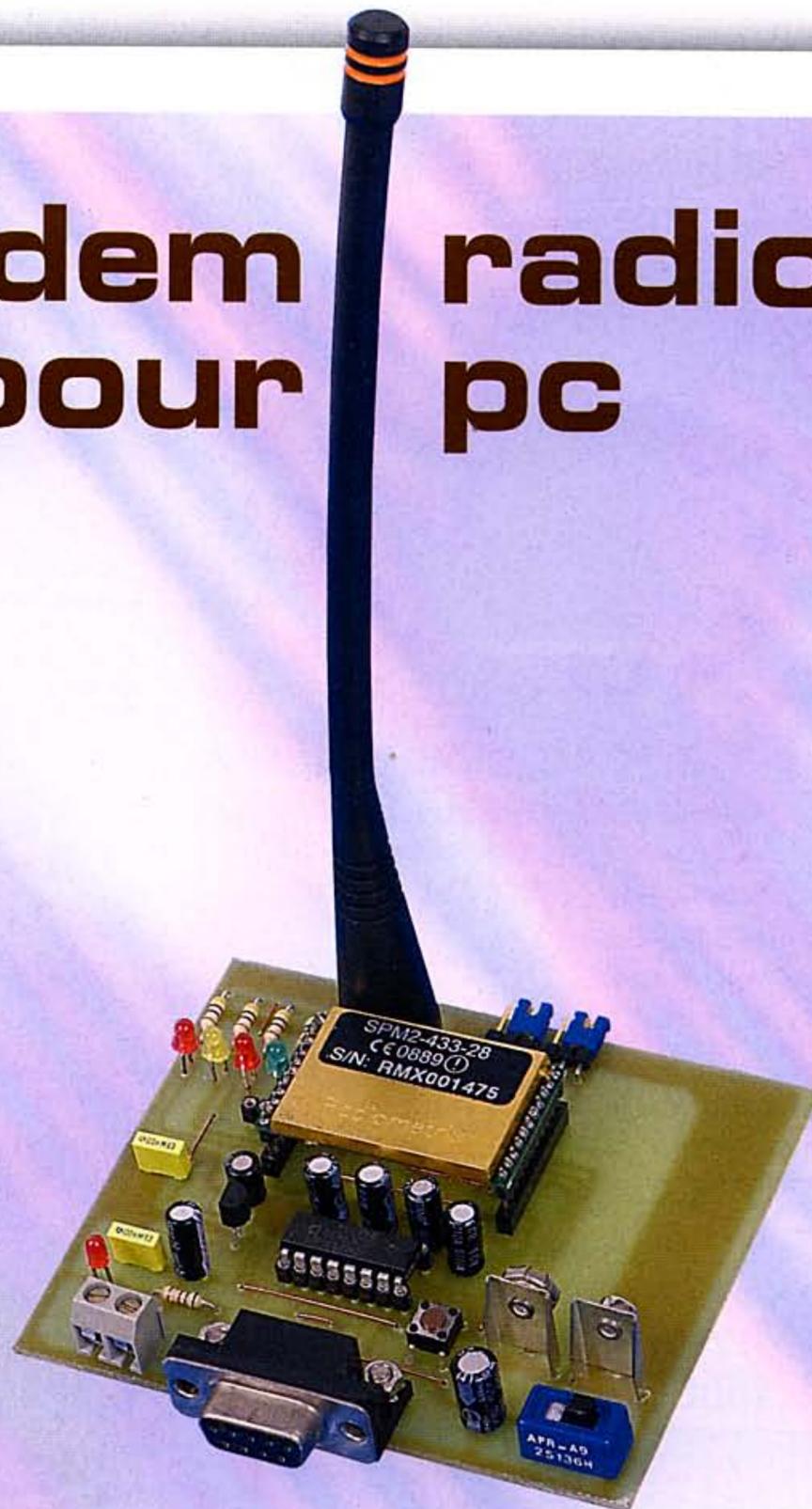


4 Exemples de circuits pouvant se raccorder aux sorties du CD 4060

ce nulle, valeur trop élevée...). La vue d'écran ci-dessous laisse apprécier la qualité du logiciel. Gageons qu'il vous rende de précieux services pour vos projets, que vous soyez enseignants, étudiants, ou simplement passionnés d'électronique.

Y. MERGY

Modem radio pour pc



Le montage que nous décrivons dans cet article permet la communication entre deux ordinateurs de type PC avec une fiabilité exceptionnelle. Les données envoyées sont en effet contrôlées et le risque d'erreurs durant la transmission est quasiment inexistant. D'autre part, la vitesse de communication est également exceptionnelle pour ce type de transmission.

Généralités

La société RADIOMETRIX, bien connue pour ses modules RF, propose un nouveau module : le **SPM2-433-28**. C'est un modem radio subminiature à faible coût pouvant facilement s'intégrer dans un nouveau projet ou dans une application déjà existante. Il permet la transmission à distance de données numériques séries au format RS232 de façon totalement transparente, rendant ainsi son utilisation très simple. En effet, le module, dont la structure interne est représentée en **figure 1**, génère les trames de préambule, la mise en "packet", le codage Manchester des données et effectue également un contrôle d'erreur

avec accusé de réception. Le SpacePortModem2 (SPM2) est constitué d'un microcontrôleur qui s'occupe de la gestion du protocole radio et d'un émetteur/récepteur RF.

La **figure 2** donne la représentation physique du module. Les fonctions de ses broches sont les suivantes :

- RF GND : masse de la partie radio.
- RF : sortie antenne.
- RX SELECT : sélection manuelle de l'état de réception.
- TX SELECT : sélection manuelle de l'état d'émission.
- TXD/AF : accès aux données transmises au transceiver RADIO.
- D3, D2 : données internes au module.
- SIGNAL : Signal de détection du préambule.
- RST : entrée RAZ du module.
- RXA : signal de contrôle d'acknowledge.
- RXR : signal de contrôle " Valid Data Packet ".
- RESET : entrée RAZ du module FRPC.
- SETUP : permet de configurer certains paramètres du module.
- WAKE/DTR : permet d'activer ou de désactiver la mode Power-Down.
- TXD : entrée TX.
- RXD : sortie RX.
- CTS : permet de gérer le flux de données.
- DEFAULT : force la communication à 9600 bds.
- VCC : entrée de l'alimentation +5 Vcc.
- GND : masse de l'alimentation reliée en interne à la masse RF.

Les caractéristiques électriques, de l'interface et de l'émetteur/récepteur sont les suivantes :

Interface

Type	Description
Interface série	Niveau TTL
Protocole série	8 bits/1 bit de stop/sans parité
Signaux série	RXD, TXD, RFFLOW, WAKE
Contrôle POWER-DOWN	Par signal WAKE
Mode d'interface	Avec ou sans gestion de flux des données
Vitesse de communication DTE	600/1200/2400/4800/9600/14400/19200/ 38400/57600/115200
Vitesse	Jusqu'à 55kbps/s sans correction d'erreur et 28kbps/s avec correction d'erreur

Caractéristiques électriques

Désignations	Minimum	Typique	Maximum	Unité
Vcc Alimentation étage HF	-----	5	-----	V
I _s Consommation	-----	40	-----	mA
I _{ss} Consom. Standby	-----	15	-----	mA
I _{pd} Consom. Power-down	-----	400	-----	µA
F _w Fréquence d'émission	-----	433,92	-----	MHz
Top Température d'utilisation	0	-----	+55	°C

Émetteur/récepteur

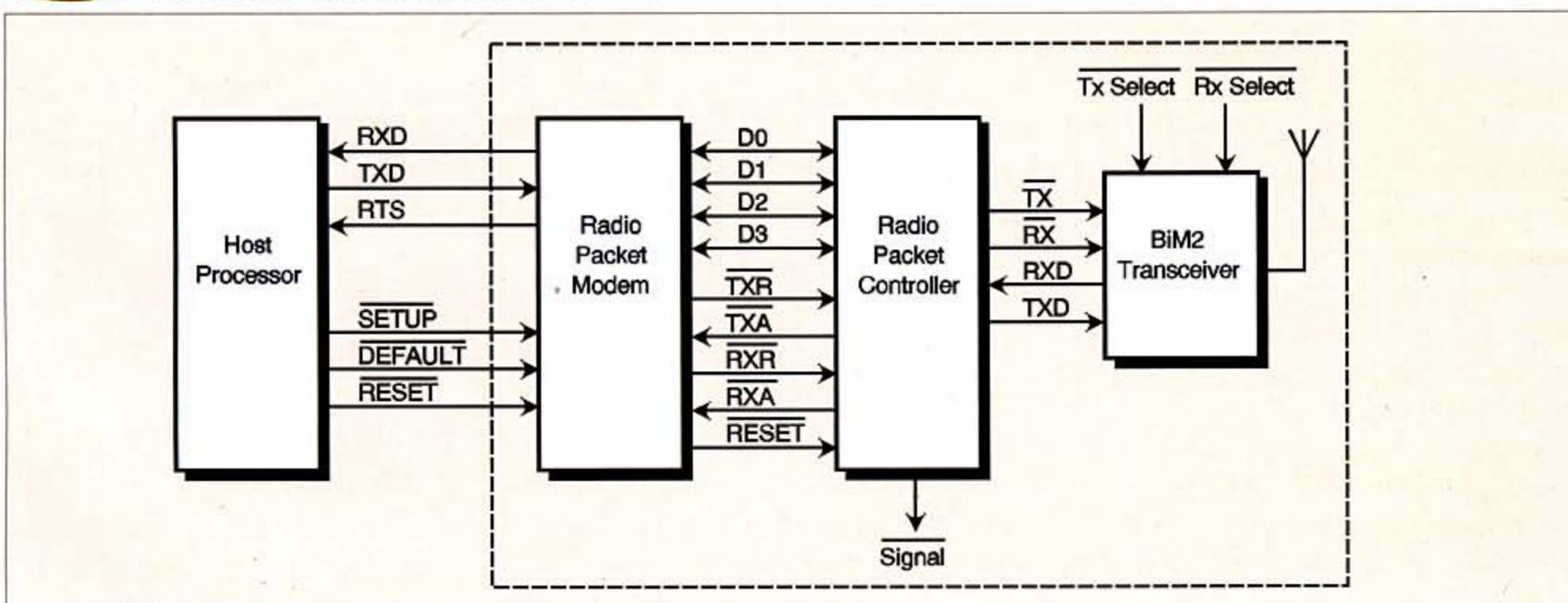
Type	Description
Puissance émetteur	10mW
Sensibilité récepteur	-100dBm
Gamme de fréquence	Existe en version 433,92 MHz, 869,85 MHz et 914,5 MHz
Conformités	EN 300220-3/EN 301489-3

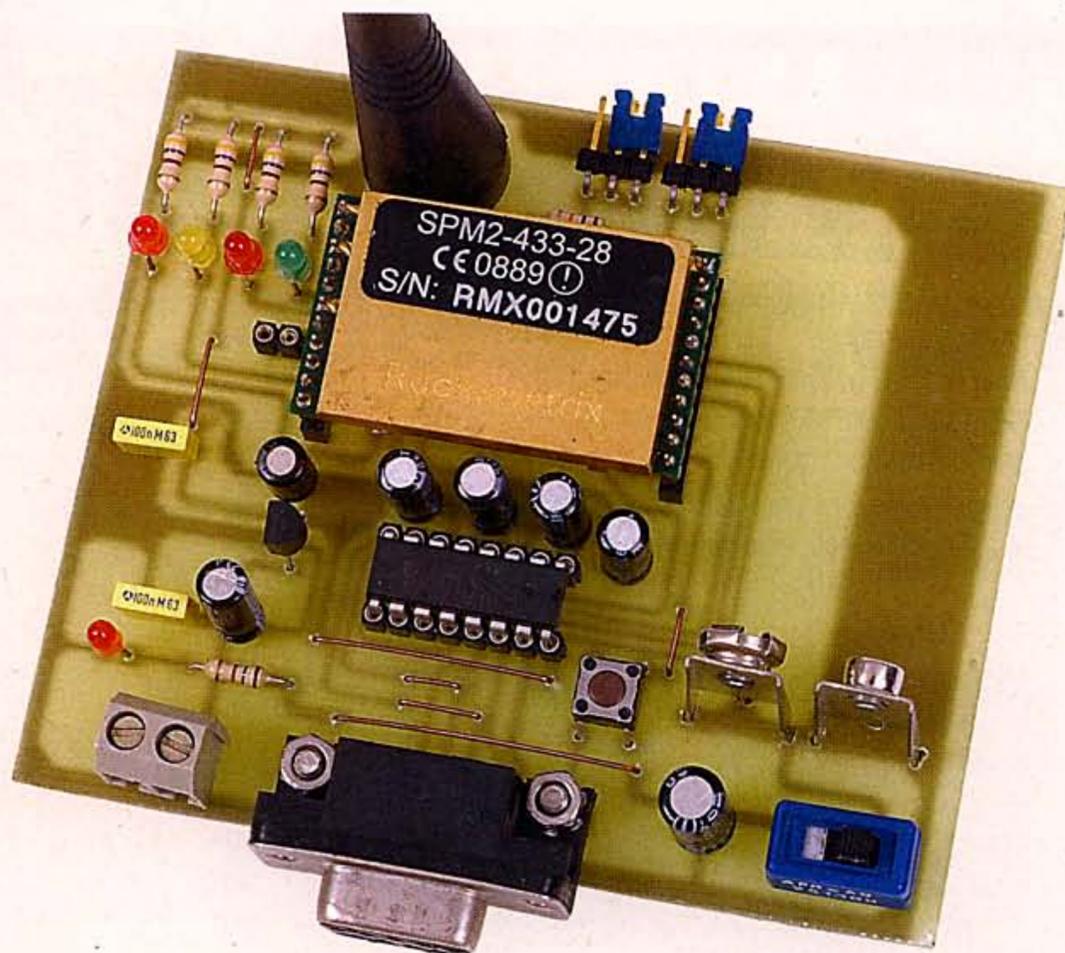
L'interfaçage du modem radio SPM2 avec un ordinateur PC ou un microcontrôleur peut se faire en mode trois fils (TX/RX/masse) sans gestion de flux de données. Dans ce cas, il convient de respecter des pauses entre chaque émission de données car le buffer interne du module est de 96 octets. Il est

également possible d'utiliser un mode avec gestion du flux de données afin d'automatiser l'accès au buffer du SPM2. Dans ce cas, on peut utiliser le signal RSFLOW que le SPM2 active lorsque son buffer est rempli à plus de 66 % et désactive lorsque son remplissage n'atteint plus que 33 %.

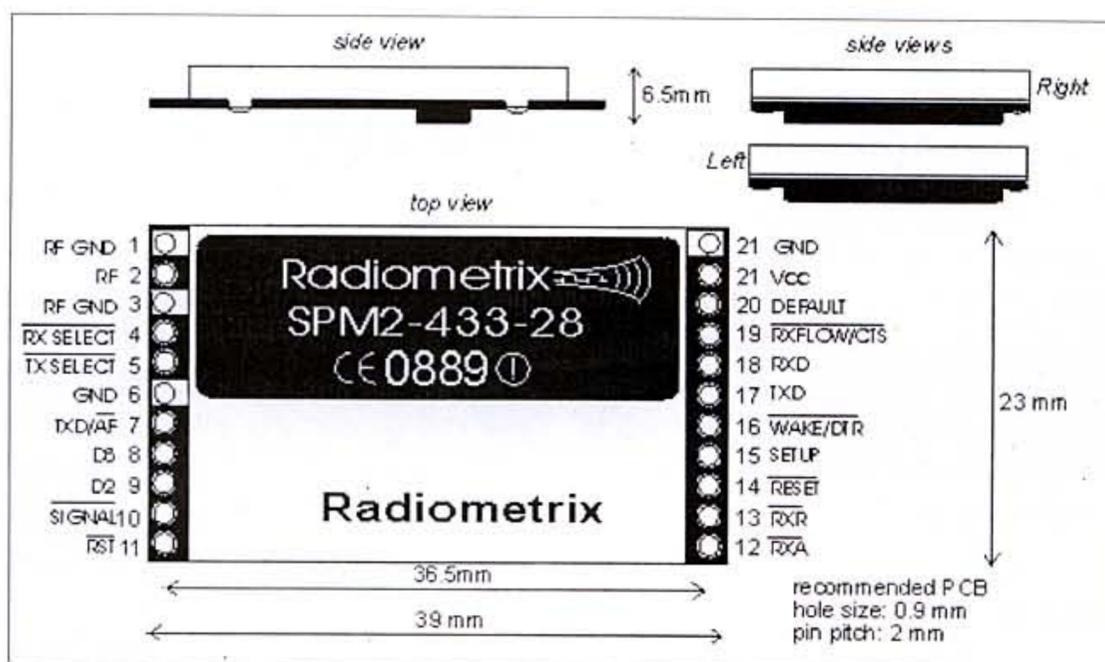
On peut utiliser les modules SPM2 sans aucune manipulation particulière. Il est cependant possible de modifier très facilement certains paramètres du SPM2. Il suffit pour cela de relier son entrée SETUP à la masse et de le connecter à un PC en utilisant le logiciel HyperTerminal™ de Windows™.

Structure interne du SPM2-433-28





Le module prêt à l'emploi. Aucun réglage n'est nécessaire



2 Représentation physique du module

Il suffit alors d'envoyer des ordres qui sont automatiquement sauvegardés dans la mémoire flash du module. La liste suivante explique les différents ordres que reconnaît le module :

config : affiche la liste des paramètres du module.

default : mise en émission permanente.

reset : effectue une RAZ du module. Tous les paramètres modifiés ne seront pris en compte qu'après cette RAZ. Il convient que la broche SETUP soit désactivée afin d'éviter de

rentrer à nouveau dans le mode de configuration.

unit : 2 modules SPM2 peuvent communiquer s'ils disposent du même numéro " unit " (0 à 15, réglage à 0 en usine).

site : 2 modules SPM2 peuvent communiquer s'ils disposent du même numéro " site " (0 à 7).

addr : permet de modifier la valeur du paramètre " unit " sans la sauvegarder dans la mémoire du SPM2 (0 à 15, réglage à 0 en usine).

baud : permet de reconfigurer le débit de communication entre la SPM2 et le PC ou le microcontrôleur (600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, réglage à 9600 en usine).

thruput : permet de sélectionner la nature du débit radio. **Max** : le débit est maximum (figure 3) ; **slots** (figure 4) : introduit des silences entre l'émission des trames afin de limiter les collisions radios si plusieurs couples de modules sont utilisés dans le même temps ; **fcc** : permet de mettre le module en conformité avec la réglementation US n'autorisant qu'une occupation radio de 10 % du temps (réglage à **max** en usine).

flow : permet de configurer le mode de communication du SPM2 avec (**hw**) ou sans (**none**) contrôle de flux de données (réglage à **hw** en usine).

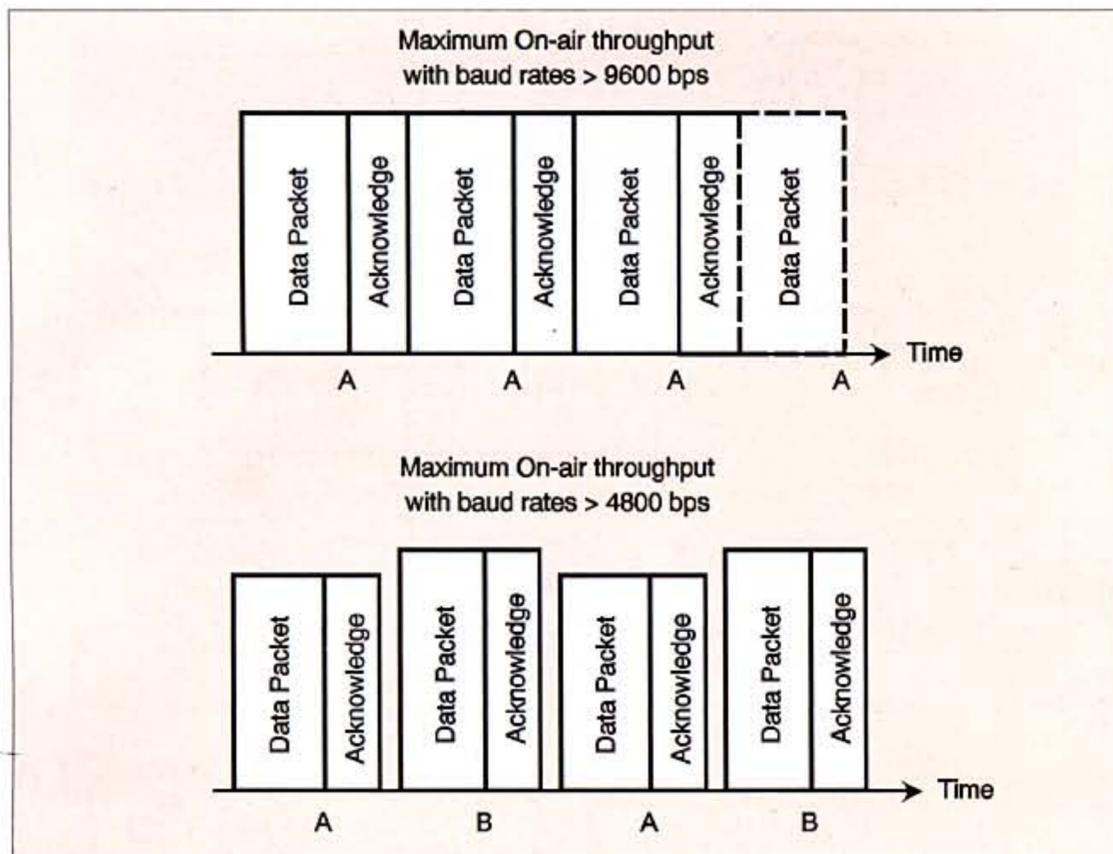
serdly : lorsque le SPM2 reçoit le premier octet émanant du système de commande externe (microcontrôleur ou PC), il attend que son buffer soit plein ou que la durée d'un timer initialisé par la valeur " serdly " X 10 ms soit arrivée à terme. Il commence alors la transmission radio proprement dite. Il est alors possible de modifier ce paramètre afin d'optimiser le débit de la transmission radio. La valeur du paramètre peut être comprise entre 10 et 255 (réglage à 30 en usine).

shdn : ce paramètre peut prendre la position " on " ou " off ". Il est réglé sur " on " en usine. Il permet d'autoriser ou non le module SPM2 à scruter la broche SHDN qui lui permet de rentrer en mode " power-down ".

retry : permet de configurer le nombre de fois qu'une trame pourra être ré-émise si celle-ci n'est pas correctement transmise. Au-delà de la valeur " retry ", si la trame n'est pas correctement transmise, elle sera considérée comme perdue. Ce paramètre peut prendre une valeur comprise entre 1 et 63 (réglée à 5 en usine).

strtmmsg : paramètre " on " ou " off " (réglé sur " on " en usine). Il permet d'autoriser ou non, à la mise sous tension du module ou après une RAZ, l'envoi d'un message par le SPM2 (SpacePortModem - Radiométrix N° de série - V1.xx) au PC ou au microcontrôleur de commande. Cela est pratique lors de la mise au point mais doit être inhibé lors de l'utilisation des modules.

ackmode : paramètre " on " ou " off " (réglé sur " on " en usine). Ce paramètre permet d'activer ou de désactiver la retransmission automatique des " packets " (contrôle d'erreur). Mis sur " off ", ce paramètre permet



3-4 Sélection de la nature du débit radio

d'accélérer le débit de la transmission ou de travailler en mode multi-récepteur, mais dans ce cas, il n'y a plus de protection contre la perte de données pouvant survenir à la suite d'un brouillage radio.

remote : paramètre " on " ou " off " (réglé sur " off " en usine). Il permet d'autoriser ou non la possibilité de pouvoir configurer le module par voie radio, sans avoir besoin de solliciter son entrée SETUP. Afin de réaliser cette configuration à distance, il convient d'envoyer la séquence suivante : remote <N° de série><commande><paramètre>. Le N° de série peut facilement être connu en raccordant le module concerné à un PC via le logi-

ciel d'émulation HyperTerminal™. Les autres paramètres sont les suivants :

Baud 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

Unit 0 à 15

Site 0 à 7

Shdn on/off

Flow rxflow/none

Serdly 30 à 255

Retry 1 à 63

radar : paramètre compris entre 0 et 15. Cette option permet d'effectuer immédiatement et très simplement un test de communication entre deux modules configurés avec le même numéro d' " unit ".

La configuration des modules SPM2 peut être effectuée de la manière que nous venons de décrire. Toutefois, RADIOMETRIX met à la disposition de tous un logiciel de configuration gratuit téléchargeable sur son site. Ce logiciel, dont une vue d'écran est donnée en **figure 5**, permet au moyen d'une interface graphique très conviviale, de modifier les paramètres du SPM2-433-28 connecté au port série du PC. Il peut également activer la fonction RADAR afin de réaliser des tests de portée entre deux modules.

Dispositif de correction d'erreur

Il existe de nombreux produits similaires au SPM2-433-28. La caractéristique essentielle qui différencie pourtant ce dernier des autres produits est qu'il possède un vrai système de correction d'erreur. En effet, les autres systèmes effectuent également un contrôle d'erreur par contrôle de parité des octets reçus. Ce système de contrôle ne délivre aucune donnée si la parité signale une erreur, ce qui est une bonne chose. Seulement, si la parité est correcte et que l'octet envoyé est erroné, la donnée est considérée comme correcte, ce qui est évidemment déjà moins bon. On ne peut alors plus parler de sécurité de transmission.

