ELECTRONIQUE PRATIQUE
291
FEVRIER 2005 ■ www.electroniquepratique.com

ELECTRONIQUE PRATIQUE

4,50€

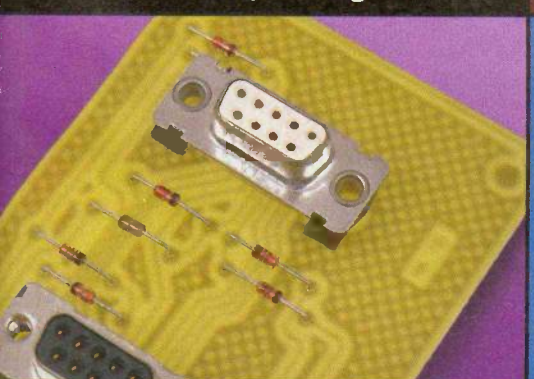
DOMOTIQUE INTELLIGENTE

Télécommande à courants porteurs et PIC

Le récepteur

L'émetteur

Mini-compas digital



Partagez les ports série

Récepteur CB performant

RETROUVEZ AUSSI :

- Une carte «PROACTIVE SIM» ouverte
- Collectionner et restaurer les GSM

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 - 291 - F: 4,50 €

SOMMAIRE

ELECTRONIQUE PRATIQUE

Fondateur : Jean-Pierre VENTILLARD

N° 291 - FÉVRIER 2005

TRANSOCEANIC
S.A.S. au capital de 574 000 €
Président : Patrick VERCHER
2 à 12, rue Bellevue, 75019 PARIS
Tél. : 01.44.84.88.26 - Fax : 01.44.84.88.18
Internet : http://www.electroniquepratique.com

RÉDACTION :
Directeur de la publication :
Patrick VERCHER
Conseiller technique :
Bernard DUVAL
Secrétaire de rédaction :
Annie LECOMTE
Mise en page :
Didier BELLANGER
Couverture :
Dominique DUMAS
Illustrations :
Alain BOUTEVEILLE SANDERS
Photos :
Isabelle GARRIGOU

Avec la participation de : Ph. André, P. Gueulle, G. Isabel, P. Mayeux, Y. Mergy, P. Morin, P. Oguic, A. Reboux, D. Rey, O. Viacava

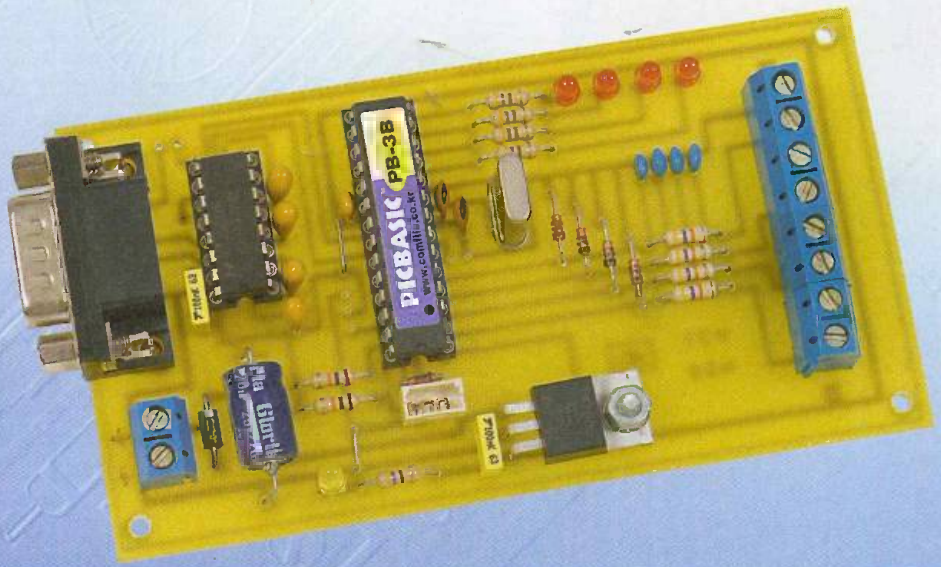
La Rédaction d'Electronique Pratique décline toute responsabilité quant aux opinions formulées dans les articles, celles-ci n'engagent que leurs auteurs.

DIFFUSION/VENTES :
Bertrand DESROCHE
Tél. : **01.44.84.84.54**
N° vert reservé aux diffuseurs et dépositaires de presse :
0800.06.45.12

PUBLICITÉ : Caroline DARSY
2 à 12, rue de Bellevue - 75019 Paris
Tél. : 01 44 84 88 26
Fax : 01 44 84 88 18
E Mail : caroline.darsy@wanadoo.fr

ABONNEMENTS/VPC :
18-24, quai de la Marne - 75019 Paris
Tél. : 01 44 84 85 16 - Fax : 01 42 00 56 92.
Préciser sur l'enveloppe «SERVICE ABONNEMENTS»
Important : Ne pas mentionner notre numéro de compte pour les paiements par chèque postal. Les règlements en espèces par courrier sont strictement interdits.
ATTENTION ! Si vous êtes déjà abonné, vous faciliterez notre tâche en joignant à votre règlement soit l'une de vos dernières bandes-adresses, soit le relevé des indications qui y figurent. Aucun règlement en timbre poste.
Forfait photocopies par article : 5,00 €.
Abonnements USA - Canada : Pour vous abonner à Electronique Pratique aux USA ou au Canada, communiquez avec Express Mag :
USA :P.O.Box 2769, Plattsburgh, N.Y. 12901-0239
CANADA : 8155 rue Larrey, Anjou (QC) H1J2L5
Téléphone : (514) 355-3333
Sans frais : 1 800 363-1310
Fax : (514) 355-3332.
Courriel : expsmag@expressmag.com
Internet : www.expressmag.com
Le tarif d'abonnement annuel pour les USA et le Canada est de 53,90 €.
I.S.S.N. 0243 4911
N° Commission paritaire : 0909 T 85322
Distribution : Transport Presse
Imprimé en France/Printed in France
Imprimerie : SIEP/Bois le Roi - 77590
DEPOT LEGAL FÉVRIER 2005

« Ce numéro a été tiré à 36 500 exemplaires »

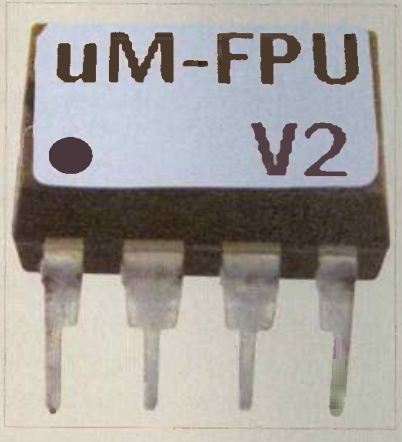


LE PROCHAIN NUMÉRO D'ÉLECTRONIQUE PRATIQUE SERA EN KIOSQUE LE 7 MARS 2005

LEXTRONIC

Coprocesseur mathématique « uM-FPU »

Doté d'un oscillateur intégré, le « UM-FPU » est un nouveau petit coprocesseur mathématique qui se présente sous la forme d'un circuit intégré 8 broches au format DIL.
Facilement interfaçable avec la plupart des microcontrôleurs, des modules PICBASIC et autres Basic-Stamp™ via une communication I2C™ ou SPI™, il vous permettra de bénéficier de possibilités de calculs avec virgule flottante (32 bits IEEE 754) telles que : Set, Add, Subtract, Multiply, Divide, Sqrt, Log, Log10, Exp, Exp10, Power, Root, Sin, Cos, Tan, Asin, Acos, Atan, Atan2, Floor, Ceil, Round, Min, Max, Fraction, Negate, Abs, Inverse, Convert Radians to Degrees, Convert Degrees to Radians, Read, Compare, Status.



Le « UM-FPU » pourra aussi effectuer de nombreuses opérations sur des valeurs de type « long integer » telles que : Set, Add, Subtract, Multiply, Divide, Unsigned Divide, Incrément, Decrement, Negate, Abs, And, Or, Xor, Not, Shift, Read 8 bits, 16 bits, 32-bit, Compare, Unsigned Compare, Status ...
D'autres fonctions de conversion multiples, de création de fonctions spécialisées ou de gestion d'un mode débug font partis de ses nombreuses possibilités.
Le circuit est vendu seul à 14,95 € TTC.

LEXTRONIC, 36 à 40 av. du Gal.-de-Gaulle 94510 La Queue en Brie - Tél. : 01 45 76 83 88
www.lextronic.fr

INFRACOM

PACK VIDÉO SANS FIL

Infracom propose un ensemble complet vidéo d'émission / réception couleur, sans fil (fréquence 2,4 GHz) pour 129,90 € pouvant couvrir de multiples applications de la surveillance discrète jusqu'au modélisme et la robotique.

Cet ensemble est composé d'une micro caméra couleur vidéo sans fil avec antenne intégrée et d'un émetteur de puissance. L'alimentation autonome de 8-9 volts sera fournie par une batterie via un connecteur pour piles ou un jack spécifique (fournie avec le pack). Par sa petite taille, cette caméra peut trouver place aisément sur des maquettes de radio modélisme, petits robots, trains miniatures, mais aussi dans des endroits discrets pour une parfaite surveillance des lieux.
Le récepteur audio vidéo à 4 canaux est muni d'une antenne SMA amovible. Ce récepteur lui aussi de petite taille dispose de sortie vidéo-composite pour une connexion directe sur moniteurs vidéo, TV, cartes de numérisation.

Des cordons audio / vidéo cinch sont fournis pour assurer les liaisons vidéo ainsi qu'un pied de montage pour la caméra vidéo.
Ce pack vidéo peut être amélioré dans ses performances avec l'acquisition d'antennes de gain plus important. Ainsi, pour 115 €, une antenne omnidirectionnelle 2,4 GHz est proposé (référence GP24002) procurant un gain de 15 dBi. Cette antenne est idéale pour l'utilisation extérieure en augmentant de manière sensible la qualité des liaisons Wifi 802.11b ou g.
Une autre antenne omnidirectionnelle 2,4 GHz (référence 24360) est aussi disponible au prix de 59,90 € augmentant le gain de 8dBi. Elle peut être liée à une carte Wifi 802.11b ou g ou à un point d'accès via un cordon de liaison N mâle / RP SMA ou RP TNC.
Enfin une troisième option d'antenne peut être aussi acquise pour 68,17 € sous la référence YF2411USB consistant en un module USB 802.11b intégrant une antenne omnidirectionnelle d'un gain de 5dB sous un design attrayant avec un pied lui assurant une bonne stabilité.
Ce module est livré avec chipset Intersil Prism 2.5, un manuel en anglais, un driver sur CD-ROM (Windows 98 / Me / XP)
Bonne nouvelle ! Pour les lecteurs d'Électronique Pratique, sachez qu'Infracom propose pour 20 € d'achats minimum, une remise de 5% sur toutes commandes pendant tout le mois de février 2005. Il suffit pour cela d'entrer le code promotionnel EP0205. Ce code promotionnel est valable une fois par client pendant la durée de l'opération hors promotions.
Adresse Internet :
http://online.infracom-france.com

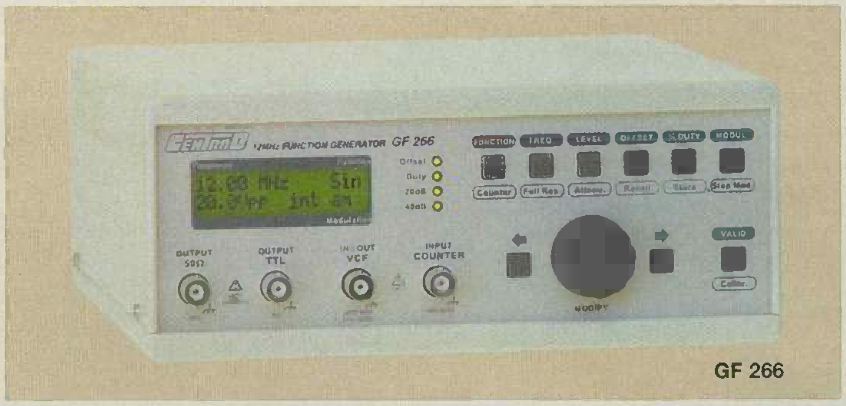
ELC :

Traditions, modernisme, précision

Fondée en 1973 par Monsieur Curri, la société ELC située à Annecy s'est spécialisée dans l'étude, la fabrication de matériel électronique de laboratoire et plus particulièrement dans les alimentations stabilisées à basse tension qui sont considérées comme des références incontournables. Cette société a su investir dans la recherche et le développement afin de répondre exactement aux demandes de l'industrie, de l'enseignement, des laboratoires, des services après-vente et des particuliers.

Les grands axes prioritaires des appareils électroniques ELC sont : la fiabilité, le respect des normes, la garantie des performances annoncées, un prix ultra-concurrentiel, une grande souplesse d'utilisation.

Grâce à une structure dynamique et réactive conjointement à de louables efforts de communication, les produits ELC sont dans leurs catégories devenus incontournables. La réalisation et le montage de ces appareils bénéficient d'unités de montage bien encadrés avec de nombreux tests de vérification à chaque étape de fabrication. La gamme actuelle ELC couvre les alimentations stabilisées de laboratoire, les alimentations stabilisées fixes industrielles, les fréquencemètres, les générateurs de fonction, mires TV, les indicateurs de tableaux analogiques et numériques, les boîtes à décades de résistance, de condensateur, d'inductance, cordons de mesures (isolement au silicone), sondes d'oscilloscopes. Ces divers produits se répartissent en deux marques, avec pour résumer, sous l'appellation Centrad, les instruments de mesure et sous celle d'ELC, les alimentations.



GF 266

Pour donner une idée plus précise de la qualité des produits électroniques, nous avons été impressionnés par les alimentations réglables stabilisées AL936L de 200 Watts. Ces différentes possibilités de configuration permettent d'obtenir des tensions ajustables de 2x0-30V 0-3A ou 1x± 0-30V 0-3A ou 1x0-60V 0-3A ou 1x0-30V 0-6A + 1x2-5,5V 3A ou 1x5,5-15V 1A.

Parmi les particularités, citons un affichage direct de la tension et de l'intensité y compris en mode série ou parallèle, une déconnexion automatique à chaque changement de configuration des voies maîtres et esclaves, un réglage Icc indépendant de la charge et enfin une troisième voie variable de la tension ou de l'intensité. Naturellement cette alimentation est protégée contre les courts-circuits par régulation de courant, contre les montées thermiques avec commande d'un ventilateur par relais et disjoncteur et contre les surintensités au primaire des transformateurs par fusible.

Dans le domaine des appareils de mesure et des générateurs de fonctions proposés par ELC, nous avons retenu le GF 266 sous la marque Centrad. Ce générateur de fonctions de 11 µHz-12 MHz à synthèse numérique directe offre une grande précision de la fréquence (0,005%) et une grande pureté de signal sinusoïdal (distorsion < 0,1%)

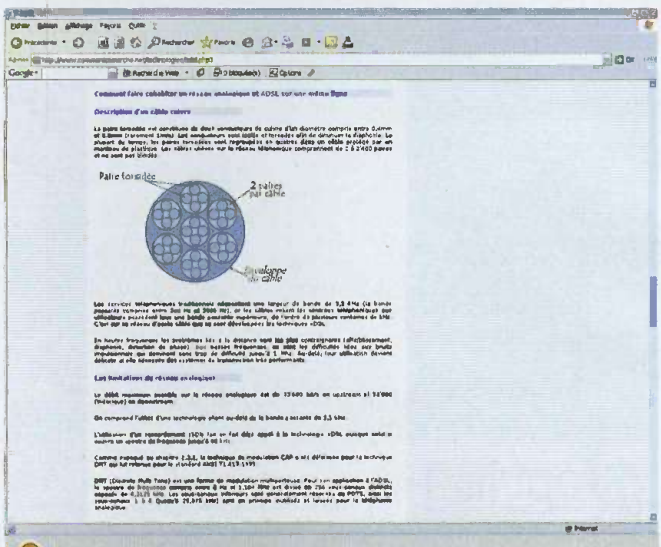
Ce générateur peut fournir des signaux carrés sur la gamme des fréquences citées plus haut avec temps de montée et de descente de 25ns, des signaux triangulaires sur une gamme de fréquence de 0,11 µHz à 5 MHz, des rampes montantes ou descendantes de 22 µHz à 5 MHz. L'affichage des différents paramètres s'effectue par écran à cristaux liquides sur 2 lignes de 16 caractères. L'interface peut être effectuée par liaison RS232.

Naturellement cela n'est qu'un aperçu des appareils proposés par ELC, nous vous conseillons d'aller sur le site internet www.elc.fr. Celui-ci bien réalisé décrit et procure les performances de ces différents matériels, véritables références dans leur domaine.

AL936N

internet PR@TIQUE

Si vous suivez cette rubrique régulièrement, vous appréciez très certainement toute la technologie qui est mise à votre service par votre fournisseur d'accès afin de vous permettre de naviguer sur Internet, à toute heure du jour et de la nuit. Aujourd'hui, les moyens d'accès au réseau Internet sont relativement variés et les connexions par modem RTC, après avoir tenu le haut du panier pendant longtemps, laissent rapidement la place aux technologies xDSL comme en témoignent les très nombreuses offres ADSL disponibles de nos jours. Les prouesses technologiques mises en œuvre pour proposer une connexion haut débit à de nombreux abonnés méritent bien que l'on y consacre cette rubrique.

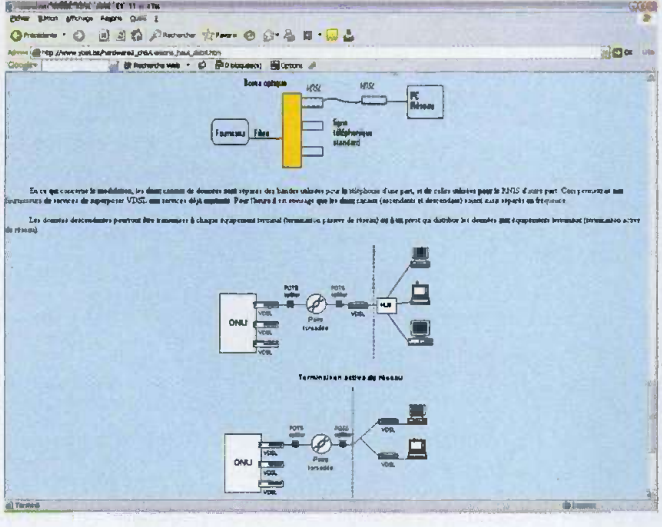


1 *http://www.commentcamarche.net/technologies/adsl.php3*

Comme vous devez vous en douter, sur Internet il n'est pas bien difficile de trouver des informations sur les technologies xDSL (terme générique employé pour regrouper toutes les technologies DSL).

Ceci dit, si vous recherchez les pages contenant le mot « ADSL » à l'aide de votre moteur de recherche favori vous risquez fort de trouver en majorité des annonces pour les très nombreuses offres de connexion haut débit, plutôt que des informations techniques sur les technologies mises en œuvre. Cependant, armés des quelques liens que nous vous proposons dans ce dossier, vous devriez pouvoir vous faire rapidement une idée sur ce sujet sans perdre un temps précieux à trier les pages proposées par un moteur de recherche.

2 *http://www.ybet.be/hardware2_ch6/Liaisons_haut_debit.htm*



Le premier site que nous vous invitons à visiter se situe à l'adresse : http://www.commentcamarche.net/technologies/adsl.php3.

On y trouve une introduction intéressante aux technologies xDSL (ADSL, HDSL, SDSL, etc.) et des explications concises et simples sur les notions de bases mises en œuvre par ces technologies.

Ce site explique, entre autre, quelles performances on peut attendre d'une connexion haut débit et quels sont les phénomènes qui entrent en jeu pour déterminer la fameuse distance maximum vis à vis du central téléphonique pour pouvoir être éligible à un abonnement haut débit par ADSL.

Ce site explique également, de façon succincte, à quoi sert le fameux filtre qu'il faut placer entre la

ligne et les équipements à raccorder au réseau téléphonique lorsque l'on souscrit à un abonnement ADSL.

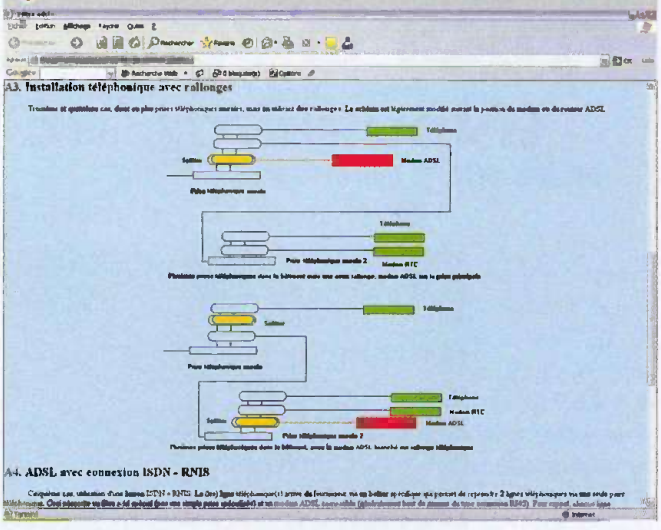
Le site suivant reprend les mêmes notions, pour l'essentiel. Ce site a retenu notre attention en raison des nombreux liens qu'il comporte sur une formation complète sur les réseaux.

Tous les documents sont accessibles gratuitement en ligne.

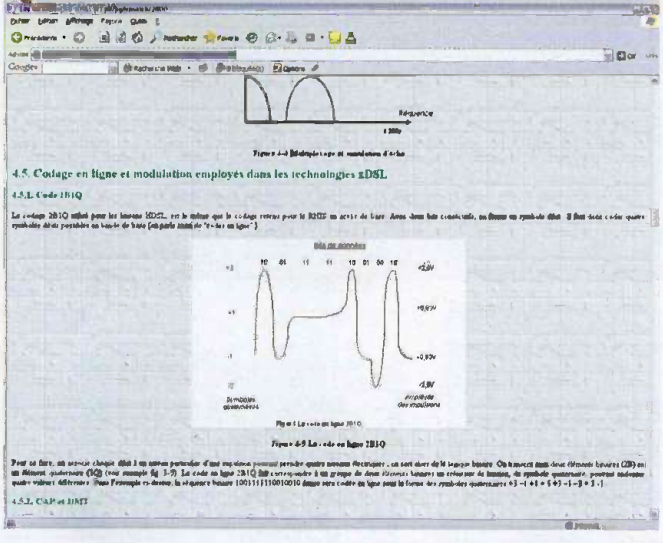
Parmi les nombreuses informations intéressantes que vous pourrez trouver sur ce site, citons la page http://www.ybet.be/hardware2_ch18/filtre_adsl.htm qui indique comment procéder pour installer des filtres ADSL (splitter), dans les configurations les plus courantes.

Pour finir, le dernier site que nous vous proposons de visiter explique d'une façon bien plus détaillée que

3 http://www.ybet.be/hardware2_ch18/filtre_adsl.htm



4 http://www.enseirb.fr/~kadionik/formation/xdsl/xdsl_enseirb.html



les précédents les techniques de modulation et de codage mis en œuvre par les technologies xDSL. L'intérêt principal de ce site est peut être de démystifier en quelques lignes les procédés de transmission qui ont réussi à détrôner nos bons vieux modems RTC pourtant très sophistiqués.

Certes la lecture des quelques pages de ce site ne permet pas de devenir subitement expert en transmission de signal, mais elle permet au moins de comprendre comment les technologies xDSL arrivent à mieux exploiter nos bonnes vieilles paires torsadées en cuivre.

Nous vous souhaitons une agréable découverte des sites proposés, et vous donnons rendez-vous le mois prochain pour de nouvelles découvertes.