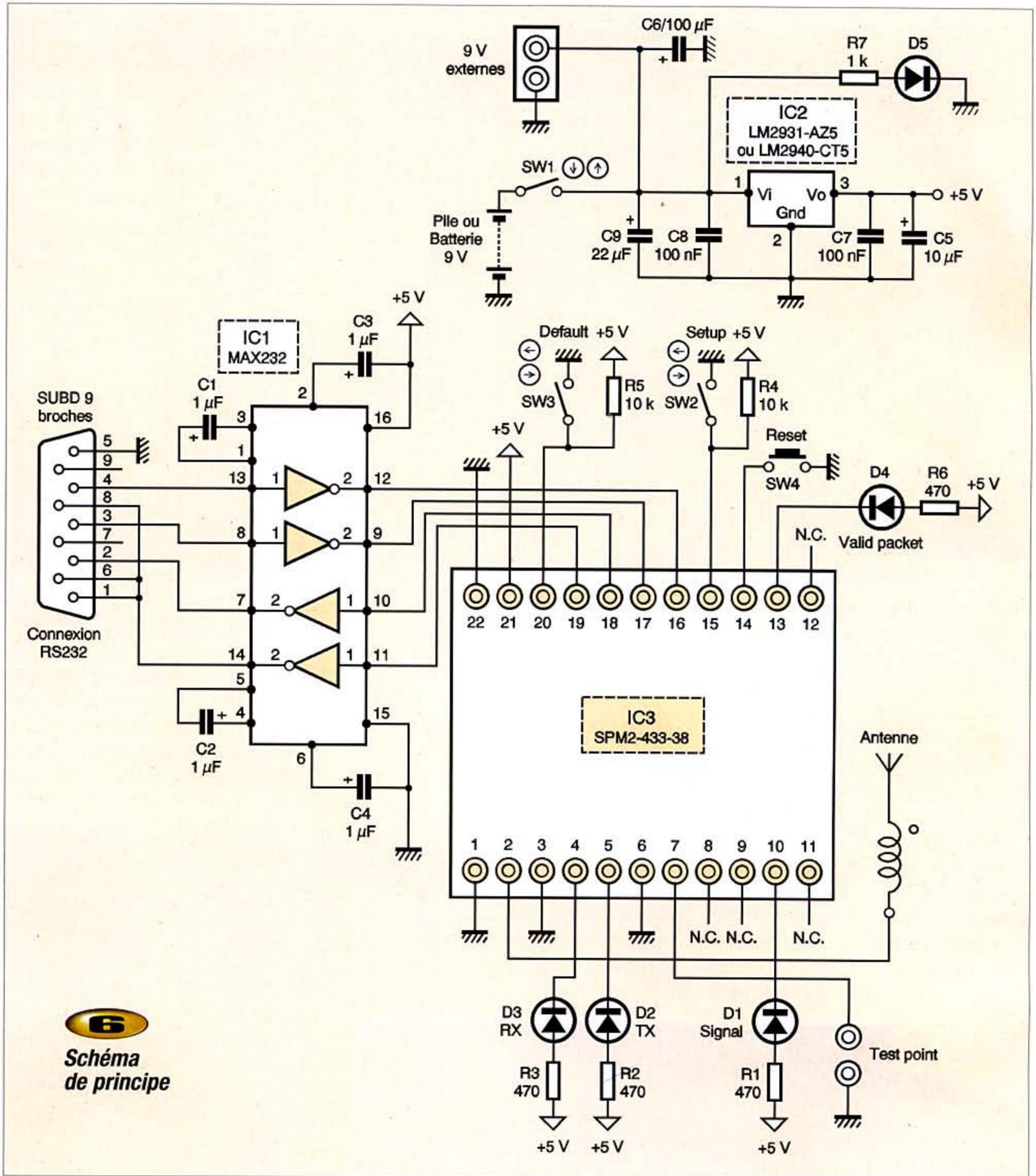
Le SPM2-433-28 dispose, quant à lui, d'un véritable système de contrôle des données envoyées avec accusé de réception. Cette vérification des données émises est effectuée de la manière suivante : lorsqu'une transmission a lieu, chaque " packet radio " est systématiquement retourné pour contrôle vers l'émetteur, de manière transparente pour l'utilisateur. Si l'émetteur qui vient d'envoyer la donnée s'aperçoit que la trame d'accusé de réception n'est pas exactement identique, il la transmet à nouveau afin qu'elle soit correctement réceptionnée par l'autre module, et ce, autant de fois qu'il sera nécessaire.

Fonctionnement en mode " point-à-point " ou " multi-points "

Lorsqu'un module SPM2 commence à émettre, il cherche à entrer en contact avec son module associé. Ce dernier doit être configuré avec les mêmes codes " unit " et " site ". En effet, un SPM2 présentant des



Le module SPM2-433-28 de Radiometrix



6
Schéma de principe

codes différents ne répondra pas à cet appel. Cette caractéristique permet de faire fonctionner jusqu'à 128 paires de modules dans le même environnement, mais à condition qu'ils ne soient pas utilisés en même temps. On peut, dans le même ordre d'idée, utiliser un module maître qui pourra facilement communiquer à tour de rôle avec 16 modules esclaves. Ces derniers devront être configurés avec des adresses " site " identiques mais avec des adresses " unit " différentes (0 à 15).

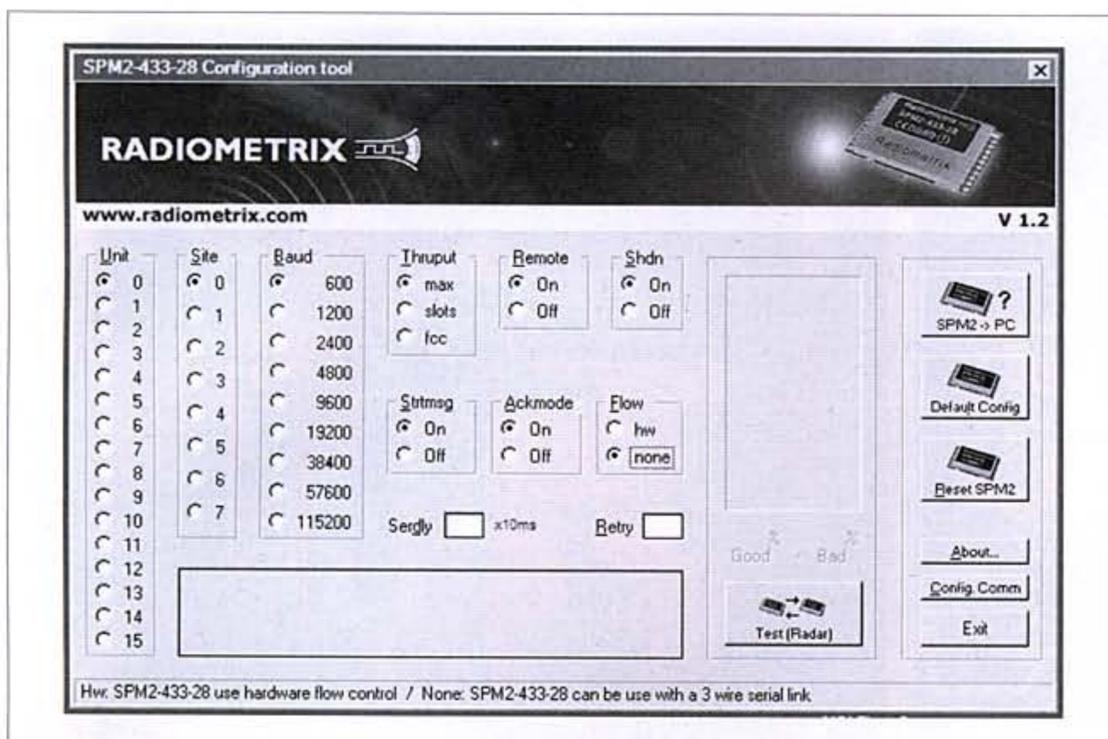
Le schéma de principe

Le schéma de principe de notre réalisation est donné en **figure 6**. Il n'est ni plus ni moins que le schéma d'application préconisé par RADIOMETRIX. Nous avons cependant éliminé la possibilité d'alimenter la platine par le port série du PC. Cette alimentation n'existe pas sur tous les ordinateurs. Nous avons préféré incorporer une petite batterie 9V sur le circuit imprimé, le circuit électronique ne

consommant pas un courant important. Un connecteur DB9 amène les signaux série du PC à un adaptateur de niveaux de tensions de type MAX232. Les signaux DCD, DSR et CTS sont ici utilisés. Quatre LED signalent les étapes du fonctionnement du module SPM2 :

- D1 visualise la réception d'un signal RF.
- D2 signale le passage en émission.
- D3 signale le passage en réception.
- D4 signale la réception d'un " packet " valide.

La LED D5 évite de laisser inutilement le



5 Une vue d'écran du logiciel de configuration

module sous tension. La platine est alimentée sous une tension de 5V générée par un régulateur à faible tension de déchet. Ainsi, lorsque la pile ou la batterie commencera à être relativement déchargée, la régulation fonctionnera encore.

Une antenne est bien évidemment nécessaire. Si l'on souhaite obtenir la portée maximale, il conviendra de choisir des antennes accordées, lesquelles seront fixées sur un bon plan de masse. Si l'on se contente d'une portée de quelques dizaines de mètres, un simple morceau de fil rigide, de longueur appropriée toutefois, conviendra fort bien.

La réalisation

Le tracé du circuit imprimé est donné en **figure 7**. et la **figure 8** donne le schéma d'implantation des composants. Il convient évidemment de réaliser deux platines identiques. Le câblage ne présente aucune difficulté, si ce n'est le nombre relativement important de straps à implanter.

Attention aux supports utilisés pour les SPM2-433-28. En effet, contrairement aux apparences, ces modules étant très miniaturisés, le pas utilisé pour leurs broches de sorties est de 2 mm et non 2,54 mm comme on pourrait le penser. Les circuits intégrés MAX232 sont également insérés dans des supports.

La place laissée sur les platines pour le régulateur de tension permet d'utiliser un régulateur de type LM2931-AZ5 (boîtier T092) ou

un LM2940-CT5 (boîtier TO220).

Les commutateurs J1 et J2 (DEFAULT et SETUP) sont constitués par des morceaux de barrettes sécables de picots à trois points sur lesquels on place des cavaliers.

Il est également prévu sur les platines un emplacement pour l'implantation d'un bornier à vis à deux points si l'on souhaite les alimenter au moyen de blocs secteurs externes.

Les essais

Aucun réglage n'est nécessaire. Il suffit simplement de connecter les modules SPM2 aux PC puis de les mettre sous tension. Pour le

Nomenclature

Résistances :

R1, R2, R3, R6 : 470Ω (jaune, violet, marron)
R4, R5 : 10 kΩ (marron, noir, orange)
R7 : 1 kΩ (marron, noir, rouge)

Condensateurs :

C1, C2, C3, C4 : 1 µF/16 V
C5 : 10 µF/16 V
C6 : 100 µF/16 V
C7, C8 : 100 nF

Semi-conducteurs :

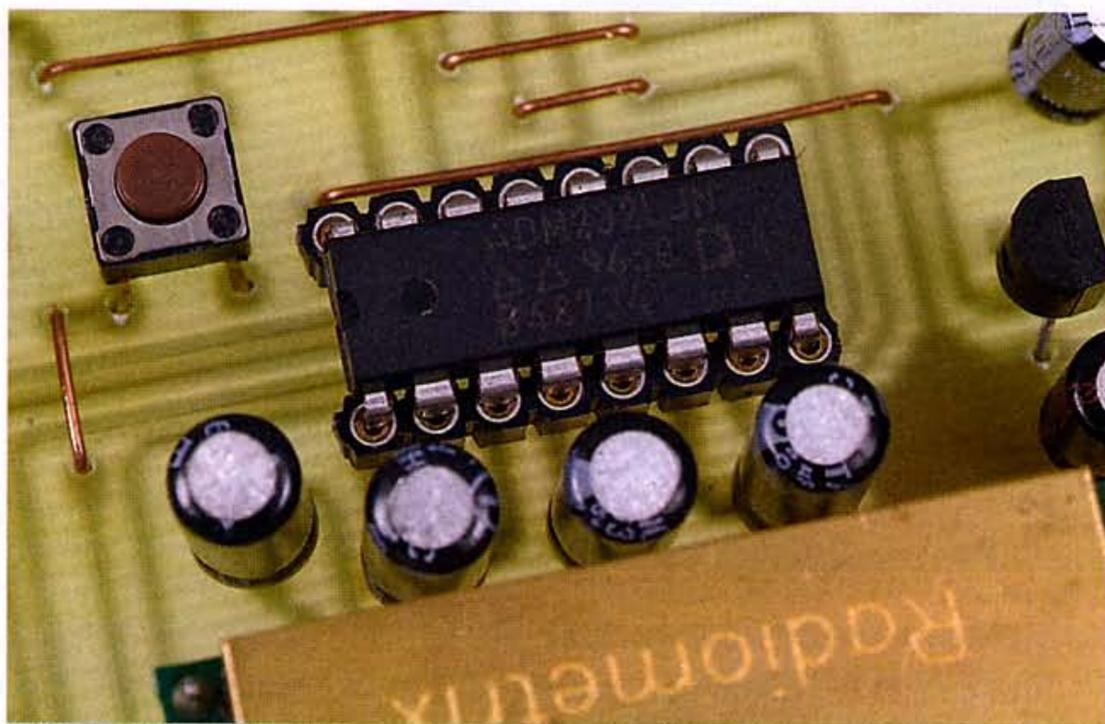
D1, D2, D3, D4, D5 : diodes électroluminescentes de couleurs différentes

Circuits intégrés :

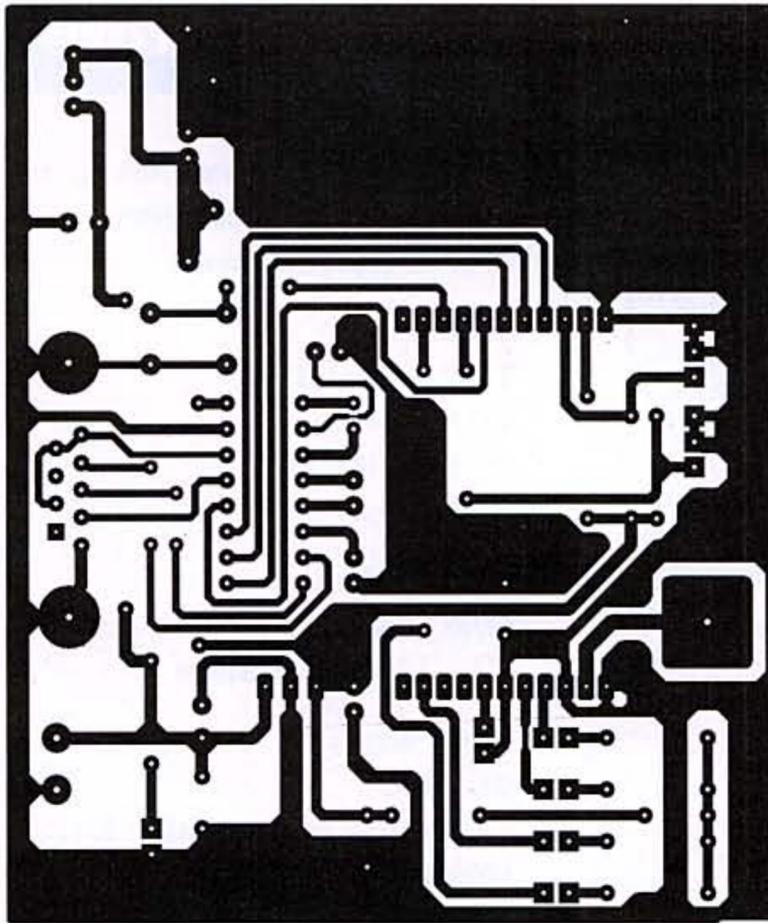
IC1 : MAX232
IC2 : régulateur de tension LM2931-AZ5 ou LM2940-CT5
IC3 : module SPM2-433-28 (LEXTRONIC)

Divers :

2 barrettes supports pour module SPM2-433-28 (LEXTRONIC)
2 morceaux de barrette sécable de picots 3 points
2 cavaliers
1 support pour circuit intégré 16 broches
1 connecteur SUBD femelle 9 broches coudées à 90° pour circuit imprimé
1 antenne accordée (voir texte)
1 connecteur pour pile 9V
1 interrupteur au pas de 2,54 mm
1 bornier à vis à deux points (voir texte)



L'adaptateur de niveaux de tensions de type MAX232



7 Tracé du circuit imprimé

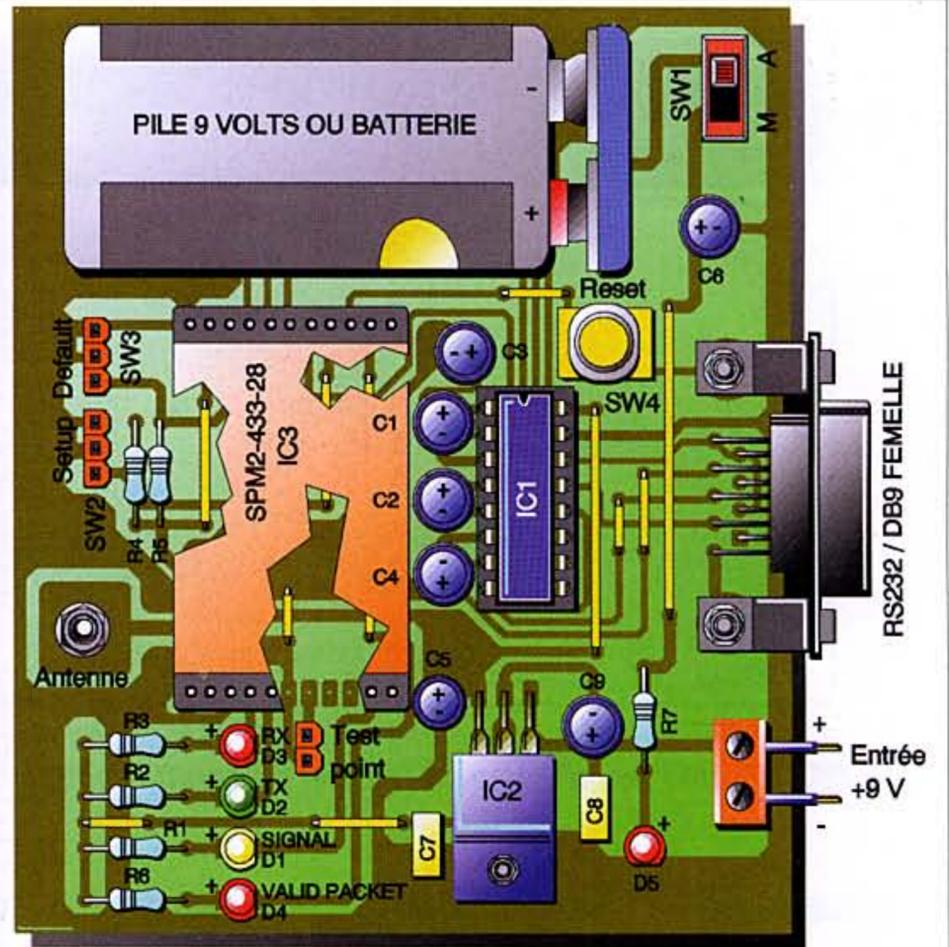
paramétrage des modules, on se référera aux indications données plus haut. Tout doit fonctionner immédiatement.

On peut télécharger le logiciel de configuration " SPM2 configuration tool " sur le site de

LEXTRONIC (<http://www.lextronic.fr>). Il suffit ensuite de le décompresser et de procéder à son installation. Pour activer le mode RADAR, le premier SPM2 doit être relié au PC tournant sous HyperTerminalTM, tandis que le

second, configuré en mode SETUP, doit être relié au second PC avec le logiciel de configuration " SPM2 configuration tool ".

P. OGUIC
Patrice.oguic@tiscali.fr



8 Implantation des éléments

La commande des téléviseurs et des moniteurs



Les moniteurs et les téléviseurs modernes disposant d'une prise péritélévision reçoivent deux sortes de signaux de nature différente : les signaux de composantes couleurs et les signaux de synchronisation. Qu'il s'agisse de moniteurs ou de téléviseurs, les signaux de composantes couleurs sont au nombre de trois, un pour chaque couleur fondamentale. Le rouge, le vert et le bleu, qui sont à la base du principe de synthèse additive des couleurs permettant d'obtenir, par simple combinaison, la totalité des couleurs visibles.

Le cas des signaux de synchronisation est différent. Un moniteur en nécessite deux: la synchronisation verticale qui marque la fin d'un écran (appelé encore trame) et le début du suivant, et la synchronisation horizontale qui, pour sa part, indique la fin d'une ligne et le début de la suivante. Ces deux signaux sont par contre mixés en un seul dans le cas du téléviseur.

Les signaux de commande des téléviseurs PAL/SECAM

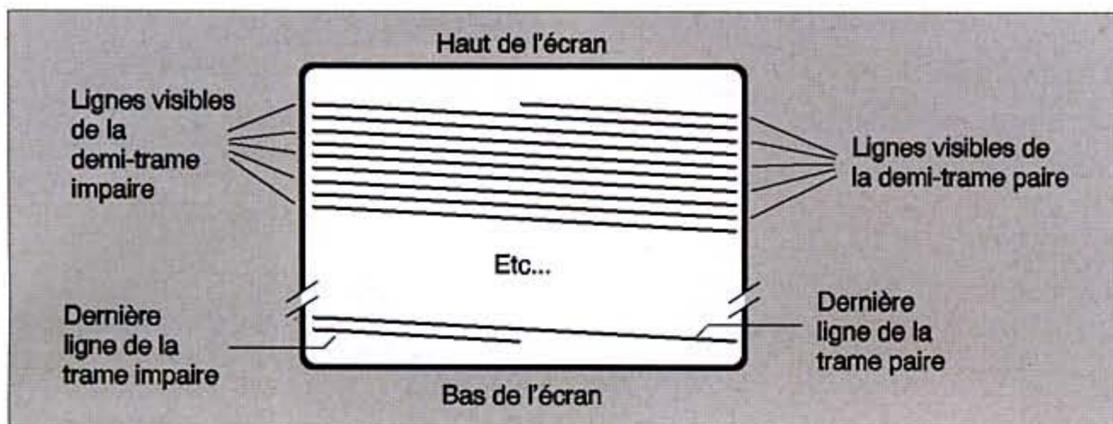
Dans ce format, une image ou trame est constituée de 625 lignes d'une

durée de 64 μ s. Cependant, pour éviter des effets désagréables de scintillement, cette image est affichée sous forme de deux demi-trames interlignées de 312,5 lignes qui se succèdent à la fréquence de 50 Hz. La **figure 1** illustre ce procédé d'interlignage.

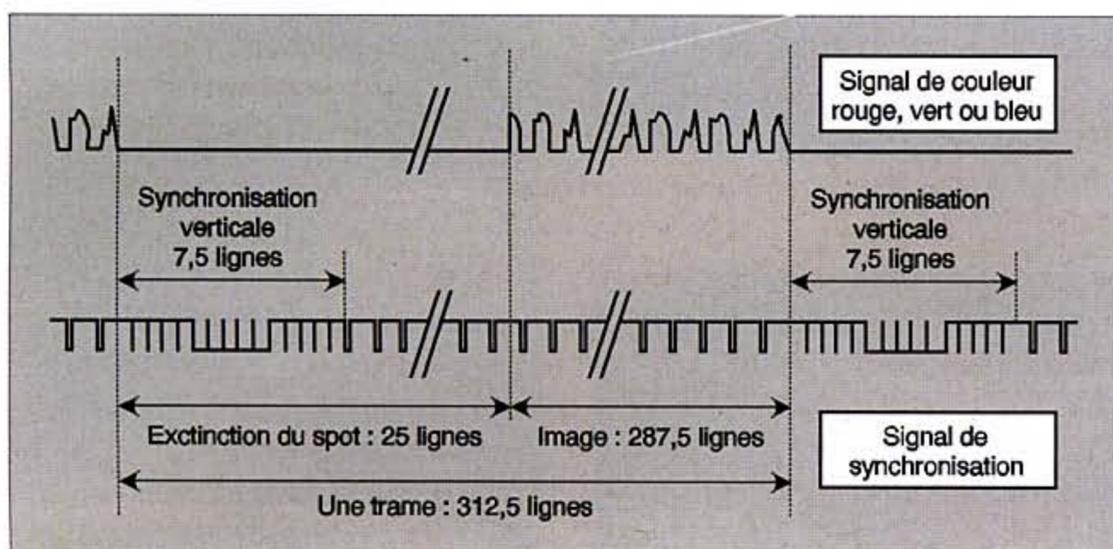
La **figure 2** décrit la composition d'une demi-trame, qui débute par une série de quinze impulsions de 32 μ s servant de signal de synchronisation verticale, dont le détail est donné sur la **figure 4**. La fréquence double de ces signaux par rapport à celle des impulsions lignes de 64 μ s, ainsi que le long niveau bas engendré par les cinq impulsions de synchronisation trame,

permettent au téléviseur de les identifier et donc de déclencher le retour du spot vers le haut de l'écran pour se préparer à l'affichage de la trame suivante. Cette préparation s'étend sur une durée de 25 lignes durant laquelle le spot doit être éteint comme on le voit sur le tracé du haut de la **figure 2**. Une fois cette phase achevée, la demi-image est transmise sur la durée restante de 287,5 lignes avant de passer à la demi-trame suivante.

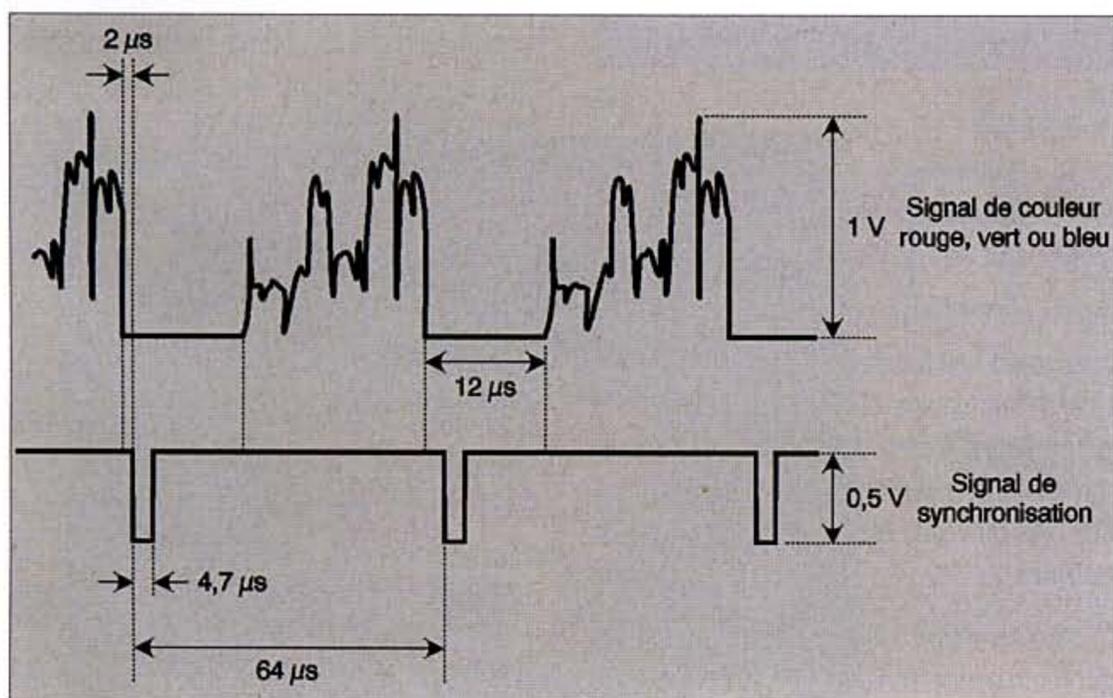
Comme on peut le remarquer sur la **figure 4a**, la demi-trame paire se termine par une ligne complète de 64 μ s suivie des impulsions de pré-égalisation et de synchronisation trame, alors qu'une demi-trame impaire



1 Principe de l'affichage entrelacé



2 Composition d'une trame d'un signal vidéo de télévision



3 Synchronisation ligne des signaux de télévision

(figure 4b) se termine par une demi-ligne interrompue par les mêmes impulsions de pré-égalisation et synchronisation trame. Symétriquement, elles débutent toutes les deux de façon différente, ces décalages servant justement à permettre de réaliser l'interlignage évoqué précédemment. On remarquera que, même si la norme fixe le

nombre de lignes de l'image à 625, toutes ces lignes ne sont pas pour autant affichées, puisque l'on en perd 25 par demi-trame, soit au total 50 par image. Il ne restera donc au final que 575 lignes réellement visibles à l'écran.