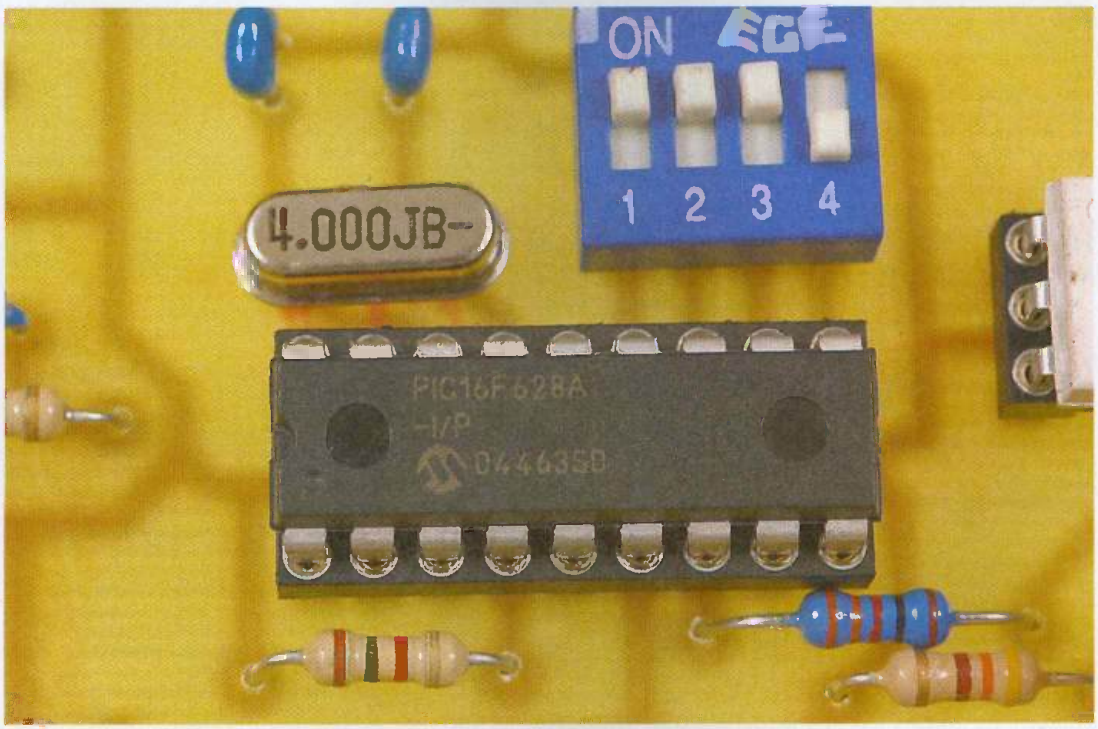
P. Morin

T1 Liste des liens

http://www.commentcamarche.net/technologies/adsl.php3
http://www.ybet.be/hardware2_ch6/Liaisons_haut_debit.htm
http://www.ybet.be/hardware2_ch18/filtre_adsl.htm
http://www.enseirb.fr/~kadionik/formation/xdsl/xdsl_enseirb.pdf
http://www.chez.com/tissier/xdsl/sommaire.htm
http://www.urec.cnrs.fr/cours/Liaison/ip_rtc/sld065.htm
http://www.urec.cnrs.fr/cours/Liaison/ip_rtc/sld066.htm
http://www.urec.cnrs.fr/cours/Liaison/ip_rtc/sld067.htm , etc.
http://www.rd.francetelecom.com/fr/technologies/ddm200306/techfiche1.php

À la découverte des microcontrôleurs PIC

(Neuvième et dernière partie)



Nous allons, dans ce numéro, aborder l'étude du TIMER et du WATCHDOG : ces deux blocs fonctionnels faisant partie intégrante du microcontrôleur PIC. Le Timer du PIC 16F84 est en fait un compteur 8 bits comptant jusqu'à 255, celui-ci est cadencé soit au rythme de l'horloge interne (issue de la fréquence du quartz divisée par quatre), soit par rapport à une entrée de comptage externe.

En effectuant une lecture de ce compteur, nous pouvons ainsi déterminer le "temps qui s'écoule" et faire des actions à des moments précis. Le TIMER sera utilisé lorsque nous aurons besoin de temporisations précises. Un autre avantage du TIMER est que celui-ci une fois initialisé, fonctionne en mode autonome et ne "ralentit" pas le programme en cours.

Synoptique interne du TIMER

Le Timer est composé principalement d'un prédiviseur, d'une entrée de comptage externe, d'un compteur ainsi que d'un système d'aiguillage piloté par des bits de configuration (**Figure 1** et **Figure 1a**).

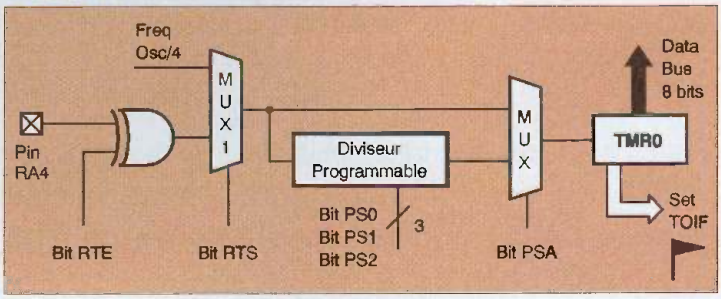
Mise en œuvre et principe de fonctionnement

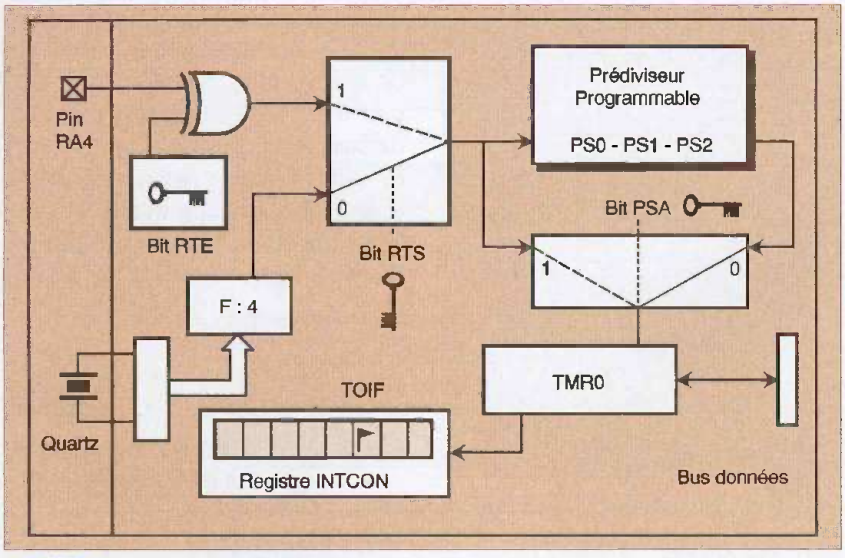
Comme représenté sur les Figures 1 et 1a, l'entrée de comptage peut être issue :

- Soit d'une fréquence appliquée sur la broche RA4, on parlera alors de mode "COUNTER".
- Soit de la fréquence du quartz divisée par 4 (le PIC divise lui même par 4 la fréquence du quartz, dans le cas d'un quartz de 4 MHz nous aurons une période de 1 µs), ce sera alors le mode "TIMER".

Le Bit RTS permet de sélectionner la

1 *Synoptique interne simplifié du TIMER*





1a *Les différents aiguillages*

source de comptage. Si ce bit est à 0, alors en sortie de multiplexeur (MUX1), nous aurons le signal Freq OSC/4, si ce bit est à "1", alors on retrouvera le signal provenant de la broche RA4 (**Figures 1** et **2**). Il est également possible d'inverser le signal issu de la broche RA4, c'est le bit RTE qui est chargé d'effectuer cette opération. Si celui-ci est à "0", alors en sortie du OU exclusif, nous aurons le signal provenant de la broche RA4, dans le cas contraire nous aurons le signal RA4 inversé (**Figure 2**).
En fait on définit ainsi le type de front (montant ou descendant) qui déclenchera le comptage.

	1	0
RTE	Sortie Xor = RA4	Sortie Xor = RA4
RTS	Sortie MUX1 = sortie Xor	Sortie MUX1 = Fosc/4
PSA	Sortie MUX = sortie MUX1	Sortie MUX = Prédiviseur

MUX : Multiplexeur
Xor : Ou exclusif

2 *Bits de Configuration*

Enfin, le bit PSA permet de sélectionner ou non le prédiviseur. Si ce bit est à "0", alors on retrouvera en sortie du multiplexeur (MUX) le signal provenant de la sortie de MUX1 prédivisé par un nombre programmable dans le prédiviseur (par les bits PS0, PS1, PS2). Si le bit PSA est positionné à "1", alors le signal en sortie de MUX sera celui issu de la sortie de MUX1 (on ne prédivise pas dans ce cas).
Tous ces bits que nous venons de voir sont dans le registre OPTION du PIC, nous verrons bien sûr comme d'habitude un exemple pour illustrer l'utilisation du TIMER.
Dans le cas d'utilisation du prédiviseur, les trois bits nommés PS0, PS1, PS2 permettent de définir une division du signal d'entrée de celui-ci. Le tableau en **Figure 3** résume les différentes possibilités de prédivision. Comme vous le constatez, le taux maximum de division sera lorsque les bits PS0, PS1 et PS2 seront positionnés à "1", ce qui signifie que le signal appliqué en entrée de prédiviseur sera dans ce cas divisé par 256 en sortie.

PS2	PS1	PS0	Division par :
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

3 *Détermination du taux de prédivision*

La programmation des différents bits de configuration du fonctionnement du TIMER sera effectuée dans le registre OPTION dont la **figure 4** rappelle les bits à positionner.
Il est à noter qu'à chaque fois que l'on remettra à jour le TIMER ou bien que l'on modifiera un bit de configuration, alors deux cycles d'instructions seront nécessaires avant que le comptage ne commence (pour synchronisation). Si, dans votre programme, vous désirez lire la valeur du compteur (TMR0), cela ne perturbera pas le comptage.

4 *Les bits à positionnement dans le registre OPTION du PIC*

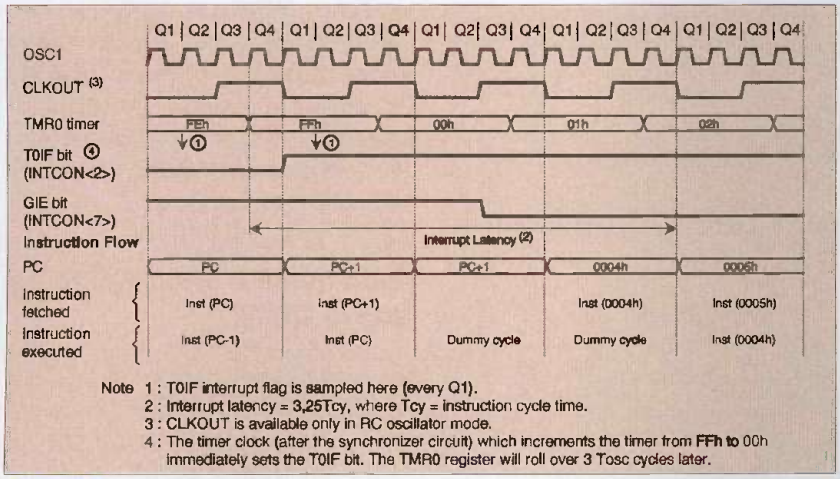
7	6	5	4	3	2	1	0
		RTS	RTE	PSA	PS2	PS1	PS0

Génération d'interruption par le TIMER

Comme nous l'avons vu dans une précédente leçon, le TIMER va pouvoir générer une interruption en passant le flag TOIF du registre INTCON à 1 et ceci dès que le compteur passera de la valeur 255 à la valeur 0. Si vous avez autorisé les interruptions générales (bit GIE) et si vous avez autorisé les interruptions sur le Timer (bit TOIE), alors le programme se déroute dès le débordement du compteur à l'adresse 04 tel que nous l'avons précédemment détaillé (**Figure 5**).
Pour illustrer le fonctionnement et l'utilisation du TIMER, nous vous proposons de réaliser un programme qui aura pour but de faire clignoter une led à une fréquence issue de l'utilisation du TIMER (**Figure 6**).

Détail du programme :

Nous passons rapidement sur les déclarations ainsi que sur les directives que nous avons déjà commentées lors de précédentes leçons. Pour réaliser une temporisation de huit secondes (tel que dans notre exemple de programme), nous allons commencer par définir les bits du registre OPTION.
Les lignes suivantes : MOVLW 0x07 et MOVWF OPTION_REG ont pour effet de positionner le registre OPTION à la valeur 7 soit en binaire 0000 0111. Si nous reprenons les explications sur les bits du registre OPTION on s'aperçoit que :
Le bit RTS = 0 donc on utilise la fréquence interne du quartz / 4,
Le bit PSA = 0 donc on utilise le prédiviseur
Les bits PS2, PS1, PS0 = 1 donc on prédivisera par 256 (voir tableau **Figure 3**)
Si on prédivise par 256 la fréquence du quartz, (elle-même divisée par 4 en interne) on aura donc si le quartz est de 4 MHz :



Note 1 : T0IF interrupt flag is sampled here (every Q1).
2 : Interrupt latency = 3,25Tcy, where Tcy = instruction cycle time.
3 : CLKOUT is available only in RC oscillator mode.
4 : The timer clock (after the synchronizer circuit) which increments the timer from FFh to 00h immediately sets the T0IF bit. The TMR0 register will roll over 3 Tosc cycles later.

5 Le mode interruption avec le TIMER

4 MHz / 4 = 1MHz en entrée de prédiviseur et 1MHz / 256 = 3906,25 Hz en sortie du prédiviseur.
Les prochaines instructions à expliquer sont les suivantes : MOVLW 0x06 et MOVWF TMR0 , ces instructions ont pour effet d'initialiser le compteur du TIMER avec la valeur 6.
Dans la suite de notre programme, on vérifiera que le compteur du TIMER n'est pas à "0", ce qui signifie que celui-ci comptera jusqu'à 256 - 6 soit 250, donc notre signal de 3906,25 Hz sera encore divisé par 250 soit une fréquence de 15,625 Hz.

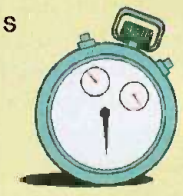
Nous utilisons ensuite un registre temporaire nommé "retard1". Celui-ci est initialisé à la valeur 7D soit 125 en décimal, cette valeur de registre nous permet d'effectuer une boucle. Pour conclure, nous avons un signal de 15,625 Hz que nous allons encore diviser par 125 (puisque nous allons effectuer la boucle 125 fois) donc en sortie de la boucle de temporisation nous aurons compté :
15,625 Hz / 125 soit 0.125 Hz ce qui représente une période de 1 / 0.125 = 8 secondes (**Figure 7**). Les leds connectées sur le port B s'allumeront pendant 8 secondes et resteront éteintes également 8 secondes. Si vous avez compris le principe, il est très facile de modifier les temps obtenus en jouant sur le prédiviseur et le nombre de boucles à effectuer pour avoir d'autres temporisations.

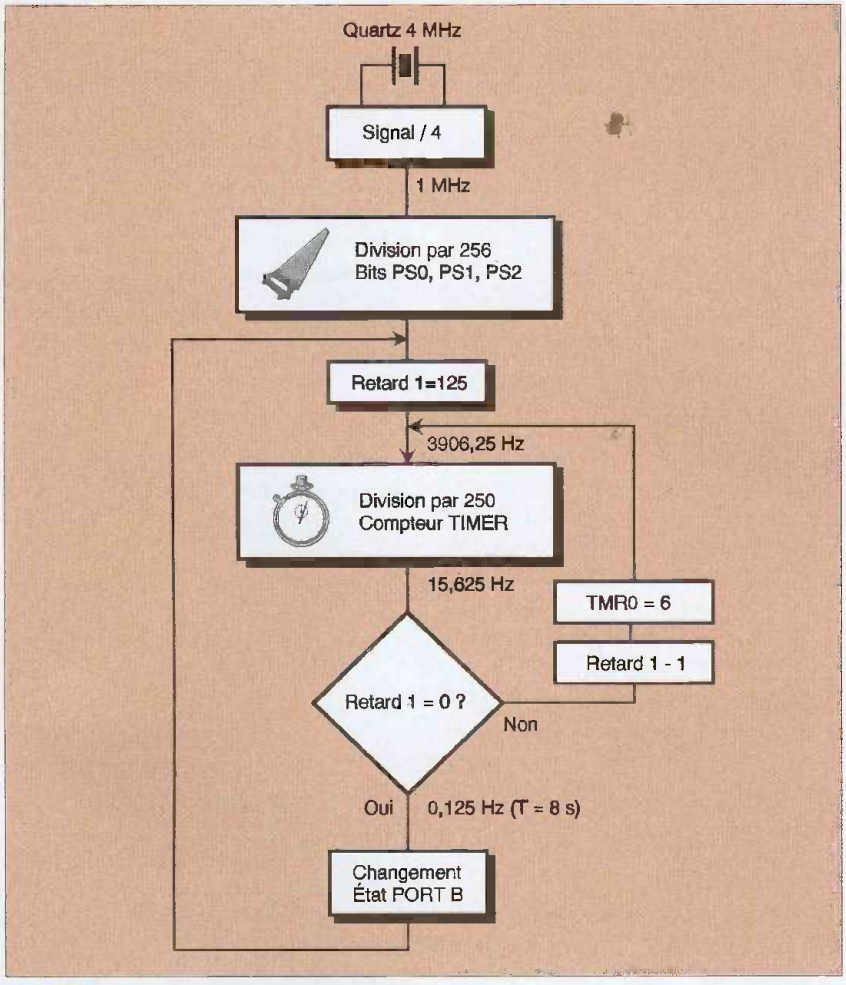
Le WATCHDOG du PIC

Le WATCHDOG ou "chien de garde" du microcontrôleur PIC peut être assimilé à un mécanisme qui va surveiller si votre programme s'exécute toujours dans de bonnes conditions et dans le cas contraire, ce processus va faire un reset du microcontrôleur. Le fonctionnement du WATCHDOG est lié comme le Timer à un compteur interne. Ce compteur est incrémenté depuis un circuit RC dont la fréquence peut évoluer selon la tension d'alimentation, le type de PIC utilisé et bien sûr selon la température (**Figure 9**). La période typique donnée par le fabricant

6 Programme illustrant l'utilisation du TIMER

```
;------ Exemple d'application avec un PIC : Essai sur le temporisateur interne ------
; Titre : TIMER
; Date : 01 - 2005
; Auteur : P.M
; PIC utilisé : PIC 16 F 84
; on utilise le timer pour avoir une temporisation multiple de la seconde
; si le registre OPTION ( bits PS0,PS1,PS2 ) =
; 04 --> 1 S ; 05 --> 4 S ; 07 --> 8 S ; 03 --> 0.5 S ; 02 --> 0.25 S
; 01 --> 0.0125 S ; 00 --> 0.00625 S
; le montage fonctionne avec un Quartz de 4 MHz

;---------- Directive d' assemblage pour PLAB ----------
list p=16f84A
#include p16f84A.inc
__config H'3FF9'
;---------- Définition des registres temporaires ----------
retard1   EQU   0x0C   ; le registre temporaire retard1 se trouve à l' @ 0C
;---------- Init des ports A et B ----------
ORG 0
bsf STATUS,5          ; on met à 1 le 5ème bit du registre status pour accéder
                      ; à la 2ème page mémoire ( pour trisb et registre OPTION )
MOVLW 0x00            ; on met 00 dans le registre W
MOVWF TRISB           ; on met 00 dans le port B il est programmé en sortie
MOVLW 0x07            ; on met 07 dans le registre W
MOVWF OPTION_REG      ; on met 07 dans le registre OPTION ( 8 secondes )
                      ; 1MHz / 256 = 3.90625 kHz
bcf STATUS,5          ; on remet à 0 le 5ème bit du registre status
;---------- Programme principal ----------
debut
MOVLW 0x06            ; on met 06 dans le registre W
MOVWF TMR0            ; on met W dans le registre du TIMER 0
                      ; le timer comptera jusqu' à 256 - 6 = 250
                      ; 3.90625 kHz / 250 = 15.625 Hz
MOVLW 0x7D            ; on met 7D ( 125 ) dans le registre W
MOVWF retard1         ; on met W dans le registre retard1
                      ; 15.625 Hz / 125 = 0.125 Hz et 1/0.125 = 8 S
call tempo            ; appel du sous programme de temporisation
COMF PORTB            ; change l 'état des leds à chaque fois que la tempo est finie
goto debut

tempo
movf TMR0,w           ;on met le contenu du registre TMR0 dans w
btfss STATUS,2        ;test bit Z (z=1 si w=0) si z=1 on saute la prochaine instruction
goto tempo

MOVLW 0x06            ; on remet 06 dans le registre W
MOVWF TMR0            ; on met W dans le registre du TIMER 0
                      ;(on recharge TMR0 avec 6)
DECFSZ retard1,F      ; on décrémente retard1 et on saute la prochaine instruction si
GOTO  tempo           ; le registre retard1 = 0 sinon retour à tempo
return
END
```





7 *Synoptique simplifié du programme*

sur ce circuit RC pour une alimentation de 5 V et une température de 25 °C est de 18 ms. Celle-ci peut varier entre 7 et 28 ms.

Si vous utilisez le WATCHDOG (nous verrons plus loin qu'il faut configurer un bit fusible dans la variable _CONFIG) et que vous lancez votre programme, alors 18 ms (période typique du WATCHDOG) plus tard, le PIC se RESET.

Pour éviter cela, il faudra insérer dans votre programme la ligne " **clrwdt** " (clear WATCHDOG) qui aura pour effet de remettre à zéro le compteur du WATCHDOG avant expiration du délai programmable (nous le verrons par la suite).

Si cette instruction n'est pas exécutée, par exemple si votre programme est "planté" suite à un parasite ou si celui-ci est dans une boucle infinie... alors le PIC sera redémarré à l'adresse de RESET (adresse 0) et donc votre programme redémarrera.

Il est à noter qu'un bit du registre STATUS (le bit TO) est remis à zéro dès que le délai du WATCHDOG a expiré, cela permet au démarrage du PIC de tester d'où vient le RESET (mise en route par l'utilisateur ou bien redémarrage sur délai d'expiration du WATCHDOG).

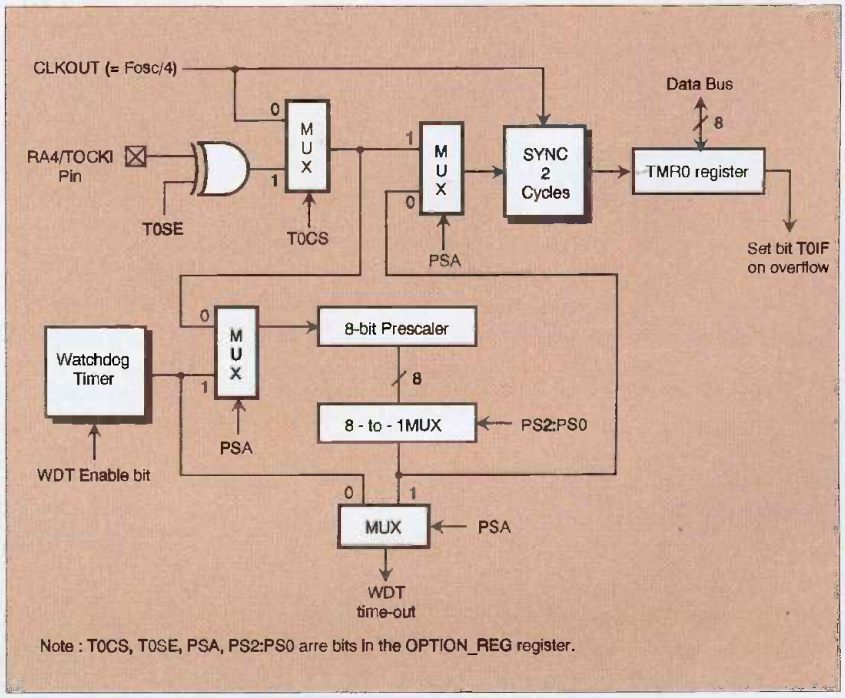
Pour conclure, le WATCHDOG est comme un compte à rebours, si celui-ci n'est pas arrêté à temps, (instruction clrwt) alors le PIC redémarre.

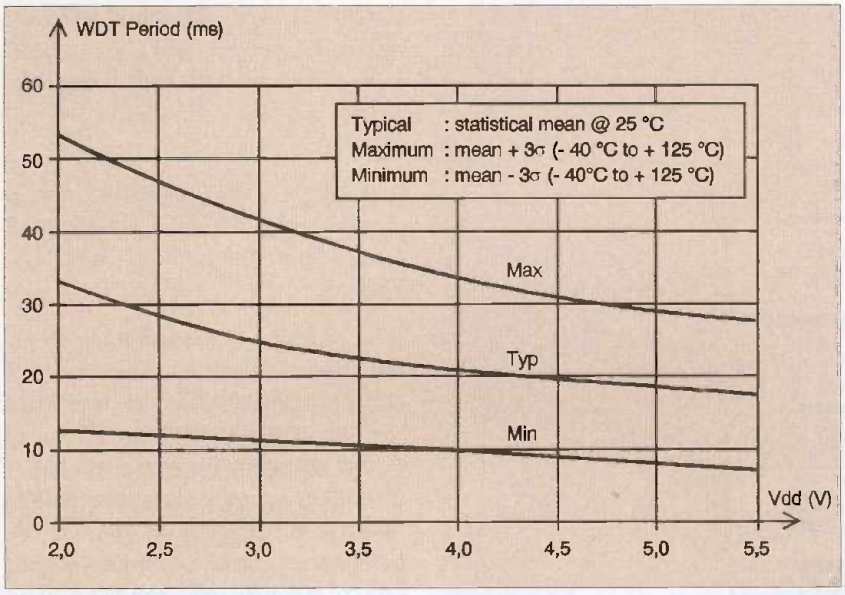
Sur le synoptique de la **Figure 8**, on peut voir que la période du WATCHDOG peut être allongée en utilisant le diviseur que nous avons déjà détaillé pour le TIMER. Sachant que ce diviseur est commun, une utilisation pour le WATCHDOG interdira l'utilisation pour le TIMER et vice-versa. Le bit PSA fera l'aiguillage pour l'utilisation du diviseur, si ce bit est à "0" le diviseur sera utilisé pour le TIMER et si ce bit est à "1", alors ce sera pour le WATCHDOG.

Attention, dans le cas de l'utilisation du diviseur pour le WATCHDOG, le taux de division est différent par rapport au TIMER. Le tableau de la **Figure 10** reprend ce taux. Comme vous le voyez, la période max (typique pour une alimentation de 5 V) sera de 18 ms x 128 = 2304 ms soit 2,3 secondes.