La synchronisation ligne est exposée sur la figure 3 et est particulièrement simple. Elle

ne comporte que des impulsions négatives d'une durée de 4,7 μ s à intervalles réguliers de 64 μ s. Tout comme dans le cas du retour du spot en fin de demi-trame, il est nécessaire d'éteindre le spot lors du retour ligne, plus précisément 1 à 2 μ s avant l'impulsion de synchronisation et environ 10 μ s après cette impulsion, soit un total de 12 μ s. La partie visible de la ligne s'étend donc sur une durée de 52 μ s, durant laquelle les signaux Rouge, Vert et Bleu sont actifs (voir figure 3).

Il faut noter que toutes les durées indiquées dans le paragraphe qui précède ne sont pas toujours rigoureusement respectées par les différents émetteurs, que ce soit des consoles de jeux, des lecteurs DVD ou des émissions de télévision. De même, si l'on veut générer les impulsions de synchronisation avec un microcontrôleur PIC cadencé à 4 MHz, il est tout à fait possible d'amener la durée des impulsions de synchronisation ligne à 5 μ s au lieu de 4,7 μ s et de ramener celles de la synchronisation verticale à 2 μ s au lieu de 2,5 μ s.

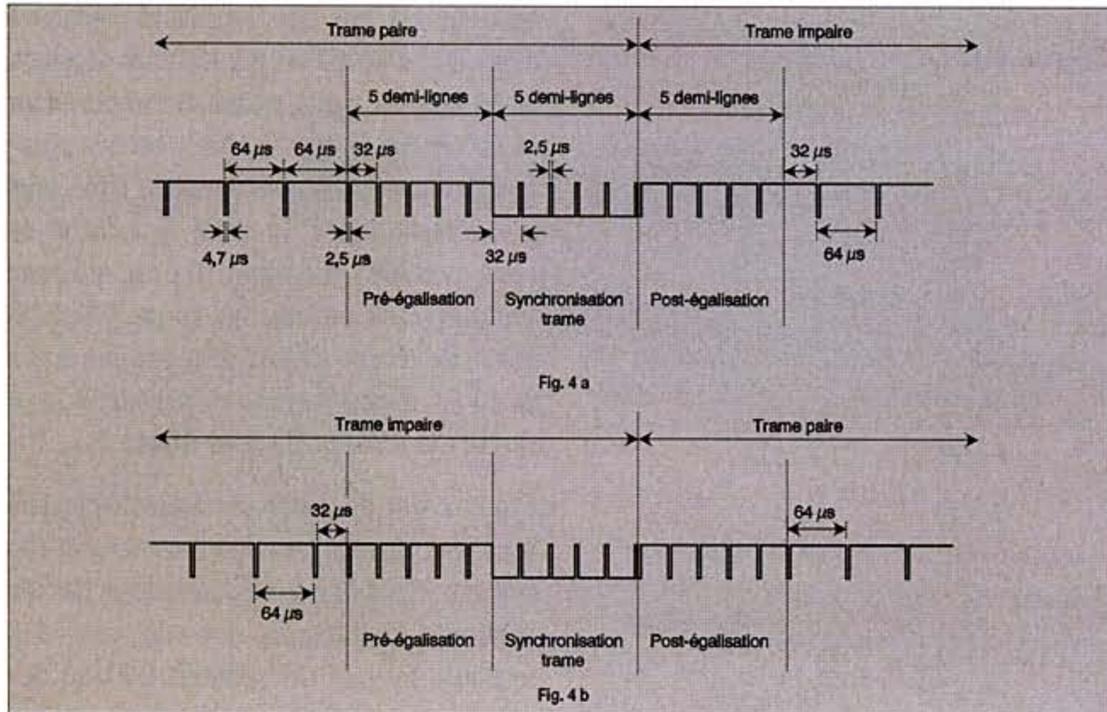
Cela fonctionne parfaitement. Par contre, la durée de 64 μ s d'une ligne et la fréquence générale des demi-trames de 50 Hz doivent être respectées, ce qui est aisé avec le microcontrôleur indiqué.

Le brochage de la prise péritel qui permet d'interfacer le téléviseur avec l'extérieur est donné sur la figure 8 en fin d'article. On notera la présence de l'entrée "commutation rapide" qui doit être alimentée sous 5 V par l'intermédiaire d'une résistance de 75 Ω afin d'obliger le téléviseur à prendre en compte les signaux de composante couleur présents sur les entrées Rouge, Vert et Bleu.

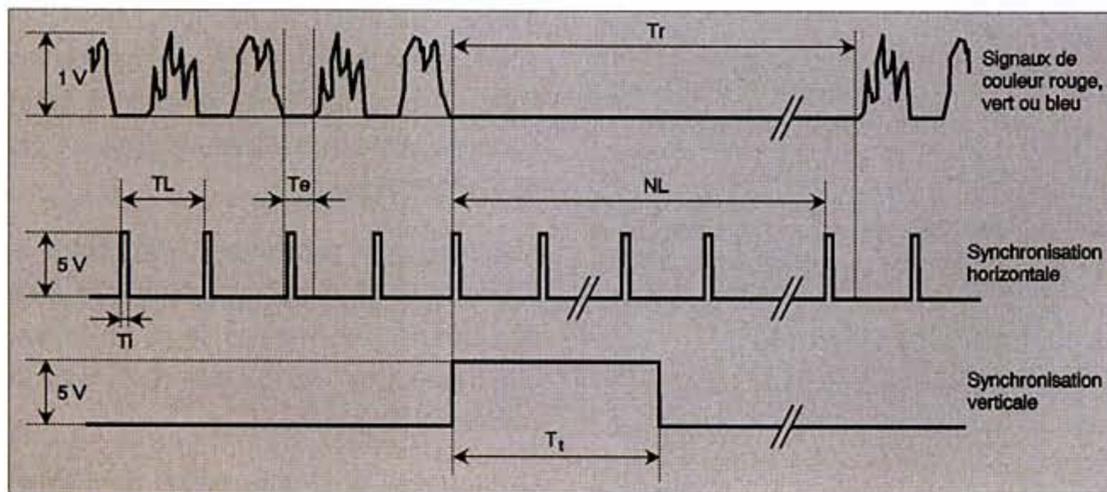
Les signaux de commande des moniteurs

Malgré les apparences, les moniteurs (actuels) et téléviseurs ne sont pas compatibles et il n'existe aucune façon simple d'adapter les signaux destinés à l'un afin d'afficher une image sur l'autre.

En premier lieu, l'augmentation des fréquences de balayage a fait que le procédé d'interlignage n'était plus nécessaire pour éviter le scintillement de l'écran et a été abandonné. A présent, l'écran est affiché en entier en une fois, à une fréquence minimale de 56 Hz et jusqu'à 85 Hz ou même 100 Hz. En conséquence, la fréquence ligne a plus



4 Synchronisation trame des signaux de télévision



5 Signaux de commande des moniteurs

que doublé par rapport à celle des signaux de télévision comme l'indique le tableau récapitulatif de la **figure 7**.

Une autre différence avec les signaux de télévision qui a déjà été mentionnée est que les signaux de synchronisation horizontale et verticale sont transmis sur des voies séparées et qu'ils sont au format TTL.

Les résolutions affichables sur un moniteur sont normalisées : les plus basses et donc les plus accessibles pour une réalisation personnelle sont 640x480 et 800x600 mais certains modèles permettent d'atteindre une résolution de 1600x1200. On notera qu'il n'y a en fait aucun lien entre la résolution verticale et horizontale et en ce qui concerne les signaux de synchronisation, seules comptent la résolution verticale et la fréquence de balayage.

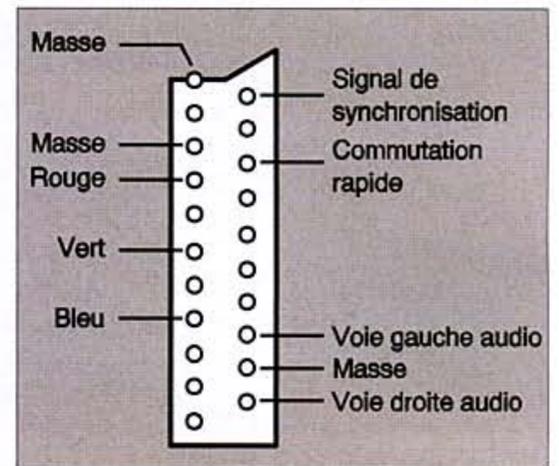
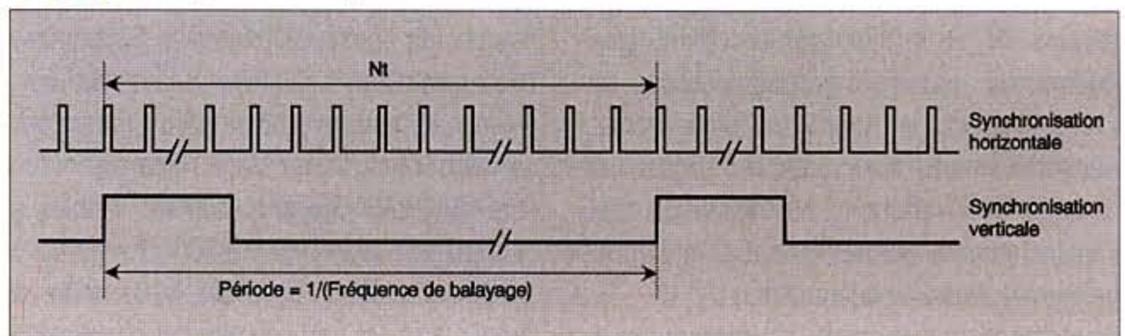
Enfin, lorsque l'on annonce une résolution verticale de 480 lignes, cette valeur correspond au nombre de lignes visibles, le nombre de lignes total par écran étant supérieur du

fait du temps perdu durant le retour du spot vers le haut de l'écran.

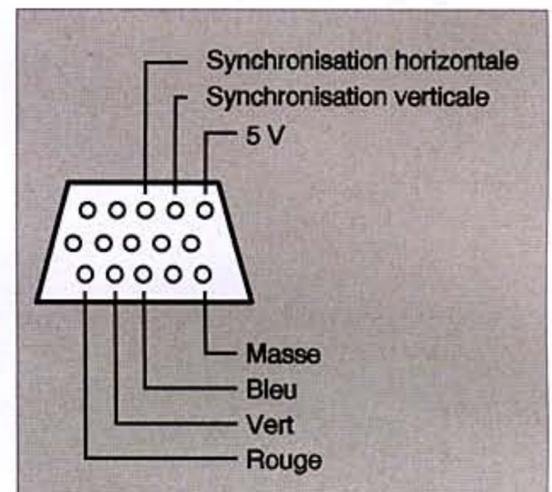
Les **figures 5** et **6** décrivent les signaux de commande des moniteurs. Ils sont de constitutions semblables à ceux de télévision et même plus simples dans le cas de la synchronisation verticale. Les différentes variables renvoient aux valeurs du tableau de la **figure 7**.

Ce tableau décrit les valeurs des différents paramètres caractéristiques en fonction de la

6 Signaux de synchronisation des moniteurs



8 Embase péritel d'un téléviseur



9 Embase SUBD haute densité

résolution et de la fréquence de balayage choisies. (Les fréquences de balayage entre parenthèses correspondent aux valeurs théoriques dont tentent de s'approcher les valeurs réellement mesurées).

Dans l'exemple donné sur les **schémas 5** et **6**, les impulsions de synchronisation horizontale et verticale sont positives c'est à dire qu'elles correspondent à des niveaux haut du signal. Cela n'est le cas que pour une résolution de 600 lignes (800x600). Dans le cas des résolutions 480 lignes (640x480) et 768 lignes (1024x768), les impulsions sont négatives, c'est à dire qu'elles correspondent à des niveaux bas. C'est ce qu'indique la rubrique "inversion de polarité" du tableau de la **figure 7**.

Toutes les valeurs de ce tableau ont été

Résolution	640x480	800x600	1024x768	640x480	800x600	640x480	800x600
Fréquence de balayage	59,96 Hz (60 Hz)	60,28 Hz (60 Hz)	60,14 Hz (60 Hz)	75,29 Hz (75 Hz)	74,85 Hz (75 Hz)	85,38 Hz (85 Hz)	84,81 Hz (85 Hz)
Fréquence des lignes	31,48 kHz	37,86 kHz	48,48 kHz	37,65 kHz	46,78 kHz	43,46 kHz	53,52 kHz
Durée de l'impulsion ligne : TL	3,6 μ s	2,8 μ s	2 μ s	1,6 μ s	1,6 μ s	1 μ s	1 μ s
Durée de l'impulsion trame : Tt	60 μ s	50 μ s	110 μ s	60 μ s	60 μ s	50 μ s	50 μ s
Durée de l'extinction retour/ligne : Te	6 μ s	6 μ s	5 μ s	5 μ s	5 μ s	4,5 μ s	4,5 μ s
Durée de l'extinction retour/écran : Tr	1 430 μ s	740 μ s	780 μ s	531 μ s	534 μ s	667 μ s	579 μ s
Inversion de la polarité des impulsions	Oui	Non	Oui	Oui	Non	Oui	Non
Nombre total de lignes : Nt	525	628	806	500	625	509	631
Nombre de lignes durant l'extinction : NL	45	28	38	20	25	29	31
Durée d'une ligne : TL	31,8 μ s	26,4 μ s	20,6 μ s	26,6 μ s	21,4 μ s	23 μ s	18,7 μ s

7 Tableau des durées des signaux de synchronisation des moniteurs

mesurées par l'auteur car il est difficile de trouver des renseignements à leur sujet. Plusieurs cartes graphiques ont été étudiées pour cela et elles ont toutes fourni les mêmes

valeurs, sauf pour l'impulsion trame (Tt) dont on n'a gardé que les valeurs minimales. Il semblerait ainsi que toute valeur de ce paramètre comprise entre 50 μ s et 110 μ s (ou

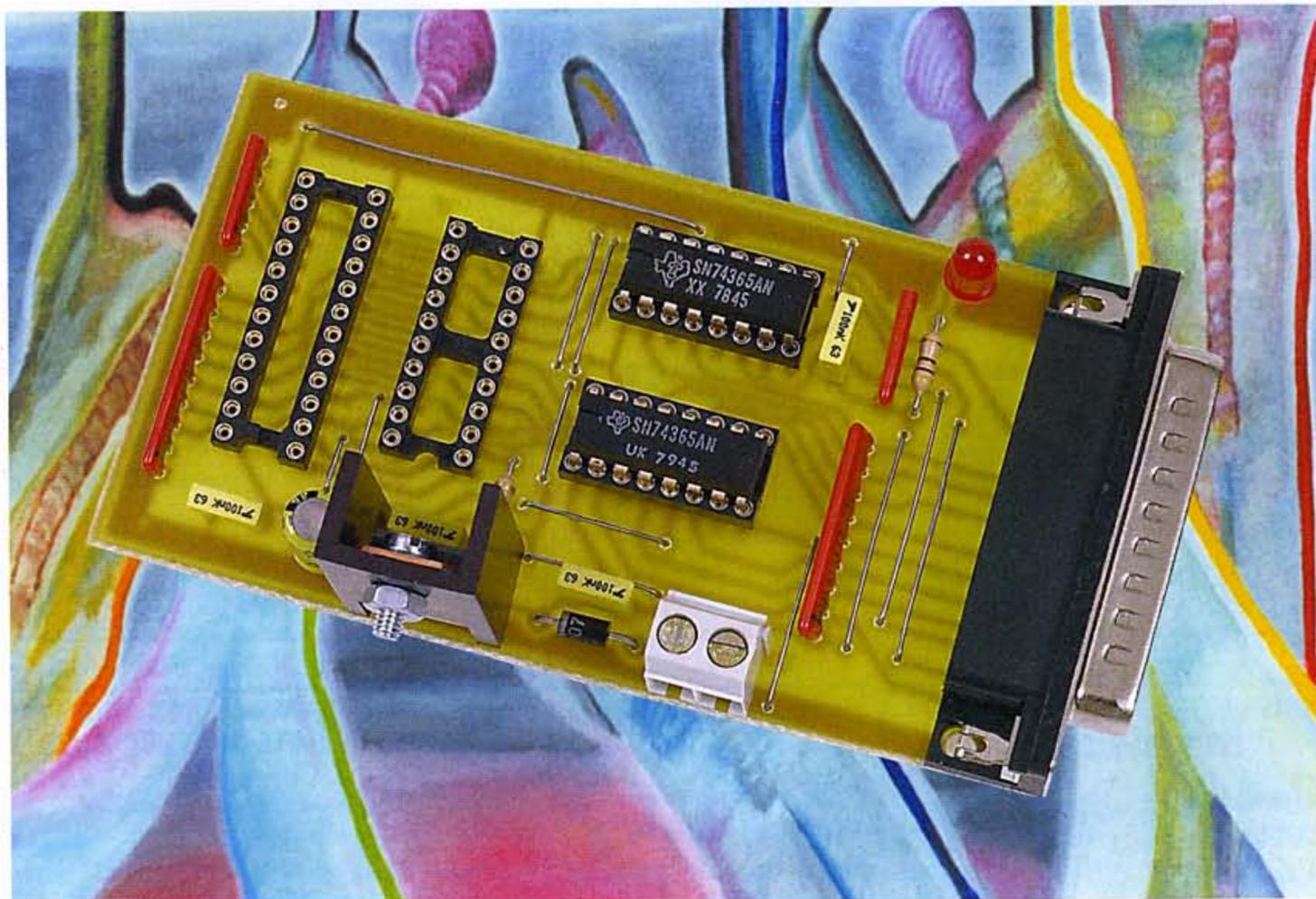
même plus) convienne. Il n'a pas été possible de déterminer le temps séparant le début de l'extinction du signal de composante couleur de la venue de l'impulsion de synchronisation horizontale qui la suit. On peut supposer que ce temps est inférieur à la microseconde.

Si l'on divise la fréquence ligne par la fréquence de balayage, on remarque qu'elles ne sont pas des multiples exacts. Il faut donc qu'à un moment donné, par exemple lors de l'impulsion de retour trame (comme cela est illustré sur la **figure 6**), les deux signaux soient re-synchronisés ensemble. Dans le cas contraire, l'image glisserait continuellement vers le haut ou vers le bas de l'écran. Dans le cas d'une réalisation d'un générateur de signaux de synchronisation pour un moniteur à l'aide d'un microcontrôleur, il sera de loin préférable de choisir des fréquences ligne et de balayage rigoureusement multiples, en tâchant de ne pas s'écarter de plus de quelques pour cents des valeurs indiquées dans le tableau.

Le brochage d'une embase Sub-D haute densité est donné sur la **figure 9**.

O. VIACAVA

Programmeur de GAL 22V10 et 16V8



Les GAL 22V10 et 16V8 en boîtier DIP sont souvent utilisés lorsque l'on a besoin de réaliser un traitement logique relativement simple ne nécessitant pas l'emploi d'un CPLD. Le montage que nous vous proposons ici vous sera d'une grande utilité pour télécharger un fichier de type jedec dans un GAL 22V10 ou 16V8 de chez LATTICE.

Les composants logiques programmables ont pris une place importante dans le monde de l'électronique. LATTICE propose sur son site le logiciel ISPlever qui permet de créer le fichier Jedec nécessaire à la programmation de ces composants.

Description du fonctionnement

Le schéma structurel est représenté aux figures 1a et 1b.

Le bornier J2 sert à alimenter la carte en +12 V. L'alimentation +5 V des circuits intégrés est assurée par le régulateur U5. La mise sous tension est indiquée grâce à la diode électroluminescente D2, limitée en intensité par

R1. La diode D1, quant à elle, prévient d'une inversion de polarité. C6 est un condensateur de filtrage et C1 à C4 des condensateurs de découplage. La communication avec le PC s'établira par le port parallèle et le connecteur J1. Les circuits U2 et U3 servent de tampons entre le GAL à programmer et le port parallèle. Les réseaux RES1 à RES4 sont des résistances de "tirage" qui fixent l'état haut à +5 V en absence de signaux sur les entrées.

La réalisation

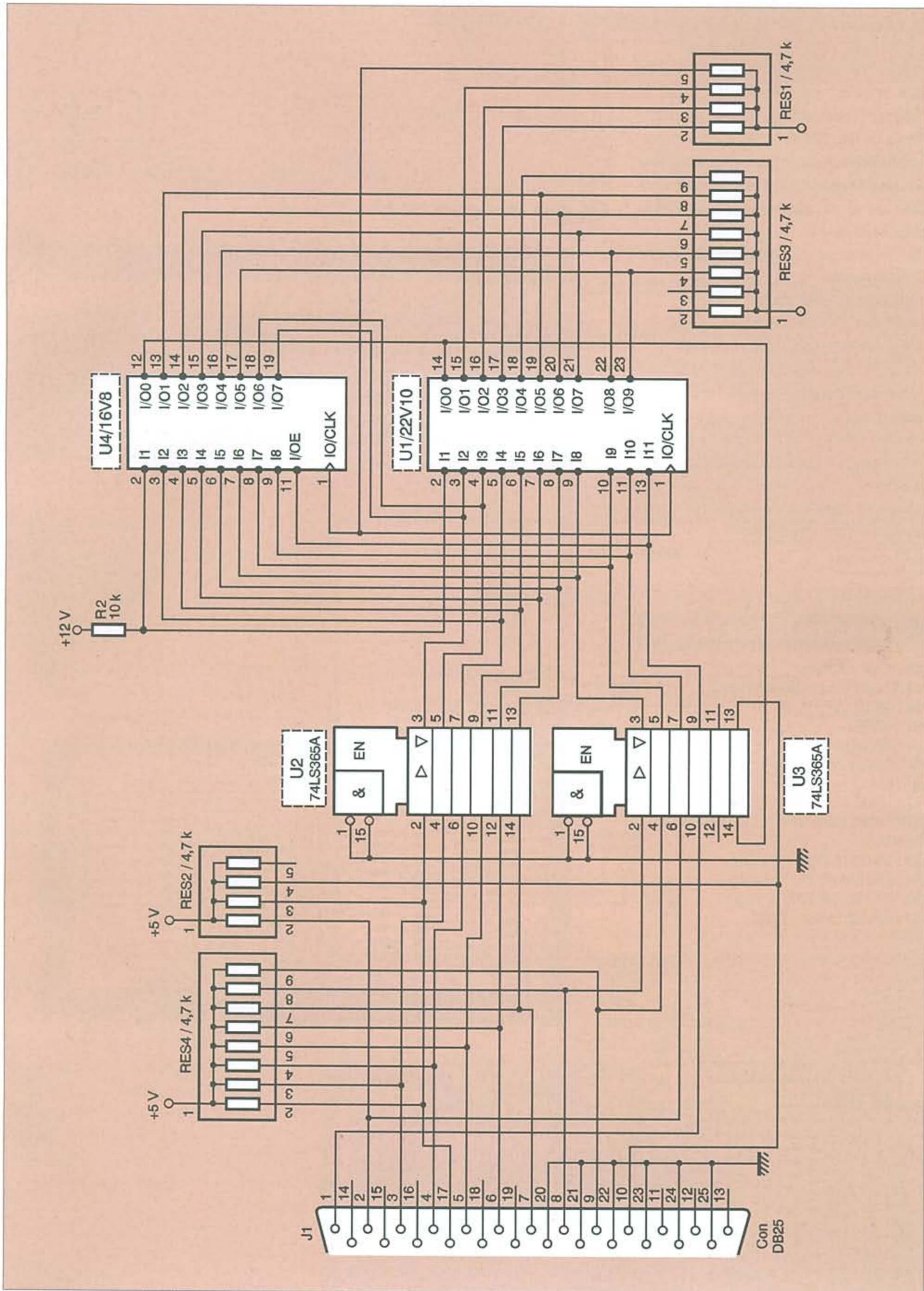
Le schéma d'implantation est donné figure 3, le typon est donné figure 2. Le montage est réalisé sur un circuit imprimé simple face et comporte 13

straps qu'il faudra câbler en premier. Montez ensuite les supports de circuits intégrés puis les composants passifs et finissez par le connecteur.

NB : vous pouvez couper facilement un réseau de résistances SIP9 (8 résistances et 1 point commun) pour le transformer en SIP5 (4 résistances et 1 point commun).

Il est conseillé d'équiper le régulateur d'un dissipateur.

Une fois la carte montée sans les circuits intégrés, alimentez-la : la LED D1 doit s'allumer. Vérifiez la tension +5V sur les supports des circuits intégrés. Coupez l'alimentation puis placez les circuits intégrés U2 et U3. Votre carte est prête à fonctionner, elle comporte tous les composants à l'exception des GAL U1 et U4.



1a Schéma de principe

Utilisation

Vous devez d'abord télécharger un logiciel libre d'accès, appelé GALBLAST.exe, à l'adresse : www.geocities.com/mwinterhoff Placez un GAL 22V10 ou 16V8 sur son support. Attention, vous ne devez pas placer deux GAL simultanément sur la carte. Connectez la carte sur le PC puis alimentez-la. Ouvrez ensuite GALBLAST.

Une fois lancé, ce logiciel se présente comme sur la **figure 4**.