Un programme utilisant le WATCHDOG est proposé en **Figure 11**, ce programme allume tout simplement les leds connectées au port

8 *Synoptique du TIMER et du WATCHDOG*





9 *Période du WATCHDOG en fonction de la tension d'alimentation*

PS2	PS1	PS0	Division par :
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

10 *Le taux de prédivision selon PS0-PS1-PS2*

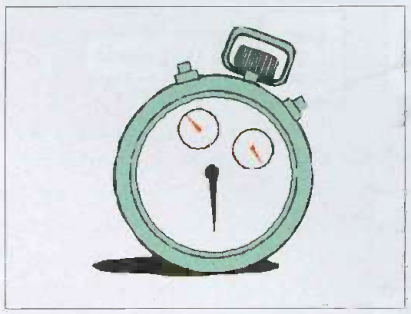
B au démarrage puis environ 2,3 secondes après, le PIC se reset et les leds changent d'état et ainsi de suite. Pour vérifier le bon fonctionnement de la RAZ du WATCHDOG, il suffira d'insérer dans la boucle "Goto debut" l'instruction ...

Clrwdt (cette instruction est en commentaire dans le programme **Figure 11**).

Dans le programme proposé, vous remarquerez que pour utiliser le WATCHDOG, il faut déclarer le fusible correspondant par la directive :

_CONFIG _WDT_ON

Pour utiliser le prédiviseur, il faut mettre le bit PSA du registre OPTION à "1" ainsi que les bits PS0-PS1-PS2 pour avoir la période la plus longue offerte par le WATCHDOG (division par 128) dans notre exemple.



Pour conclure ce dernier chapitre

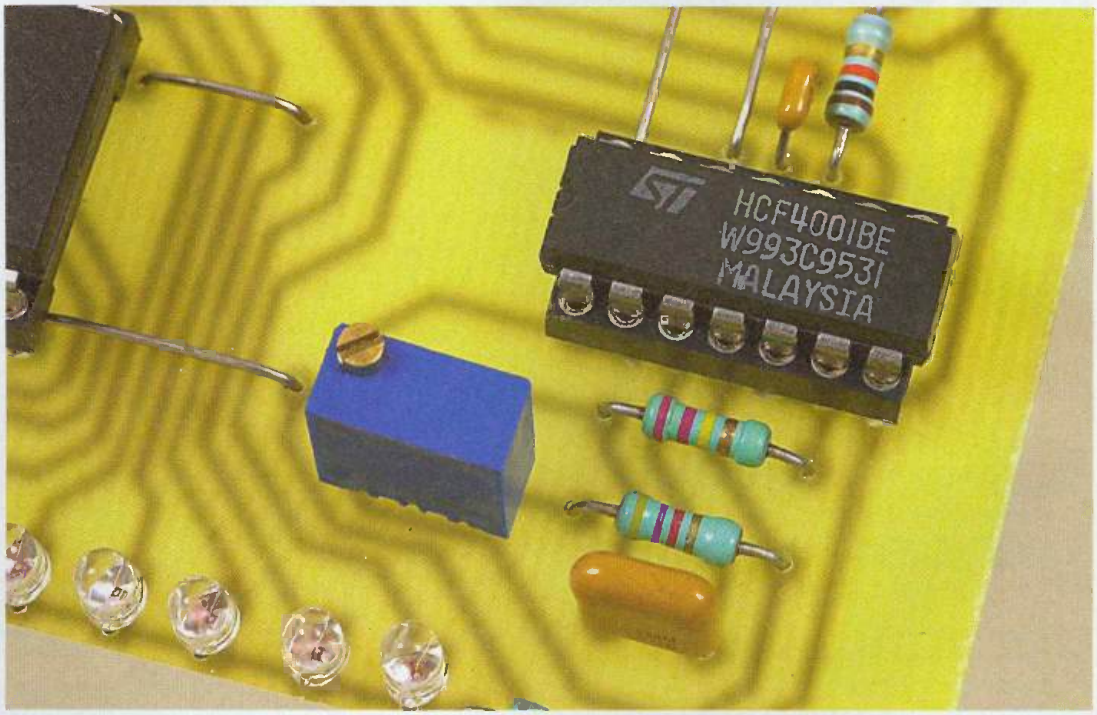
Avec cette ultime partie, nous terminons cette découverte des microcontrôleurs PIC. Nous espérons que ces leçons vous auront donné entière satisfaction et surtout vous auront donné envie d'en savoir plus sur ces composants, dignes de toute notre attention. Vous pouvez télécharger les fichiers sources, comme pour les précédentes leçons, sur le site de l'auteur.

P. MAYEUX
http://perso.libertysurf.fr/p.may

11 *Le programme du WATCHDOG*

```
;------ Exemple d'application avec un PIC : Essai sur le WATCHDOG ------
; Titre : WATCHDOG
; Date : 01 - 2005
; Auteur : P.M
; PIC utilisé : PIC 16 F 84
; essai du WATCHDOG période typique = 2.2 seconde
; le montage fonctionne avec un Quartz de 4 MHz

;---------- Directive d' assemblage pour PLAB ----------
list p=16f84A
#include p16f84A.inc
__config _CP_OFF & _WDT_ON & _PWRTE_ON & _HS_OSC

;---------- Init des ports A et B ----------
ORG 0
bsf STATUS,5        ; on met à 1 le 5ème bit du registre status pour accéder
                    ; à la 2ème page mémoire ( pour trisa et trisb et OPTION )
MOVLW 0x00          ; on met 00 dans le registre W
MOVWF TRISB         ; on met 00 dans le port B il est programmé en sortie

MOVLW 0x0F          ; on met 0F dans le registre W
MOVWF OPTION_REG    ; on met 0F dans le registre OPTION
                    ; PSA= 1 PS2=1 PS1=1 PS0=1

bcf STATUS,5        ; on remet à 0 le 5ème bit du registre status pour accéder
                    ; à la 1ère page mémoire
COMF PORTB          ; on change d'état les leds du port B

;----------- Programme principal ----------
debut
  ;clrwdt           ; pour mettre à "0" le compteur du watchdog
                    ; en commentaire pour essayer
  goto debut

END
```

Pratiquez la logique séquentielle



Pour faire suite à l'article sur la maîtrise des fonctions logiques de base paru dans le numéro 290 d'Électronique Pratique, nous vous proposons une étude des fonctions plus complexes. Nous avons vu précédemment qu'il est possible de combiner plusieurs portes entre-elles et que l'état logique de la sortie est directement lié au niveau des entrées, en temps réel. Un changement d'état d'une entrée se traduit par l'analyse et éventuellement par le basculement immédiat de la sortie.

En logique séquentielle, nous assemblons également plusieurs portes, mais nous faisons intervenir parfois quelques composants passifs (résistances et condensateurs). Il en résulte que l'état logique des entrées et des sorties se modifie dans le temps. Le niveau d'une sortie ne dépend plus de l'état immédiat de ses entrées, mais également de leur état antérieur. Il s'agit des monostables, des oscillateurs astables et des bascules bistables.

La bascule bistable (type RS)

Commençons par l'étude de la bascule bistable. Comme un relais bistable électromécanique, cette fonction logique peut prendre deux états stables. Le type RS pour « Reset, Set » sous-entend qu'une impulsion de niveau haut sur l'entrée « R » initialise la sortie « Q » de la bascule à 0 et « Q/ » à 1. La même action sur l'entrée « S » provoque la situation inverse.

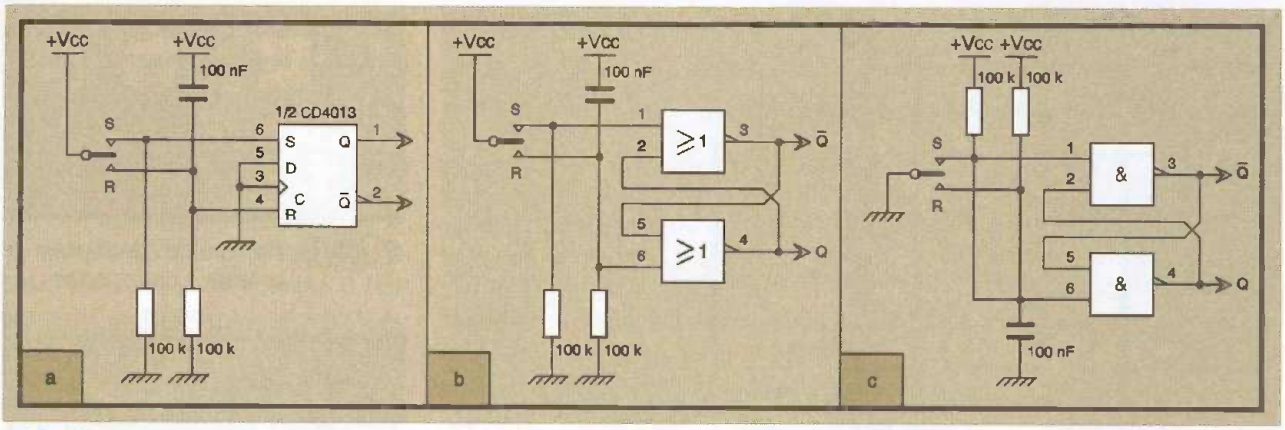
La bascule est aussi appelée fonction mémoire. Il existe plusieurs manières de réaliser cette fonction. La plus simple (**figure 1a**) consiste à utiliser un demi circuit CD4013. Vous pouvez également la câbler (**figure 1b**) à l'aide de deux portes NON-OU.

La logique inverse (impulsion 0 sur les entrées) s'obtient au moyen de deux portes NON-ET (**figure 1c**). Le réseau RC sert à choisir la position « R » à l'initialisation.

Pour aider à comprendre le principe de la bascule RS, le **tableau 1** donne les tables de vérité de cette fonction. Notez deux états particuliers: le cas où les deux entrées sont à 0 n'influe pas sur les sorties ; et le cas où les deux entrées seraient simultanément à 1 tendrait à vouloir mettre les deux sorties complémentaires au même potentiel, chose impossible en pratique!

La bascule bistable (type D)

La seconde bascule, de type « D », est aussi appelée verrou. Son fonctionnement est directement asservi à l'état de l'entrée D. Si celle-ci est au niveau logique 1, le front montant d'une impulsion sur l'entrée C positionne la sortie Q à 1 et Q/ à 0. Dans le cas contraire, la même action portera la sortie Q à 0 et Q/ à 1. D'où l'idée de



1 *La bascule type " RS "*

Situations 1a et 1b

Entrées		Sorties	
R	S	Q	$\bar{Q}$
0	0	Inchangées	
↓	0	0	1
0	↓	1	0
↓	↓	Interdit	

Situation 1c

Entrées		Sorties	
R	S	Q	$\bar{Q}$
0	0	Inchangées	
↓	0	1	0
0	↓	0	1
↓	↓	Interdit	

Tableau 1
Tables de vérité d'une bascule " RS "

relier l'entrée D à la sortie Q/, cette situation a pour effet de faire basculer l'état des deux sorties à chaque front montant sur l'entrée C. Ce principe s'apparente, en électromécanique, au télérupteur. Le circuit intégré CD 4013 comporte deux bascules pouvant être utilisées en type « RS », mais également, comme le montre la **figure 2**, en type « D ».

L'entrée D, au moyen de S2, peut être portée au niveau 0, 1, ou être reliée à la sortie Q/. Afin d'éviter un état aléatoire à la mise sous tension, dû uniquement au temps de réponse des portes intégrées, un circuit RC agit sur l'entréé R pour mettre la sortie Q à 0.

Si nous appliquons une fréquence sur l'entrée C, lorsque l'entrée D est reliée à la sortie Q/, nous sommes en présence d'un diviseur par 2 car la sortie Q présentera un niveau logique haut tous les deux fronts montants.

Afin d'éviter un comportement indésirable de la bascule sur une commande manuelle, l'entrée "C" doit être raccordée à un monostable anti-rebonds, nous étudierons ce type de circuit d'ici quelques lignes. Le **tableau 2** donne la table de vérité très complète de la bascule "D", et de la bascule "RS" contenues dans un demi CD 4013.

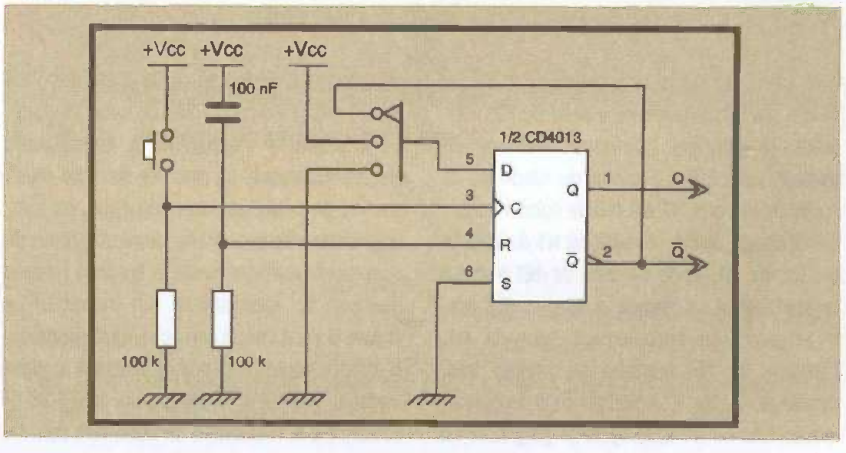
L'oscillateur astable

Cette fonction très courante destinée à produire une oscillation permanente, donc une fréquence, peut se concevoir de plusieurs manières. Le schéma le plus simple n'utilise qu'un inverseur à seuil. Une porte NON-ET à seuil, intégrée dans un circuit CD4093, ou un inverseur d'un CD40106 en logique CMOS convient parfaitement. Le schéma de la **figure 3a** montre le premier cas. Étudions plus précisément son fonctionnement. À la mise sous tension, le condensateur C déchargé se comporte comme un court-circuit. Il envoie donc une impulsion négative sur l'entrée de la porte logique à seuil. La sortie présente alors un niveau haut, et le condensateur se charge progressivement à travers la résistance R. Lorsque sa tension arrive au niveau du seuil de basculement haut de la porte logique, l'entrée passe à 1 et la sortie bascule à 0. La résistance R sert maintenant au cycle de décharge du condensateur. Quand la tension de seuil bas est atteinte, le système bascule à nouveau et ceci indéfiniment. La formule approchée permettant de calculer la fréquence de l'oscillateur est la suivante :

```
F(en Hz) = 1/0,8 . R(en MΩ) . C(en µF)
   Exemple avec R=560kΩ et C=2,2µF :
   F(en Hz) = 1 / (0,8 . 0,56 . 2,2)
            = environ 1Hz
```

Tel qu'il est conçu sur la **figure 3a**, l'oscillateur fonctionne indéfiniment. Le seul moyen, peu élégant de l'interrompre, consiste à couper son alimentation. Si vous souhaitez le

2 *Mise en œuvre d'une bascule de type " D "*



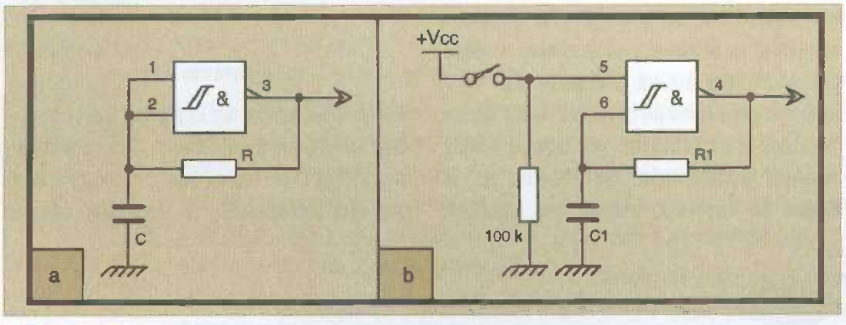
État	Entrées				Sorties	
	R	S	D	C	Q	$\overline{Q}$
Mise à 0	1	0	X	X	0	1
Mise à 1	0	1	X	X	1	0
Interdit	1	1	X	X	Selon fabricant	
Basculement	0	0	0	↑	0	1
	0	0	1	↑	1	0
	0	0	X	0	Sans changement	
	0	0	X	1	Sans changement	
	0	0	X	↓	Sans changement	

Tableau 2
Tables de vérité des bascules " D " et " RS "

commander par un dispositif numérique par exemple, vous devrez vous conformer au schéma de la **figure 3b**. La porte NON-ET s'utilise alors dans sa fonction initiale et le basculement de sa sortie est soumis à l'état de l'entrée positionnable. Référez-vous à la table de vérité parue dans le numéro 289 d'Électronique Pratique.

Le second schéma d'oscillateur astable évite l'emploi de portes à seuil et peut se concevoir à l'aide de portes NON-OU, NON-ET, ou simplement d'inverseurs logiques. Ce montage est couramment employé au sein même de certains circuits intégrés comportant un oscillateur et des diviseurs de fréquence ; voyez le CD4060, CD4521, etc. Le schéma de la **figure 4** montre le câblage de cette fonction. La valeur de la résistance R1 doit être cinq à dix fois supérieure à celle de R2. Étudions le principe de fonctionnement de manière simplifiée. Partons du stade où la sortie de la porte N2 est à 0, le condensateur C1 déchargé porte l'entrée de N1 à 0 via la résistance R1, donc l'entrée de N2 à 1. Le condensateur se charge à travers R2 jusqu'au seuil de basculement haut de N1. L'entrée de N2 passant au niveau bas, décharge C1 par R2 jusqu'au seuil de basculement bas de N1 et le cycle se perpétue. La résistance R1 n'influe pas sur les calculs, elle sert juste à maintenir l'oscillation. De plus, il convient de ne pas dépasser certaines valeurs trop élevées pour C1. La formule approchée permettant de calculer la fréquence de l'oscillateur est la suivante :

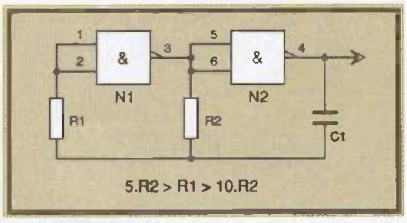
```
F(Hz.) = 1 / 2,2 . R2(en MΩ) . C1(en µF.)
  Exemple avec R2=470kΩ ; R1=3,3MΩ et
                C1=220nF :
  F(Hz.) = 1 / (2,2 . 0,47 . 0,22)
         = 4,395 Hz
```



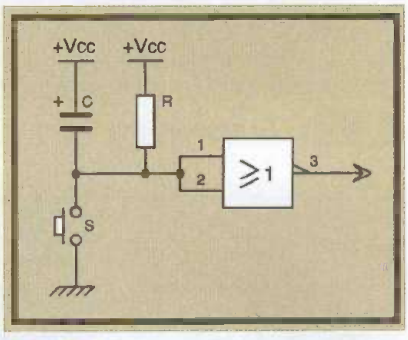
3 ***Schéma de l'oscillateur astable à une porte***

Le monostable à effets multiples

Comme son nom le laisse deviner, le monostable, contrairement à l'oscillateur, ne fournit qu'une impulsion en sortie. L'anti-rebonds est un des principaux rôles de cette fonction. Le monostable s'apparente à la minuterie électromécanique. Le premier de cette étude permet de prolonger une impulsion, ou l'action sur une touche. Si une autre impulsion de commande survient avant la fin de la temporisation, le monostable est relancé. La **figure 5** vous donne son schéma de principe. N'importe laquelle des trois portes à logique inverse: NON-OU, NON-ET ou INVERSEUR convient à la réalisation de cette fonction. Au repos, les deux bornes du condensateur C sont au potentiel de +Vcc (via la résistance R), la sortie présente alors un niveau logique 0. Une action sur le bouton poussoir, ou tout signal de niveau logique 0 appliqué sur les entrées de la porte charge instantanément le condensateur C. Celui-ci ne commencera sa décharge à travers la résistance R qu'après relâchement du bouton poussoir. Il s'en suit un niveau logique 1 en sortie tant que les entrées n'auront pas retrouvé leur seuil de tension inférieur. Si une nouvelle action se produit sur la touche avant la fin du cycle, l'impulsion positive en sortie est prolongée car le condensateur est rechargé. La formule approchée donnant le calcul du temps T en sortie, après relâchement du bouton poussoir, est la suivante :



4 ***Schéma de l'oscillateur astable à deux portes***



5 ***Monostable à effets multiples***

```
T(en S) = 0,7 . R2(en MΩ) . C1(en µF)
  Exemple avec R=470kΩ et C=2,2µF :
T(en S) = 0,7 . 0,47 . 2,2 = 0,7238 S.
```

Ce circuit prend en compte chaque impulsion de niveau logique bas. Il peut aussi servir d'intégrateur pour détecter les impulsions manquantes d'un cycle continu: la sortie passe à 0 lors du défaut. Le diagramme de la **figure 6** clarifie ce principe.

Le monostable simple effet à logique positive

Ce monostable génère une impulsion de niveau logique haut sur un front montant en

entrée. Contrairement à son homologue à effets multiples, il ne prend pas en compte une autre commande en entrée avant la fin de la temporisation. La durée du créneau positif en sortie est invariable. La **figure 7** dévoile le schéma du monostable à logique positive.

Ce circuit utilise deux portes NON-OU d'un CD 4001. Au repos, la sortie de N2 est au niveau bas, l'entrée de N1 (broche 2) l'est donc aussi. L'autre (broche 1) est positionnée à 0 par la résistance R1. La sortie de N1 présente donc un niveau haut sur la patte négative du condensateur C1. Celui-ci reste déchargé car son extrémité positive est portée au niveau 1 par la résistance R2. La sortie de la porte N2 est stable au niveau bas car ses deux entrées sont également à 1.

Une action sur le bouton poussoir S1, ou toute impulsion de niveau logique 1 à ce point, fait basculer la sortie de N1 au niveau logique 0 lors du front montant, en forçant l'entrée (broche 1) à 1. Il s'en suit une charge instantanée de C1 portant les entrées de N2 à l'état bas, et par conséquent sa sortie à 1. Tant que le condensateur ne s'est pas déchargé via la résistance R2, cet état est auto-maintenu par l'entrée de N1 reliée à la sortie de N2, même si l'action sur S1 est relâchée. La résistance R3, facultative, est une protection dans le cas où la valeur de C1 serait trop élevée. Voici la formule approchée du calcul de la durée « T » de l'impulsion :

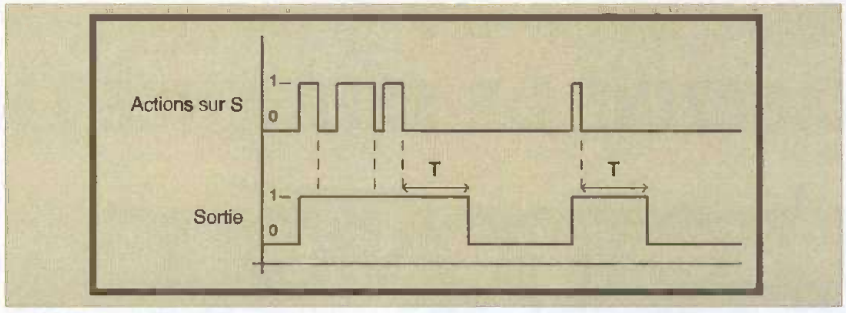
```
T(en S) = 0,7 . R2(en MΩ) . C1(en µF)
 Exemple avec R2=220kΩ et C1=680nF :
T(en S) = 0,7 . 0,22 . 0,68 = 0,1 S
```

Avec les valeurs de cet exemple, un appui sur la touche S1 génèrera toujours une impulsion d'environ un dixième de seconde, quelque soit le temps d'action sur S1. Ce circuit anti-rebonds se prête particulièrement bien à l'élimination des multiples pics indésirables lors de l'appui sur une touche. Le diagramme de la **figure 8** concrétise cette étude.

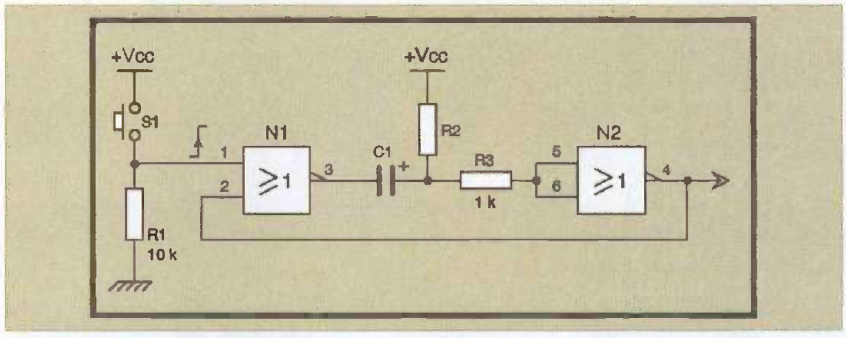
Le monostable simple effet à logique négative

Pour obtenir la logique inverse, il suffit d'employer des portes NON-ET, et de se conformer au schéma de la **figure 9**. Ce monostable se comporte de manière identique au précédent, mais en logique inverse. Au repos, la sortie présente un état positif permanent. Une impulsion de niveau 0 en entrée génère, en sortie, un créneau de niveau 0 et de durée invariable. La formule de calcul reste identique.

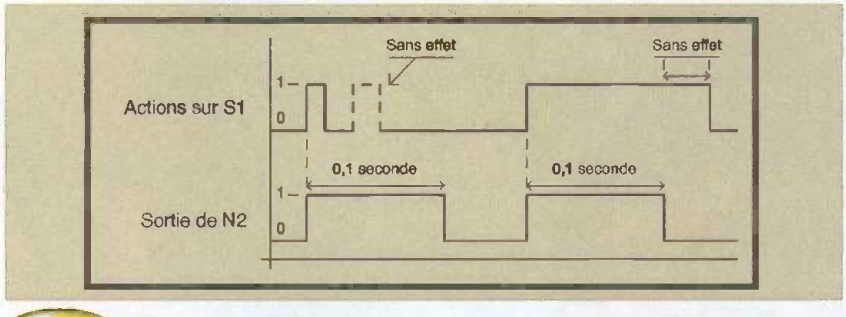
Y. MERGY



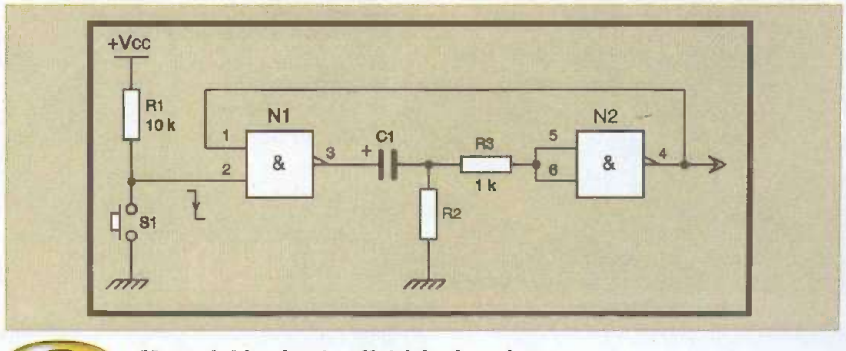
6 *Diagramme de fonctionnement du monostable à effets multiples*



7 *Monostable positif à simple effet*



8 *Diagramme de fonctionnement du monostable positif à simple effet*



9 *Monostable simple effet à logique inverse*

Bibliographie :

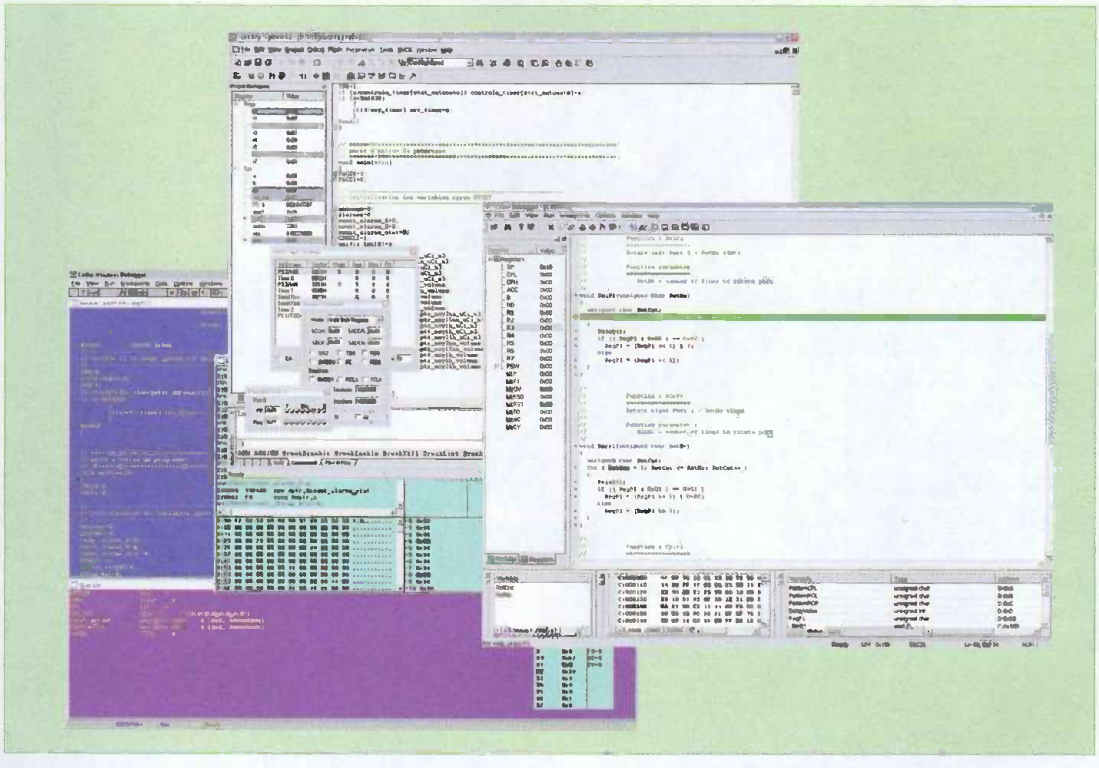
« Pour s'initier à l'électronique logique et numérique » de Y. MERGY chez Dunod ETSF.

« Applications C MOS » de P. WALLERICH chez Dunod ETSF.

« Circuits numériques (théorie et applications) » de Ronald J. TOCCI chez Dunod.

« 27 Modules d'électronique associatifs » de Y. MERGY chez Dunod ETSF (épuisé).

Découvrir le rôle des outils de développement pour microcontrôleurs.



Il suffit de feuilleter les pages de cette revue pour constater à quel point les microcontrôleurs sont présents dans l'électronique moderne. Si la mise en œuvre matérielle d'un microcontrôleur est relativement simple de nos jours, la conception et la mise au point d'un logiciel pour microcontrôleur fait intervenir de nombreux outils et un cycle de développement bien précis que nous vous proposons de découvrir.

Face à la multitude de produits disponibles sur le marché, l'amateur qui souhaite débuter dans la mise en œuvre d'un microcontrôleur a de quoi être rebuté. Tout d'abord, avant même d'avoir pu se faire une opinion sur les performances de tel ou tel composant, il va devoir choisir une famille de microcontrôleurs qui convienne aux futures applications qu'il souhaite développer. En effet, même si dans un premier temps le choix du microcontrôleur ne semble pas important (puisque l'utilisateur souhaite simplement se familiariser avec les outils de développement), il vaut mieux choisir tout de suite un microcontrôleur adapté aux futurs besoins pour ne pas avoir à repartir de zéro (ou presque).

Certes, il existe sur le marché de nombreux kits de développement qui sont assez adaptés à une première prise en main. Mais ils sont parfois d'une utilisation très limitée ce qui oblige ensuite à investir dans un nouveau matériel beaucoup plus coûteux pour pouvoir réaliser une application complète.

C'est le cas, par exemple, avec certains kits organisés autour des microcontrôleurs qui embarquent un interpréteur Basic. La prise en main peu sembler rapide et abordable, car le coût des kits est attractif. Mais ensuite, le coût des microcontrôleurs qu'il faut acheter devient vite prohibitif. De plus, il faut acquérir un appareil (parfois coûteux) pour mettre en place le programme dans la mémoire morte (EPROM ou FLASH) du microcontrôleur. Ajoutons à cela que les performances de ce type de microcontrôleur est généralement médiocre et vous comprendrez qu'il est très facile de se tromper dans le choix d'un kit de développement si le seul critère examiné est le prix.

Pour vous permettre de vous y retrouver dans la multitude des offres (commerciales ou gratuites), nous vous proposons de décrire le cycle de développement habituel et d'introduire les outils nécessaires (logiciels ou matériels) au fur et à mesure.

Pour commencer, imaginons que vous avez défini la spécification du logiciel que vous souhaitez concevoir. En toute logique, vous pouvez donc commen-

cer l'écriture du programme. Il vous faut cependant décider du langage avec lequel vous allez travailler. Le choix peut être assez vaste, mais pour les microcontrôleurs susceptibles d'intéresser le grand public, on peut estimer que le choix se limite aux langages suivants : Le langage BASIC, le langage C (quelques fois C++) et l'assembleur.

Le langage BASIC reste très prisé par de nombreux amateurs parce qu'il est largement enseigné dans le milieu scolaire. Il est vrai qu'il s'agit d'un langage relativement simple à apprendre et qui a su évoluer pour s'adapter au monde des microcontrôleurs. L'inconvénient de ce langage reste principalement l'utilisation des instructions « GOTO » qui déstructurent totalement le code source et conduisent bien souvent à la création de véritables labyrinthes. Ajoutons à cela que le langage BASIC offre peu de souplesse dans la gestion des variables internes du programme, sans parler du problème de la visibilité globale des variables (n'importe quelle ligne du programme peut modifier accidentellement une variable par erreur, ce qui complique énormément la tâche du concepteur pour identifier la source du problème).

Il existe deux familles d'outils radicalement différentes pour travailler en langage BASIC. Les compilateurs pour le langage BASIC transforment le code source (les lignes de programme que vous écrivez) en code machine (le programme final à transférer dans la mémoire du microcontrôleur). Le code machine obtenu peut être très performant et compact mais chaque modification du programme nécessite une nouvelle compilation (l'analyse syntaxique est réalisée une seule fois, par le compilateur). De plus, cette méthode de travail ne permet pas de bénéficier des fonctions de mise au point inhérentes au langage BASIC (interrogation des variables en mode console, etc.). En contre partie, le code machine généré contient l'ensemble du programme nécessaire au microcontrôleur de sorte qu'au final il suffit d'acheter des modèles vierges moins coûteux.

A l'inverse, lorsque le langage BASIC est interprété, les lignes de programmes sont analysées à la volée par un logiciel embarqué dans le microcontrôleur (l'interpréteur). Ce mode de fonctionnement offre plus de souplesse pour la mise au point mais il nécessite une quantité de mémoire beaucoup plus importante (c'est le code source qui est transféré dans la mémoire du microcontrôleur) et le temps d'exécution est extrêmement lent (la syntaxe de la ligne de programme en cours d'exécution est analysée à chaque passe). Certains environnements en BASIC compilé proposent de charger un interpréteur en plus du programme en code source de l'utilisateur pour permettre de bénéficier des facilités de mise au point en mode interprété. Puis, lorsque la mise au point de l'application est terminée, il suffit de compiler le programme pour obtenir un code cible performant épuré de tout interpréteur. Ce mode de fonctionnement pose souvent des problèmes de mise au point en raison de la différence flagrante du temps d'exécution du programme final entre le mode compilé et le mode interprété.

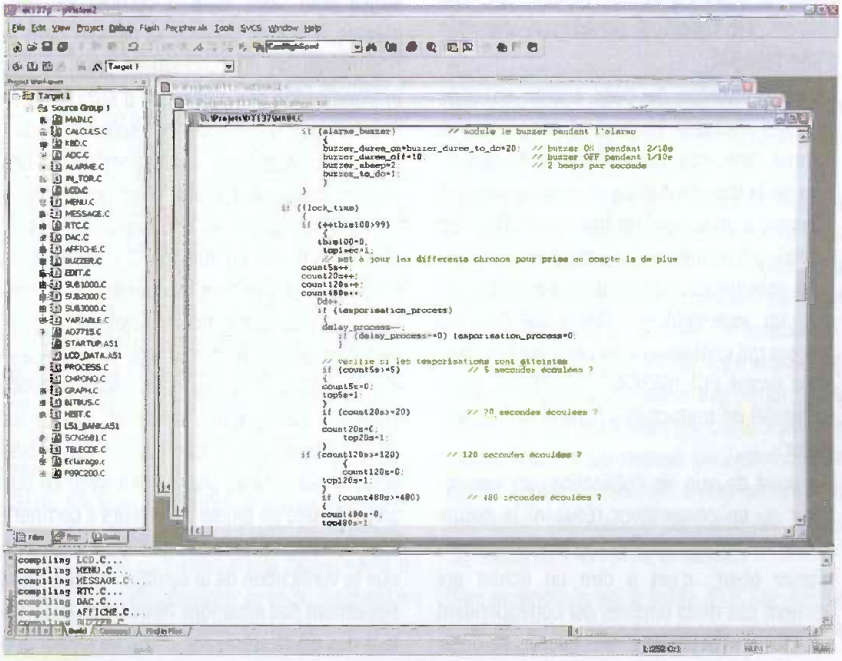
Le langage assembleur permet d'obtenir les logiciels les plus performants qui soient (très rapides et très compacts) mais il est aussi très contraignant car totalement dépendant du microcontrôleur choisi. Pour pouvoir programmer en assembleur, il faut donc commencer par étudier le jeu d'instructions du microcontrôleur. Ensuite, il faut décomposer le fonctionnement du programme jusqu'au plus bas niveau, pour écrire des instructions compréhensibles directement par le microcontrôleur. Le jeu d'instructions étant généralement très limité, la réalisation de certains calculs devient vite un casse tête si l'on ne possède pas une bibliothèque de fonctions adaptées. Par exemple, les opérations élémentaires sur des nombres réels deviennent vite très compliquées en langage assembleur (multiplication, division, etc.). Alors ne parlons pas des racines carrées ou des fonctions trigonométriques.

En langage assembleur, il incombe au concepteur de définir la position des variables et structures de données de son programme dans la mémoire du microcontrôleur. Le langage assembleur souffre également du manque de structure intrinsèque aux instructions de sauts inconditionnels. Il existe des aménagements dans le langage assembleur pour permettre de regrouper une série d'instructions sous formes de MACRO et créer des agglomérats de variables pour former des structures. On parle alors de langage Macro-Assembleur. Cependant, on est encore loin des facilités offertes par les langages structurés.

L'assembleur étant généralement gratuit, il est tentant de vouloir démarrer son apprentissage avec ce langage. S'il est vrai que l'utilisation de l'assembleur permet de beaucoup apprendre lorsque l'on est correctement encadré, un tel choix peut s'avérer très décourageant et totalement démotivant pour l'amateur isolé car les bibliothèques de fonctions standards sont payantes et parfois fort chères.

Dans l'industrie, il est assez rare de concevoir une application en langage assembleur dans sa totalité, étant donné la difficulté et le temps de développement que cela représente. Il est bien plus efficace de n'utiliser le langage assembleur que pour quelques portions du logiciel pour lesquelles les performances doivent être maximales. On préfère alors uti-

1 ***Exemple d'interface IDE***



liser un langage de haut niveau pour écrire le corps principal du programme et ne conserver que quelques fonctions critiques écrites en langage assembleur.

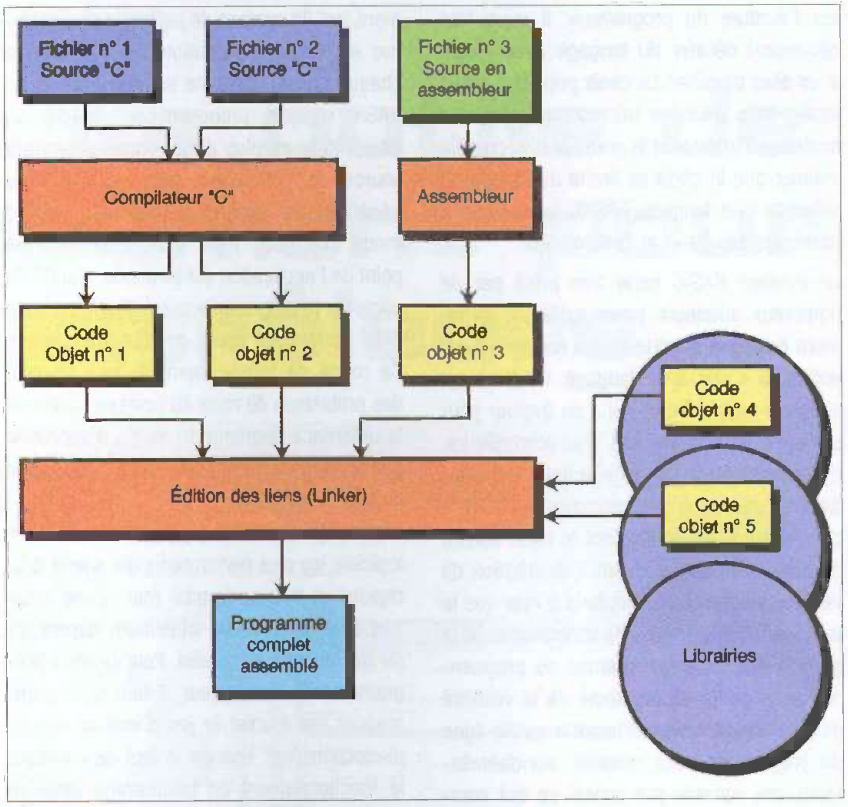
Comme vous l'avez sûrement deviné, le langage 'C' s'est largement répandu car il est très bien adapté au développement de logiciels sur microcontrôleurs. Le langage 'C' offre de nombreux avantages sans pour autant pénaliser les performances du logiciel final. Le langage 'C' étant fortement structuré, il impose au concepteur une certaine rigueur bénéfique. De plus, les librairies standards qui accompagnent tout compilateur 'C' offrent des services qui répondent à la majorité des besoins des petites applications.

Une fois le langage de programmation déterminé, il ne reste plus qu'à écrire le code source du logiciel à l'aide de n'importe quel éditeur de texte. Même le programme de bloc notes de Windows peut faire l'affaire ! Cependant il faut noter que les outils de développements pour microcontrôleurs sont de plus en plus livrés avec un éditeur de texte adapté au langage retenu (coloration des mots clés du langage par exemple).

Notez qu'il faut faire la distinction entre un éditeur de texte adapté au langage et un environnement de développement intégré (IDE). Les environnements de type IDE regroupent au sein d'une seule interface graphique tous les logiciels nécessaires à la création du programme final. Les interfaces IDE permettent généralement de regrouper un ensemble de fichiers sources au sein d'un même projet et de définir les options de compilateurs à appliquer en fonction de vos besoins. La **figure 1** dévoile un exemple d'interface IDE.

La transformation du code source en code machine nécessite toujours plusieurs étapes comme cela vous est dévoilé en **figure 2**. Lors de la transformation, le code source est transmis à un logiciel de traduction. Pour les fichiers qui contiennent du code écrit en langage assembleur, le logiciel en question s'appelle un «assembleur » tandis que pour les fichiers qui contiennent du code écrit en langage évolué ('C', 'PASCAL', 'FORTRAN', etc.), le logiciel de traduction s'appelle un 'compilateur'.

Du point de vue de l'utilisation, un assembleur ou un compilateur réalisent la même chose. Ils transforment un 'fichier source' en 'fichier objet', c'est à dire un fichier qui contient des mots binaires qui correspondent au jeu d'instructions du microcontrôleur



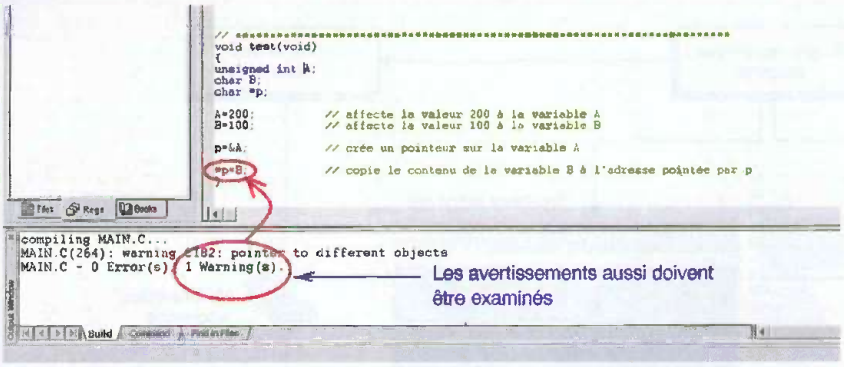
2 *Transformation du code source en code machine*

'ciblé'. Les fichiers objets se reconnaissent généralement grâce à l'extension '.obj' ou '.o' qui leur est attribuée par le compilateur (ou assembleur).

Pendant la traduction du fichier source, le compilateur et l'assembleur procèdent à une vérification des règles de syntaxe correspondant au langage choisi. Si des erreurs sont trouvées, le traducteur affichera des messages pour vous aider à corriger les problèmes. Il faut en général se plonger dans la documentation du compilateur pour trouver la signification des messages d'erreurs, mais avec un peu d'habitude, on reconnaît facilement les messages correspondant à des erreurs fréquentes (oubli d'un point-virgule en fin d'instruction ou accolades mal appairées dans le cas du langage 'C'). Lorsque les erreurs détectées sont bloquantes, le compilateur ne crée pas le fichier objet, ce qui permet d'empêcher le démarrage du processus d'édition des liens que nous allons détailler dans les paragraphes suivants, sans pour autant interrompre la compilation des autres fichiers associés au projet (pour avoir en une passe la liste de toutes les erreurs à corriger). Les compilateurs modernes vont plus loin que la vérification de la syntaxe. Ils analysent également des situations fréquentes et émettent des avertissements lorsqu'ils jugent que les instructions demandées peuvent être ambiguës.

L'exemple de la **figure 3** est intéressant à étudier. Ici, le compilateur 'C' nous informe qu'il y a un risque pour que les échanges de variables risquent d'être mal interprétées selon leur contenu, car les variables n'ont pas le même format (c'est un peu comme si on essayait de garer sa voiture dans un local à vélo). Dans notre exemple, la variable A occupe 16 bits non signés. Elle peut donc prendre des valeurs comprises entre 0 et 65535. En revanche la variable B occupe 8 bits non signés. Elle peut donc prendre des valeurs comprises entre –127 et +127. Il semble donc tout à fait possible de pouvoir transférer le contenu de la variable B vers la variable A. Si on ignore l'avertissement, notre programme d'exemple fonctionnera parfaitement. Cependant si un jour la variable B prend une valeur négative, le résultat obtenu ne sera plus valide (faites l'essai en écrivant les nombres en binaire et vous verrez bien). Il est donc indispensable d'analyser pourquoi le compilateur émet des avertissements, même si le programme semble bien fonctionner jusque là.

On pourrait penser que la traduction des 'fichiers sources' en 'fichiers objets' suffit pour obtenir le programme final. Il n'en est



3 ***Le compilateur " C " informe qu'il y a un risque***

rien. Lors de la traduction d'un fichier source en code machine, le compilateur n'est pas en mesure de faire le lien avec des fonctions ou des variables qui se trouvent localisées dans un autre fichier source. On pourrait être tenté de regrouper toutes les lignes du programme dans un fichier unique pour surmonter cette difficulté mais cela ne changerait rien car le problème resterait entier vis à vis des fonctions qui pourraient être localisées dans une bibliothèque standard.

Pour permettre une conception modulaire des programmes, le compilateur réserve la place nécessaire pour coder les instructions (en code machine) qui permettront au logiciel final d'accéder aux fonctions ou variables des autres modules. Au moment de la traduction du fichier source, le compilateur laisse donc des 'blancs' qu'il faudra remplir plus tard par un second logiciel chargé de faire le lien entre les différents modules qui forment le programme complet. Pour faciliter la tâche du programme chargé de l'édition des liens, le compilateur ajoute dans le fichier objet une table qui indique où sont positionnés les espaces vides et le nom des variables ou des fonctions qu'ils sont censés référencer. Le compilateur ajoute également une seconde table qui indique où sont situés tous les éléments connus dans le fichier objet. Cela permettra de faciliter la tâche du programme chargé de l'édition des liens et de permettre aux outils de mise au point de pouvoir remonter jusqu'au code source.