L'utilisation est très simple :

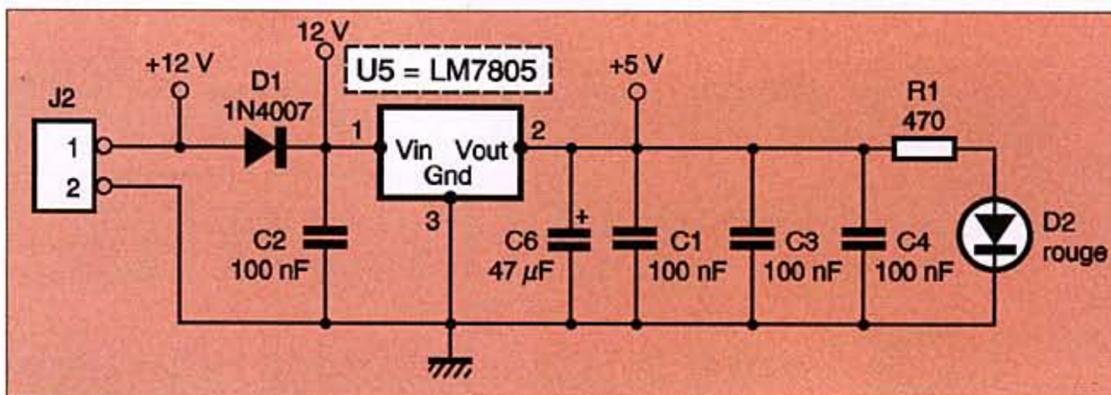
- sélectionnez le port parallèle et le type de GAL que vous avez placé sur la carte,
- chargez le fichier Jedec (load JEDEC),
- procédez à l'écriture du fichier dans le GAL (write GAL),
- vérifiez l'écriture du fichier (verify GAL).

Vous pouvez aussi effacer le GAL (erase GAL) avant de le reprogrammer. Nous laisserons les autres fonctions aux utilisateurs plus avertis. Bonne programmation !

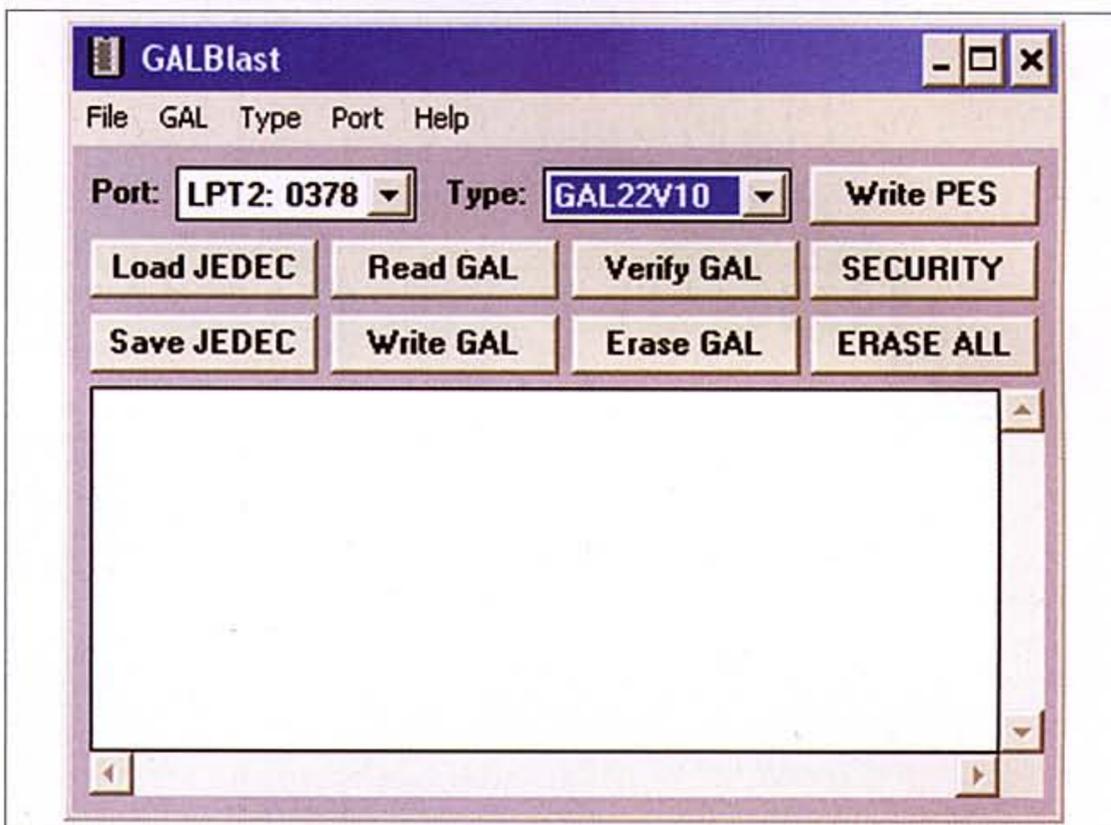
L. JOSSE

Nomenclature

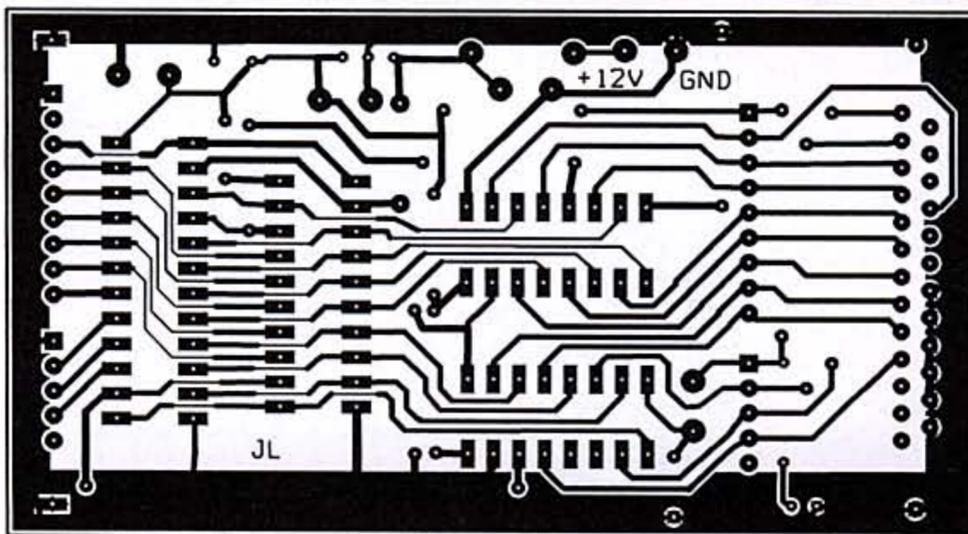
- C6 : 47 μ F chimique
- C1, C2, C3, C4 : 100 nF céramique
- D2 : Led rouge
- D1 : 1N4007
- J1 : DB25M coudé
- J2 : Bornier 2 plots
- R1 : 470 Ω
- R2 : 10 k Ω
- RES1, RES2, RES3, RES4 : 4.7 k Ω Réseau SIP9 et SIP5
- U1 : GAL22V10 DIP24 + support
- U4 : GAL16V8 DIP20 + support
- U2, U3 : 74LS365 DIP16 + support
- U5 : LM7805 (boîtier TO220)



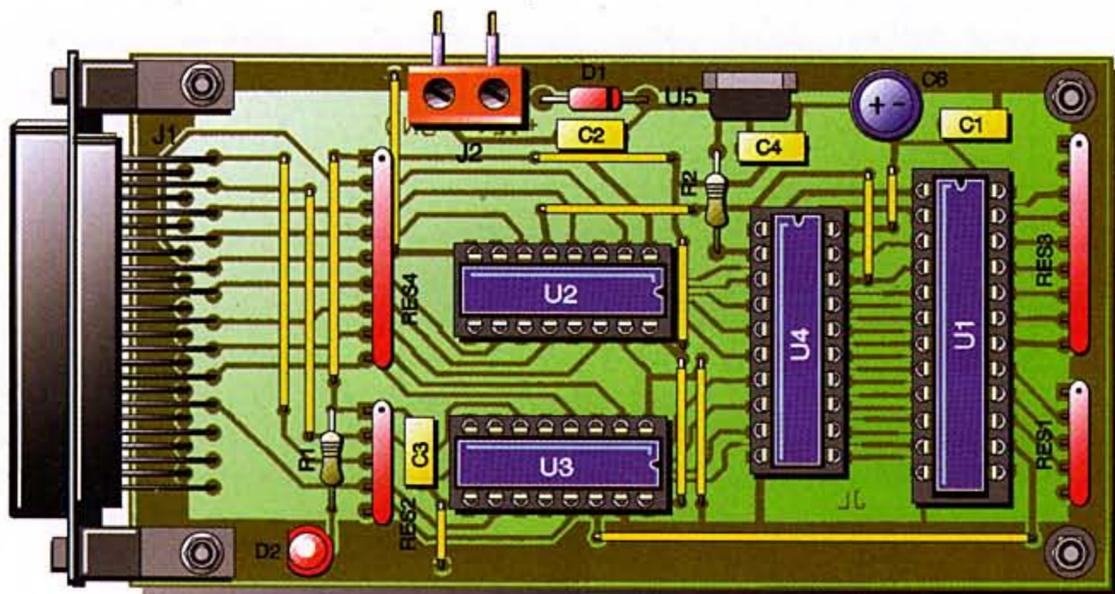
1b Alimentation stabilisée de +5V



4 Logiciel GALBLAST.exe

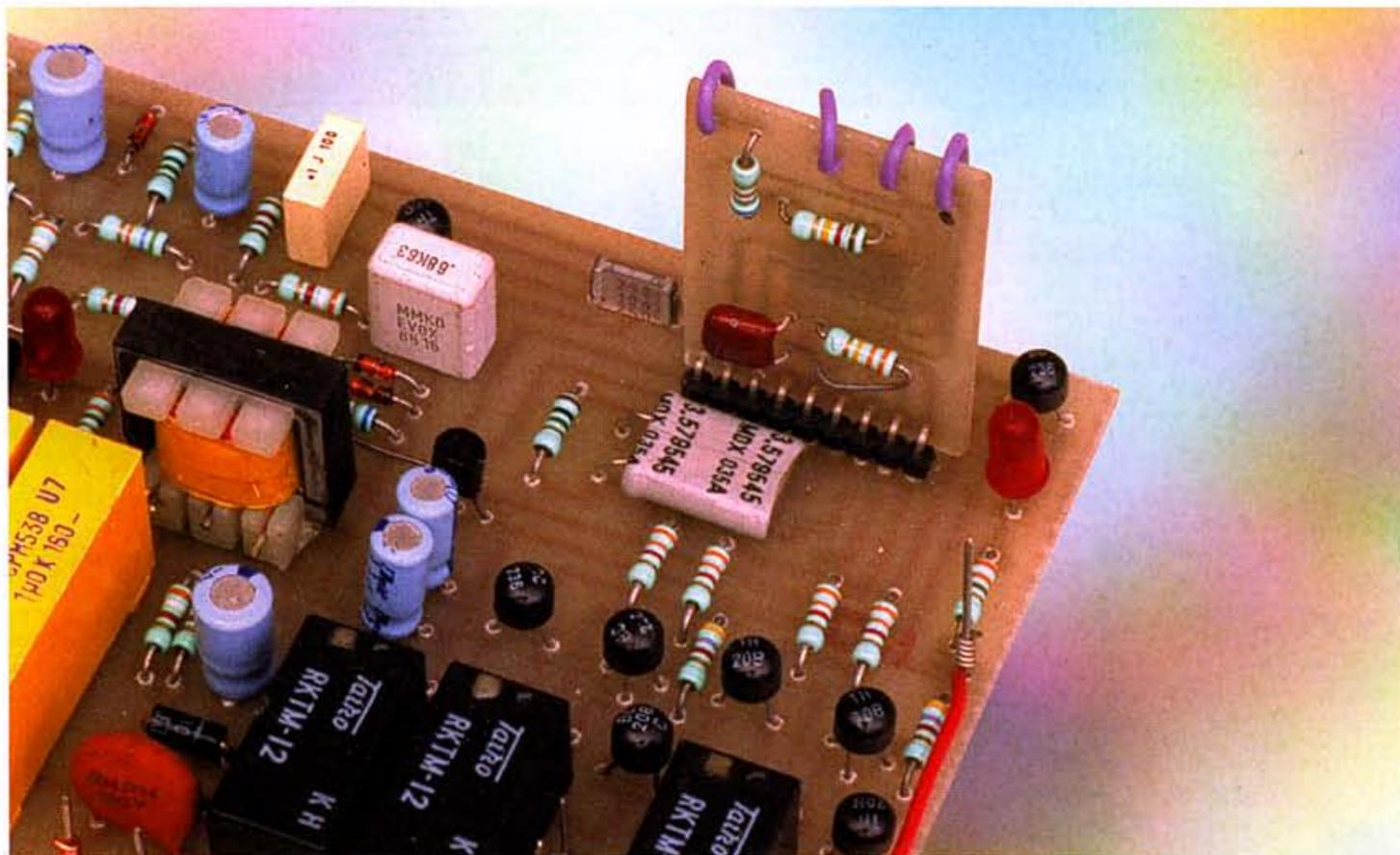


2 Tracé du circuit imprimé



3 Implantation des éléments

Codage et décodage dtmf



Développé à l'origine pour composer des numéros de téléphones, le codage à "fréquences vocales" DTMF a su s'imposer dans d'innombrables autres applications, notamment la commande à distance. Face à la concurrence d'approches purement logicielles, les circuits intégrés codeurs et décodeurs se font plus rares sur le marché, obligeant souvent à remettre en cause les schémas les plus éprouvés.

Une norme incontournable

C'est dès le milieu des années soixante que les postes téléphoniques américains ont commencé à bénéficier de la modernisation consistant à remplacer la numérotation "décimale" (celle des antiques cadrans rotatifs) par la transmission de paires de fréquences audibles, propres à chaque chiffre. Initialement mis au point à l'initiative des laboratoires Bell, le codage correspondant a été adopté en tant que norme internationale par des organismes officiels comme le CCITT ou, pour l'Europe, la CEPT. Rappelons que huit fréquences situées dans la bande passante "téléphonique" 300-3400 Hz se répartissent en deux groupes :

- Groupe "bas" (697, 770, 852, et 941 Hz).
- Groupe "haut" (1209, 1336, 1447, et 1633 Hz).

Même en laissant de côté la fréquence assez peu utilisée en téléphonie

de 1633 Hz, cela permet de définir douze paires composées d'une tonalité du groupe bas, mélangée à une du groupe haut. Le saviez vous ? C'est ainsi que sont apparues les touches "dièse" et "étoile" des claviers téléphoniques, en complément des chiffres de 0 à 9 !

L'affectation des ces fréquences, selon la **figure 1**, n'est en aucune façon le fruit du hasard : des choix très précis ont été opérés, en évitant notamment les fréquences possédant des harmoniques en commun, et en privilégiant au contraire des combinaisons apparaissant le moins possible dans la parole, la musique, ou le bruit ambiant. Il en résulte une grande "robustesse" de codage, même sur des voies de transmission plus difficiles que les lignes téléphoniques (canaux radio, pour des applications d'appel sélectif, par exemple). Cela suppose, en revanche, une précision très stricte des codeurs et des décodeurs, presque toujours "pilotes quartz".

Les moyens de codage

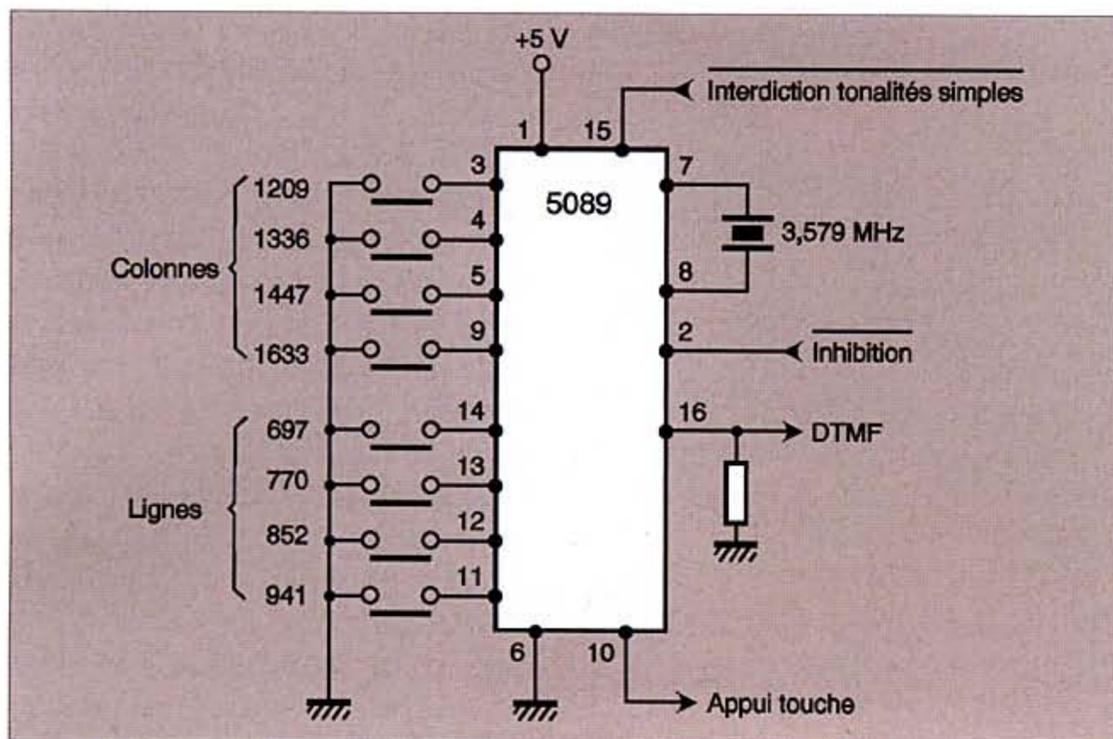
Bien entendu, la production en masse de claviers téléphoniques à bas prix ne pouvait s'envisager sans circuits intégrés spécifiquement dévolus à cette fonction, dont le plus connu est le 5089, disponible sous diverses marques (comme le TCM5089 de Texas Instruments).

Placé sous le contrôle d'un quartz (ou d'un résonateur céramique) de 3,579 MHz, ce composant s'interface directement à un clavier possédant deux contacts par touche (ou à un port de sortie de microcontrôleur !) et syn-



L'affectation des tonalités DTMF.

	Haut	1209	1336	1447	1633
Bas					
697		1	2	3	A
770		4	5	6	B
852		7	8	9	C
941		*	0	#	D



2 Mise en œuvre du TCM5089.

thétise numériquement des signaux quasiment sinusoïdaux.

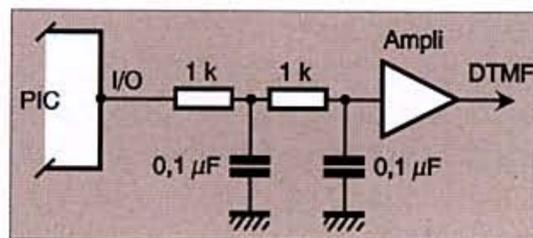
Il est toutefois possible de lui faire produire des tonalités individuelles et non plus des paires de fréquences et également de le "désaccorder" volontairement en utilisant un quartz d'une valeur différente (par exemple 4,43 MHz).

Tout cela lui ouvre, évidemment, d'innombrables horizons...

Moyennant certaines précautions, un résonateur de 455 kHz permet même d'explorer le domaine "subsonique" (appels sélectifs CTCSS, par exemple).

Le schéma d'application de la **figure 2** montre que l'appui sur une touche du clavier est censé mettre simultanément à la masse une entrée "ligne" et une entrée "colonne". Si la broche 15 n'est pas maintenue au niveau bas, on peut mettre l'une ou l'autre individuellement à la masse pour produire une tonalité simple. Dans les deux cas, un niveau haut apparaît sur la broche 10 dès qu'une tonalité est produite, ce qui permet notamment de commander le passage en émission d'un équipement radio (poste CB ou simple module genre TX-FM AUDIO...). Toujours disponible au détail (Ref. 50-7374 du catalogue Sélectronic), le 5089 subit cependant de plein fouet la concurrence de la production logicielle de signaux DTMF. Pour une carte "son" de PC, il s'agit d'une tâche on ne peut plus simple, tandis qu'un microcontrôleur suffisamment rapide peut aussi faire l'affaire.

La **figure 3** montre ainsi comment exploiter, au plus simple, une sortie de PIC pilotée en



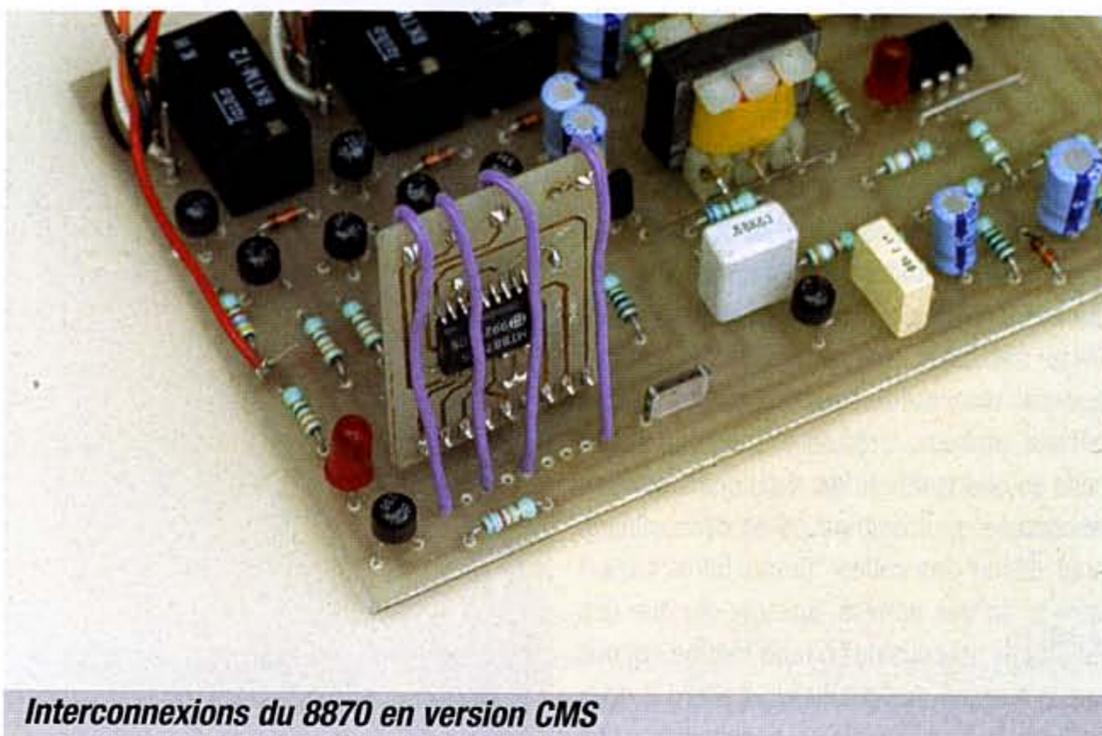
3 Un filtre de sortie rudimentaire pour PIC.

PWM (modulation de largeur d'impulsion), au travers de l'instruction "DTMFOUT" du compilateur Basic "PROTON+" de Crownhill (<http://www.picbasic.org>). C'est alors entièrement par calculs qu'est élaboré le signal, dont l'amplitude instantanée est convertie en rapport cyclique d'un signal tout ou rien de fréquence élevée qu'il est indispensable de

filtrer énergiquement avant usage. À une fréquence d'horloge de 4 MHz, on obtiendra un résultat tout juste acceptable, à vrai dire comparable à ce que donnerait l'association de deux 555. Pour rivaliser avec un 5089, il faut envisager une fréquence d'horloge d'au moins 10 à 12 MHz et un filtre actif passe-bas réglé aux environs de 2 kHz. Cette relative lourdeur de mise en œuvre ne se justifie donc que si un microcontrôleur performant est de toute façon nécessaire pour d'autres tâches. Un mot, enfin, sur la génération de signaux DTMF par les téléphones portables GSM : comme la qualité de transmission n'est pas toujours suffisante pour garantir l'intégrité des paires de tonalités, ce n'est pas le mobile qui produit celles-ci, mais le réseau ! Lorsque l'on presse une touche du clavier en cours de communication, le mobile transmet numériquement au réseau l'ordre de générer la paire de tonalités correspondante. Pour peu que la réception se fasse sur une ligne fixe, la qualité du signal est donc parfaite, puisqu'à aucun moment celui-ci n'a transité par une voie radio exposée à toutes sortes de perturbations.

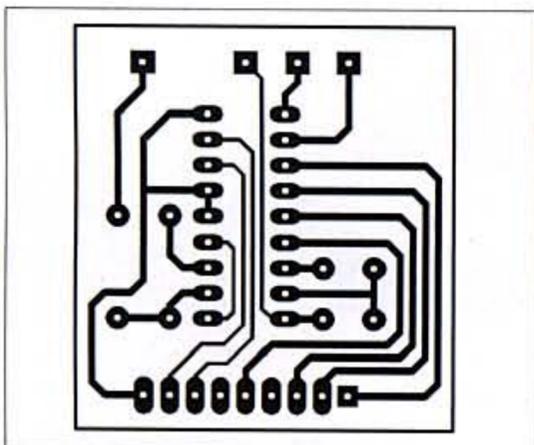
Les moyens de décodage

N'hésitons pas à le dire : dans l'état actuel des choses, il demeure hélas présomptueux de vouloir décoder convenablement des signaux DTMF avec un simple PIC, ou tout autre microcontrôleur comparable. Il est d'ailleurs bon de remarquer que l'instruction "DTMFIN" n'existe dans aucun des compilateurs que nous avons passés en revue ! Pas question, en effet, d'accéder ainsi à la quali-

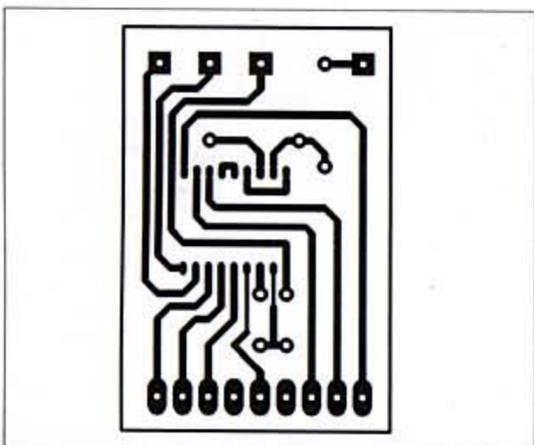


Interconnexions du 8870 en version CMS

agir sur les valeurs de ces composants, guidé par la note d'application du fabricant, s'il souhaite optimiser telle ou telle caractéristique du décodeur (sensibilité d'entrée, temps de réponse, etc.) pour des besoins particuliers. D'un point de vue pratique, la réalisation épouse la forme d'un très petit circuit imprimé muni d'une rangée de 9 picots rigides (broches 10 à 18 du regretté SSI202). Quatre fils assurent la correspondance avec les broches 1, 3, 5, et 9, ce qui permet de positionner le module soit debout (utiliser une barrette sécable à picots soudés), soit à plat (préférer alors des picots droits, de type "à wrapper"). Pas de fausse pudeur : ce

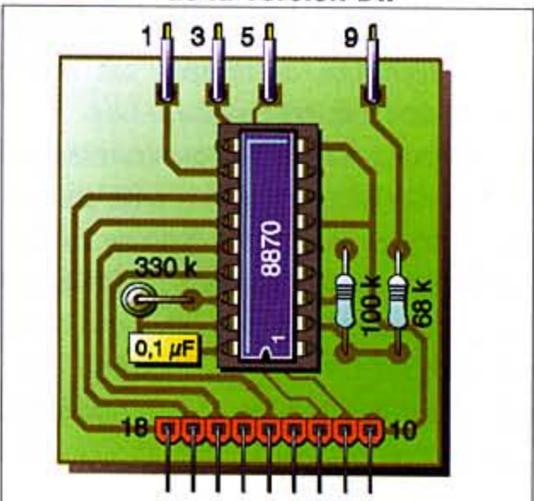


5 Le circuit imprimé de la version DIP.



6 Le circuit imprimé de la version CMS

7 L'implantation de la version DIP



F _{LOW}	F _{HIGH}	KEY (ref.)	OE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
ANY	ANY	ANY	L	Z	Z	Z	Z

9 Le format de sortie du M8870.

genre de bricolage dit "piggy-back" se pratique couramment dans l'industrie !

Le tracé du cuivre est fourni à la **figure 5**, et le plan de câblage à la **figure 7**.

Dans certaines situations, on pourra toutefois être gêné par le fait que la largeur de la carte excède notablement l'empattement de l'empreinte DIP18. On pourra alors se tourner vers une version CMS du 8870, pour laquelle nous avons dessiné le circuit imprimé de la **figure 6**. Bien entendu, le circuit intégré sera cette fois soudé côté cuivre, les composants conventionnels demeurant sur l'autre face, ce qui contribue au gain de place constaté.

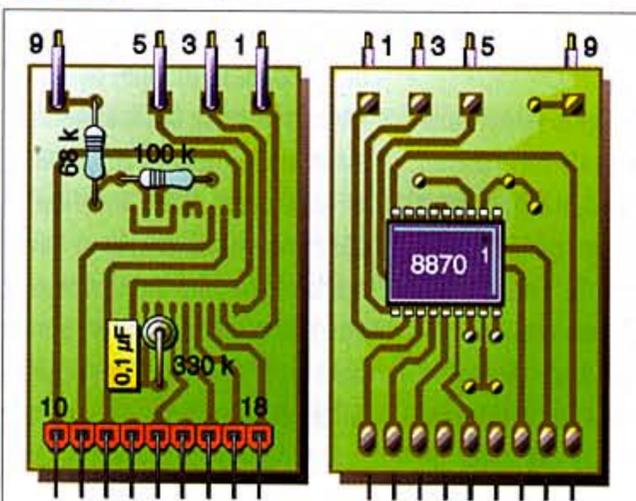
Reste à évoquer le comportement fonctionnel du 8870 par rapport au SSI202. On se souviendra que ce dernier offrait le choix entre deux formats de données : hexadécimal, et "2 parmi 8", à vrai dire assez rarement utilisé. Le 8870, pour sa part, fonctionne exclusivement en mode "hexa", selon la table de vérité de la **figure 9**.

Une différence importante, toutefois, qui ne saute pas aux yeux mais mérite d'être prise en considération : en l'absence de toute tonalité (autrement dit en état de "veille"), le SSI202 délivrait le code 0000 (affecté à la touche "D" des claviers).

Le M8870, lui, maintient sur ses sorties le code du dernier chiffre reçu et cela jusqu'à ce qu'il en reçoive un nouveau. Cela ne pose généralement pas de problème, dans la mesure où une broche de "strobe" (StD) signale le moment où les données peuvent être considérées comme valides.

Il faudrait, par contre, prévoir une petite mise à niveau logicielle (ou très exceptionnellement matérielle) si l'on avait sciemment exploité cette particularité du SSI202.

8 L'implantation de la version CMS.



P. GUEULLE

Une carte SIM "minimum"



Les téléphones portables GSM pourraient bien être les lecteurs de cartes à puce les plus répandus qui soient, dont les plus anciens sont aujourd'hui massivement désaffectés. Munis d'un clavier et d'un afficheur, ils ont encore plus d'un tour dans leur sac, même si on ne s'en sert plus pour téléphoner.

Moyennant une programmation très particulière, une simple BasicCard pourra leur faire reprendre du service, qu'ils soient ou non restés verrouillés sur tel ou tel opérateur.

Un "sésame" incontournable

Sans carte SIM, un téléphone portable GSM n'est que l'ombre de lui-même : si on le met sous tension, il affiche tout juste un message du genre "SIM absent" ou "Vérifier carte", sans donner accès à aucun de ses menus. En principe, il devrait tout de même permettre d'appeler le 112, le numéro gratuit international des services de secours, mais tous les opérateurs ne

jouent hélas pas le jeu, loin s'en faut. En fait, le logiciel du téléphone attend de reconnaître une carte SIM conforme à une normalisation très précise (GSM 11.11), pour ouvrir ce que l'on appelle une "session GSM", donnant alors accès aux principales fonctionnalités de l'appareil.

Normalement émise par un opérateur de téléphonie mobile, la carte SIM est surtout là pour authentifier son client, avant de lui donner accès au réseau. Si une carte SIM "invalide" (correspondant par exemple à un abonnement résilié, ou à un compte prépayé que l'on a laissé expirer) permet bel et bien au téléphone d'ouvrir une session GSM, celle-ci tournera court dès que son émetteur reconnaîtra, à distance, qu'elle n'est pas valable : un message

du genre "SIM non enregistrée" ou "Problème abonnement" s'affichera, et aucun menu ne sera généralement accessible.

Les principales exceptions à cette règle sont l'utilisation d'une carte SIM invalide en dehors de son pays d'émission, ou dans une zone non couverte (le téléphone reste en mode "recherche de réseau", ses menus étant alors accessibles).

Pousser plus loin les manipulations, en toute indépendance vis-à-vis des opérateurs de réseaux, suppose l'utilisation d'une carte SIM spécialement paramétrée, assimilable à une "carte de test".

Des quantités de choses deviennent alors possibles, sauf téléphoner : contourner le verrouillage de l'opéra-

teur ayant fourni le téléphone, ausculter les relais des différents réseaux, mettre certains téléphones en mode "test", etc.

Mais le fin du fin est encore de développer des applications personnelles que l'on programmera dans la carte : à condition de mettre en oeuvre la technologie "SIM Application Toolkit", qui permet réellement de "récupérer" le clavier, l'afficheur, et même le haut-parleur du téléphone !

Non, il ne s'agit nullement de spéculations théoriques, et encore moins de "piratage" : la société française Célavie a même remporté le trophée de la meilleure application du salon CARTES 2003 avec sa "Lifecarte", une "carte d'alerte médicale" que n'importe quel urgentiste peut lire en l'insérant simplement dans un téléphone portable !

Et même si cela peut paraître trop beau pour être vrai, ce genre d'application peut fort bien être développée avec une simple BasicCard...

Un noyau logiciel pour BASICCARD

Pour se lancer (progressivement !) dans ce genre d'aventure, le point de passage obligé est la mise au point d'un noyau logiciel permettant de "déguiser" une BasicCard en carte SIM conforme à la spécification GSM 11.11.

Ce document, de nature publique, explique de façon détaillée toutes les étapes de l'ouverture d'une session GSM : les "questions" que le téléphone pose à la carte, et les "réponses" que celle-ci doit lui retourner.

A force d'expérimenter, on finit par cerner le minimum incompressible de fonctionnalités qu'une carte SIM doit supporter pour être reconnue par un téléphone GSM, à priori quelconque.

Avec un peu d'audace, on arrive même à éliminer des fonctions que la norme considère comme "obligatoires", tout en offrant la possibilité de les désactiver dans ce que l'on appelle la "SIM Service Table" (voir notre ouvrage "Téléphones portables et PC", paru aux éditions ETSF).

Le résultat de ces "élagages" successifs est le code source MINISIM.BAS, téléchargeable sur le site de la revue.

Destiné aux BasicCard "Professional" ZC 5.4 (Rev.C ou supérieure) ou ZC 5.5, il exploite leur aptitude à fonctionner en mode "T=0". Rappelons, en effet, que les BasicCard "Compact" et "Enhanced" (ZC 3.3, ZC 3.9, etc.) ne supportent que le protocole "T=1",

incompatible avec les téléphones GSM.

Compilé, pour un type de carte donné, avec le kit logiciel BasicCard version 4.52 ou supérieure (dernière version téléchargeable sur <http://www.basiccard.com>), il occupe moins de 2500 octets d'EEPROM dans la carte, soit 15% d'une ZC 5.4, ou 8% d'une ZC 5.5.

Autant dire qu'il reste énormément de place pour les applications ou les données que chacun voudra bien venir greffer sur ce noyau, qu'il n'est d'ailleurs pas interdit de modifier... en sachant tout de même exactement ce que l'on fait !

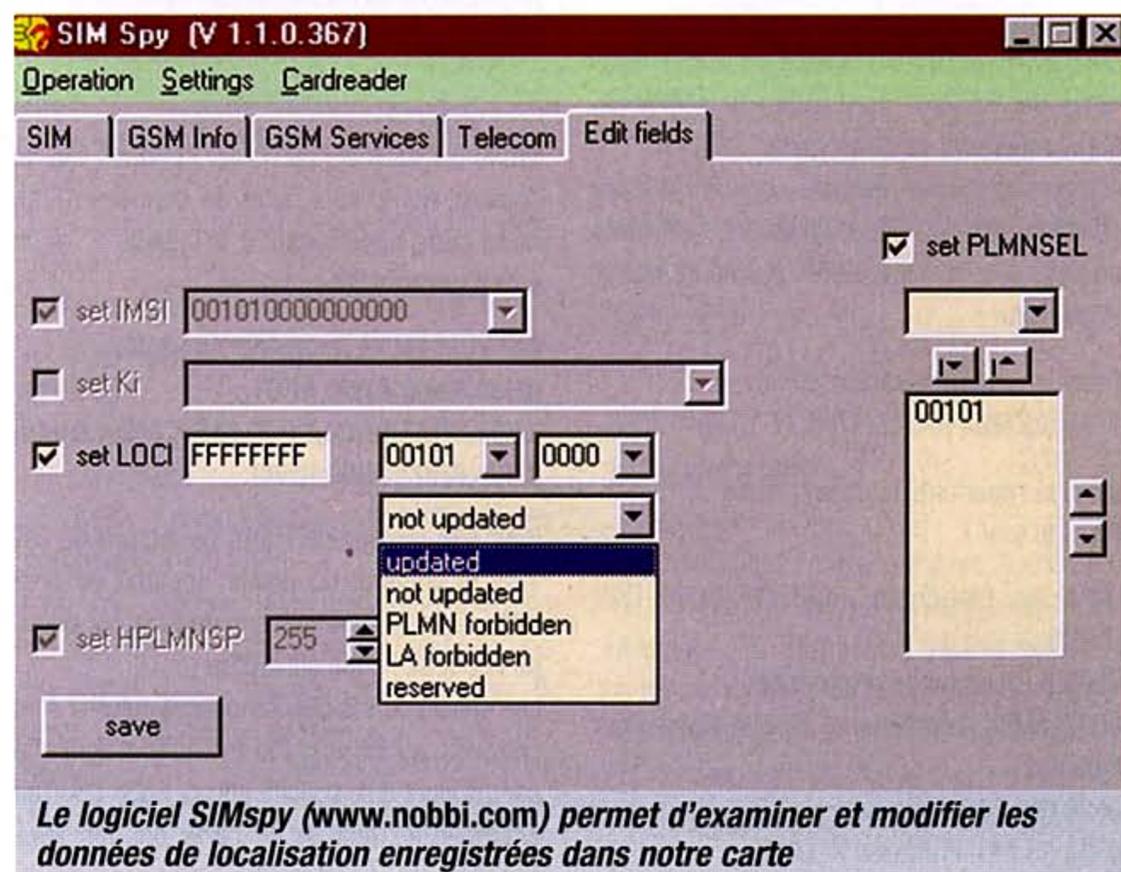
Le fichier objet MINISIM.IMG, pour sa part, constitue une version "clef en main", à télécharger exclusivement dans une ZC 5.4 Rev.C, au moyen de l'utilitaire BCLOAD du kit BasicCard, et d'un lecteur compatible (PC/SC, CyberMouse, ACR20, ACR30, etc.)

- SEEK
- INCREASE
- INVALIDATE
- REHABILITATE
- READ BINARY
- UPDATE BINARY
- Toutes les commandes de gestion des codes confidentiels "PIN".

Dans cette liste, les connaisseurs remarqueront l'absence des grands classiques "READ RECORD" et "UPDATE RECORD".

Cela pour la bonne et simple raison que nous n'avons créé dans la carte que des fichiers "transparents", à l'exclusion de tout fichier "linéaire" ou "cyclique".

Notre carte ne contient donc ni répertoire de numéros, ni SMS, fichiers particulièrement encombrants et nullement indispensables dans notre contexte de carte SIM "minimum".



Des commandes et des données

Comme toute application "carte à puce", le présent projet se caractérise avant tout par le jeu de commandes ISO 7816 que reconnaît la carte.

Parmi toutes les commandes définies par les spécifications GSM, il est apparu souhaitable d'implémenter au minimum celles-ci :

- SELECT (Sélection de fichier)
- GET RESPONSE
- STATUS
- RUN GSM ALGORITHM (authentification cryptographique)
- SLEEP (mise en veille)

Bien mieux, plusieurs de ces commandes sont simplement simulées, à commencer par celles qui concernent les codes confidentiels. Prenons le cas de "VERIFY CHV", qui n'occupe ainsi que deux lignes de code source :

**Command &HA0 &H20 VCHV(\$\$,Disable Le)
End Command**

Cette simple "déclaration" évite que la carte réponde "instruction inconnue" si le téléphone lui adresse une commande contenant le code opératoire 20h, ce qui interromprait très probablement la session GSM.

En fait, la carte répondra 9000h (exécution correcte) quel que soit le contenu de la commande, trompant ainsi grossièrement le sys-

tème d'exploitation du téléphone : on pourra ainsi jouer à modifier, activer, ou désactiver le code PIN, celui-ci restera imperturbablement inactif !

Même chose en ce qui concerne l'algorithme d'authentification cryptographique, qui construit une réponse R\$ de longueur conforme (0Ch = 12 octets), mais au contenu parfaitement factice :

```
Command &HA0 &H88 RUN(SS,Disable Le)
SW1SW2=&H9F0C
R$=RIGHT$(SS,12)
End Command
```

Remarquons, au passage, la valeur 9F0Ch du compte-rendu que l'on fait renvoyer par la carte, afin d'indiquer au lecteur (le téléphone) qu'il doit venir chercher une réponse de 12 octets au moyen d'une commande "Get response".

Mais à vrai dire, c'est surtout dans son "système de fichiers" que réside le secret du fonctionnement de cette carte.

Il a pu être mis en évidence que les fichiers suivants étaient nécessaires et suffisants pour qu'une "session GSM" puisse démarrer et perdurer :

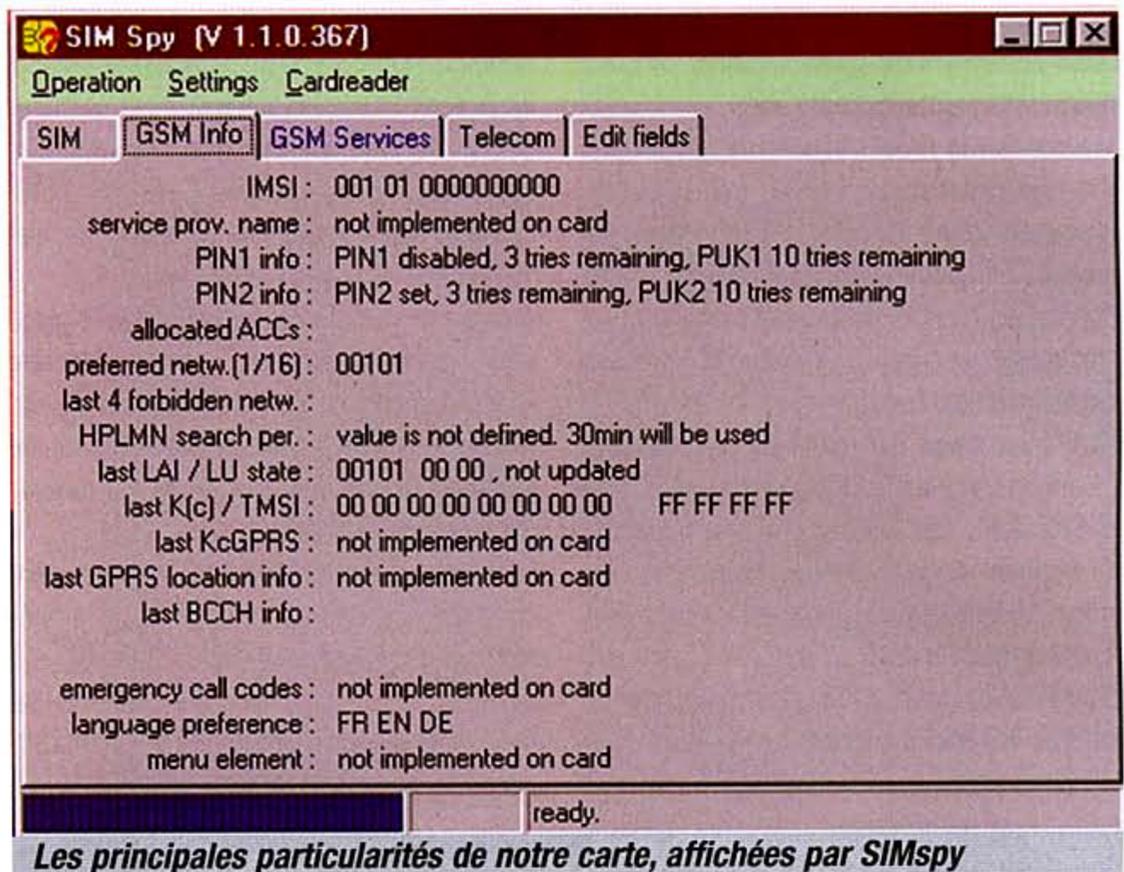
Dans le répertoire racine (3F00) :
2FE2 (ICCID)

Dans le répertoire Telecom (7F10) :
Rien du tout !

Dans les répertoires GSM (7F20) et DCS (7F21) :

- 6F05 (LP, Language preference)
- 6F07 (IMSI, International Mobile Subscriber Identity)
- 6F20 (Kc, Cipharing Key)
- 6F30 (PLMN, Preferred PLMNs)
- 6F31 (HPLMN search period)
- 6F38 (SST, SIM Service Table)
- 6F74 (BCCH, Broadcast Control Channels)
- 6F78 (ACC, Access Control Class)
- 6F7B (FPLMN, Forbidden PLMNs)
- 6F7E (LOCI, Location Information)
- 6FAD (AD, Administrative Data)
- 6FAE (Phase)

Plutôt que de mettre à contribution le système de fichiers "natif" de la BasicCard, dont la structure est sensiblement différente, nous avons tout simplement affecté une chaîne de caractères à chacun des fichiers "transparents" que nous venons de passer en revue. De longueur fixe, celle-ci est obtenue par concaténation des données et du "descripteur" du fichier.



Les principales particularités de notre carte, affichées par SIMspy

Prenons l'exemple du fichier "Phase", qui ne contient qu'un seul octet de données (02h), notre carte appartenant à la "phase 2" de la spécification GSM :

```
EEPROM PHA$ As String*16=&H02, _
&H00,&H00,&H00,&H01, _
&H6F,&HAE,&H04,&H00,&H04,&HF4,&H44,
&H03,&H02,&H00,&H00
```

Pour des raisons de clarté de lecture du listing, le contenu du fichier apparaît en première ligne, tandis que son descripteur occupe les deux lignes suivantes.

Les quinze octets qui composent celui-ci sont critiques, au bit près : la moindre irrégularité par rapport aux spécifications GSM pourrait entraîner le rejet de la carte par les téléphones les plus sourcilleux.

Sans entrer dans des détails qui dépasseraient largement le cadre de cet article, notons simplement que la seconde ligne du descripteur commence par les deux octets identifiant le fichier (ici 6FAEh).

D'une façon générale, les chaînes abritant les fichiers sont déclarées "EEPROM" : initialisées, lors de la programmation de la carte, avec le contenu spécifié dans le code source, elles peuvent être modifiées à volonté par des commandes "update binary" envoyées soit par le téléphone, soit par n'importe quel bon logiciel de manipulation de cartes SIM (par exemple le programme NAVSIM de notre ouvrage précité).

C'est particulièrement intéressant avec une telle carte "de développement", dont aucun

fichier n'est évidemment protégé en écriture ! Bien entendu, de telles modifications survivront à une mise hors tension de la carte, puisqu'elles résident dans une zone non volatile de sa mémoire, et même à un "reset".

Une exception a toutefois été volontairement introduite, à titre d'illustration de la liberté de manoeuvre qu'offre une telle carte : le fichier FPLMN (liste des réseaux interdits) est déclaré "Public", et réside donc dans une zone volatile (RAM) de la mémoire.

Son contenu s'effacera par conséquent (retour à FFh) lors de chaque reset de la carte.

Cela signifie que d'éventuelles tentatives infructueuses d'inscription sur tel ou tel réseau ne laisseront aucune trace dans ce fichier, pourtant prévu à cet effet pour éviter que cela ne se renouvelle !

En pratique, notre code source est cependant prévu pour que la carte ne tente pas de s'inscrire spontanément, et même pour qu'elle ne donne pas suite à une éventuelle tentative imposée manuellement par l'utilisateur.

Cela, tout simplement du fait que le contenu du fichier ACC (classes d'accès) est 0000h, petite astuce à laquelle ne semblent même pas avoir pensé certains développeurs d'applications "professionnelles", qui encombreront donc allégrement les réseaux...

Reste la question des identifiants qui, dans une "vraie" carte SIM, correspondent nominativement à un client donné.

L'IMSI (International Mobile Subscriber

Identifiant) a été fixé à 80 90 01 01 00 00 00 00 00, autrement dit au client N°0 d'un opérateur fictif dont le code serait 001 01.

Cette valeur, explicitement réservée aux cartes de test, permet de passer par dessus un éventuel verrouillage du téléphone par l'opérateur qui l'a fourni.

Le contenu du fichier AD (premier octet à 81h) joue un rôle complémentaire, dans la mesure où il autorise la mise en mode "test" de certains téléphones (Motorola et Philips, notamment), soit automatiquement, soit par une commande "cachée".

Quant à l'ICCID (80 90 01 02 03 04 05 06 07 F2), il est librement inspiré de celui d'une "Lifecarte" specimen (80 90 01 01 00 00 00 00 00 F2), afin d'éviter toute usurpation involontaire du code d'un opérateur existant, tout en restant vraisemblable.

Selon la recommandation E.118 concernant les cartes internationales de facturation des télécommunications, un véritable ICCID devrait commencer par les chiffres 89, suivis d'un indicatif de pays (33 pour la France), et de l'identification de l'opérateur (01 pour Orange, 10 pour SFR, 20 pour Bouygues), les

chiffres restants constituant le numéro de série de la carte chez l'opérateur en question.

Une question d'imagination

Reste maintenant à manipuler avec cette carte, ce qui peut se faire de bien des façons. Le plus simple est de commencer par s'en servir uniquement de "clef" pour ramener à la vie un téléphone portable de récupération, sans doute "simlocké", et en tout cas dému-ni de carte SIM ou équipé d'une carte périmée.

Il n'en faut pas davantage pour exploiter le mode "trace" de la plupart des GSM Alcatel : taper six fois "0" et une fois "étoile" pour accéder à un menu "secret" permettant de repérer les relais de tel ou tel opérateur que l'on aura au préalable sélectionné lors d'une recherche des réseaux disponibles.

Avec certains téléphones Motorola, il suffira d'appuyer longuement sur la touche "dièse" pour accéder à un mode "test", dont le mode d'emploi détaillé circule fort à propos sur Internet.

Mais on peut aller bien plus loin en modifiant hardiment tel ou tel fichier de la carte, avec l'aide d'une solide documentation (par exemple les livres écrits par l'auteur !).

Il sera alors intéressant d'examiner ce que le téléphone lui-même aura inscrit ici ou là, par exemple dans les fichiers LOCI (informations de localisation) ou BCCH (liste des canaux "balise" des relais du voisinage).

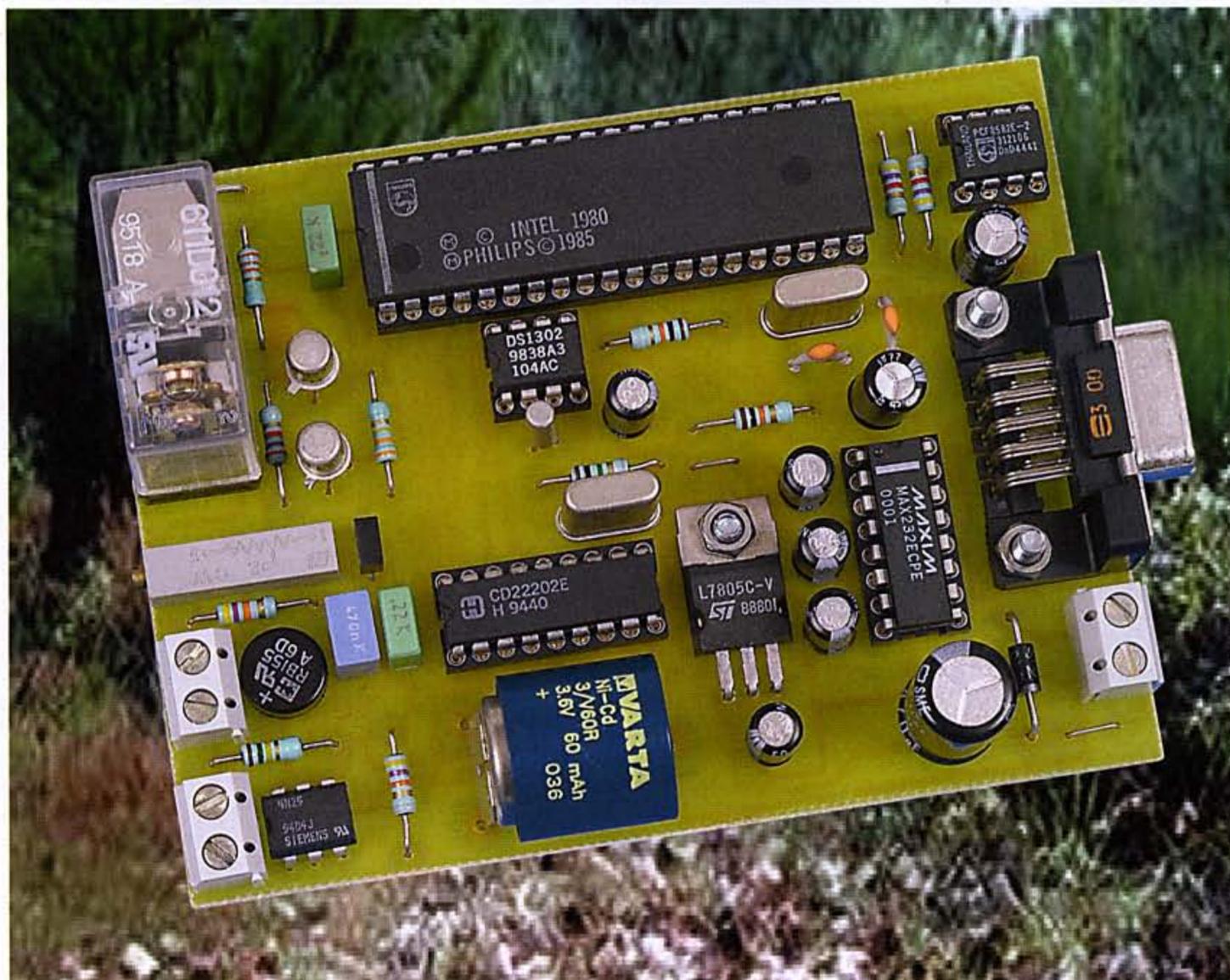
Certains logiciels du domaine public, comme SIMspy (<http://www.nobbi.com>) le font très bien !