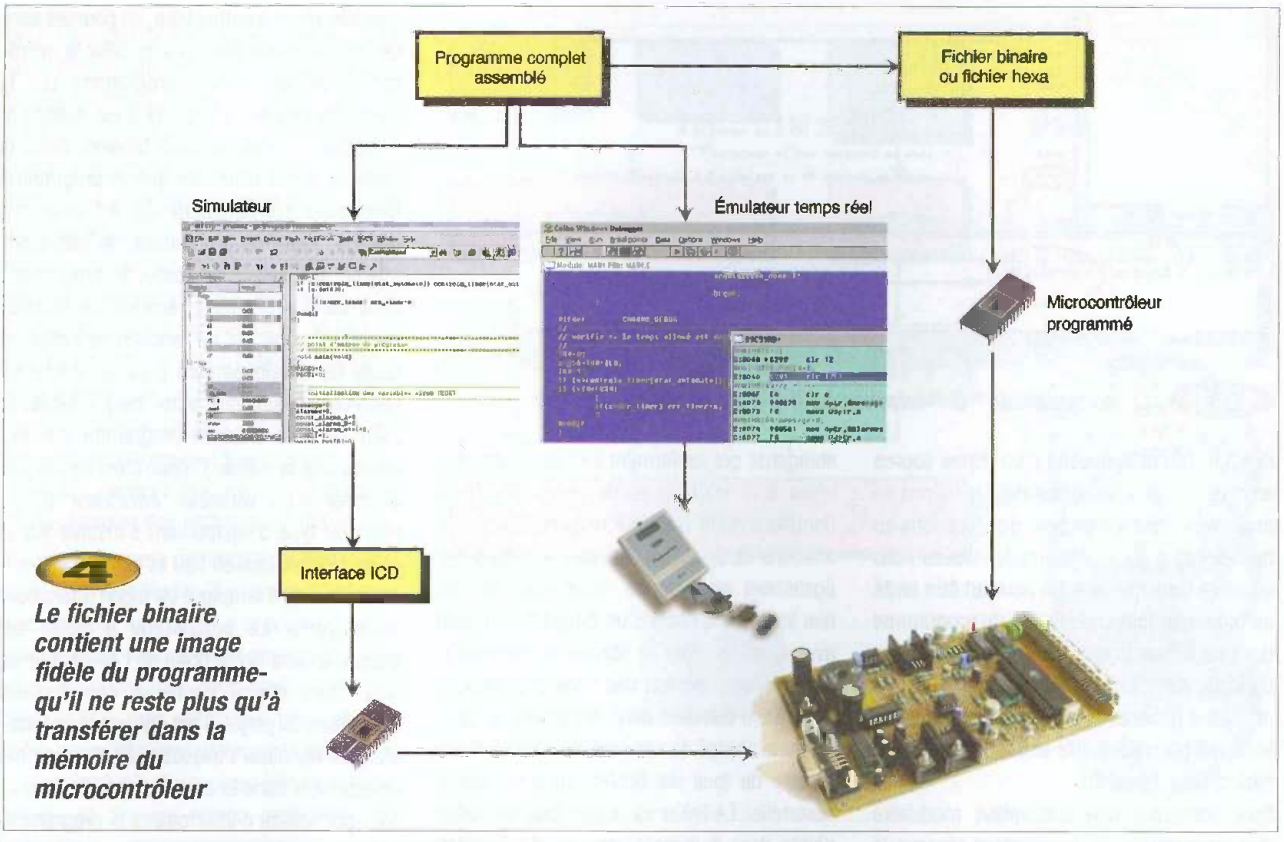
Dans l'état, un fichier objet est déjà parfaitement représentatif d'un code source et des fonctions qu'il renferme. Selon la nature des fonctions qui sont regroupées au sein du fichier objet, il est intéressant de former des collections de sous programmes prêts à l'emploi, en vue de pouvoir les réutiliser ou les échanger facilement. C'est justement le rôle des librairies. Les compilateurs 'C' sont tous livrés avec un ensemble de librairies standards qui renferment des fonctions très utiles à la majeure partie des applications (fonctions d'affichage sur un port série ou un afficheur LDC, lecture d'un clavier, etc.). Il est également possible de créer vous-mêmes des librairies à l'aide d'un programme fourni avec le compilateur (le libraire ou 'librarian'). Pour réaliser l'édition des liens d'un logiciel complet, il convient donc d'indiquer au programme chargé de cette opération (le 'linker') la liste de tous les fichiers qu'il va devoir assembler. Le linker va ranger tous les codes objets dont il dispose les uns derrière les autres (ou dans un ordre imposé par les options de compilation ou des directives spéciales ajoutées dans le code source) pour former le programme final. Il va ensuite examiner les tables pour combler les vides laissés par le compilateur. Si jamais il reste des vides qui n'ont pas pu être comblés, le linker va rechercher dans les librairies associées au projet (options dans l'environnement IDE ou options de la ligne de commande) le code objet qui lui manque pour l'ajouter automatiquement à votre programme.

Au final, si tous les espaces laissés vides par le compilateur ont été comblés par le linker, il n'y a plus qu'à enregistrer l'image binaire du programme ainsi assemblé. Dans le cas contraire, le linker émettra des messages d'erreurs qui identifient les liens qui n'ont pu être résolus. Le fichier binaire, qui porte souvent l'extension '.bin', contient une image fidèle du programme qu'il ne reste plus qu'à transférer dans la mémoire du microcontrôleur, comme l'indique la **figure 4**. Très souvent, le fichier binaire est également converti en un fichier hexadécimal portant l'extension '.hex' (format Intel-HEX ou Motorola S19). Ce type de fichier est plus approprié lorsque le transfert vers l'outil de programmation se fait au travers d'une liaison série. Il permet aussi de consulter le code machine à l'aide d'un simple éditeur de texte.

Une fois arrivé à cette étape, on pourrait penser qu'il ne reste plus qu'à installer le microcontrôleur fraîchement programmé sur la carte électronique à laquelle il est destiné et à mettre le montage sous tension. Dans la pratique, il est bien rare que le programme fonctionne correctement du 1er coup. En général, avant de programmer le microcontrôleur, il convient de tester le programme dans un environnement adapté. En fonction de votre budget et en fonction de l'offre en outils de développement pour la famille de microcontrôleurs que vous avez choisie, la méthode pour tester le programme ne sera pas du tout la même. L'idéal, c'est de pouvoir disposer d'un véritable 'émulateur temps réel'. Ce type d'équipement s'installe sur la carte électronique en lieu et place du microcontrôleur qu'il remplace de façon totalement transparente. Le programme à tester est transféré dans la mémoire de l'émulateur et, moyennant d'avoir configuré correctement les options du projet, il est généralement possible de visualiser l'exécution du programme directement dans le code source. Ces appareils permettent d'interrompre le programme à tout moment et de placer des points d'arrêts dans le programme pour contrôler le déroulement des instructions. Un émulateur permet également d'examiner le contenu des variables manipulées par le programme et de consulter le contenu de la mémoire et des registres du microcontrôleur. Certains modèles permettent même de consulter toutes ces données sans avoir besoin d'interrompre le déroulement du programme. Ces appareils sont malheureusement très coûteux (plusieurs milliers d'euros) et restent cantonnés au milieu professionnel.

Fort heureusement, pour les microcontrôleurs récents, sont apparus des outils de développements plus adaptés à la bourse des étudiants et des amateurs (certains sont gratuits). Parmi les outils généralement fournis à moindre coût avec le compilateur, nous pouvons citer les simulateurs. Ces outils sont entièrement logiciels. Ils permettent de tester vos programmes en simulant le fonctionnement du microcontrôleur sur votre ordinateur. Ces outils peuvent s'avérer très performants car ils ne sont pas limités par des possibilités matérielles. Ils permettent généralement de placer autant de points d'arrêts que peut le souhaiter l'utilisateur et de manipuler toutes les zones de mémoires simulées.

Bien entendu, le point faible d'un simulateur c'est qu'il ne permet pas de prendre en



4 ***Le fichier binaire contient une image fidèle du programme-qu'il ne reste plus qu'à transférer dans la mémoire du microcontrôleur***

compte les interactions entre le logiciel et le matériel. Un simulateur est cependant très précieux lorsqu'il s'agit de mettre au point des routines de calculs qui ne traitent que des informations déjà stockées dans la mémoire du microcontrôleur.

Selon le simulateur, il est parfois possible de simuler manuellement (ou à l'aide de scripts) l'activité des ports d'entrées/sorties du microcontrôleur. Par exemple, le simulateur associé au compilateur de la marque Keil permet de simuler complètement le fonctionnement de l'uart à l'aide d'un terminal intégré à l'outil de simulation. Dès lors, il devient très facile de tester l'interaction entre les données échangées avec le monde extérieur et le programme sous test.

Malgré tout, lorsque les interactions entre le matériel et le logiciel sont complexes, le simulateur arrive vite à ses limites. Il faut alors faire appel à un peu de matériel pour permettre de simuler également les interactions physiques avec le monde réel. Dans ce domaine, il existe aujourd'hui des solutions très intéressantes et à faible coût. Ces solutions sont possibles parce que les fabricants de microcontrôleurs, conscients du frein à la diffusion de leurs produits que posent les questions de coût des émulateur temps réel, ont intégré des fonctions de mise au point à l'intérieur des microcontrôleurs. C'est le cas notamment des microcontrôleurs Microchip au travers des interfaces ICD ou des microcontrôleurs Motorola au travers des liaisons BDM. Les interfaces ICD et les interfaces BDM ont accès au cœur du microcontrôleur final qui est installé sur la carte d'étude (soit au moyen d'une liaison RS232 ou USB soit au moyen d'une liaison spécifique qui monopolise des broches du microcontrôleur).

Les solutions techniques mises en œuvre sont fondamentalement différentes, mais au final, tous ces petits boîtiers permettent à l'utilisateur de réaliser les mêmes travaux : Simuler le fonctionnement du logiciel et programmer la mémoire du microcontrôleur. Certes, ces solutions sont moins performantes qu'un véritable émulateur temps réel puisque leur utilisation monopolise une petite partie des ressources du microcontrôleur et que le nombre de points d'arrêts est pratiquement limité à 1.

Mais le coût de ces solutions est 100 fois (voir 1000 fois) moins cher que l'achat d'un véritable émulateur temps réel. Votre bourse appréciera !

Pour les microcontrôleurs qui sont dépourvus de fonctions internes de mise au point, il existe des solutions purement logicielles. Ces solutions font appel à des fonctions de mise au point regroupées dans un programme appelé 'Moniteur'. Pour utiliser ces solutions, il vous faut réserver quelques ports du microcontrôleur (généralement le port de l'uart, si le microcontrôleur en possède une) pour créer une liaison de communication avec votre ordinateur. Le code objet du moniteur doit être lié avec votre programme (par le linker), ce qui augmente la taille du programme final. De plus, il vous faut vous réserver certaines ressources pour le fonctionnement du moniteur (quelques octets sur la pile, un peu de mémoire RAM, routage de certains vecteurs d'interruptions et souvent l'utilisation exclusive d'un timer).

Ces solutions s'avèrent souvent performantes, mais malheureusement, il faut déjà avoir une bonne expérience du cycle de développement pour pouvoir les mettre en œuvre. L'utilisation d'un moniteur est donc à déconseiller en début d'apprentissage, à moins de pouvoir être aidé par des personnes ayant un peu plus d'expérience.

Nous espérons que ce petit tour d'horizon des outils de développement associés aux microcontrôleurs vous aura permis de mieux comprendre le rôle des différents outils actuellement disponibles sur le marché (assembleur, compilateur, simulateur, émulateur, programmateur, etc.) et que cela vous permettra de choisir sereinement la solution la mieux adaptée à vos besoins.

P. MORIN

Collectionner et restaurer les GSM

SAGEM RC712 : un mode " trace " très complet, mais protégé par un code réputé " secret " (en fait, les 8 derniers chiffres de l'IMEI !)

Au rythme d'un renouvellement tous les deux ans en moyenne, les téléphones portables font désormais partie des déchets à recycler, au même titre que les piles usagées ou les voitures accidentées!

Cela peut donner l'idée d'en faire collection à bon compte, quitte à restaurer au passage ceux qui ne seraient plus tout à fait en état de marche. Cela, sans oublier les passionnantes expériences auxquelles un GSM "anonyme" permet de se livrer...

Une abondante matière première

Compte tenu des prix purement symboliques (entre un franc et un euro) auxquels les opérateurs ont pris l'habitude de fournir les téléphones portables à leurs clients "captifs", il n'est guère surprenant que l'on puisse en trouver d'occasion pour quelques dizaines de centimes ! C'est le cas depuis quelques années sur toutes les bonnes "foires à tout", sans grand risque d'acquérir des appareils volés si l'on se limite à des modèles franchement dépassés, voués à finir leur carrière dans le coffre à joujoux des enfants en bas âge. Même s'ils sont toujours "simlockés" sur le réseau de l'opérateur qui les a fournis, ils accueilleront "les yeux fermés" une carte SIM réalisée à partir d'une BasicCard "Professional" (voir EP N° 288), et pourront même servir, en cas d'urgence, à appeler le 112 (surtout s'ils acceptent des piles en lieu et place de leur batterie, certainement bien fatiguée). Mais on se risquera peut-être à tenter de les déverrouiller, ou de "faire parler" la carte SIM de l'ancien propriétaire qui, en cas de changement d'opérateur, est bien souvent abandonnée dans l'ancien appareil, car réputée inutilisable.

Parallèlement aux téléphones, on recherchera bien volontiers leurs accessoires (chargeurs, cordons "allume-cigare", kits piéton, antennes de voiture, etc.) que l'on pourra éventuellement cannibaliser pour se procurer les connecteurs nécessaires à certaines expériences.

Au delà du très réel plaisir de rassembler des GSM de marques et modèles divers et variés, on ne résistera sans doute pas longtemps à la tentation de les faire fonctionner, afin de découvrir les particularités et singularités de chacun (avec parfois une radio FM ou des jeux "en prime"). Après tout, ne fait-on pas rouler les voitures de collection, parfois au son d'un auto-radio à lampes ?

Encore faut-il prendre soin de ne pas perturber les réseaux des opérateurs, surtout si l'on a procédé à des remises en état hasardeuses, et s'abstenir de mettre inutilement sa carte SIM personnelle en danger. Cerise sur le gâteau, les GSM datant des "temps héroïques" sont souvent équipés d'excellents étages radio, si ce n'est d'antennes télescopiques dont le gain se compare plus que favorablement à celui des actuelles solutions "intégrées".

À l'époque, en effet, la couverture des réseaux étant bien moins dense qu'actuellement, chaque décibel en plus dans le "bilan de liaison" était bon à prendre. Ajoutons enfin qu'ils

PHILIPS "Fizz" : l'idéal pour envoyer des SMS très spéciaux (voir notre ouvrage "Plus loin avec les cartes à puce")

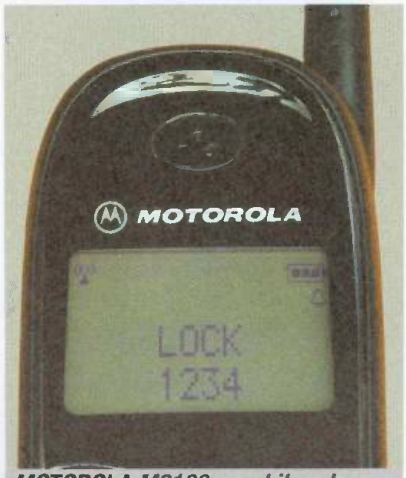
sont infiniment moins enclins à la "censure" de certaines manipulations, que les modèles plus récents.

Le "tour du propriétaire"

Par quel bout faut-il prendre un téléphone portable que l'on vient d'acheter à vil prix, ou même de sauver de la poubelle, et qui reste très probablement insensible à tout appui sur sa touche "marche" ? Assurément par sa batterie : quelle qu'en soit la technologie (NiCd, NiMH, ou même Li-ion), elle est forcément "à plat" si le téléphone est resté sans servir pendant plusieurs mois ou davantage. N'oublions pas que la consommation d'un GSM à l'arrêt n'est jamais complètement nulle, raison pour laquelle il est souhaitable de retirer sa batterie (ou d'en isoler les contacts avec un morceau de sac plastique) avant tout stockage prolongé. L'ennui, c'est qu'une batterie qu'on laisse se décharger aussi profondément risque de se détériorer (l'élément le plus faible se trouvant chargé "à l'envers" par les autres). Si le chargeur d'origine du téléphone (à supposer qu'on en dispose) n'arrive pas à ressusciter la batterie en quelques dizaines d'heures, alors il faudra recourir au "chargeur de choc" que nous avons décrit dans EP N° 285. Après avoir "réveillé" ainsi les éléments les plus abîmés, en désassemblant peut-être même partiellement l'accu, quelques jours de charge à 30 mA devraient rééquilibrer suffisamment l'ensemble pour le rendre de nouveau utilisable. Une autre approche consiste à supprimer les éléments 1,2 V rechargeables, et à les remplacer par des piles 1,5 V, mais alors, il ne faudra plus se servir d'aucun chargeur !

De la discrétion S.V.P.

La question de l'alimentation étant supposée réglée, on retirera toute carte SIM pouvant être restée dans le téléphone. A défaut, celle-ci déclinerait sagement son identité à son réseau d'origine dès la mise sous tension du mobile, ce qu'il vaut mieux éviter tant que l'on n'en sait pas un peu plus long sur son compte... C'est donc avec un logiciel d'exploration de cartes SIM (voir notre ouvrage "Téléphones portables et PC" aux éditions ETSF) qu'on pourra lui faire subir, si on le souhaite, des investigations finalement assez proches de celles effectuées dans les laboratoires de police scientifique. Si son code PIN est activé, on dispose en principe de trois essais avant que la carte ne se bloque, mais le logiciel nous dira exactement ce qu'il en est. Bien souvent, on aura la chance d'être tombé sur une carte dont le code "par défaut" n'a pas été changé. Il est alors payant d'essayer 0000 et 1234, les valeurs les plus courantes, en conservant la possibilité d'un ultime essai si cela ne donne pas le résultat escompté. Admettons que nous ayons trouvé le bon code, ou que l'ancien propriétaire nous l'ait fourni : plutôt que de le noter, puis de l'entrer lors de chaque mise sous tension, il sera plus commode de le désactiver une fois pour toutes, toujours à l'aide d'un lecteur de cartes à puce connecté à un PC. Reste alors à décider, en conscience, de la conduite à tenir : ou bien effacer, sans les lire, les données personnelles que sont les répertoires de numéros et les mini-messages (SMS), ou alors les examiner en quête d'une information potentiellement utile (par exemple pour tenter de vérifier que le vendeur était bien le légitime propriétaire). Notons d'ailleurs que certains modèles de téléphones disposent d'un annuaire "interne", venant en complément de celui qui réside dans la carte, et qu'il faudra également penser à vider ou à inspecter. La question se pose maintenant de savoir si l'on peut, sans risque, insérer sa propre carte SIM dans un tel téléphone de récupération. À moins que l'on envisage vraiment de l'utiliser pour de bon, il est vivement conseillé de s'abstenir.



MOTOROLA M3188 : un bibande compatible " SIM Toolkit ", au mode " test " facilement accessible avec une BasicCard

ALCATEL "One Touch Club" : un puissant mode "trace" à peine caché, et une excellente partie radio

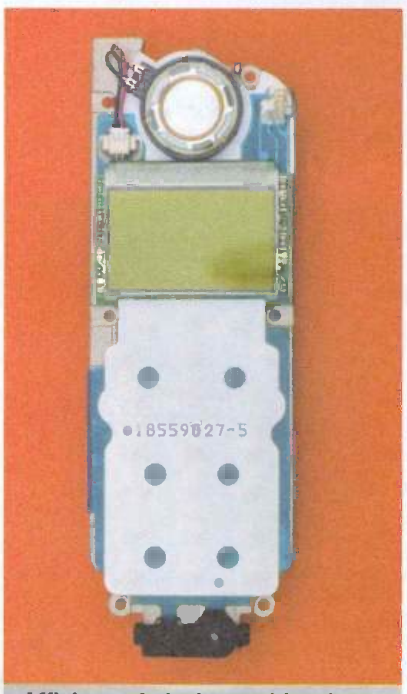
Il faut savoir, en effet, que les opérateurs relèvent couramment, à distance, le numéro d'identification (IMEI) des téléphones qui s'inscrivent sur leurs réseaux : il est là pour cela, en vue notamment de dépister les mobiles volés ou douteux. C'est même sans doute systématiquement fait en cas d'appel vers le 112, un numéro digne du plus grand respect, à n'utiliser sous aucun prétexte pour de simples essais gratuits ! Or, rien n'est plus facile que d'établir une corrélation entre cet IMEI et l'IMSI (le numéro matricule) de la carte SIM, puis de remonter à l'identité de son titulaire : fini l'anonymat ! Techniquement parlant, cela permet d'optimiser (pas toujours à bon escient...) certains services en fonction du modèle de téléphone utilisé par le client. Il ne faudra, par exemple, pas s'étonner si un service "USSD" (le #123# ou le #111# d'Orange, par exemple) ne fonctionne plus après quelques jours d'utilisation de la carte SIM dans un mobile de type ancien ! Mais alors, comment explorer toutes les fonction-

PANASONIC GD90 : très pratique pour "bricoler" des fonctions "SIM Toolkit" dans une BasicCard !

nalités de nos GSM de collection, sans buter sur d'irritants messages du genre "SIM absent" ? Tout simplement en programmant astucieusement une BasicCard ou une Gold Card, ou encore en se servant d'une carte SIM de collection, choisie parmi celles d'opérateurs étrangers : il suffit normalement d'ajouter les codes de tous les opérateurs locaux dans sa liste des "réseaux interdits" pour que le téléphone (à supposer qu'il ne soit pas "simlocké") fonctionne sans chercher à s'inscrire, et donc en toute discrétion.

Et la restauration ?

Mis à part les inévitables problèmes de batteries, les téléphones portables de récupération sont parfois en assez piteux état : antenne arrachée, afficheur cassé, boîtier fêlé, clavier hors d'usage, etc. S'agissant d'équipements des télécommunications soumis à un agrément, seuls les services techniques habilités par le constructeur sont normalement autorisés à intervenir. Cela se comprend fort bien si l'on considère, par exemple, que le remplacement d'une antenne par une autre, radioélectriquement mal adaptée, peut transformer l'appareil en un puissant perturbateur. On voit mal, par contre, en quoi le fait de reconstituer un téléphone convenable à partir de deux ou trois épaves du même modèle pourrait poser problème. À condition d'opérer soigneusement (éviter absolument l'apparition de "pièces en trop" !), il est finalement fort simple de remplacer un afficheur ou un clavier, en général sans avoir à faire la moindre soudure, mais en se méfiant des adhésifs double faces, dont certains sont conducteurs. Tout au plus faut-il disposer d'un jeu de tournevis miniatures à empreintes "Torx" (N° 6 à 10), et cruciformes (N° 00 à 1). Dans certains cas, on pourra même être tenté d'échanger une carte électronique complète, sur laquelle il est pratiquement hors de question de réparer quoi que ce soit. Attention, toutefois, au fait que ce genre d'opération risque de modifier l'IMEI du téléphone bénéficiant de la "greffe". Pour en avoir le coeur net, on tapera *#06# (même sans carte SIM !), et on comparera ce qui s'affiche avec ce qui est inscrit sur l'étiquette signalétique, si celle-ci n'a pas disparu dans la bataille (car elle recouvre souvent une vis qu'il faut pourtant bien retirer). Rappelons, car c'est important, que falsifier l'IMEI d'un téléphone portable est généralement considéré comme un délit (mais le cas de figure envisagé est tout de même un peu particulier...).



Afficheur et clavier sont les deux organes les plus vulnérables après l'antenne et la batterie

Ausculter les réseaux

Nombreux sont les GSM "grand public" qui disposent des mêmes outils de diagnostic réseau que ceux équipant les techniciens des opérateurs. Simplement, le menu qui y donne accès est "caché", ou bien protégé par un code plus ou moins "secret" ou par une carte SIM spéciale. Autant dire qu'il suffit d'aller sur Internet pour percer ces secrets de Polichinelle, quand on ne met pas le doigt dessus par hasard (qui pourrait interdire au possesseur d'un GSM Alcatel, par exemple, d'appuyer six fois de suite sur la touche "zéro", puis une seule fois sur l'étoile ?). Schématiquement, le mode "trace" permet de repérer les stations relais d'un opérateur donné, et d'en relever certaines caractéristiques que l'utilisateur n'a, il faut bien l'avouer, nul besoin de connaître pour téléphoner. Et pourtant, l'expérience est immensément instructive, voire édifiante ! Menée avec une carte SIM normale et en cours de validité, elle ne renseignera en général que sur les relais appartenant à l'opérateur qui l'a émise. Par contre, avec une carte SIM réalisée à partir d'une BasicCard, et donc paramétrée pour que le mobile ne puisse s'inscrire sur aucun réseau, on pourra choisir en toute liberté (faire d'abord une recherche manuelle de réseau, et sélectionner celui auquel on s'intéresse). Il faut savoir (et les spécifications GSM, librement téléchargeables sur www.etsi.org, l'expliquent clairement), qu'à chaque relais est affecté un identifiant unique. Celui-ci (le "CGI" pour Cell Global ID) se décompose en quatre champs de données : MCC, MNC, LAC, CI. MCC (code pays) et MNC (code réseau) identifient l'opérateur et son pays : 208-01 pour Orange,

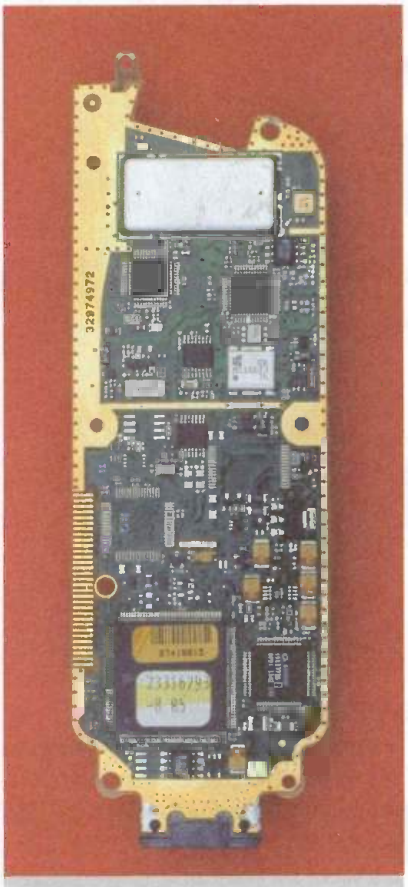
208-10 pour SFR, et 208-20 pour Bouygues, dans le cas particulier de la France. LAC est un code de deux octets indiquant la zone locale (Local Area) dans laquelle se situe le relais. Par "zone", il faut entendre une étendue relativement importante (chaque département français n'en compte qu'un petit nombre) regroupant de multiples relais. CI (Cell Identity), pour sa part, est également un code de deux octets, mais qui identifie individuellement le relais (en toute rigueur, la "cellule" desservie par celui-ci). Contrairement à une idée reçue, ce n'est que lorsqu'il change de zone locale qu'un mobile en "veille" est tenu de se signaler au réseau, et non pas à chaque fois qu'il change de relais. Lorsqu'il est appelé, tous les relais de la zone locale où il est censé se trouver transmettent donc un "paging", c'est-à-dire une invitation à se signaler. Toutefois, le réseau peut fort bien envoyer un "paging" à sa propre initiative, voire périodiquement, et donc "géo-localiser" assez finement le mobile à tout moment, en général à l'insu de son utilisateur. Certaines applications pratiques, pas toujours très respectueuses de la vie privée, sont d'ailleurs basées sur ce mécanisme. Chaque "secteur" d'un relais (entendons par là chaque groupe d'antennes orientées dans une direction donnée) émet en permanence, et au maximum de la puissance autorisée, ses identifiants sur un canal "balise" dit BCCH (Broadcast Channel). C'est ce qui permet aux mobiles de tenir à jour une liste des relais du voisinage (voir fichier 7F20:6F74 ou 7F21:6F74 de la carte SIM), et d'en changer aussi souvent qu'il le faut pour maintenir une bonne qualité de transmission ou une fluidité optimale du trafic. C'est essentiellement par leurs numéros de canaux BCCH que l'on repèrera les relais avec un GSM dont on aura activé le mode "trace". Les canaux de la bande 900 MHz sont ainsi numérotés de 0001 à 0124, et ceux de la bande 1800 MHz de 0512 à 0885 (selon la spécification GSM 05.05). S'y ajoutent les 49 canaux supplémentaires de la bande 900 MHz "étendue" (0975 à 1023), utilisés notamment par Bouygues Télécom pour son déploiement du "bibande". Si donc nous activons le mode "trace" d'un GSM complaisant, nous accèderons à un groupe de données qu'il est fort intéressant de décoder.