Cela, en attendant de développer de véritables applications autonomes, qui n'utiliseront finalement le téléphone qu'en tant que terminal "clavier-écran" : c'est bien plus complexe (voir spécification GSM 11.14), mais cela ouvre des horizons pratiquement infinis : transformation d'un GSM en calculatrice hautement spécialisée, ou même en générateur de valeurs d'authentification cryptographiques...

Nous y reviendrons certainement !

P. GUELLE

Filtre d'appels téléphoniques



Si vous souhaitez pouvoir analyser finement vos factures téléphoniques, et si vous souhaitez pouvoir filtrer certains numéros, alors ce petit montage devrait vous intéresser.

Le montage que nous vous proposons permet de filtrer et d'interdire les appels des numéros programmés dans l'appareil. Il permet aussi d'obtenir la liste de tous les appels passés sur un poste téléphonique, ainsi que la durée exacte de ces appels.

Contrairement au service de facturation détaillée des opérateurs en téléphonie publique qui ne vous donne pas l'intégralité des numéros composés, notre montage vous révélera absolument tout sur l'usage du poste téléphonique sous surveillance.

Rappelons que ce type de montage ne doit pas être raccordé au réseau téléphonique public sans autorisation (ce qui n'est jamais accordé aux particu-

liers). Ce montage peut néanmoins rendre de grands services dans le cadre d'une petite entreprise qui souhaite filtrer les appels passés à partir d'un poste téléphonique en libre service, lorsque le central téléphonique n'est pas pourvu de cette fonction.

Dans ce cas, notez que pour des raisons éthiques et législatives, il est indispensable d'informer les utilisateurs qu'un système de surveillance des numéros appelés est installé sur ce poste.

Schéma

Le schéma de ce montage est reproduit en **figure 1**. On y découvre le microcontrôleur P89C51RD2 (U2) qui

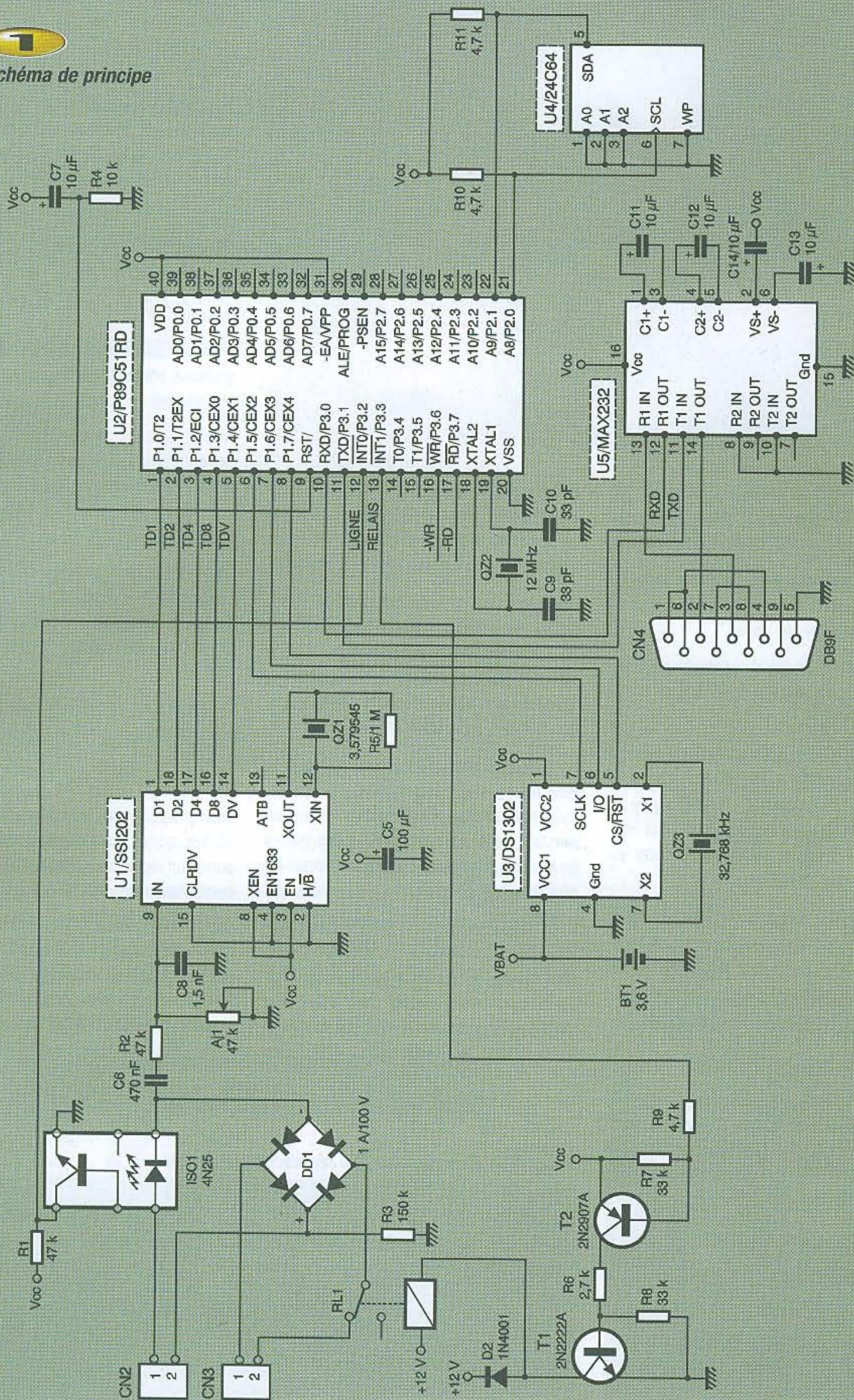
est le cœur du montage. Il dispose également d'une mémoire à accès série (U4 sur bus I2C) qui servira à enregistrer les règles de filtrage et les numéros appelés.

Pour permettre le dialogue du montage avec un PC, l'uart du microcontrôleur est mis en oeuvre simplement en adaptant les niveaux des signaux RXD et TXD. Dans ce but, nous avons fait appel à un circuit MAX232 (U5). En ce qui concerne l'horloge du microcontrôleur et la cellule de remise à zéro, leur mise en oeuvre est tout à fait classique.

Abordons maintenant l'interface de la ligne téléphonique. Comme vous pouvez vous en douter, pour décoder les



Schéma de principe





Décodage des signaux DTMF par un circuit CD2220 2E

signaux DTMF qui servent à composer les numéros téléphoniques, nous avons fait appel à un circuit SSI202 ou CD2220E (U1).

Il n'est pas nécessaire de vous présenter ce circuit puisqu'il fait régulièrement l'objet de réalisations dans ces pages.

Notez simplement que la résistance ajustable AJ1 permet de régler la sensibilité du circuit U1.

Les sorties du circuit U1 aboutissent directement sur le port P1 du P89C51RD2. Le microcontrôleur se chargera de scruter régulièrement les signaux fournis par U1 pour surveiller les numéros composés à partir du combiné sous surveillance.

Pour permettre au microcontrôleur de déterminer le début d'un numéro, nous disposons de l'information issue de l'optocoupleur IS01.

Cette partie du montage permet au microcontrôleur de faire la distinction entre un long silence entre deux numéros composés, et un long silence suite au raccroché du combiné.

L'interface avec la ligne téléphonique se fait via le pont de diode DD1. Ceci est nécessaire pour se prémunir des changements de polarité de la ligne qui peuvent survenir sans que vous en soyez informé (suite à une intervention sur votre ligne téléphonique).

De plus, cela permet de ne pas se casser la tête, au moment de raccorder le montage. Vous noterez avec intérêt que la masse du montage se trouve reliée à l'une des bornes de la ligne téléphonique via la résistance R3 pour fixer les potentiels.

Il faut savoir que pour une ligne téléphonique, le +48V du générateur est relié à la terre. Cela signifie que la ligne téléphonique présente un potentiel négatif par rapport à la terre. Il suffit, pour s'en convaincre, de mesurer la tension qui existe entre les lignes du

téléphone et la terre d'une prise de courant. Cette remarque n'est pas sans implications sur le comportement des montages que l'on connecte sur une ligne téléphonique.

Tant que la masse du montage n'est pas en liaison avec la terre, il n'y a pas de problème. Si l'appareil est relié à un PC (par un port série par exemple) il en va tout autrement.

Pour permettre au montage de libérer la ligne lorsque le numéro qui est composé doit être filtré, nous avons utilisé un relais dont le contact repos est en série avec la ligne téléphonique.

C'est une solution simple, mais qui nécessite quelques précautions logicielles pour éviter de perturber le central.

En effet, la durée de libération de la ligne doit être suffisamment longue pour éviter certains dysfonctionnements. Dans notre cas, la durée est réglable par le logiciel, pour un temps compris entre 10 s et 60 s.

Pour permettre d'horodater les événements mémorisés par notre appareil, nous avons fait appel à une petite horloge 'temps réel' qui est réalisée autour du circuit DS1302 (U3).

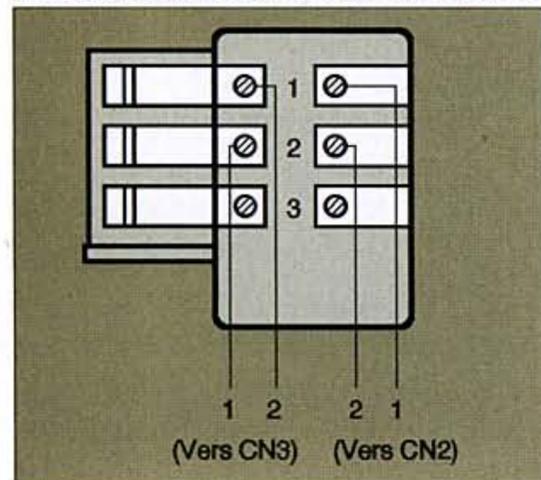
Ce circuit dispose des éléments nécessaires pour maintenir chargé un petit accumulateur et il s'occupe de tout lors des coupures de la tension d'alimentation.

Le montage sera alimenté par une tension de 12 Vdc qui n'a pas besoin d'être stabilisée puisqu'un régulateur LM7805 (REG1) fournit la tension Vcc nécessaire au montage (**figure 3**).

Une tension correctement filtrée fera très bien l'affaire. Cependant, le bloc d'alimentation devra pouvoir fournir au moins 300 mA. Ajoutons que la diode D1 permet de protéger le montage en cas d'inversion du connecteur d'alimentation.

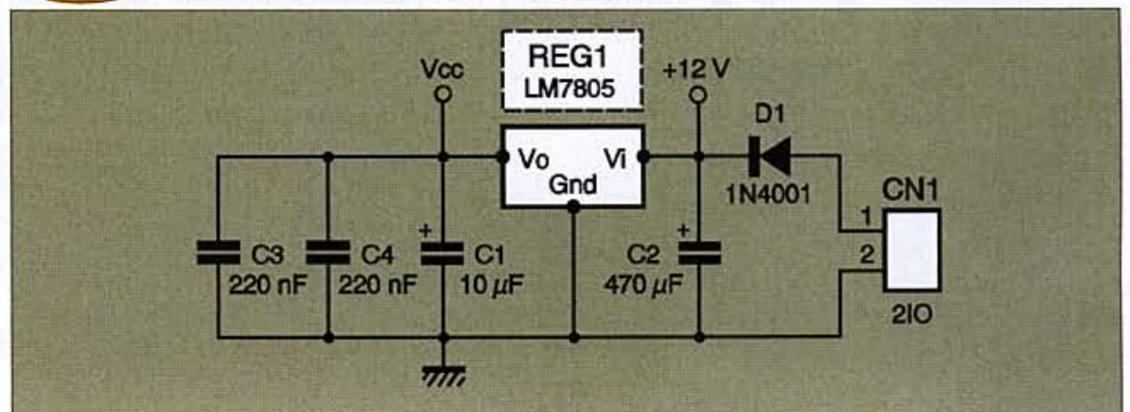
2

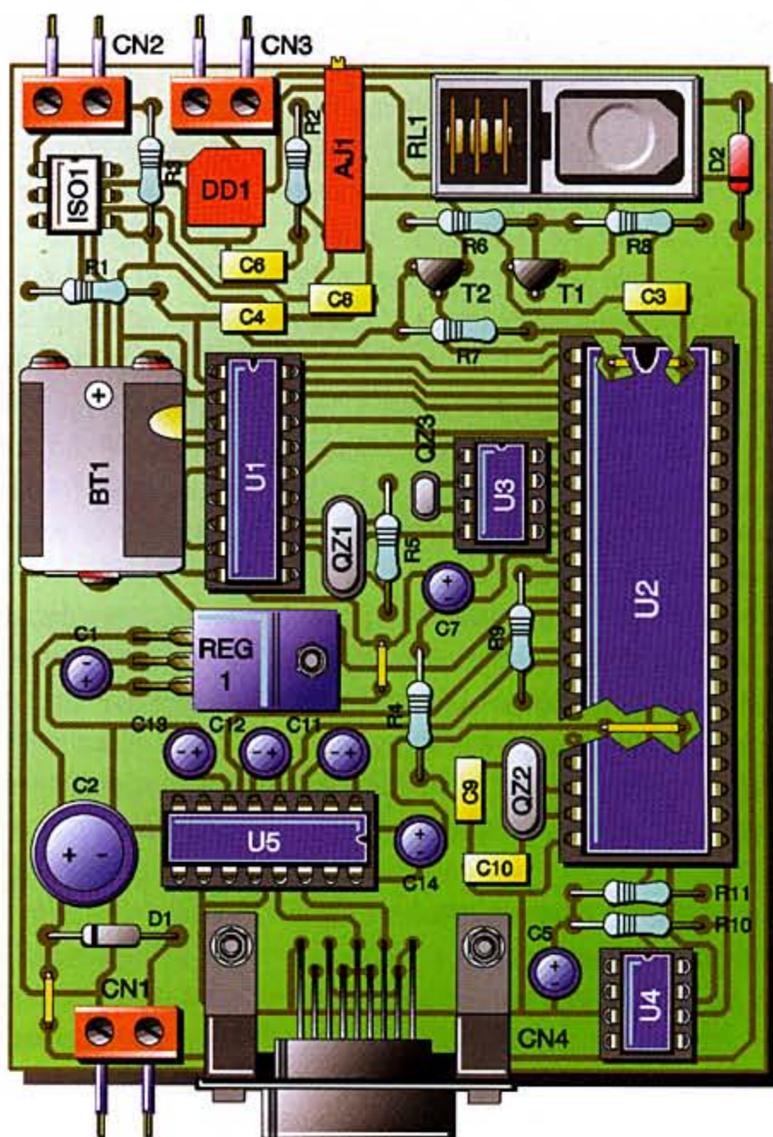
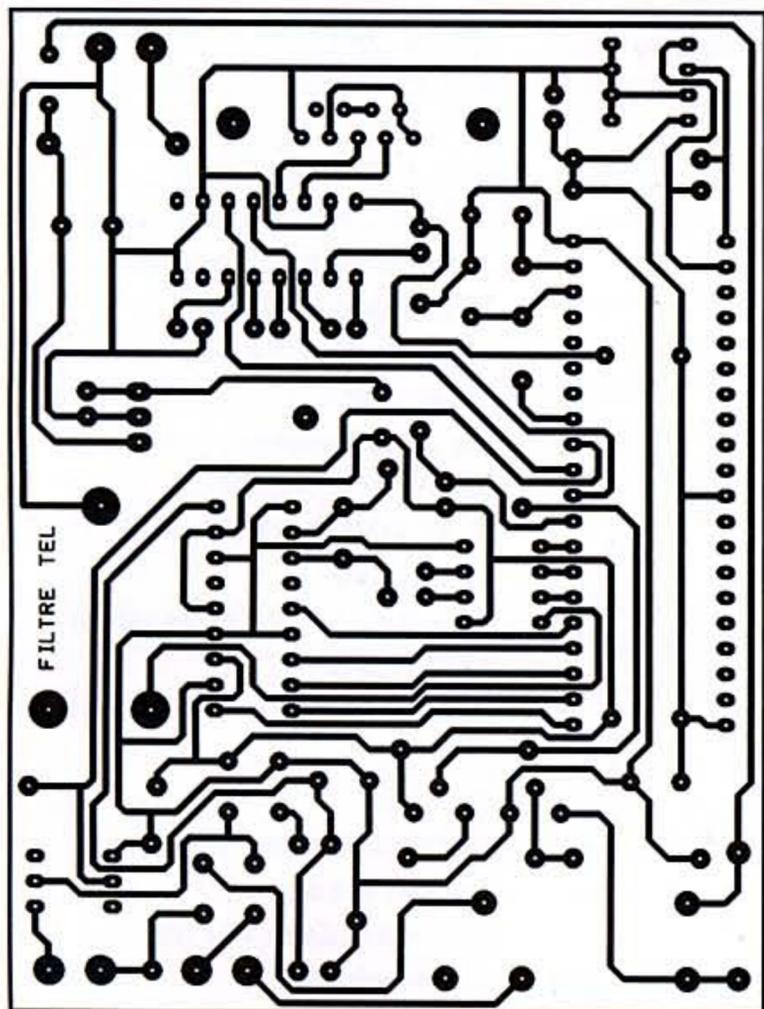
Prise téléphonique mâle à raccorder aux connecteurs CN2 et CN3 du module



3

Une alimentation régulée classique de + 5 V





4 Tracé du circuit imprimé

5 Implantation des éléments

Réalisation

Le dessin du circuit imprimé à reproduire est visible en **figure 4**. La vue d'implantation associée est indiquée en **figure 5**. Les pastilles seront percées à l'aide d'un foret de 0,8 mm de diamètre pour la plupart. En ce qui concerne CN1, CN2, CN3, REG1, D1, D2 et DD1, il faudra percer les pastilles avec un foret de 1 mm de diamètre.

Enfin, en ce qui concerne le relais RL1 et l'accumulateur BT1, il faudra percer les pastilles avec un foret de 1,3 mm de diamètre. N'oubliez pas également de percer le passage des vis pour CN4 et REG1 avec un foret de 3,5 mm de diamètre.

Comme d'habitude, procurez-vous les composants avant de réaliser le circuit imprimé pour vérifier que l'implantation est possible. Cette remarque concerne essentiellement le relais. Pour le reste, il n'y a pas de difficulté particulière. Soyez tout de même attentifs au sens des condensateurs et des circuits inté-

grés. Vous noterez la présence de quatre straps qu'il est préférable d'implanter en premier pour des raisons de commodité (surtout pour les deux straps situés sous le microcontrôleur).

Attention au connecteur CN4 qui est un modèle femelle. Un modèle mâle pourrait s'implanter sur le circuit imprimé, mais les points de connexions se retrouveraient inversés symétriquement par rapport à l'axe vertical.

Dans ce cas, il n'y aurait aucune chance pour que le 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. Le régulateur REG1 pourra être monté sur un dissipateur thermique, mais ce n'est pas une nécessité.

Le microcontrôleur U1 sera programmé avec le contenu d'un fichier que vous pourrez vous procurer par téléchargement sur le serveur Internet de la revue (<http://www.eprat.com>).

Le fichier d'archive que vous téléchargerez contient également un programme conçu pour fonctionner sous Windows. Le programme exploite l'un des ports série de votre PC pour dialoguer avec le montage.

Lors du premier lancement du programme 'WfiltreTel.exe', la première opération à effectuer consiste à renseigner le programme sur le port série à utiliser (menu 'OPTION').

De plus, lors de la première mise en service de l'appareil, il convient d'initialiser la mémoire du montage. Utilisez la fonction 'Initialisation complète de l'appareil distant', à partir du menu 'action distante', puis utili-

sez la fonction de mise à l'heure de l'horloge toujours à partir de ce menu.

Notez également qu'à la première mise en service du montage, il est probable que l'accumulateur BT1 soit déchargé. Prévoyez donc de laisser le montage sous tension au moins 4 à 5 heures pour recharger l'accumulateur. Sinon, suite à une coupure d'alimentation, vous devrez recommencer la procédure de mise à l'heure.

L'appareil est désormais prêt à fonctionner, mais les filtres ne sont pas encore programmés. Utilisez la zone de saisie de la fenêtre principale pour renseigner les numéros que vous souhaitez interdire.

Vous pouvez utiliser le symbole '?' pour masquer des chiffres dans le numéro, afin de constituer un filtre générique. Le montage est programmé pour accepter des filtres sur 16 numéros, ce qui permet d'interdire des appels dont le numéro comporte une extension.

Si vous souhaitez filtrer un numéro à partir d'un préfixe uniquement, vous n'êtes pas obligé de saisir le symbole '?' pour remplir les 16 positions. Le programme s'en chargera pour vous. Ajoutons que le montage accepte 16 filtres maximum, ce qui devrait permettre de couvrir les besoins les plus simples.

Une fois que la liste des filtres est constituée, vous pouvez l'enregistrer dans un fichier en vue d'une édition ultérieure. Si vous n'enregistrez pas la liste maintenant, vous pourrez toujours interroger l'appareil afin de la reconstituer. Pour informer l'appareil des

filtres qu'il doit appliquer, utilisez la fonction 'exporter la liste des filtres' à partir du menu 'action distante'. Précisez ensuite la durée pendant laquelle la ligne sera inhibée en cas de filtrage d'un appel.

Pour cela utilisez les fonctions du menu 'action distante'. Vous avez le choix entre 10 s, 20 s, 30 s, 45 s et 60 s. Par défaut, la valeur appliquée est de 10 s.

À l'issue de ces manipulations, l'appareil est enfin prêt à surveiller et filtrer les appels passés à partir du poste auquel il est raccordé. Pour connaître la liste des appels qui ont été passés sur le poste surveillé, utilisez la fonction 'importer la liste des appels' à partir du menu 'action distante'.

Vous pourrez ensuite imprimer cette liste ou l'enregistrer dans un fichier texte. Notez que le fait d'importer la liste n'efface pas la mémoire du montage.

La place mémoire étant limitée (270 numéros appelés maximum en mémoire), pensez à effacer la liste des appels dès que vous aurez fini de la consulter, toujours à l'aide des fonctions du menu 'action distante'.

Pour terminer cet article, rappelons que ce montage ne doit pas être raccordé à une ligne du réseau téléphonique public sans autorisation.

Ce montage a été conçu pour être raccordé derrière une installation privée, en vue d'étendre les possibilités d'un autocommutateur dépourvu des fonctions de filtrage.

P. MORIN



Le MAX 232 adapte les niveaux des signaux RXD et TXD

Nomenclature

AJ1 : Ajustable 47 k Ω multitours

BT1 : Accumulateur 3,6 V / 60 mA/H à souder sur circuit imprimé (par exemple référence Varta 53306 603 059).

CN1, CN2, CN3 : Bornier de connexion à vis, 2 plots, au pas de 5,08 mm, à souder sur circuit imprimé, profil bas

CN4 : Connecteur Sub-D, 9 points, femelle, sorties coudées, à souder sur circuit imprimé (par exemple référence HARTING 09 66 112 7601).

C1, C7, C11, C12, C13, C14 : 10 μ F / 25 Volts, sorties radiales

C2 : 470 μ F / 25 Volts, sorties radiales

C3, C4 : 220 nF

C5 : 100 μ F / 25 Volts, sorties radiales

C6 : 470 nF

C8 : 1,5 nF

C9, C10 : Condensateur céramique 33 pF, au pas de 5,08 mm

DD1 : Pont de diodes 1 A / 100 V

(par exemple RB51)

D1, D2 : 1N4001 (diode de redressement 1 A / 100 V)

ISO1 : Optocoupleur 4N25 (en boîtier DIP 6 broches)

QZ1 : Quartz 3,579545 MHz en boîtier HC49/U

QZ2 : Quartz 12 MHz en boîtier HC49/U

QZ3 : Quartz 32,768 kHz en boîtier Mini-Cyl

REG1 : Régulateur LM7805 (5 V)

en boîtier TO220

RL1 : Relais FBR611 ou équivalent

(1 contact R/T)

R1, R2 : 47 k Ω 1/2 W 5%

(Jaune, Violet, Orange)

R3 : 150 k Ω 1/2 W 5% (Marron, Vert, Jaune)

R4 : 10 k Ω 1/2 W 5% (Marron, Noir, Orange)

R5 : 1 M Ω 1/2 W 5% (Marron, Noir, Vert)

R6 : 2,7 k Ω 1/2 W 5% (Rouge, Violet, Rouge)

R8, R7 : 33 k Ω 1/2 W 5%

(Orange, Orange, Orange)

R9, R10, R11 : 4,7 k Ω 1/2 W 5%

(Jaune, Violet, Rouge)

T1 : 2N2222A

T2 : 2N2907A

U1 : Décodeur DTMF SSI202 ou CD22202E

U2 : Microcontrôleur P89C51RD2

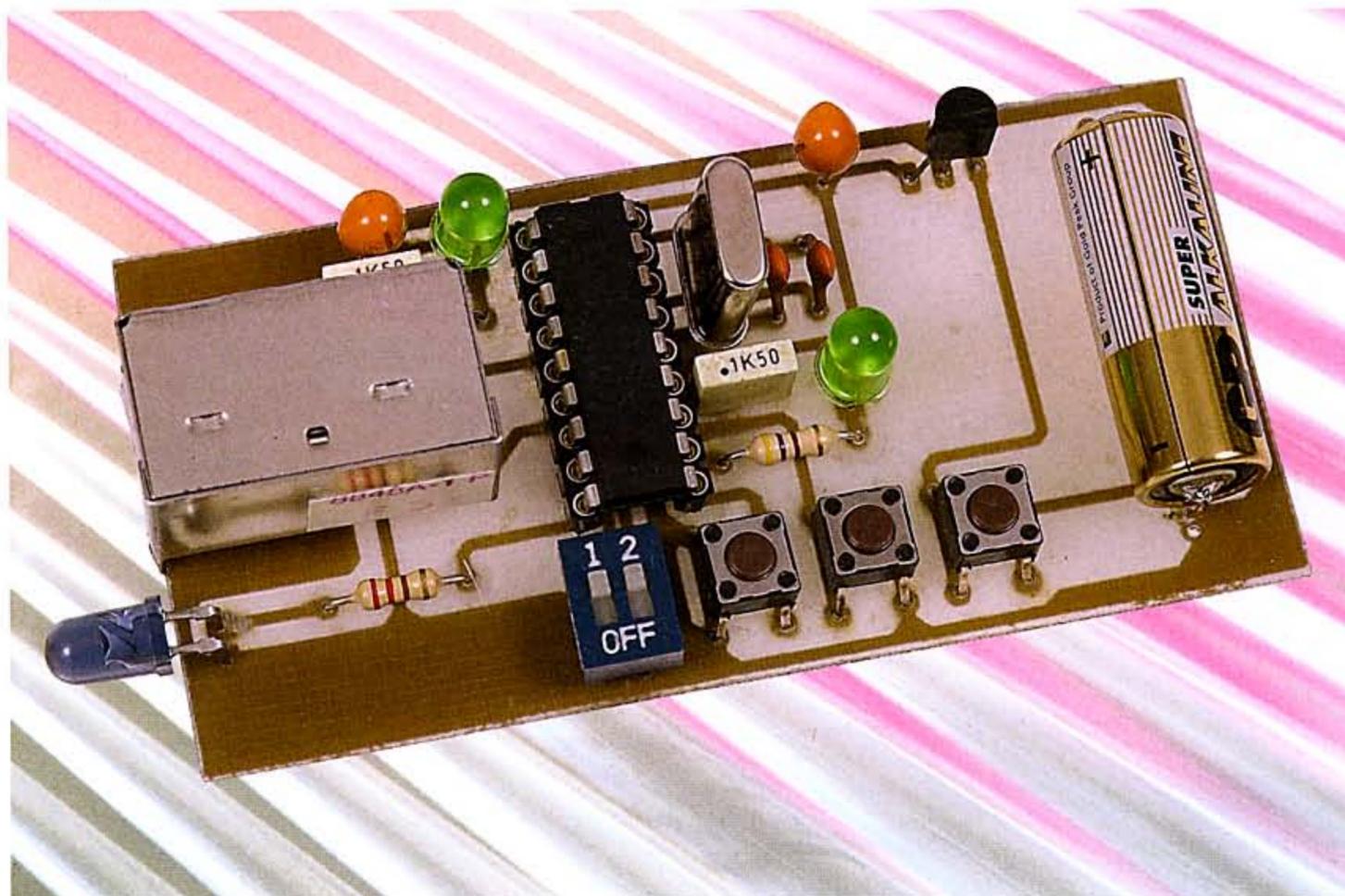
(ou P89C51RC2)

U3 : Horloge temps réel DS1302

U4 : Mémoire 24C64

U5 : Driver de lignes MAX232

Lecteur/copieur de télécommandes infrarouges



Il peut s'avérer très intéressant de disposer d'une seconde télécommande infrarouge, qu'elle soit utilisée pour le téléviseur, l'automobile ou une porte de parking. Le montage que nous vous proposons aujourd'hui est un lecteur/copieur de télécommande qui peut lire et reproduire quatre ordres d'une télécommande infrarouge quelconque.

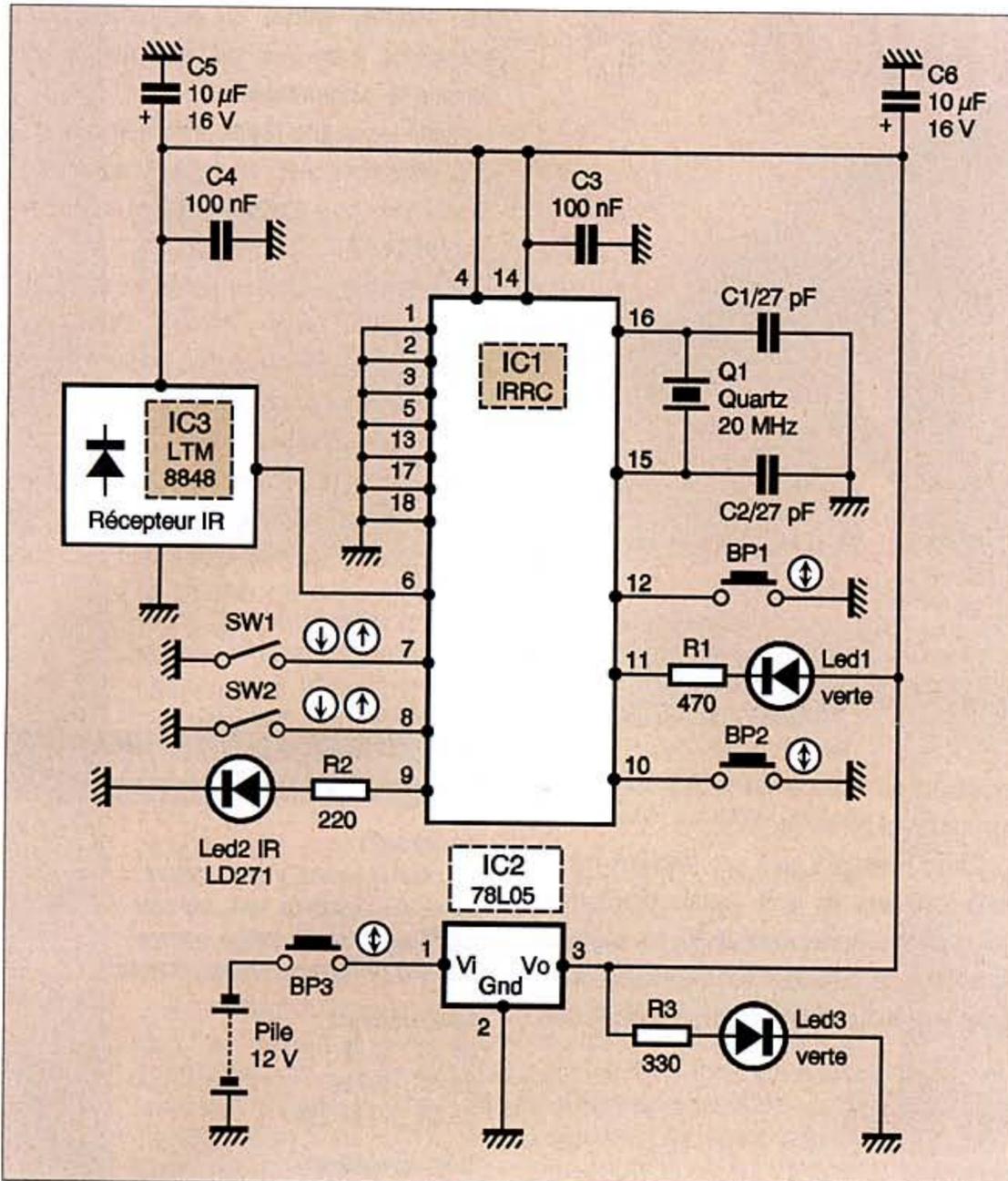
Lorsque l'on constate le prix prohibitif des télécommandes infrarouges de remplacement, on voit l'intérêt du montage que nous présentons. Bien sûr, notre lecteur/copieur ne peut lire que quatre ordres, ce qui est nettement insuffisant dans le cas d'une télécommande de téléviseur, à moins de ne vouloir utiliser que le changement de chaînes et le contrôle du son. Cependant, il conviendra parfaitement dans le cas d'une commande de véhicule automobile ou pour l'ouverture d'une porte. Dans ce cas, on peut alors regrouper la commande de divers systèmes dans un même boîtier. Le circuit intégré utilisé, l'IRRC, est un microcontrôleur préprogrammé. Ses caractéristiques électriques sont données dans le tableau de la **figure 1**, figure qui contient également son brochage.

Ses caractéristiques générales sont les suivantes :

- Lecture de quatre ordres transmis par rayonnement infrarouge,
- Stockage des données reçues dans une mémoire non volatile,
- Reproduction des codes lus et stockés,
- Quatre nouveaux ordres peuvent être stockés en effaçant les précédents,
- Très peu de composants externes nécessaires,
- Connexion directe d'un récepteur infrarouge à sortie au niveau TTL,
- Connexion directe d'une diode d'émission infrarouge,

Les fonctions de ses différentes broches sont données ci-dessous :

- **Broches 1, 2, 3, 5, 13, 17, 18, GND**, connexions à la masse de l'alimentation,
- **Broches 4, 14, +VCC** : connexions au + (plus) de l'alimentation,
- **Broche 6, IRI, IR Input, entrée** : entrée des ordres reçus par la diode infrarouge,
- **Broche 7, CS0, entrée** : entrée de sélection du bit 0 qui permet de définir l'adresse de l'ordre stocké en mémoire,
- **Broche 8, CS1, entrée** : entrée de sélection du bit 1 qui permet de définir l'adresse de l'ordre stocké en mémoire,
- **Broche 9, IRO, IR Output, sortie** : broche de connexion de la LED infra-



Code à transmettre	CS1	CS0
1	→ masse	masse
2	→ masse	+VCC
3	→ +VCC	masse
4	→ +VCC	+VCC

Le schéma de principe

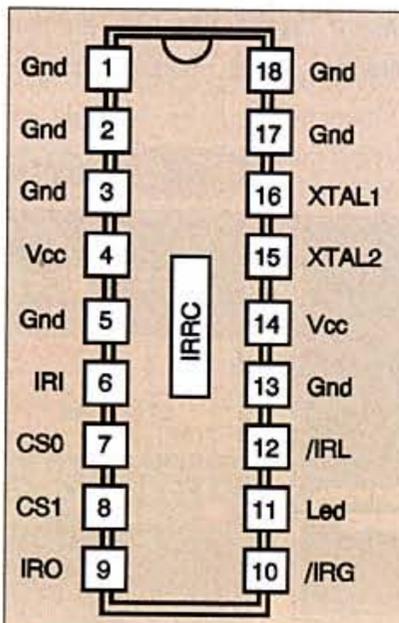
Le schéma de principe de notre réalisation est donné en **figure 2**. Il est d'une très grande simplicité puisque c'est le microcontrôleur qui gère toutes les opérations. Un quartz d'une fréquence de 20 MHz associé à ses deux capacités de 27 pF permet le cadencement de l'IRRC. Des poussoirs connectés entre les broches 10, 12 et la masse, permettent respectivement par leur manœuvre de transmettre les ordres lus et de mémoriser ces ordres. Deux commutateurs miniatures permettent la sélection de l'ordre à envoyer. La diode LED connectée à la broche 11 permet de visualiser la réception d'un ordre mais également de visualiser son émission lorsque le poussoir BP2 est actionné. La diode infrarouge émettrice est connectée en broche 9, IRO. La résistance R2 limite le courant qui la traverse. Lorsqu'elle a une valeur de 220 Ω, le rayonnement infrarouge est relativement faible et la portée assez restreinte. On peut descendre cette valeur à 150 Ω. Un troisième poussoir permet de mettre le montage sous tension. Ce dernier est alimenté au moyen d'une pile miniature de 12 volts. Un régulateur de tension de type 78L05 amène cette tension à +5V. Une diode LED associée à sa résistance de limitation indique la mise sous tension du montage. Signalons que ces deux derniers composants sont optionnels.

2 Schéma de principe

- rouge émettrice, modulée à 38,46 kHz et qui peut débiter un courant de 20mA,
- **Broche 10, IRG, IR Generate, entrée** : lorsqu'un niveau bas lui est appliqué, la diode infrarouge émettrice est alimentée et transmet le code choisi,
- **Broche 11, LED, sortie** : normalement à l'état bas. C'est la broche de connexion de la LED indicatrice qui peut débiter un courant de 20mA. Son état logique oscille entre la masse et +VCC lorsque l'IRRC lit un code ou le transmet,
- **Broche 12, IRL, IR Learn, entrée** : lorsqu'un niveau bas lui est appliqué pendant au moins 10 ms, c'est la validation d'apprentissage des codes reçus par la diode réceptrice infrarouge connectée sur la broche IRI (broche 6),
- **Broche 16, XTAL1, entrée** : connexion d'un quartz de 20 MHz,
- **Broche 15, XTAL2, sortie** : connexion d'un quartz de 20 MHz,

la masse, c'est le code dont l'adresse est sélectionnée par les entrées CS0 et CS1 qui est généré par la diode infrarouge émettrice. Le tableau donné ci-contre indique ces adresses :

Caractéristiques électriques du microcontrôleur IRCC



Parameter	min	max	unit
Supply voltage	3,0	5,25	V
Current*	1	4	mA
Operating Temp.	0	70	°C
Input Low Voltage	Gnd	0,2 Vcc	V
Input High Voltage	0,8 Vcc	Vcc	V
Output Low Voltage	-	0,6	V
Output High Voltage	Vcc - 0,7	-	V
XTAL1, 2	20	20	MHz

Note : lorsque la broche IRG est connectée à



La diode réceptrice est enfermée dans un boîtier métallique

La diode réceptrice est un modèle intégré qui comporte en interne, dans un petit boîtier métallique assurant son blindage, un petit circuit amplificateur et démodulateur dont la sortie est au standard TTL. Cela simplifie nettement les réglages.

La réalisation

Le tracé du circuit imprimé est donné en **figure 3** tandis que la **figure 4** représente le schéma d'implantation des composants. Peu de choses sont à préciser tant le circuit est simple à câbler. Le circuit intégré IRRC doit être placé dans un support par mesure de sécurité. La pile est soudée directement sur le circuit imprimé au moyen de deux petits morceaux de fil de câblage rigides.

Le boîtier de la diode infrarouge réceptrice possède quatre pattes sur son blindage. Les deux se trouvant à l'arrière, donc vers le circuit intégré, sont coupées à ras du boîtier. Les deux restantes sont pliées vers le recto du circuit imprimé et y sont soudées. Il ne reste

plus alors qu'à souder les trois broches d'alimentation et de sortie.

Les trois poussoirs sont des modèles miniatures courants. Ils sont soudés directement sur la platine. Après avoir vérifié les soudures et veillé à ce qu'aucun court circuit n'existe entre les soudures, on peut passer aux essais.

Les essais

Il convient de bien respecter la procédure que nous décrivons au risque de ne pas pouvoir mémoriser les codes envoyés.

On positionne la télécommande que l'on désire copier face à la diode réceptrice de notre montage, à environ une trentaine de centimètres. Il convient alors d'appuyer sur le poussoir de mise sous tension (BP3) puis sur le poussoir de lecture (BP1) un court instant. A ce moment, la diode D1 se met à clignoter et l'on envoie alors l'ordre que l'on souhaite mémoriser. La LED D1 reproduit, en clignotant rapidement, l'ordre reçu. Le premier code est mémorisé. Si l'on souhaite enregis-

trer d'autres ordres, on recommence ces opérations, mais bien sûr, sans mettre hors tension le lecteur/copieur.

Lorsque les quatre codes sont mémorisés, la LED indicatrice reste allumée durant environ 5 secondes puis s'éteint. On peut alors couper l'alimentation du montage.

Quand le lecteur/copieur est remis sous tension, si aucun ordre émanant d'une autre télécommande n'est reçu, les codes mémorisés auparavant peuvent être générés.

Il suffit de choisir le code que l'on veut utiliser au moyen de SW1 et SW2 puis d'appuyer sur les poussoirs BP3 puis BP2. Le code est alors généré continuellement tant que dure l'appui.

P. OGUIC

Nomenclature

Résistances :

- R1 : 470 Ω (jaune, violet, marron)
- R2 : 150 Ω (marron, vert, marron) à 220 ohms (rouge, rouge, marron)
- R3 : 330 Ω (orange, orange, marron)

Condensateurs :

- C1, C2 : 27 pF
- C3, C4 : 100 nF
- C5, C6 : 10 μ F/16V

Semi-conducteurs :

- LED1, LED3 : diodes LED vertes
- LED2 : diode émettrice infrarouge LD271

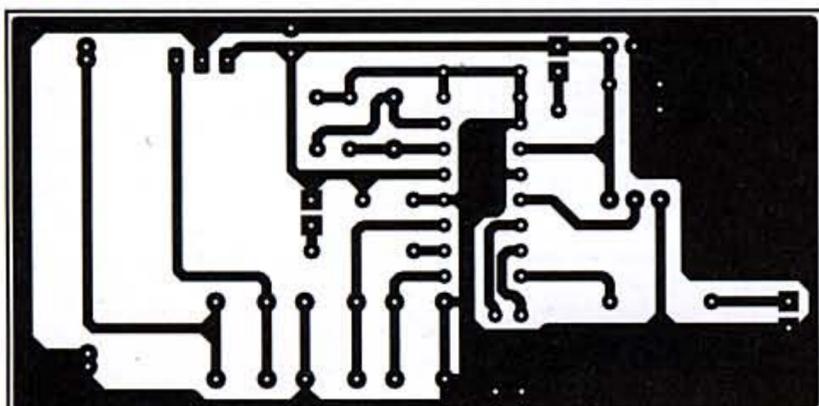
Circuits intégrés :

- IC1 : IRRC (LEXTRONIC)
- IC2 : régulateur de tension 78L05
- IC3 : récepteur infrarouge intégré LITEON8848 (LEXTRONIC)

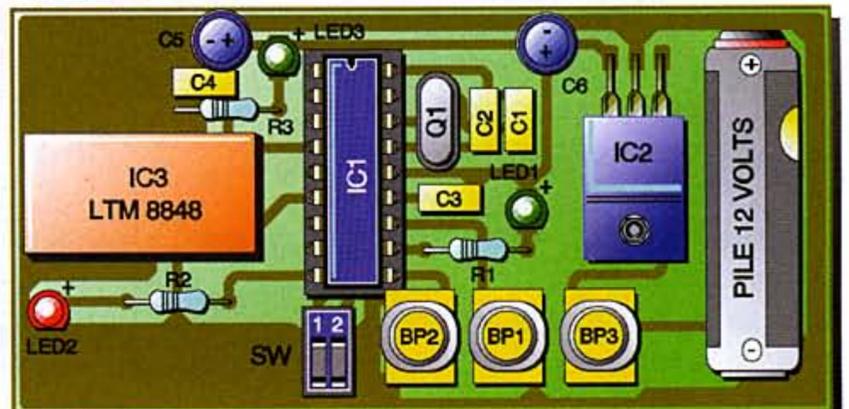
Divers :

- BP1, BP2, BP3 : boutons poussoirs miniatures pour circuit imprimé
- SW1, SW2 : dip switches pour circuit imprimé
- 1 quartz 20 MHz
- 1 pile miniature 12 volts
- 1 support pour circuit intégré 18 broches

3 Tracé du circuit imprimé



4 Implantation des éléments



Une signalisation de détresse



Sur la route, il est indispensable de respecter une règle simple : voir et être vu. L'observation de ce précepte revêt une importance encore plus grande de le cas d'un arrêt inopiné sur le bas côté de la route suite à une panne ou à une crevaison. Le montage proposé, branché sur la prise allume-cigare du véhicule, émet périodiquement de puissants éclairs signalisant le danger.

1. Le fonctionnement (figures 1 et 2)

Alimentation

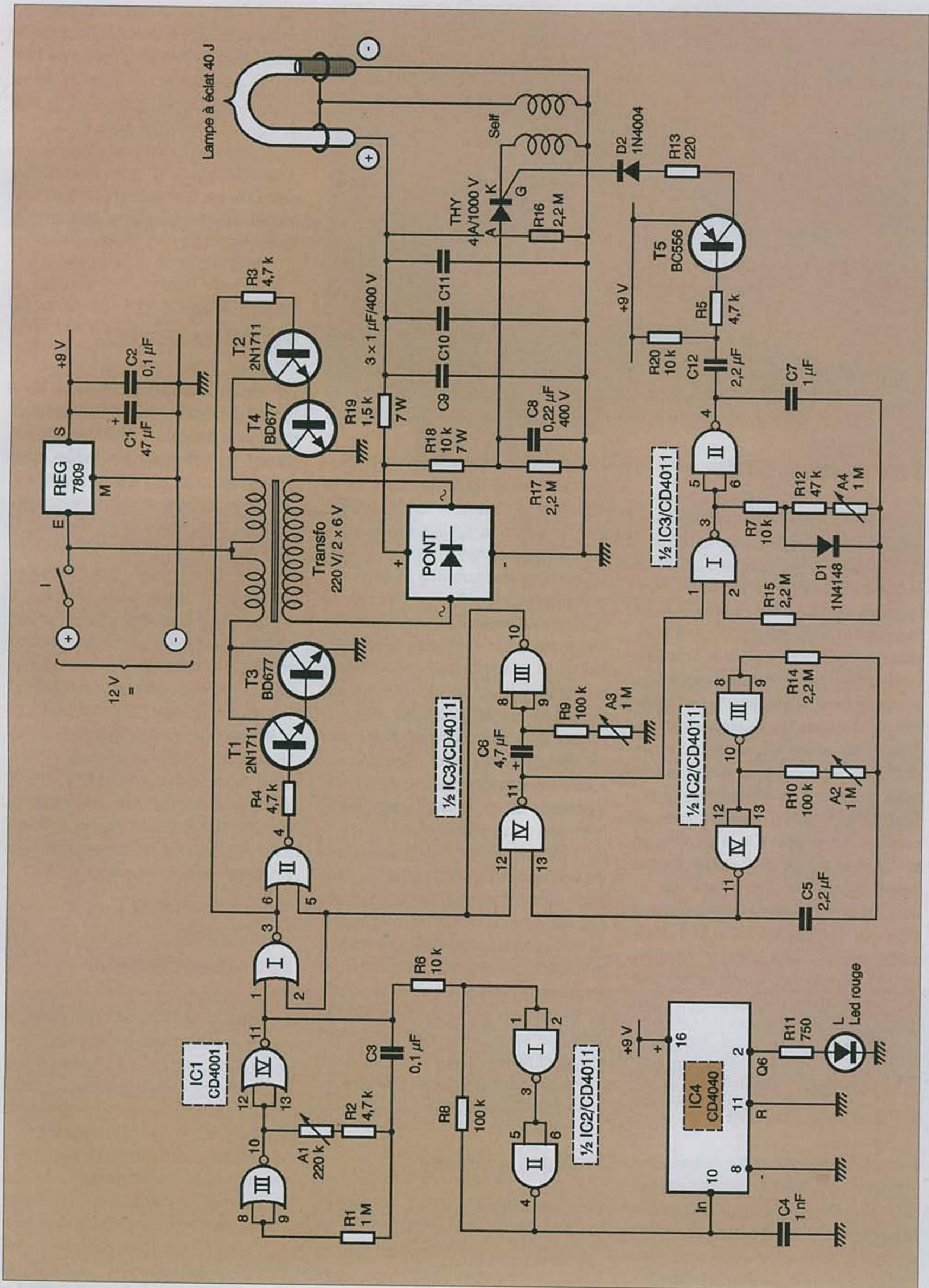
Le montage fonctionne sous une source de courant continu de 12 volts. L'interrupteur permet sa mise en ser-

vice. Ce potentiel de 12 volts est directement mis à contribution pour fournir la puissance nécessaire à la production des éclairs, comme nous le verrons ultérieurement.

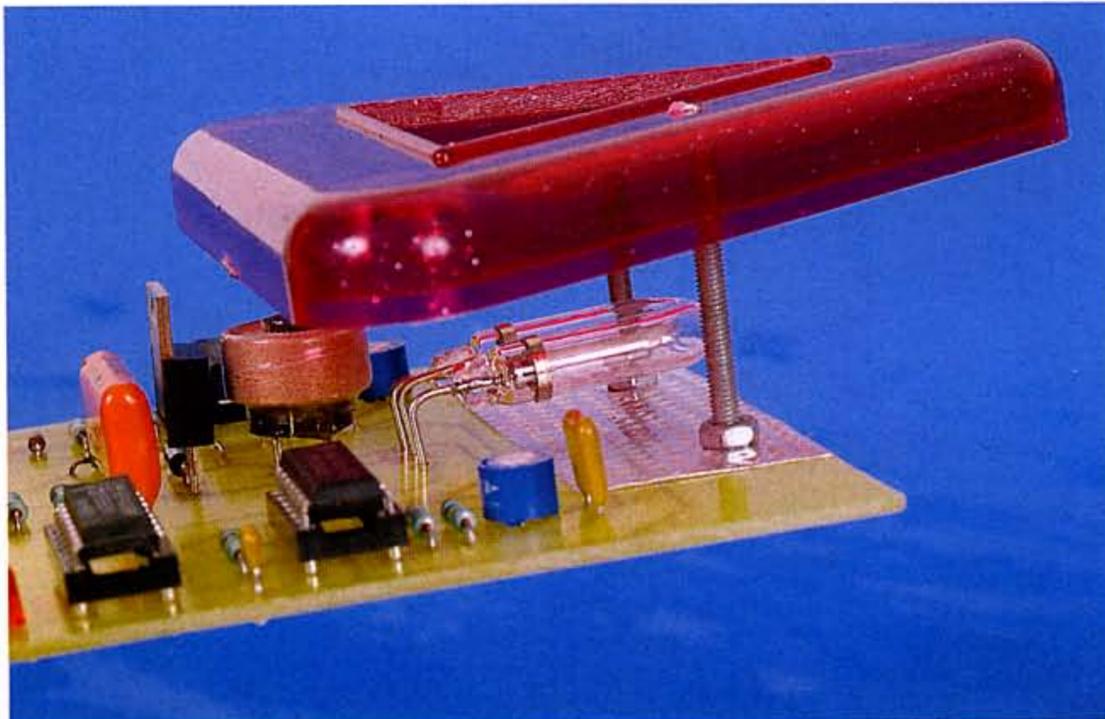
Quant au potentiel de pilotage du montage, il est disponible sur la sortie d'un régulateur 7809 lequel délivre

une tension continue stabilisée à 9 volts.

La capacité C1 réalise le complément de filtrage nécessaire compte tenu des perturbations introduites par la répétition des décharges à tension élevée, tandis que C2 fait office de capacité de découplage.