Prenons le cas d'un Alcatel BE4 (One Touch 300 à 303), affichant ce qui est reproduit à la figure 1 (choix "network" ou "réseau" du menu "traces").

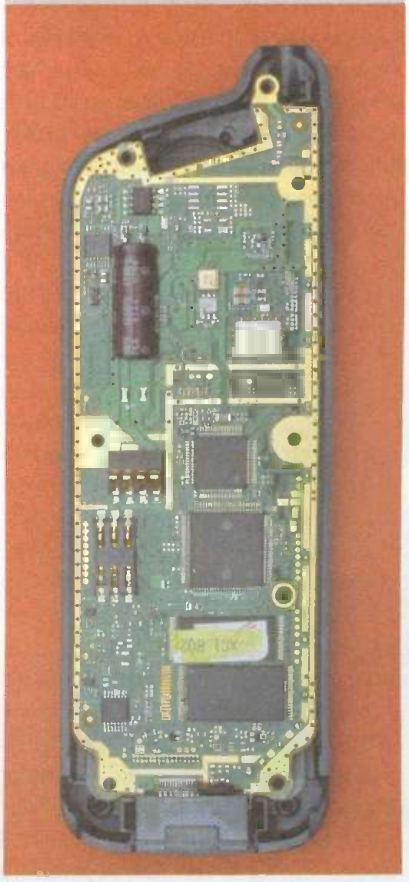
0554 47 04
2 0 18
9102 6FC5

Un exemple de rapport de "trace"

0554 est le numéro de canal BCCH du relais qu'utilise actuellement le mobile, ou qu'il "fait semblant" d'utiliser si la carte SIM est conçue pour ne pas lui donner accès au réseau (aucune "classe d'accès" valide dans son fichier 7F20:6F78 ou 7F21:6F78). 04 est un "code de couleur" (BSIC) permettant au mobile de faire la distinction entre plusieurs relais utilisant le même canal BCCH, et susceptibles d'être reçus en un même lieu. 9102 est le code de la zone locale à laquelle appartient le relais, MCC et MNC étant implicitement supposés connus (suite à un choix



Côté pile et côté face de la carte principale d'un RC712



Pas une seule soudure pour remonter ce GSM aux sous-ensembles aisément interchangeables !

Un relais à trois secteurs, dont il est fort instructif de faire le tour, mode "trace" du GSM activé.

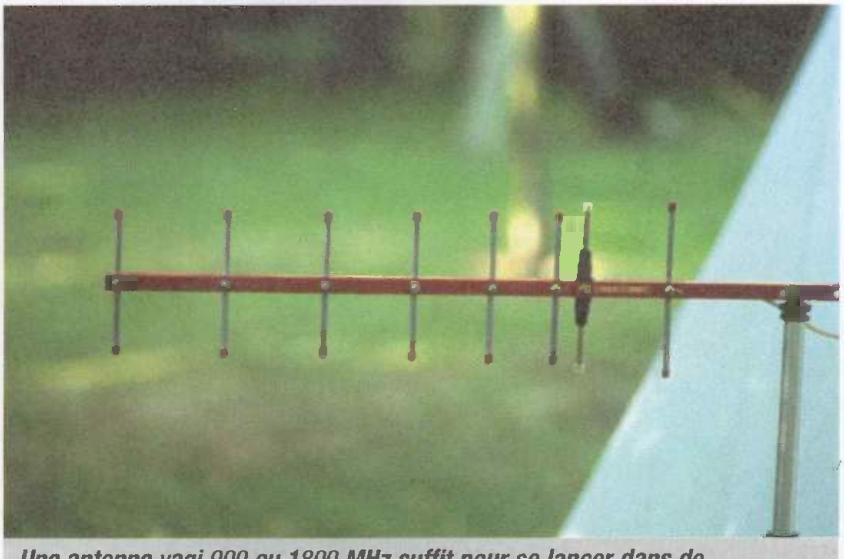
manuel ou automatique du réseau ausculté). S'agissant, en l'occurence, d'un relais exploité par Orange, l'identifiant complet de la cellule s'écrit donc 208 01 9102 6FC5. 47 est une valeur fort intéressante à surveiller, dans la mesure où il s'agit du niveau du signal radio reçu, exprimé en décibels (maximum 64). On suivra avec intérêt ses variations lors de déplacements du mobile, à l'extérieur ou à l'intérieur (par exemple en s'approchant d'une fenêtre). A vrai dire, nous tenons là un instrument, certes d'une précision limitée, mais très suffisant pour évaluer les champs électromagnétiques reçus de tel ou tel relais dont on redoute peut-être (c'est à la mode !) les "rayonnements" supposés nocifs. Or, il n'est pas rare de constater que l'on reçoit 10 à 20 dB plus fort un relais situé à plusieurs centaines de mètres, voire davantage, que celui installé sur le toit même du bâtiment où l'on se trouve, ou dont on est le voisin immédiat. L'explication est simple : l'extrême directivité des antennes, et la grande rigueur avec laquelle les ingénieurs radio des opérateurs définissent leur pointage. Ce genre d'investigation se trouve encore facilité par d'autres sous-menus, qui énumèrent les canaux BCCH des cellules voisines, avec leurs niveaux de réception respectifs (RXLEV) ou même des critères de qualité de liaison (C1 et C2). On pourra ainsi "tourner" autour d'un relais préalablement repéré, et identifier les canaux BCCH de leurs différents "secteurs" d'émission (bien souvent au nombre de trois). Et avec un minimum de pratique, il est même facile de déterminer avec certitude à quel opérateur appartient un relais donné ! Enfin, pour peu que l'on équipe le téléphone d'une antenne directive (de type "yagi", par exemple), il est très divertissant d'organiser des "chasses au renard" que ne renieraient pas nos amis radio-amateurs. Repérer des relais parfois bien camouflés, se situer par goniométrie sur une carte topographique en recoupant, à la boussole, les directions de relais préalablement identifiés, voici des activités de plein air inédites et instructives, pouvant être menées aussi bien en ville qu'à la campagne...

Des énigmes a résoudre...

Collectionner des GSM de différents modèles et s'en servir (en tout bien tout honneur) avec une carte SIM "maison", cela permet de mettre en évidence une foule de choses intéressantes. Opérer une recherche manuelle de réseau avec un téléphone purement 1800 MHz (en l'occurence un des premiers modèles Bouygues) permet de vérifier si une zone donnée est oui ou non couverte en bibande par Orange et/ou SFR. Inversement, un modèle 900 MHz révèlera si le "GSM étendu" de Bouygues est disponible en un lieu donné. Bien que les opérateurs rétorquent volontiers que "cela ne regarde pas le client" (sic.), ce n'est pas sans intérêt, loin de là. À l'évidence, une zone couverte en bibande craindra moins la saturation, aux heures de pointe, que si elle n'était desservie que par des relais 900 MHz. Au surplus, la puissance d'émission étant moindre en 1800 MHz qu'en 900, l'autonomie de batterie peut se trouver sensiblement augmentée. En général, un mobile bibande cherchera à utiliser, de préférence, des relais 1800 MHz s'il en trouve. Cela libère, soit dit en passant, de la capacité 900 MHz pour les possesseurs de simples mobiles mono-bande. Mais il se passe parfois des choses étranges : un mobile 1800 MHz qui devient, du jour au lendemain et dans une "Local Area" bien précise, incapable de détecter les relais d'un certain opérateur, tout en fonctionnant à la perfection avec ceux des deux autres. Et curieusement, dans la même zone, un mobile bibande s'inscrira systématiquement en 900 MHz, bien qu'il repère toujours des relais 1800... Interrogés, les opérateurs n'apprécient généralement pas du tout que leurs clients s'aperçoivent de ce genre de situation (embarrassante ?), tout en les incitant fermement à mettre le mobile rétif au rancart (sans se douter qu'il en vient probablement !). Mais au bout d'un certain temps (quatre mois jour pour jour, dans un exemple vécu), tout rentre spontanément dans l'ordre. Bizarrement, cela coïncide à peu près avec de discrets travaux de déploiement de l'UMTS dans la zone concernée. Et ce n'est là qu'un exemple, parmi tant d'autres, des opportunités que peut offrir une collection de GSM, soigneusement entretenue en état de marche, et régulièrement sollicitée !

Gageons qu'il pourrait bien y avoir là matière à faire éclore des vocations...

P. GUEULLE



Une antenne yagi 900 ou 1800 MHz suffit pour se lancer dans de passionnantes "chasses au renard" (pardon, aux relais !)

Une carte "PROACTIVE SIM" ouverte

Voici venu le moment d'implanter dans une BasicCard des applications qui "tourneront" sur un téléphone portable faisant office de lecteur de cartes à puce muni d'un clavier et d'un écran. Ceci en exploitant hardiment le mécanisme "Proactive SIM" prévu par la spécification GSM 11.14 et destiné, à l'origine, aux seuls opérateurs de téléphonie mobile.

Quelques lignes de basic...

Nous avons déjà vu comment moins de deux cent lignes de code source ZCBasic suffisent pour développer une carte SIM "minimum" capable d'être reconnue par quasiment n'importe quel téléphone portable et "bricolée" à volonté.

Même s'il n'en faut pas davantage pour démarrer une "session GSM" et manipuler avec les menus (visibles ou cachés) du téléphone, il serait bien plus motivant de venir y greffer des applications personnelles. Aussi étonnant que cela puisse paraître, quelques lignes supplémentaires de code source permettent de parvenir à un tel résultat pourvu que le téléphone soit compatible "SIM Toolkit", autrement dit âgé de quatre à cinq ans et guère plus.

Comme son nom l'indique, la technologie "Proactive SIM" permet à une carte SIM convenablement programmée de donner des ordres au téléphone dans lequel elle est insérée, et non plus seulement d'en recevoir. Elle peut ainsi "récupérer" son clavier et son écran pour dialoguer avec l'utilisateur, et bien plus encore lorsqu'elle est émise par un opérateur de téléphonie mobile et est en cours de validité. Il faut pourtant bien admettre que ceux-ci n'ont plus le monopole du développement de cartes de type SIM : à condition de ne pas chercher à s'inscrire indûment sur leurs réseaux, chacun est libre d'exploiter les spécifications GSM pour faire tout autre chose que téléphoner.

Le meilleur exemple de cette (r)évolution est la "Lifecarte", dans laquelle la société Célavie a imaginé d'enregistrer tous les détails médicaux et personnels de son porteur, qui sont ainsi lisibles en cas d'urgence, en insérant tout simplement la carte dans un téléphone portable GSM. Normalement, le développement de cartes de ce genre nécessite l'usage de kits logiciels très spécialisés et fort coûteux mais, nous allons le découvrir (il suffit d'oser) que le kit BasicCard (gratuit...) n'est pas en reste !

Un mécanisme astucieux

Rendons à César ce qui est à César : il faut bien reconnaître que les concepts "SIM Application Toolkit" et "Proactive SIM" sont de véritables traits de génie de la communauté GSM. Selon les normes ISO 7816, en effet, une carte à puce est fondamentalement "esclave" du terminal dans lequel elle est insé-

rée : celui-ci lui envoie des commandes qu'elle exécute et auxquelles elle répond, un point c'est tout.

Dans le contexte particulier des applications GSM, il a été imaginé qu'une carte SIM puisse retourner un compte-rendu **91 Le** au lieu de **90 00** après l'exécution correcte d'une commande. Grâce à cette adaptation mineure du protocole T=0, la carte peut prévenir le terminal (en l'occurrence un téléphone portable) qu'elle souhaite lui donner un ordre, sous la forme d'un bloc de **Le** octets. C'est alors au téléphone qu'il appartient d'en prendre connaissance (mais seulement si tel est son bon plaisir !) en envoyant une commande "FETCH" (**A0 12 00 00 Le**). Celui-ci rendra alors généralement compte à la carte de son exécution (c'est bel et bien "le monde à l'envers" !), en lui envoyant une commande "ENVELOPE" (**A0 C2 00 00 Lc**), contenant un bloc de **Lc** octets.

Mais dans ce bloc d'octets, il est bien entendu que le téléphone peut à son tour donner des ordres à la carte, selon une syntaxe complètement indépendante du jeu de commandes, forcément restrictif, de la spécification GSM 11.11. Ce sera notamment le cas si l'utilisateur sélectionne une option dans un menu qui lui a été présenté par la carte. Car c'est cela, la grande force du concept "Proactive SIM" : la carte peut insérer ses propres menus dans celui du téléphone et prendre le contrôle de l'afficheur de ce dernier. Cela permet aux applications qu'elle héberge de dialoguer directement avec l'utilisateur en toute indépendance vis-à-vis des réseaux des opérateurs, même si elles ont la possibilité d'établir des communications ou d'envoyer des SMS, parfois de façon fort discrète, pour ne pas dire furtive...

Bien entendu, tous ces mécanismes ont été conçus de façon à ne dérouter en rien les téléphones qui, en raison de leur âge, ne sont pas compatibles "SIM Toolkit". De même, une carte ne supportant aucune fonctionnalité STK ou Proactive SIM doit, en principe, fonctionner convenablement dans un téléphone de la toute dernière génération. Différentes étapes permettent au téléphone et à la carte de s'informer mutuellement de ce qu'ils savent faire, afin d'éviter tout malentendu susceptible de déclencher des comportements erratiques.

En tout premier lieu, le téléphone vient lire l'octet "Phase" de la carte SIM (dans son fichier 7F20:6FAE ou 7F21:6FAE).

Si cet octet est supérieur à 02h (il sera alors en général à 03h), la carte appartient à la "Phase 2+" et supporte donc au moins un sous-ensemble des fonctionnalités STK.

S'il est lui-même compatible "Phase 2+", le téléphone envoie spontanément à la carte une commande "TERMINAL PROFILE" (**A0 10 00 00 Lc**) contenant un bloc de **Lc** octets décrivant ce qu'il sait faire (un bit y étant affecté à chaque fonctionnalité supportée ou non).

Parallèlement, il ira lire le fichier "SIM Service Table" (7F20:6F38 ou 7F21:6F38) de la carte, où il trouvera le détail des "services" que sait gérer cette dernière (à raison de deux bits par fonctionnalité supportée ou non, et activée ou non).

Notons qu'en présence d'une carte de Phase 2 ou 1 (octet Phase respectivement égal à 02h ou absent), le téléphone doit s'abstenir d'envoyer la commande "TERMINAL PROFILE", qui serait de toute façon rejetée avec un message d'erreur **6D 00** (instruction inconnue).

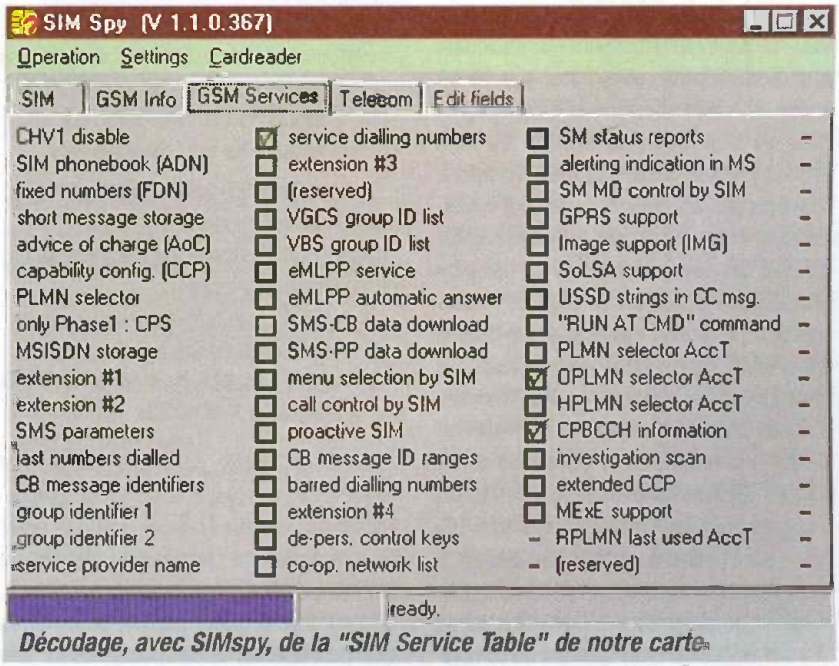
Passons à la pratique

Il est naturellement logique de prendre pour point de départ le code source MINISIM.BAS qui nous a déjà servi à développer une carte SIM "minimum" (EP n° 288).

Rappelons qu'il nécessite, pour sa compilation, une version 4.52 ou supérieure du kit logiciel BasicCard (téléchargeable gratuitement sur www.basiccard.com), et qu'il est destiné aux versions "Professional" de la BasicCard (ZC 5.4 ou ZC 5.5).

Pas énormément plus long (moins de trois cents lignes au total, y compris plusieurs petites applications de démonstration), le fichier MINISTK.BAS offert sur le site de la revue présente tout de même quelques différences ponctuelles. La fonction AscW() permet notamment de le compiler pour une ZC 5.4 Rev. A, et non plus seulement pour une Rev. C ou supérieure supportant directement la fonction ASC du Basic. Mais c'est dans la "SIM Service Table" que se situe une astuce assez particulière : un premier octet à 03h au lieu de 00h. Cela signifie que, bien que le mécanisme de gestion du code PIN ne soit pas réellement implémenté dans notre carte, celle-ci reconnaîtra néanmoins les commandes d'activation ou de désactivation du code confidentiel.

Tout l'intérêt de la chose est que nous utiliserons la fonction "activation du PIN" du téléphone pour mettre en route, uniquement à notre initiative, notre "moteur" Proactive SIM. Insérée dans un téléphone compatible STK, la carte sera reconnue d'emblée comme appartenant à la "Phase 2+", et recevra donc une commande "TERMINAL PROFILE" si le mobile est compatible STK. Son contenu sera mémorisé (dans la chaîne TP$) en vue d'une utilisation que nous détaillerons plus loin. Une "vraie" carte SIM répondrait normalement à cette commande par un compte-rendu **91 XX** au lieu de **90 00**, afin de prier le téléphone de venir chercher un ordre d'insertion d'un nou-



Décodage, avec SIMspy, de la "SIM Service Table" de notre carte.

T (tag)	L (length)	V (value)		
D0h Tag "BER-TLV"	18h 24 octets suivent	... (les lignes suivantes)		
81h Tag "description commande"	03h 3 octets suivent	01h commande N° 1	21h Type commande "Display Text"	81h Qualifier "Priorité haute" + "Effacement par l'utilisateur"
82h Tag "identités"	02h 2 octets suivent	81h origine = SIM	02h destination = afficheur	
8Dh Tag "texte"	0Dh 13 octets suivent	04h codage sur 8 bits	TP$ (12 octets)	

1 ***Structure du programme TLV "Terminal profile"***

veau menu. La nôtre pourrait faire de même si on enlevait le mot REM précédant l'instruction Call CRD(), et cela pourrait effectivement fonctionner avec certains mobiles, notamment les plus anciens supportant le STK. L'ennui, c'est que les téléphones les plus récents sont tellement surchargés de travail, à ce stade de leur initialisation, qu'ils n'ont souvent pas le temps de répondre par une commande "FETCH". C'est expressément toléré par la spécification GSM 11.14, qui admet que cette opération soit remise à plus tard, à charge pour la carte de continuer à répondre **91 XX** au lieu de **90 00** jusqu'à ce que le téléphone veuille bien réagir. Dans la plupart des cas, le mobile se borne à guetter l'apparition d'un compte-rendu **91 XX** en réponse aux commandes "STATUS" (**A0 F2 00 00 Lc**) ou "GET RESPONSE" (**A0 C0 00 00 Lc**) qu'il envoie fréquemment à la carte. Mais si nous programmions la BasicCard pour qu'elle réponde autre chose que **90 00** à une telle commande, elle s'abstiendrait aussi de transmettre le bloc de données attendu (comme si **Lc** était à 0). Il s'agit là d'une particularité de son système d'exploitation, que l'on ne peut contourner à coup sûr qu'avec des versions récentes du compilateur ZCBasic et des cartes (ZC 5.5, notamment). Or, la plupart des téléphones récents considèrent cela comme une anomalie, et mettent fin à la session GSM sur un message d'erreur (carte SIM non valide) ! On pourrait songer à faire émettre le compte-rendu **91 XX** en réponse à une autre commande, n'ayant pas de données à retourner au terminal (par exemple "UPDATE BINARY"). Malheureusement, rien ne dit qu'une telle occasion se présentera à bref délai, tandis que certains téléphones ignorent un compte-rendu **91 XX** reçu à ce stade. Quasiment tous, par contre, interprètent correctement un compte-rendu **91 XX** reçu en réponse à une commande d'activation ou de désactivation du code PIN, et c'est finalement par ce biais que nous avons pu résoudre le problème.

Notre carte n'exécutera donc aucune fonction "Proactive SIM" tant que l'utilisateur n'aura pas effectué, depuis le menu du téléphone, une tentative d'activation du code PIN. Il entrera alors quatre à huit chiffres absolument quelconques, mais une modification triviale du code source (ajout d'un simple IF S$ = ...) suffirait pour que seule une valeur bien précise soit reconnue. Bien que cette demande d'activation soit en réalité ignorée (on se souvient que les fonctions de gestion du code PIN sont simplement simulées), le téléphone signalera sa bonne exécution ! Mais à partir de ce moment, on doit enfin disposer d'une nouvelle entrée (baptisée P. Gueulle) dans le menu principal (ou "Services") du téléphone. Si on la sélectionne, un sous-menu doit apparaître, proposant les cinq choix suivants :

- *Term profile* (affichage du "Terminal Profile" précédemment mémorisé).
- *Clear LOCI* (réinitialisation des informations de localisation du mobile).
- *LP = FRE* (langue préférentielle : français).
- *LP = ENG* (langue préférentielle : anglais).
- *SIM Reset* (réinitialisation de la carte SIM par le téléphone).

Pour autant qu'ils occupent exactement douze caractères (espaces compris), tous ces libellés peuvent être librement modifiés avant la compilation du code-source, permettant une large personnalisation de ce menu. Changer leur longueur (dans la limite de ce que peut afficher le téléphone) serait une toute autre affaire, supposant de modifier avec précision plusieurs octets disséminés un peu partout dans le "programme" construit autour de la commande proactive "SET UP MENU".

Ceci du fait de l'utilisation d'un langage dit "TLV" (Tag, Length, Value), dans lequel apparaissent régulièrement des octets indiquant la longueur des données qui suivent, et dont la structure permet l'imbrication, parfois complexe, de plusieurs niveaux (comme on le verra à la **figure 1**).

C'est dans le dernier des **Lc** octets de données d'une commande "ENVELOPE" (**A0 C2 00 00 Lc D3** etc.) que le mobile indique à la carte quelle option du menu "P. Gueulle" vient d'être sélectionnée par l'utilisateur. La valeur de la variable STN est alors chargée, s'il y a lieu, avec le numéro (1, 2, ou 3) de l'application "Proactive SIM" à exécuter et un compte-rendu approprié est élaboré par le sous-programme Sub CRD(). Si STN = 2, par exemple, la carte renverra un compte-rendu **91 1A** en réponse à la commande "ENVELOPE". Le téléphone saura alors qu'il doit venir chercher vingt-six (1Ah) octets par une nouvelle commande "FETCH" (**A0 12 00 00 1A**). Lorsqu'elle recevra cette commande, la carte se souviendra que STN = 2 (car STN est une variable "publique"), et placera dans ces 1Ah octets une structure TLV demandant au téléphone d'afficher, en hexadécimal, les quatre premiers octets du "Terminal profile", mémorisés dans TP$: nous y voilà ! Dans la foulée, STN sera remis à zéro afin que l'ordre ne soit pas exécuté à plusieurs reprises.

Programmer en "TLV"

Analysons donc un peu le contenu de ce "programme" en miniature, qui met en oeuvre la commande proactive "DISPLAY TEXT". Le langage "TLV" est basé sur l'emploi de "tags", codes tenant sur un octet et fixant la signification des octets de données ("value") qui le suivent.

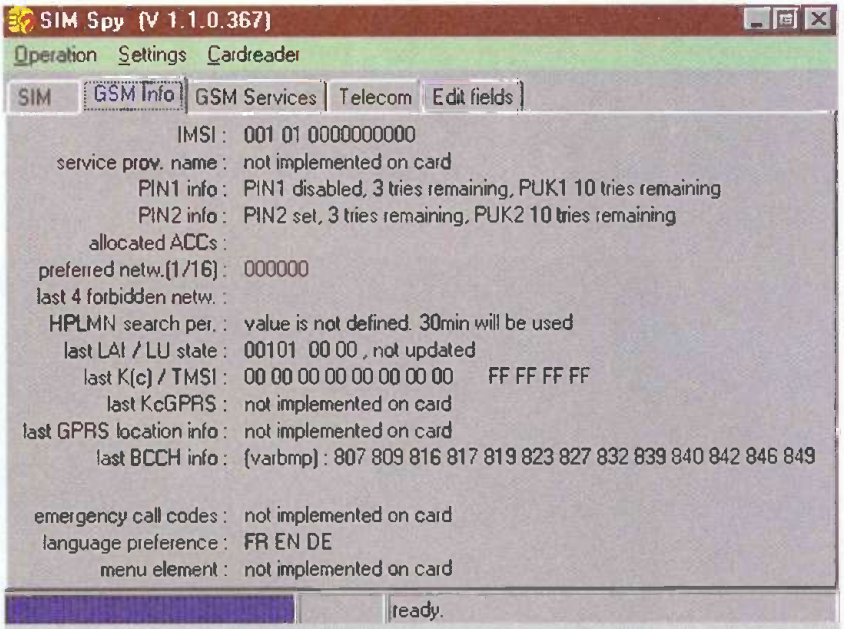
Entre les deux, un octet "length" précise le nombre d'octets (longueur) du champ "value". Bien entendu, plusieurs de ces structures "SIMPLE-TLV" peuvent se suivre, comme le montre la **figure 1**.

Le premier octet est à D0h, valeur de "tag" signalant que tout ce qui suit est une commande "Proactive SIM", écrite en code "BER-TLV" (Basic Encoding Rules). Le second indique la longueur du bloc d'octets venant après lui, ici vingt-quatre (18h), y compris les

douze octets de TP$ (soit quatre groupes de deux caractères hexa, suivis chacun d'un espace). Les douze octets qui se situent entre deux ont la signification suivante :

- Description de la commande (tag 81h) : 03h (3 octets suivent), 01h (commande N°1), 21h (type = "Display Text"), 81h (priorité haute, effacement manuel par l'utilisateur).
- Identités (tag 82h) : 02h (2 octets suivent), 81h (expéditeur = SIM), 02h (destinataire = afficheur).
- Texte alphanumérique (tag 8Dh) : 0D (13 octets suivent), 04 (codage sur 8 bits).

Bien évidemment, tout cela ne s'invente pas, mais on "s'y met" finalement assez vite, un peu comme au langage machine d'un microcontrôleur. La spécification GSM 11.14 (téléchargeable gratuitement sur www.etsi.org) décrit avec précision comment coder chacune des commandes "proactives" que recense le tableau ci-dessous.



Les principales caractéristiques de notre carte, vues par SIMspy

Commande	Type	En ligne
Refresh	01h	
More Time	02h	
Poll interval	03h	
Polling Off	04h	
Set Up Call	10h	*
Send SS	11h	*
Send USSD	12h	*
Send SMS	13h	*
Play Tone	20h	
Display Text	21h	
Get Inkey	22h	
Get Input	23h	
Select Item	24h	
Set Up Menu	25h	
Provide local information	26h	*

Les principales commandes "proactives"

Si l'on considère que chacune de ces commandes peut admettre de multiples variantes, selon la valeur de l'octet "qualifier" qui la suit (dans notre exemple, pour fixer la priorité et le mode d'effacement du texte affiché), il est clair que les possibilités sont immenses. Ceci même en se limitant aux commandes utilisables "hors ligne", c'est-à-dire sans inscription sur le réseau d'un opérateur.

Encore quelques octets

Imaginons maintenant que l'utilisateur ait sélectionné l'option "LP = ENG" : apprenant que c'est l'entrée numéro quatre du menu qui a été choisie, la commande "ENVELOPE" met à 01h le premier octet de la chaîne LP$ stockée dans la mémoire EEPROM de la carte (fichier GSM "Language Preference").

Nous sommes cette fois en présence d'un programme strictement interne à la BasicCard (STN reste à 0, et le compte-rendu à **90 00**), qui pourrait bien sûr être infiniment plus complexe (par exemple de type cryptographique).

À partir de maintenant, donc, la langue préférentielle de la carte SIM est l'anglais, mais rien ne transparaît encore au niveau de l'affichage : ce changement ne sera effectif qu'après un arrêt du mobile, ou bien après un "reset" de la carte commandé par la cinquième option du menu.

Cela si, et seulement si, le choix de la langue a été programmé sur "automatique" dans le menu du téléphone (si une langue donnée a été présélectionnée dans le mobile, elle sera en effet prioritaire par rapport à celle spécifiée comme "préférentielle" dans la carte SIM). Là encore, le "reset" de la SIM est effectué (par le téléphone) à la demande de l'application résidant dans la carte qui répond à une commande "FETCH" (cas STN = 3), par une structure TLV contenant une commande proactive "REFRESH" (code opératoire 01h), suivie d'un "qualifier" 04h précisant l'ampleur de la réinitialisation demandée.

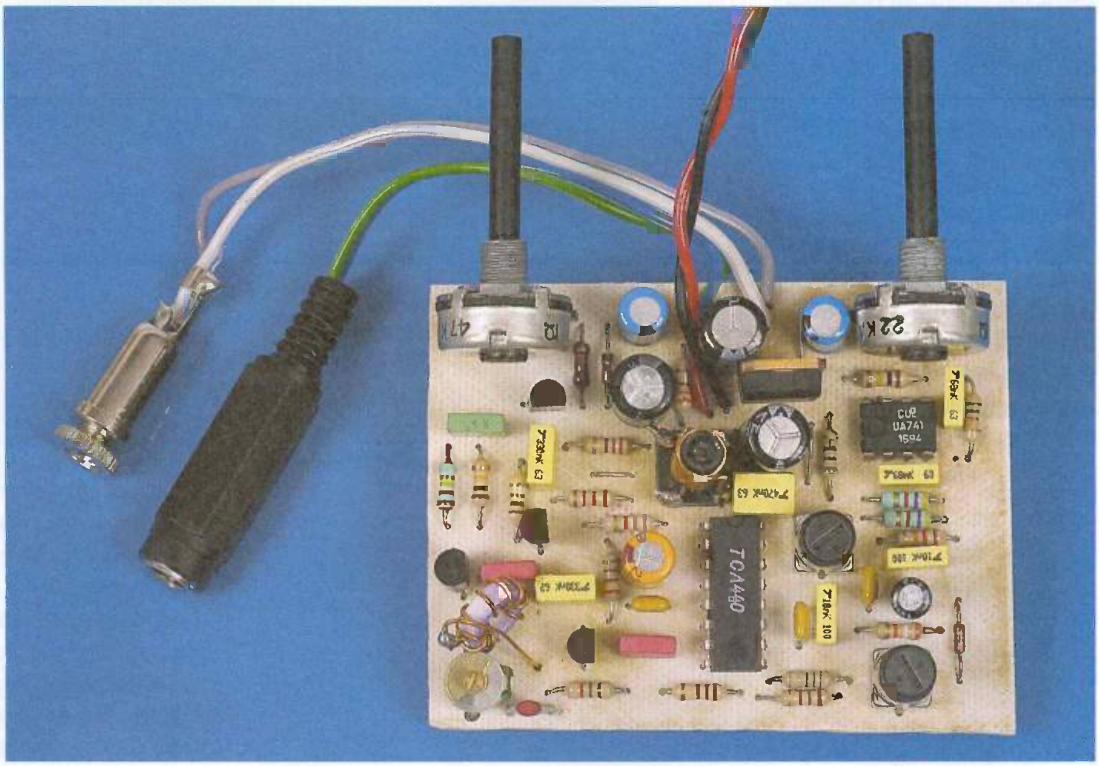
On aura compris qu'il ne s'agit évidemment là que de quelques exemples, volontairement très simples : l'essentiel de la puissance d'une telle carte tient au fait qu'il reste, dans une BasicCard ZC 5.4 ou ZC 5.5, énormément de place pour les propres applications du développeur.

En fait, celui-ci peut fort bien ne se servir des fonctionnalités "Proactive SIM" que pour communiquer, comme nous venons de le faire, avec le clavier et l'afficheur du téléphone, qui servira alors de lecteur de cartes à puce évolué.

Et si l'on s'en tient là, ce n'est vraiment pas aussi compliqué qu'on pourrait le croire...

P. GUEULLE

Récepteur 27MHz pour la CB



Grâce à ce montage simple à mettre en oeuvre et facile à régler, vous pourrez écouter le trafic sur la CB (Citizen Band), qui, si elle est moins populaire qu'au début des années 80, reste encore assez fréquentée.

Il s'agit d'un récepteur superhétérodyne dont la fréquence intermédiaire est de l'ordre de 455 kHz environ. Il a été conçu pour être très simple à mettre en œuvre puisqu'il ne demande qu'un seul véritable réglage à effectuer et malgré cela, possède une sensibilité et une sélectivité tout à fait suffisantes pour un récepteur dédié à la surveillance de l'activité de la bande des 11 mètres (27 MHz).

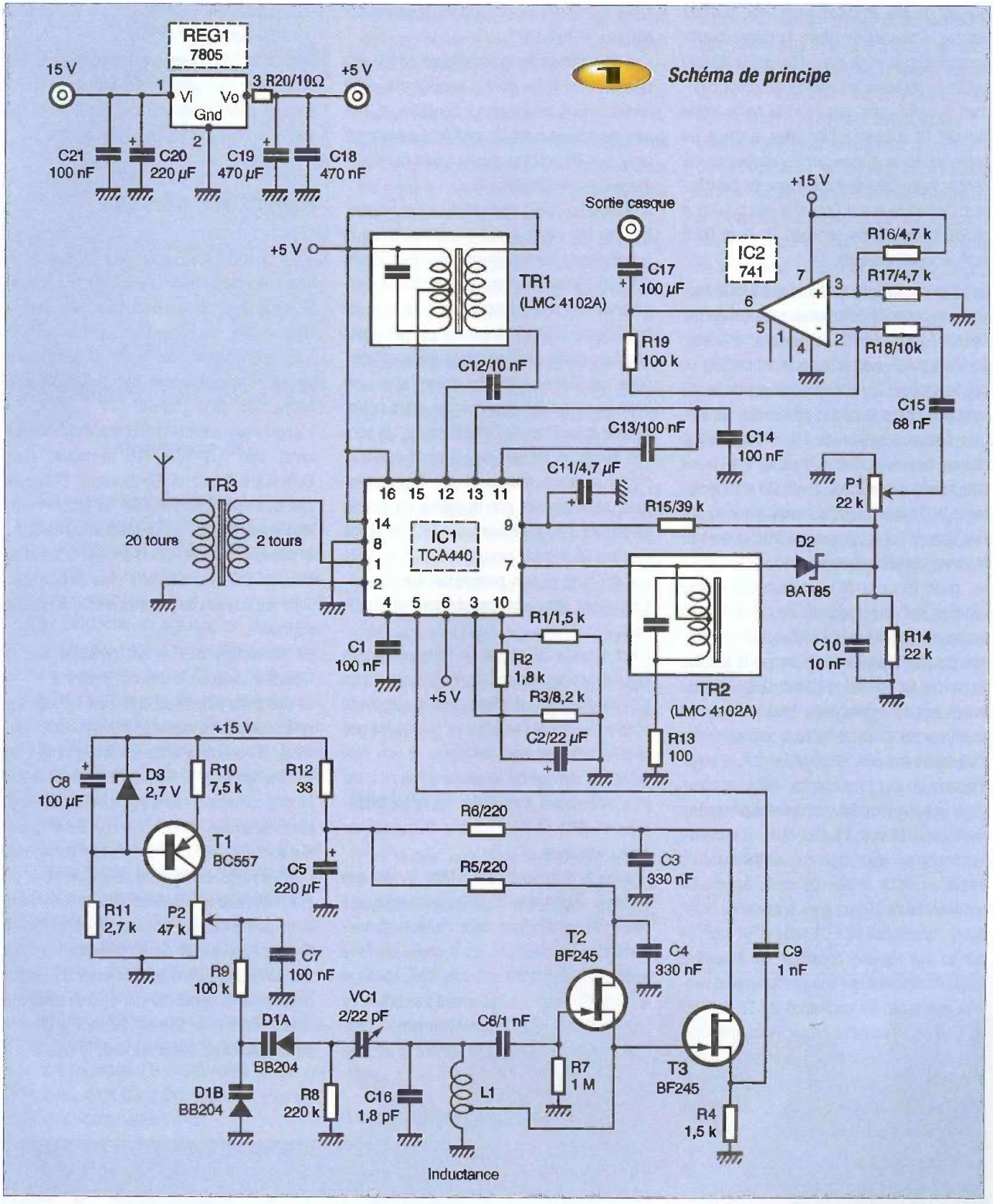
Le cœur du montage est un circuit spécialisé dans la réception en modulation d'amplitude (AM), le TCA 440, certes un peu ancien mais parfaitement suffisant pour cette réalisation et toujours disponible.

Le schéma de principe (**figure 1**) suit dans ses grandes lignes les notes d'applications fournies par le constructeur :

Le signal capté par l'antenne est amené au premier amplificateur radio-fréquence (noté RF sur les schémas constructeurs) IC1 par l'intermédiaire d'un transformateur adaptateur d'impédance TR3. Le rapport des spires primaire/secondaire, en l'occurence d'une valeur de 10, est donné par le fabricant du circuit.

On remarquera que dans le but de simplifier au maximum le réglage du récepteur, le circuit d'accord habituel situé au niveau de l'entrée d'antenne a été omis. Cela a deux conséquences : premièrement, les fréquences parasites (et en particulier celles de la FM 88/108 MHz) ne trouvent pas d'obstacle à l'entrée du montage et peuvent venir perturber un peu la réception (surtout si l'antenne est mal adaptée), mais ce problème reste mineur dans la bande de fréquences qui nous intéresse ici.

La seconde conséquence de l'absence de circuit d'accord d'entrée est à la fois un avantage et un inconvénient : comme ce filtre d'entrée est omis, pour une fréquence de l'oscillateur local donnée que nous noterons Fosc, il sera possible de recevoir à la fois Fosc+FI et Fosc-FI (avec FI=455 kHz environ). Cela permet d'étendre la bande basse et haute reçue par le récepteur pour une plage de variation de la fréquence Fosc donnée. Le risque de brouillage est faible puisqu'il est très rare que les deux fréquences Fosc+FI et Fosc-FI soient occupées en même temps, surtout dans cette bande de fréquences. Par contre, cela crée un effet de recouvrement de la bande qui fait que certaines stations seront audibles sur deux points d'accords différents mais cela est sans importance. Afin d'éviter tout risque de saturation, le gain de l'amplificateur radio-fréquence est contrôlé par le niveau de réception, par l'intermédiaire du réseau de résistances R1,R2 et R3 et du condensateur C2 qui en extrait la valeur moyenne.



Une fois le signal radio d'entrée amplifié par le TCA 440, il parvient au mélangeur où il est mixé avec le signal issu de l'oscillateur local afin d'opérer au changement de fréquence.

Cet oscillateur local est bâti à l'aide de transistors indépendants T2 et T3 et non à l'aide des sous systèmes spécialisés du TCA440, ceci parce que l'oscillateur interne de ce circuit s'est révélé fort capricieux et instable aux environs de 27 MHz, sans réelle possibilité d'y remédier du fait du manque complet de données pratiques sur le fonctionnement de cet oscillateur dans la fiche technique du fabricant.

Cet oscillateur est une variante du classique oscillateur Hartley, assemblé autour du FET T2 : l'oscillation est dûe à la réinjection d'une fraction du signal (entre 1/5 et 1/3) à la source de T2, grâce à une prise intermédiaire insérée sur le bobinage L1.

La capacité d'accord du circuit oscillant est déterminée par une diode varicap double : la BB204, couplée au bobinage par l'intermédiaire du condensateur ajustable VC1 dont le

rôle est double. Placé en série avec la diode varicap, il permet de régler la plage de fréquences reçue, et de plus, il bloque la tension continue présente au niveau du point commun à L1/C6 afin que l'anode de la diode varicap D1.A puisse bien être amenée au potentiel de la masse par la résistance R8 dont la valeur élevée évite d'amortir l'oscillateur. On notera que du point de vue du circuit oscillant, les diodes varicaps D1.A et D1.B sont montées en série.

La tension continue d'accord qui sert à faire varier la capacité des diodes est amenée par l'intermédiaire de R9. Ici encore, la valeur élevée de R9 évite d'amortir le circuit. Le condensateur C7 évite des retours de haute fréquence dans le circuit générateur de tension construit autour de T1. Il s'agit d'une source de courant fixe dont la valeur est déterminée par la diode zener D3 et la résistance R10 qui permet d'obtenir environ 12,5 V aux bornes du potentiomètre P2 qui sert au réglage. Lorsque la tension en sortie de P2 est nulle (curseur à la masse), les diodes varicaps ont leur capacité maximale, on se trouve alors dans la zone d'accord des fréquences les plus basses. Lorsque la tension en sortie de P2 est de 12,5 V, les diodes voient leur capacité chuter fortement et l'on se trouve alors dans la zone d'accord des fréquences les plus élevées. En fait, la plage d'accord du montage une fois réglé sera plus large que la bande CB proprement dite qui va de 26,960 Mhz à 27,410 MHz. On pourra ainsi capter des signaux radioamateurs (hélas en BLU) autour de cette bande. Le condensateur C16 a son importance : en effet, l'ensemble VC1,D1.A,D1.B et tout ce qui lui est attaché constitue un ensemble assez hétérogène qui entrave fortement l'entrée spontanée en oscillation de T2. Le rôle de C16 est donc de stabiliser l'ensemble par son comportement purement capacitif tout en ayant une capacité suffisamment faible pour ne pas diminuer l'action des diodes varicaps sur l'accord du circuit oscillant.

Le transistor T3 est un simple tampon qui permet d'éviter une trop grande influence de l'entrée du TCA440 sur l'oscillateur T2.

Le signal disponible en sortie du mélangeur se trouve sur la broche 15 de IC1. Il faut à présent l'amplifier pour gagner en sensibilité et le filtrer pour gagner cette fois en sélectivité. La partie amplificatrice est entièrement intégrée dans IC1, le CAG (contrôle automatique de gain) ne fait appel qu'à R15 et C11. Par contre, le filtrage demande deux composants externes plus importants, les transformateurs TR1 et TR2.

Ces deux composants constituent en fait une charge accordée dont l'impédance croît brusquement aux alentours de 455 kHz. TR1 sert de charge pour la sortie du mélangeur alors que TR2 charge pour sa part le dernier étage de l'amplificateur FI.

Afin de simplifier la réalisation de l'ensemble, il a été fait appel à des transformateurs FI (=Fréquence Intermédiaire) conventionnels (TR1 et TR2) pour cet usage. Cependant, l'impédance des transformateurs FI courants est trop élevée (15000 ohms), c'est pour cela que l'on utilise la prise intermédiaire disponible sur TR1 et TR2. On obtient ainsi une impédance de 7500 ohms qui, plus faible, évite la saturation de l'amplificateur. Un second avantage est que cela amortit moins le circuit oscillant interne de ces transformateurs et augmente par là même un peu la sélectivité. On remarquera que TR2 est utilisé comme un simple circuit oscillant accordé sur 455 kHz et non comme un transformateur, selon les recommandations du fabricant.

Il est d'usage de placer un filtre céramique entre le condensateur C12 et la broche 12 de IC1, dont le but est d'accroître la sélectivité mais on a préféré l'omettre ici, ceci parce que la sélectivité est déjà suffisante et que cela facilite le réglage du montage puisqu'il n'est plus nécessaire d'accorder les transformateurs FI (TR1 et TR2) avec la fréquence du filtre céramique.

Une fois le signal amplifié et filtré, il reste à le détecter. Plutôt que d'utiliser pour cela une diode au germanium dont l'approvisionnement devient aléatoire, on a choisi de faire appel ici à une diode schottky très répandue (D2) dont le seuil est également très faible. Le signal traverse alors le potentiomètre P1 destiné au réglage du volume sonore et attaque un amplificateur de gain 10 construit autour d'un classique LM 741 (IC2).

Une sortie pour casque a été préférée à une sortie sur haut-parleur 8 ohms mais il est toujours possible de brancher le montage sur des enceintes amplifiées.

Réalisation pratique

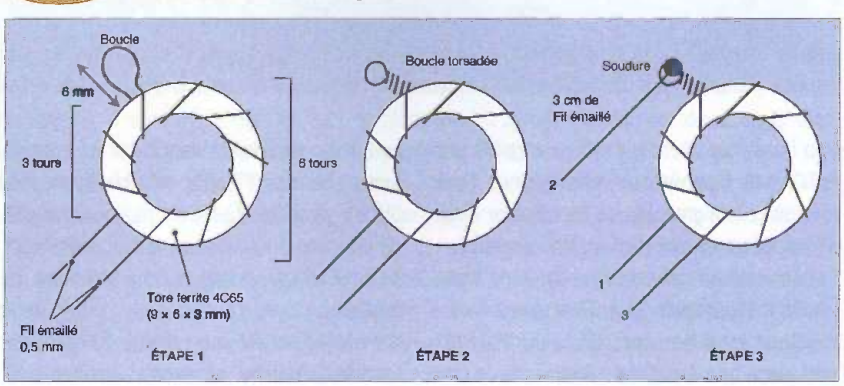
Il est souvent préférable, pour la réalisation d'un récepteur radio, d'utiliser un circuit imprimé avec un plan de masse mais dans le cas présent, le récepteur fonctionne bien avec un circuit simple face. Les pistes sont larges et ne devraient pas poser de problèmes lors de la gravure.

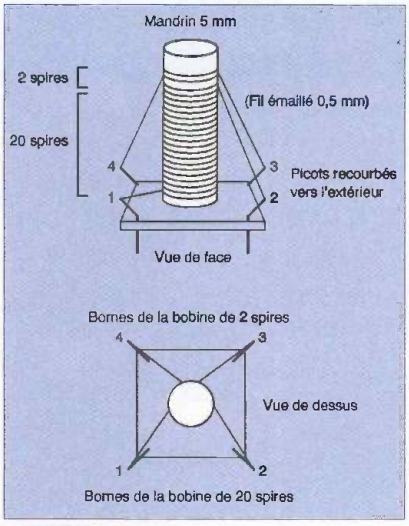
L'approvisionnement des composants est lui aussi aisé : le TCA 440 se trouve chez Conrad,Go Tronic et Electronique Diffusion. Les transformateurs FI, TR1 et TR2,le mandrin 5 mm destiné à la construction de TR3 , le tore ferrite 4C65 et le fil émaillé 0,5 mm se trouvent tous par exemple chez Sélectronic. Tous les autres composants sont d'un usage courant.

La confection de L1 est détaillée sur la **figure 2**, celle de TR3 sur la **figure 3**.

L1 comporte 9 tours de fil émaillé 0,5 mm sur un tore ferrite en matériau 4C65 (couleur violette). Si l'on commence à bobiner le tore en commençant par le fil de sortie numéroté 3, il faudra compter 6 tours avant de laisser le fil faire une petite boucle de 6 mm de longueur qui sera destinée au branchement de la prise intermédiaire. En laissant la boucle libre, on poursuit alors le bobinage des trois derniers tours puis on laisse dépasser environ 2 cm de fil pour former le fil de sortie numéro 1. Une fois cette première étape achevée, il faut torsader le fil de la boucle afin de bien tendre le fil enroulé sur le tore en laissant juste une petite ouverture au sommet de la boucle. La

2 ***Confection du bobinage L1***





3 ***Confection du bobinage TR3***

dernière étape consiste à souder dans cette ouverture un brin de fil émaillé pour disposer alors du fil de sortie numéro 2. Normalement, le fil émaillé est soudable directement sans ponçage préalable, il faut juste chauffer de façon insistante en ajoutant de la soudure jusqu'à ce que l'émail soit portée à ébullition et que la soudure prenne enfin. Il faudra faire de même lorsque l'on soudera la bobine L1 sur le circuit imprimé en prenant garde de ne pas surchauffer les pistes du circuit. Un peu d'expérimentation sur des chutes de fils pourra toujours servir.

L'avantage d'utiliser un tore est que le rayonnement de la bobine sera minimal, ce qui est important à ces fréquences. Par contre, la valeur de l'inductance ne sera pas réglable.

Le transformateur TR3 est construit sur un mandrin de diamètre 5 mm. Afin de faciliter sa confection, il sera utile de plier vers l'extérieur le haut des quatre picots ce qui permettra de fixer aisément les brins de fil émaillé avant soudure. Le primaire de ce transformateur est constitué d'une bobine de 20 spires (0,5 mm), dont les extrémités sont connectées aux picots notés 1 et 2.

Le secondaire est bobiné juste au dessus de cette première bobine et ne comporte que deux spires. Les extrémités de cette seconde bobine seront reliées aux picots notés 3 et 4.

Lors du placement des composants, il ne faudra pas oublier de souder le strap placé sous IC1 avant de souder ce circuit. Il n'est pas nécessaire de placer les circuits IC1 et IC2 sur des supports mais il faut être sûr de leur fonctionnement et ne pas les endommager en les soudant.