1 Schéma de principe



Le triangle rouge est maintenu par deux tiges filetées

Base de temps 50 Hz

Les portes NOR III et IV de IC1 forment un oscillateur astable délivrant sur sa sortie un signal carré se caractérisant par une période de 20 millisecondes, ce qui correspond à une fréquence de 50 Hz. Par l'intermédiaire de l'ajustable A1, il est possible d'obtenir cette valeur de façon simple grâce à un artifice consistant à diviser la fréquence afin de pouvoir observer et chronométrer la période de clignotement d'une LED de signalisation. Les portes NAND I et II de IC2, montées en trigger de Schmitt, fournissent en sortie des créneaux caractérisés par des fronts rendus verticaux et donc aptes à attaquer l'entrée de comptage du compteur IC4, un CD 4040. Un tel compteur comporte 12 étages binaires montés en cascade. Sur la sortie Q6, on obtient ainsi un créneau de forme carrée dont la période est de $0,020 \text{ s} \times 2^6 = 0,02 \times 64 = 1,28 \text{ s}$. La LED L, dont le courant est limité par R11, clignote donc à cette période pour une position correcte du curseur de l'ajustable A1.

Élaboration de la haute tension

Les portes NOR I et II de IC1 réalisent deux inversions successives du créneau présenté sur l'entrée de la première à condition toutefois que les entrées 2 et 5 soient soumises à un état bas. Nous verrons au paragraphe suivant comment cette condition est satisfaite. Les transistors T1/T3 d'une part et T2/T4 d'autre part, forment deux groupements

Darlington dont les fonctionnements alternent au rythme de 50 Hz. Rappelons que de tels montages réalisent de fortes amplifications en courant. Au niveau des deux enroulements 6 volts du transformateur, on observe alors un courant partant du point commun et se dirigeant alternativement vers l'une ou l'autre des extrémités de ces enroulements. Cela se traduit par un flux magnétique alternatif de 50 Hz s'établissant dans l'armature du transformateur. En particulier et aux bornes de l'enroulement 220 volts, on recueille un potentiel alternatif dont l'allure est loin d'être sinusoïdale. Mais cela n'a aucune importance étant donné que les deux alternances de ce potentiel sont aussitôt redressées par un pont de diodes.

Par l'intermédiaire de R19, le groupement des trois capacités C9 à C11 se charge et on peut mesurer un potentiel continu de près de 400 volts sur les armatures positives. Il en est de même pour la capacité C8 qui se charge à travers R18. Les résistances R16 et R17, de valeurs élevées, ont pour mission de décharger les capacités une fois le montage arrêté. Cette précaution peut éviter à l'amateur imprudent de bien désagréables secousses s'il venait, par inadvertance, à toucher les armatures des capacités.

Périodes d'activité

Les portes NAND III et IV de IC2 forment un oscillateur astable délivrant des créneaux de forme carrée à une période pouvant varier de 0,5 à 5,5 secondes, suivant la position angulaire du curseur de l'ajustable A2. Les fronts descendants du signal commandent à leur

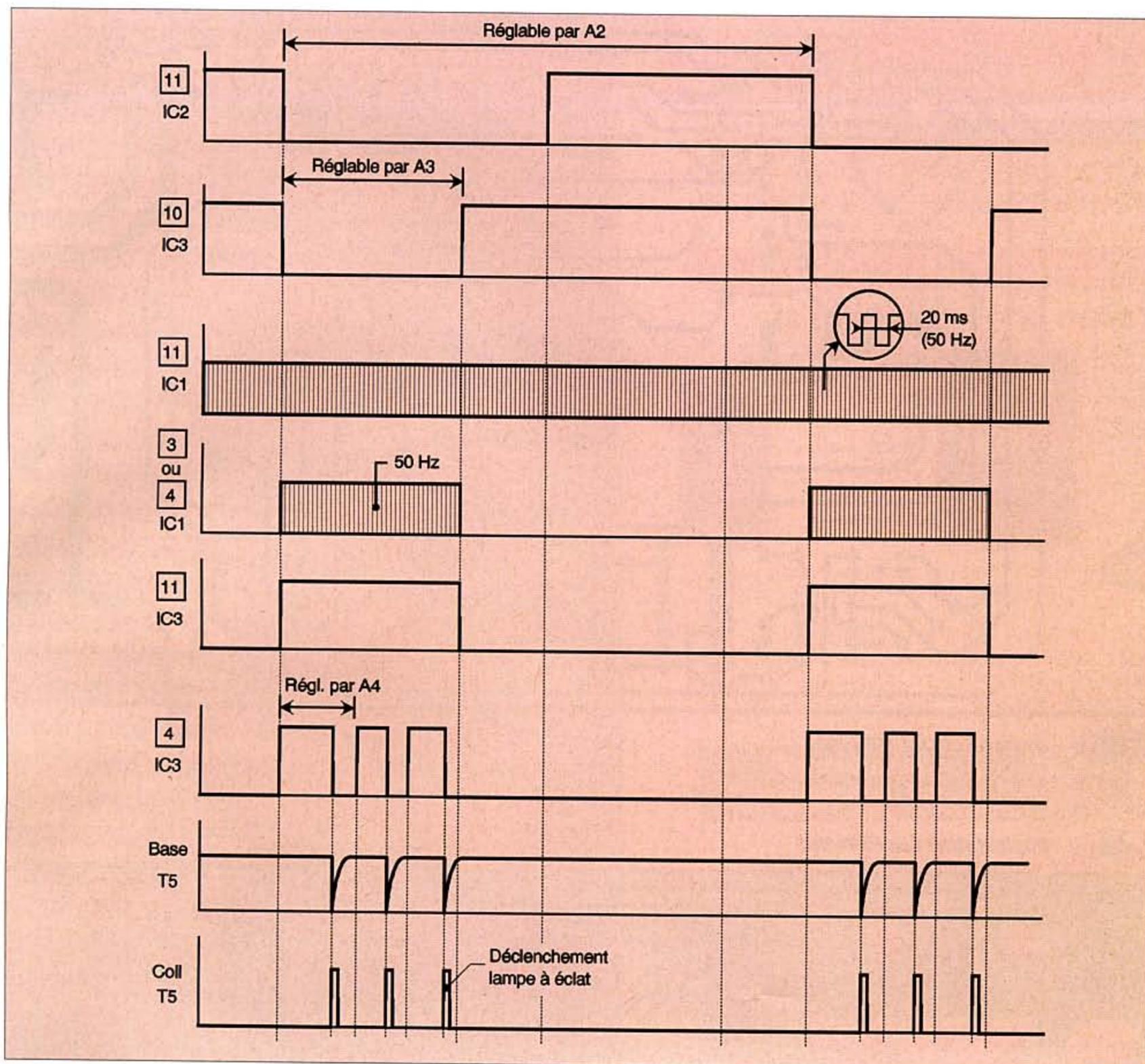
tour une bascule monostable constituée des portes NAND III et IV de IC3. A l'occasion de chaque front descendant présenté sur l'entrée de cette bascule, on enregistre sur sa sortie un état bas dont la durée est réglable de 0,33 à 3,3 secondes grâce à l'ajustable A3. À noter que lors de ces états bas, les portes NOR I et II de IC1 sont opérationnelles c'est à dire que la production de la haute tension évoquée au paragraphe précédent est active.

Commande des déclenchements

Lors des états bas issus de la sortie de la bascule monostable, on recueille des états hauts sur la sortie de la porte IV de IC3. Ces derniers activent l'oscillateur astable formé par les portes I et II de IC3. Ce dernier présente alors sur sa sortie des créneaux asymétriques étant donné la présence de la diode D1. En fait, on observe de brefs états bas d'une durée d'environ 10 millisecondes à une période réglable de 0,1 à 1 seconde (10 à 1 Hz) suivant la position angulaire du curseur de l'ajustable A4. C'est cette période qui caractérise la succession des éclairs. À l'occasion de ces brefs états bas, la capacité C12 se charge rapidement par l'intermédiaire de la jonction émetteur-base du transistor PNP T5 et par R5. Il en résulte la saturation périodique de ce dernier. La capacité C12 se décharge lors des états hauts disponibles sur la sortie de l'oscillateur astable, à travers R20, de manière à se trouver prête pour affronter le signal de commande suivant.

Réalisation des déclenchements

La saturation de T5 a pour effet l'établissement d'un courant dans la jonction gâchette-anode du thyristor THY. Il en résulte un courant important mais bref, provenant de la capacité C8, dans l'un des enroulements de la self de déclenchement. Cet enroulement a un nombre de spires nettement moins important que l'enroulement de sortie. Il en résulte une élévation du potentiel, lequel est aussitôt acheminé sur l'électrode de déclenchement de la lampe à éclat. Cette dernière est alors soumise au déclenchement caractérisé par une brutale décharge du groupement des capacités C9 à C11. Les capacités C9 à C11,



2 Chronogrammes de fonctionnement

ainsi que C8, se rechargent entre deux déclenchements consécutifs de la lampe à éclat.

2. La réalisation

Circuit imprimé (figure 3)

La réalisation du circuit imprimé n'appelle pas de remarque particulière. On notera cependant que les pistes destinées à véhiculer le courant dans les enroulements 6 volts du transformateur se caractérisent par une largeur plus importante. Avant de débuter la fabrication du circuit imprimé, il est plus prudent de se procurer les composants nécessaires au montage de façon à pouvoir éven-

tuellement en adapter le brochage au niveau dimensionnel. Cela est particulièrement conseillé pour le transformateur, la lampe à éclats, la self de déclenchement et l'interrupteur de mise en service.

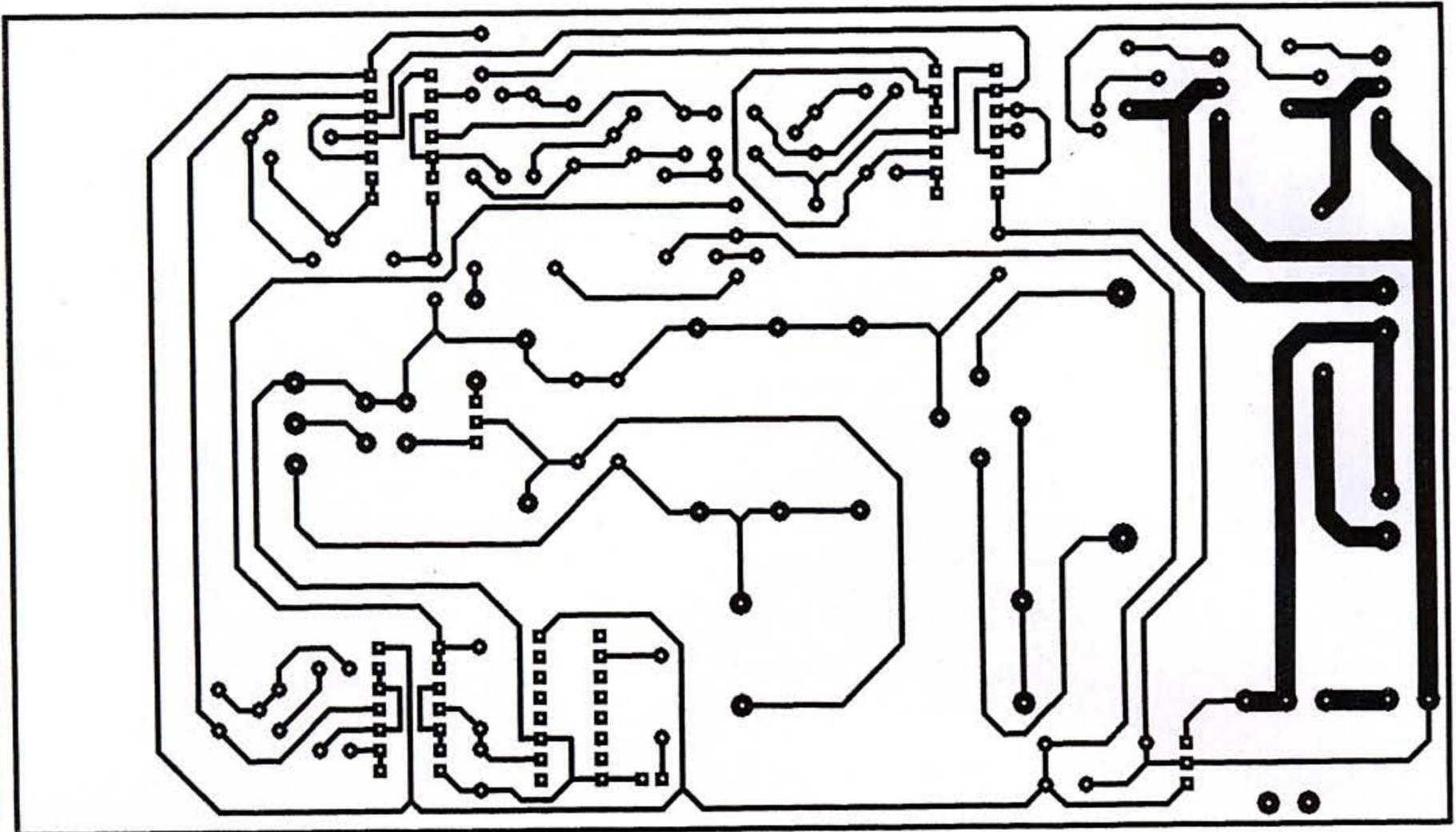
Implantation des composants (figure 4)

Après la mise en place des straps de liaisons, on plantera les résistances, les diodes, les supports de circuits intégrés et les capacités. On terminera par les composants les plus volumineux. Tous les curseurs des ajustables sont à placer en position médiane. Il va sans dire qu'il convient d'apporter un soin particulier au respect de l'orientation des compo-

sants polarisés. Dans un premier temps, il est conseillé de ne pas implanter le transformateur. Cette précaution autorisera les différents réglages sans risquer de toucher aux connexions soumises à des potentiels élevés.

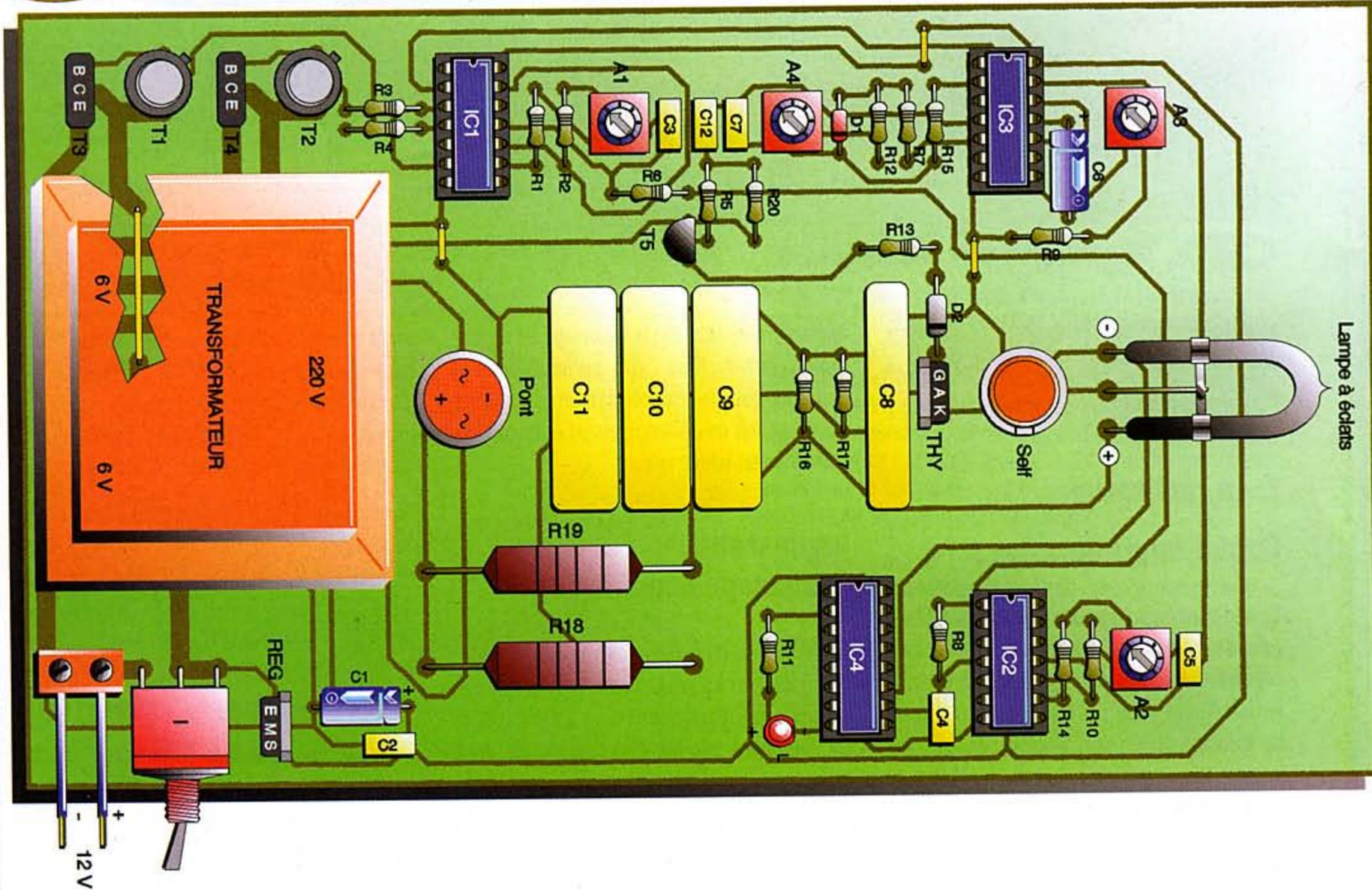
Concernant la self de déclenchement, il est nécessaire de déterminer l'enroulement comportant le nombre de spires le moins élevé. C'est celui qui présente la résistance ohmique la plus faible. La self est à orienter de façon à ce que cet enroulement se trouve alimenté par la cathode du thyristor.

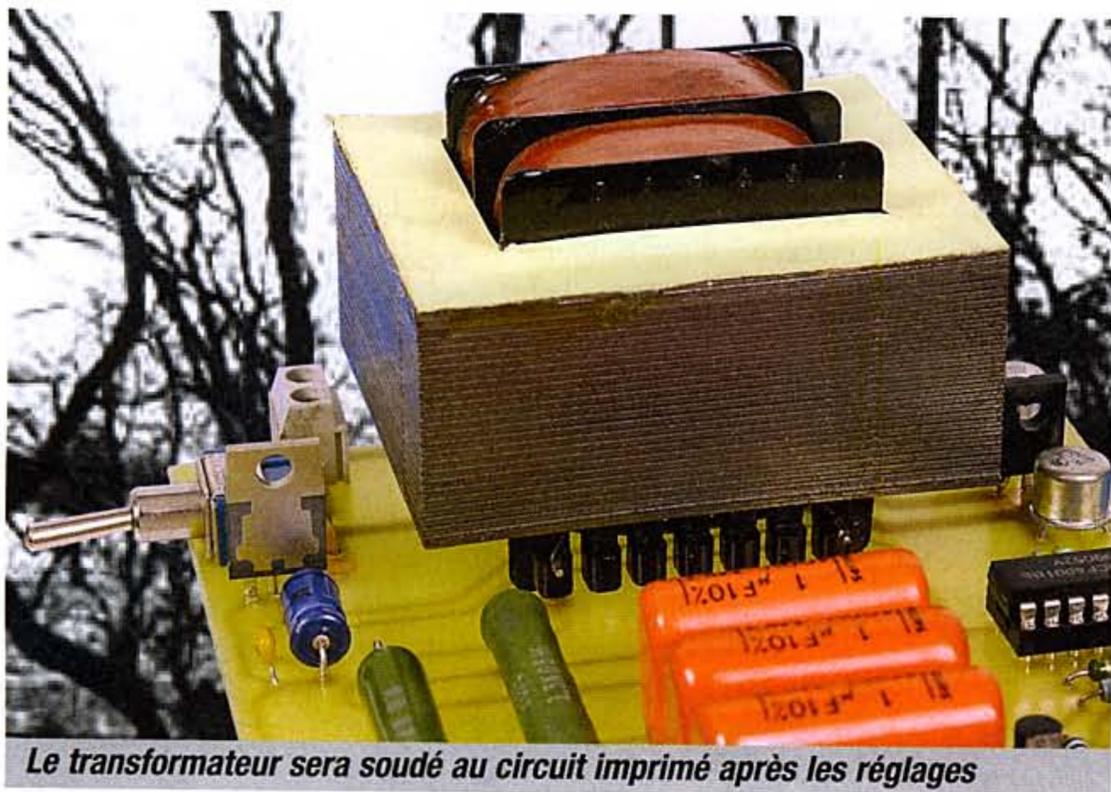
Attention également à l'orientation de la lampe à éclats. Le " plus " est souvent repéré par un point rouge. Si tel n'était pas le cas, on distingue le " moins " par l'opacité du verre au niveau de l'électrode.



3 Tracé du circuit imprimé

4 Implantation des éléments





Le transformateur sera soudé au circuit imprimé après les réglages

Réglages

Il est important de bien respecter la polarité de l'alimentation. La solution de l'allume-cigare est pertinente parce que celui-ci fait office de "détrompeur".

Ajustable A1

Cet ajustable sert à régler la base de temps 50 Hz. La fréquence augmente quand on tourne le curseur dans le sens horaire et inversement. Le réglage est correct quand la LED "L" clignote 10 fois en 12,8 secondes.

Ajustable A2

Il détermine le rythme de succession des périodes actives du montage. Une valeur de l'ordre de 3 secondes convient généralement. La période diminue quand on tourne le curseur dans le sens horaire.

Ajustable A3

Ce réglage sert à déterminer la durée d'une période active. Par exemple, on peut la régler à 1,5 ou 2 secondes. Elle augmente en tournant le curseur dans le sens anti-horaire.

Ajustable A4

Cet ajustable permet de définir la fréquence de succession des éclairs lors d'une période

active. Un réglage à 3 ou 5 Hz donne de bons résultats. La fréquence augmente lorsque l'on tourne le curseur dans le sens horaire.

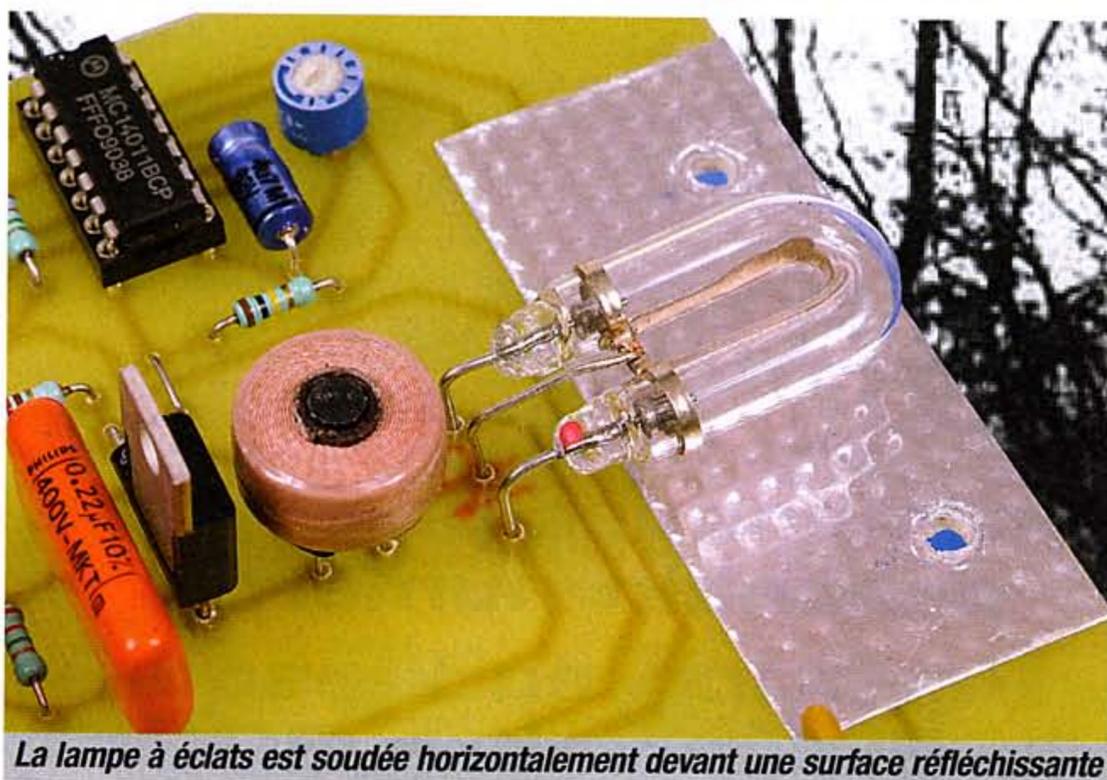
R. KNOERR

Nomenclature

4 straps

- R1 : 1 M Ω (marron, noir, vert)
- R2 à R5 : 4 x 4,7 k Ω (jaune, violet, rouge)
- R6 et R7 : 2 x 10 k Ω (marron, noir, orange)
- R8 à R10 : 3 x 100 k Ω (marron, noir, jaune)
- R11 : 750 Ω (violet, vert, marron)
- R12 : 47 k Ω (jaune, violet, orange)
- R13 : 220 Ω (rouge, rouge, marron)

- R14 à R17: 4 x 2,2 M Ω (rouge, rouge, vert)
- R18 : 10 k Ω / 7 W (bobinée vitrifiée)
- R19 : 1,5 k Ω / 7 W (bobinée vitrifiée)
- R20 : 10 k Ω (marron, noir, orange)
- A1 : Ajustable 220 k Ω
- A2 à A4 : 4 ajustables 1 M Ω
- D1 : Diode-signal 1N 4148
- D2 : Diode 1N 4004
- L : LED rouge diamètre 3 mm
- Pont de diodes
- REG : Régulateur 9 volts - 7809
- C1 : 47 μ F / 16 V - Electrolytique
- C2 et C3 : 2 x 0,1 μ F - Céramique multicouches
- C4 : 1 nF - Céramique multicouches
- C5 : 2,2 μ F - Céramique multicouches
- C6 : 4,7 μ F / 16 V - Electrolytique
- C7 : 1 μ F - Céramique multicouches
- C8 : 0,22 μ F / 400 V - COGECO
- C9 à C11 : 3 x 1 μ F / 400 V - COGECO
- C12 : 2,2 μ F - Céramique multicouches
- T1 et T2 : 2 transistors NPN 2N 1711
- T3 et T4 : 2 transistors NPN BD 677
- T5 : Transistor PNP BC 556
- THY : Thyristor 4 A / 1000V - TYN 1004
- IC1: CD 4001 (4 portes NOR)
- IC2 et IC3: 2 x CD 4011 (4 portes NAND)
- IC4 : CD 4040 (compteur binaire 12 étages)
- 3 supports 14 broches
- 1 support 16 broches
- Lampe à éclats 40 Joules
- Self de déclenchement
- Transformateur 220 V / 2 x 6 V / 24 VA
- I : Interrupteur unipolaire (pour circuit imprimé)
- Bornier soudable 2 plots



La lampe à éclats est soudée horizontalement devant une surface réfléchissante