Les liaisons avec les fiches jack du casque et d'alimentation (vérifiez la polarité au moment du branchement!) sont détaillées sur le schéma d'implantation des composants. Le montage sera alimenté sous 13,5 V à 15 V, la consommation ne doit pas dépasser 100 mA.

Le dernier branchement à effectuer concerne l'antenne. Il se fait par l'intermédiaire de deux fils à souder juste à côté de TR3 et désignés sur le schéma d'implantation par les lettres A (pour antenne) et B (pour blindage). Le lecteur trouvera en **figure 4** un exemple d'antenne filaire qui donne des résultats corrects sans être complexe à installer. Elle est constituée par un simple fil souple (multibrins) de longueur totale L1 + L2 = 417 cm qui possède une prise intermédiaire au point indiqué sur le schéma (attention,le fil n'est pas coupé en ce point.). C'est sur cette prise intermédiaire que sera connecté le fil issu du point noté A sur la plaquette évoqué ci-dessus. Le fil issu du point B sera tout simplement enroulé autour de ce fil issu du point A jusqu'au niveau du point de jonction avec la prise intermédiaire de l'antenne mais son extrémité restera libre et isolée, sans aucune connexion électrique puisqu'il ne s'agit que d'un blindage.

(On aura intérêt à ce que les deux sections L1 et L2 du fil d'antenne soient bien coplanaires.)

Réglages

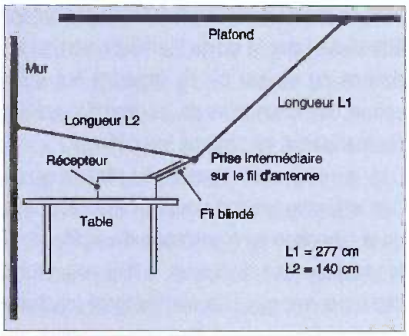
Avant la mise sous tension, il faut régler TR1 et TR2 à mi-course (voir le réglage de la maquette présentée), de même que pour VC1. Le noyau de TR3 devra tout simplement être vissé jusqu'à ce qu'il affleure un peu au dessus du niveau où se trouvent les deux spires du secondaire de ce transformateur. Normalement, ce réglage est définitif.

Dès la mise sous tension du récepteur, on doit entendre un fort souffle, caractéristique de la réception en modulation d'amplitude. Si le récepteur est silencieux, il faut en premier lieu s'assurer que l'oscillateur local fonctionne bien. Pour cela, il faut mesurer avec un multimètre possédant une forte impédance d'entrée la tension aux bornes de la résistance R7. On doit trouver une valeur proche de -1,5 V (si l'on a branché le cordon noir du multimètre à la masse du montage). Toute valeur nettement inférieure, comme -0,5 V ou tout simplement une tension nulle indique que T2 n'oscille pas correctement : parmi les solutions possibles, si la bobine a été correctement réalisée, on peut tenter d'augmenter la valeur de C16 jusqu'à 2,2 pF ou bien encore déplacer ce condensateur et le souder juste aux bornes de L1.

Le seul réglage du montage consiste à ajuster la valeur de VC1 pour se placer juste sur la bande des 11 mètres. Normalement, lorsque cet ajustable est à mi-course, on doit pouvoir capter quelques émetteurs CB lorsque le potentiomètre P2 est en fin de course. Il faudra alors diminuer encore la valeur de VC1 en prenant garde cependant à ne pas atteindre une valeur trop faible ce qui, en plus d'une bande d'accord très restreinte, mène parfois au décrochage de l'oscillateur local.

Bien entendu, il n'y a pas toujours des émis-

Mise en situation du bobinage TR3 sur son mandrin



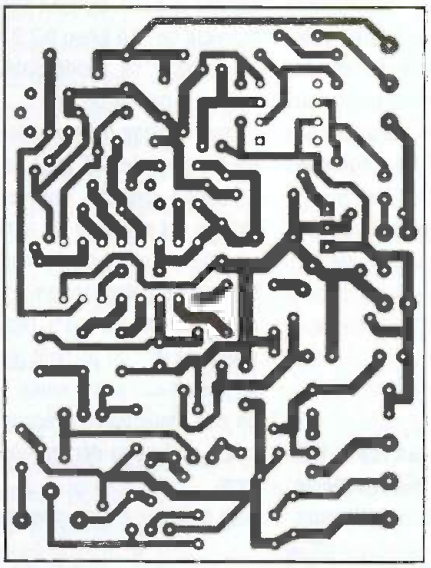
4 *Exemple d'antenne filaire*

sions à capter sur la bande des 27 MHz et il faut savoir être patient. Juste en dessous de cette bande se trouvent des fréquences dédiées aux radio-amateurs où l'on capte souvent des signaux de toutes sortes, ce qui permettra au moins de vérifier le bon fonctionnement du récepteur. On pourra tenter de régler TR2 pour améliorer la qualité de la réception mais ce réglage n'est pas critique. L'auditeur entendra également certaines stations FM dont les fréquences d'émission auront interféré avec les harmoniques du signal fourni par l'oscillateur local pour donner un battement à 455 kHz, mais ces images sont rares autour de 27MHz, beaucoup plus fréquentes en dessous.

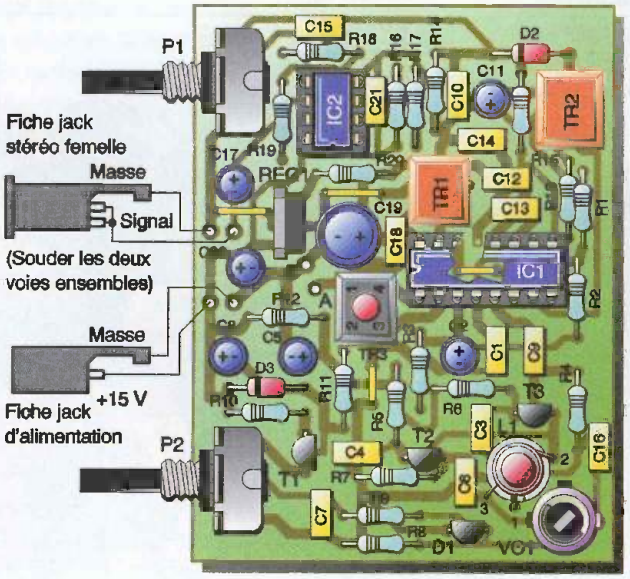
La bande CB est divisée en 40 canaux d'une largeur de 10 kHz et l'un de ces canaux est entièrement dédié aux usagers de la route (le canal 19), c'est certainement sur celui-ci que l'on aura le plus de chances de capter un signal.

O. VIACAVA

5 *Tracé du circuit imprimé*



6 *Implantation des éléments*



Nomenclature

Résistances :
R1, R4 : 1,5 kΩ
R2 : 1,8 kΩ
R3 : 8,2 kΩ
R5, R6 : 220 Ω
R7 : 1 MΩ
R8 : 220 kΩ
R9, R19 : 100 kΩ
R10 : 7,5 kΩ
R11 : 2,7 kΩ
R12 : 33 Ω
R13 : 100 Ω
R14 : 22 kΩ
R15 : 39 kΩ
R16, R17 : 4,7 kΩ
R18 : 10 kΩ
R20 : 10 Ω

Condensateurs :
C1, C7, C13, C14, C21 : 100 nF
C2 : 22 µF
C3, C4 : 330 nF
C5, C20 : 220 µF
C6, C9 : 1 nF
C8, C17 : 100 µF
C10, C12 : 10 nF
C11 : 4,7 µF
C15 : 68 nF
C16 : 1,8 pF
C18 : 470 nF
C19 : 470 µF

Diodes :
D1 : BB204
D2 : BAT85
D3 : Diode Zéner 2,7 V

Transformateurs :
TR1, TR2 : LMC4102A ou MCS 10735
Transformateur FI
TR3 (voir texte) : 20 tours et deux tours sur mandrin 5mm en fil émaillé 0,5 mm

Circuits intégrés :
IC1 : TCA 440
IC2 : LM 741

Inductance :
L1 (voir texte) : 9 spires sur tore ferrite violet de grade 4C65, Taille C : 9_6_3 mm

Potentiomètres (axe 4 mm) :
P1 : 22 kΩ LOG
P2 : 47kΩ LIN

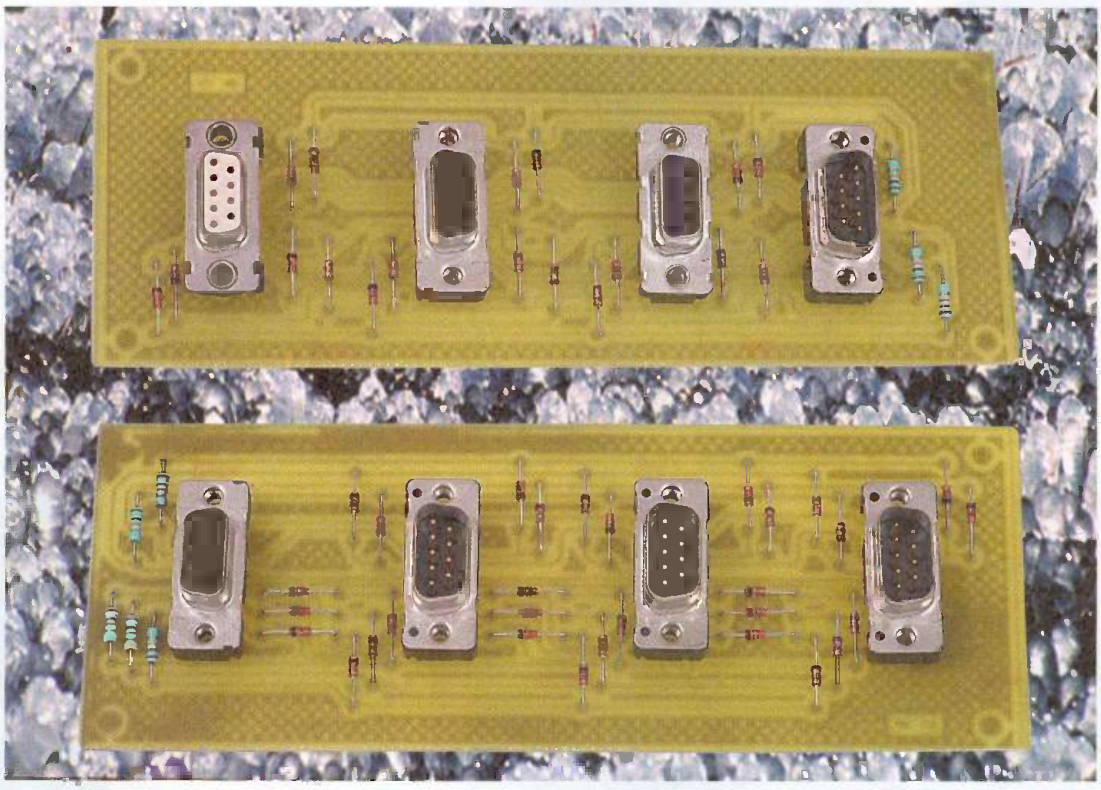
Régulateur :
REG1 : 7805

Transistors :
T1 : BC557
T2, T3 : BF245 B ou C

Condensateur ajustable :
VC1 : 2/22 pF (vert)

Divers : fil émaillé 0,5 mm, 4 m de fil multibrins 0,22 mm_ pour l'antenne, prise jack d'alimentation 5,5/2,5, connecteur jack stéréo 3,5 mm femelle pour cordon.

Partagez automatiquement les ports série de votre PC



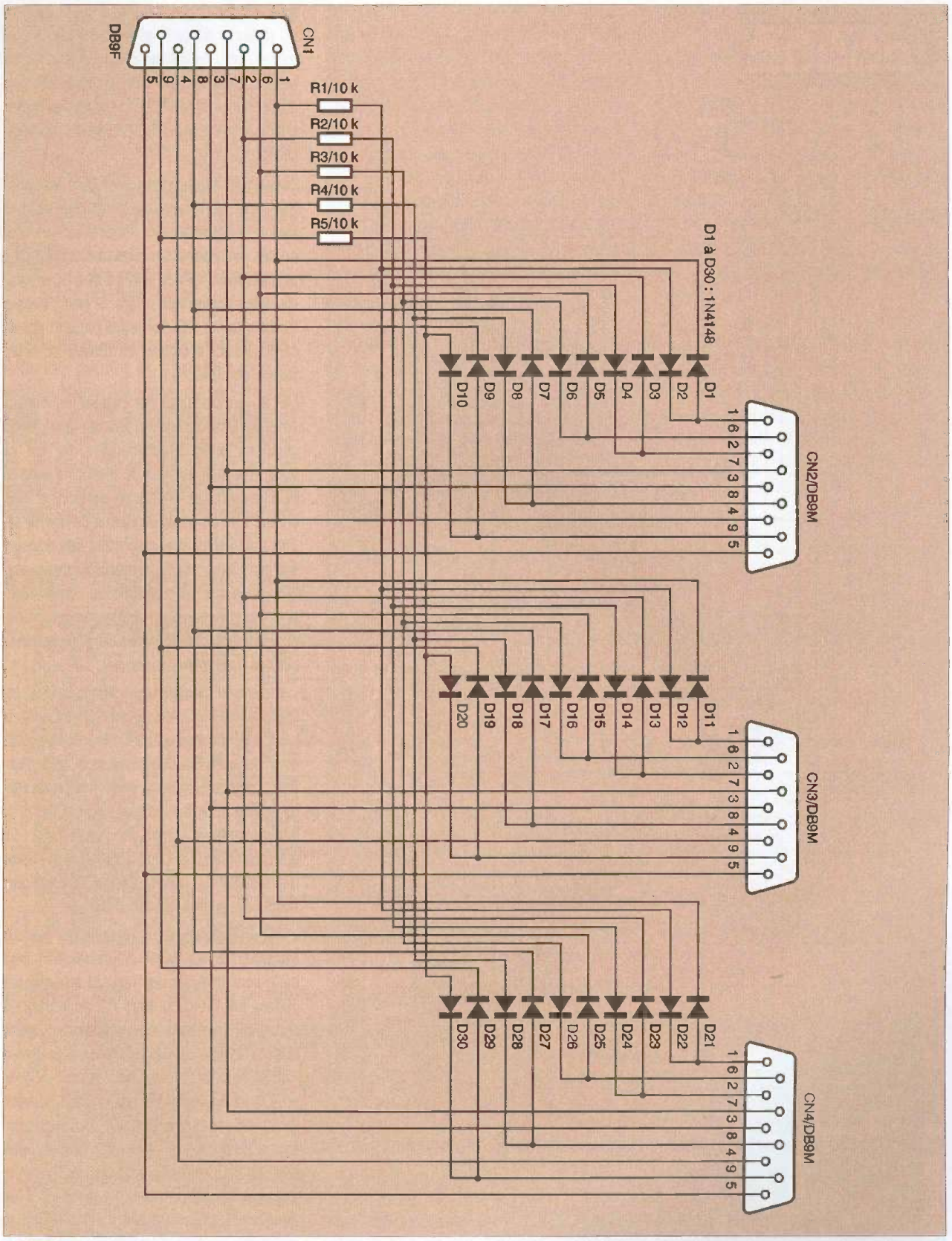
Bien qu'il soit aujourd'hui possible d'équiper un PC de très nombreux ports série, grâce à des adaptateurs USB/RS232, il peut être intéressant de disposer d'un commutateur bon marché pour partager les ports RS232.

Il est souvent nécessaire de brancher et débrancher toutes sortes d'équipements aux ports série de son PC. C'est le cas notamment avec les nombreux montages que nous vous proposons régulièrement dans ces pages. Dans certains cas, il est possible de partager la liaison RS232 en raccordant les équipements en " étoile ". Bien entendu, cela sous-entend qu'il faut veiller à ce qu'il n'y ait qu'un seul équipement qui transmette les données à la fois. Cela est possible, par exemple, en ne laissant qu'un seul équipement alimenté, ce qui est parfois plus simple que de se mettre à quatre pattes pour accéder à l'arrière du PC. Dans d'autres cas, la nature même des équipements permet le partage de la liaison RS232 dans une configuration en étoile (appareils de mesure, etc ...).

Schéma

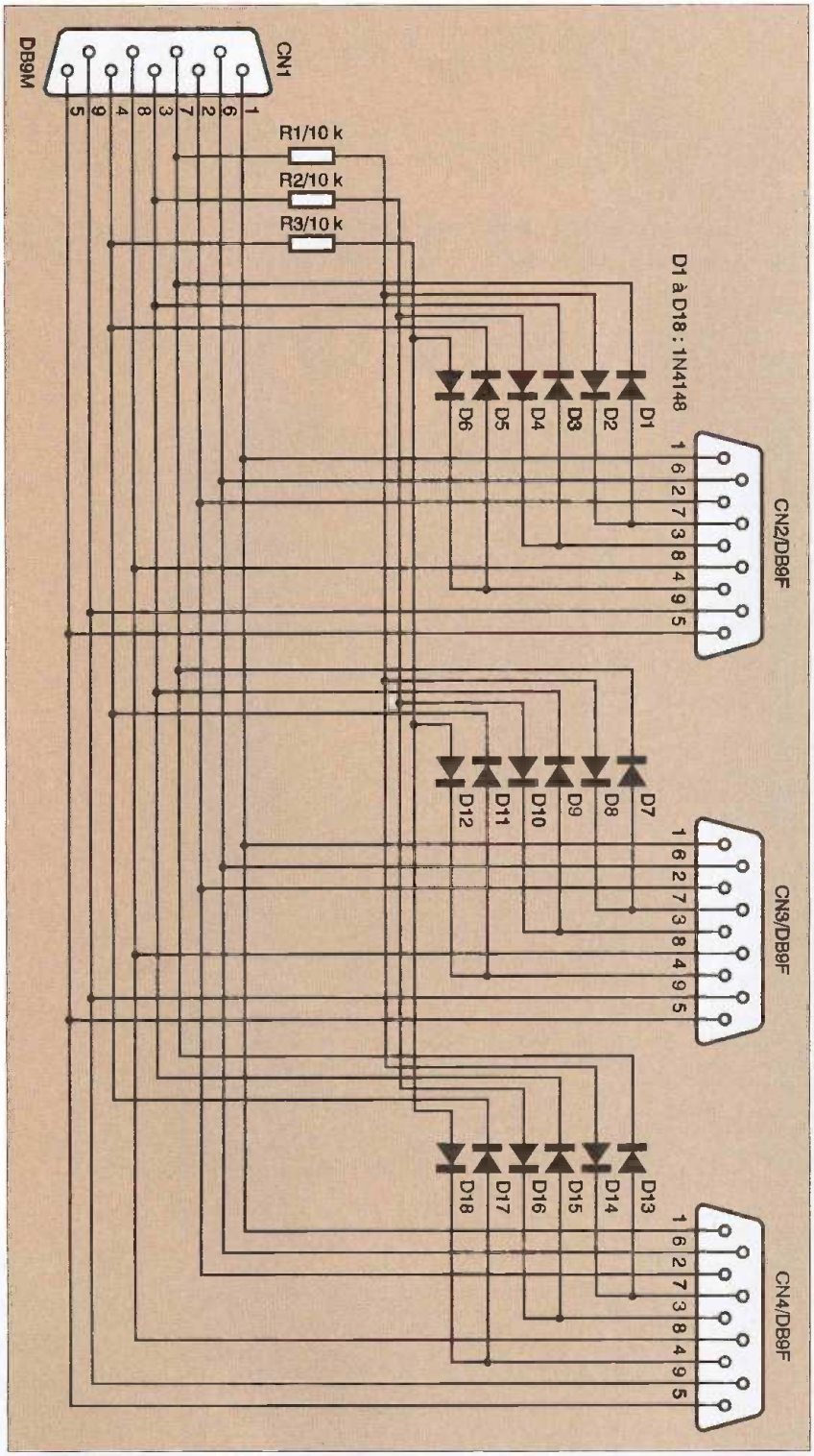
Sous la dénomination RS232 peut se cacher deux types d'équipements : DTE ou DCE. Les équipements de type DTE (Data Terminal Equipement) prennent généralement l'initiative du dialogue sur la liaison RS232. Pour ces équipements, par exemple, le signal dénommé TX transmet les données de l'équipement tandis que le signal dénommé RX reçoit les données. A l'inverse, un équipement de type DCE (Data Communication Equipement) se comporte généralement comme un équipement récepteur (terminal, imprimante, etc ...). Pour ces équipements, le signal dénommé TX reçoit les données émises par un équipement DTE tandis que le signal dénommé RX transmet les données.

En toute logique, il est donc possible de connecter directement un équipement DTE et un équipement DCE à l'aide d'un câble direct (liaison fil à fil). A l'inverse, pour connecter ensemble deux équipements de type DTE, il faut faire appel à un " cordon croisé ". Ajoutons que les équipements DTE sont généralement équipés d'un connecteur mâle (Sub-D 9 points ou



1 ***Commutateur permettant de relier un port série de type DTE vers trois terminaux équipés de type DCE***

25 points) tandis que les équipements DCE sont équipés d'un connecteur femelle. Il faut cependant se méfier car certains fabricants d'équipements prennent des libertés avec ces règles. Il faut donc toujours vérifier les informations fournies dans la notice de l'équipement. Ceci étant dit, vous comprendrez qu'il n'est pas aussi simple que cela de réaliser un commutateur pour port série qui



2 ***Commutateur permettant de relier un terminal (DCE vers trois PC (DTE)***

couvre toutes les combinaisons possibles. Dans notre cas, nous avons préféré décomposer les problèmes en deux au travers de deux interfaces dont les schémas sont reproduits en **figures 1** et **2**. Le schéma de la **figure 1** correspond à un commutateur permettant de relier un port série de type DTE (port RS232 d'un PC) vers trois terminaux équipés de type DCE (une imprimante série, par exemple). Le schéma de la **figure 2** correspond à la situation inverse et permet de relier un terminal (DCE) ou une imprimante vers trois PC (DTE). Ceci peut, par exemple, permettre de visualiser sur une console centrale les messages envoyés à partir de trois PC distincts ou bien de partager une imprimante série. Rappelons qu'un PC peut facilement remplir le rôle d'un terminal télétype moyennant l'usage d'un programme approprié (par exemple HYPERTERMINAL de WINDOWS).

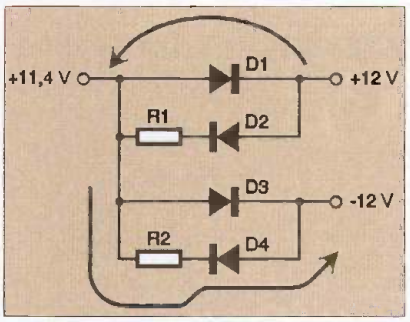
Comme vous pouvez le constater, les schémas sont particulièrement simples puisque tout le fonctionnement repose sur les propriétés des diodes. On ne peut donc pas faire plus simple. L'idée consiste à distribuer librement les signaux émis par le port principal tandis que les signaux reçus passent par un commutateur à diodes, au travers de résistances de protection.

Les drivers de lignes qui pilotent les signaux émis par le port principal devront donc piloter plusieurs ports simultanément, ce qui ne pose généralement pas de problème avec un PC standard. Les drivers de lignes dont il est équipé sont en général capable de fournir +/- 100 mA. Par contre, cela n'est pas toujours vrai pour les ports série des ordinateurs portables. En cas de problème de communication, il faudra penser à vérifier les caractéristiques exactes des drivers de lignes utilisés par vos appareils.

Les signaux positifs sont transmis directement au travers d'une diode câblée dans le sens passant, tandis que les signaux négatifs sont transmis via une résistance de protection. Lorsque plusieurs ports RS232 tentent d'imposer un état différent sur la ligne, le signal résultant restera au niveau haut . En effet, eu égard à la valeur relativement élevée des résistances (voir la **figure 3**), c'est bien l'état haut qui l'emporte.

Il est intéressant de noter que les signaux transmis par les lignes physiques d'un port série sont inversés par rapport aux signaux utilisés par l'uart de votre PC. Au repos, les lignes de l'uart sont au niveau haut, de sorte que les lignes correspondantes sur la liaison RS232 sont au niveau bas (entre –9 V et –15 V). Avec les schémas retenus, lorsque tous les ports secondaires sont au repos, les lignes correspondantes seront bien à l'état bas. Si un des ports secondaires souhaite transmettre des données, il sera alors en mesure de transmettre les états hauts au moment approprié, de sorte que les signaux arriveront sur le port principal sans qu'il soit nécessaire de désigner lequel des ports secondaires est actif. La sélection est donc automatique.

Bien entendu, cela pose un problème lorsque



3 *La valeur élevée des résistances impose un état "haut"*

plusieurs ports secondaires tentent de transmettre des données en même temps car les états hauts sont prioritaires sur les états bas. Si cela ce produit, les données reçues seront erronées. Ceci n'est pas toujours un problème gênant. N'oublions pas que le but des montages que nous vous proposons est de permettre à un utilisateur de transmettre des données à partir de plusieurs équipements, sans se soucier de choisir le port série actif. Cela signifie qu'à un instant précis, l'utilisateur n'utilise qu'un seul port secondaire. Bien entendu, si vous souhaitez utiliser ce montage pour monter un mini réseau afin de réaliser des transferts de façon automatique vous aurez sûrement des soucis. C'est cependant possible si vous prenez la peine de mettre au point un protocole qui désigne à chaque instant quel est le port secondaire qui a la parole.

Réalisation

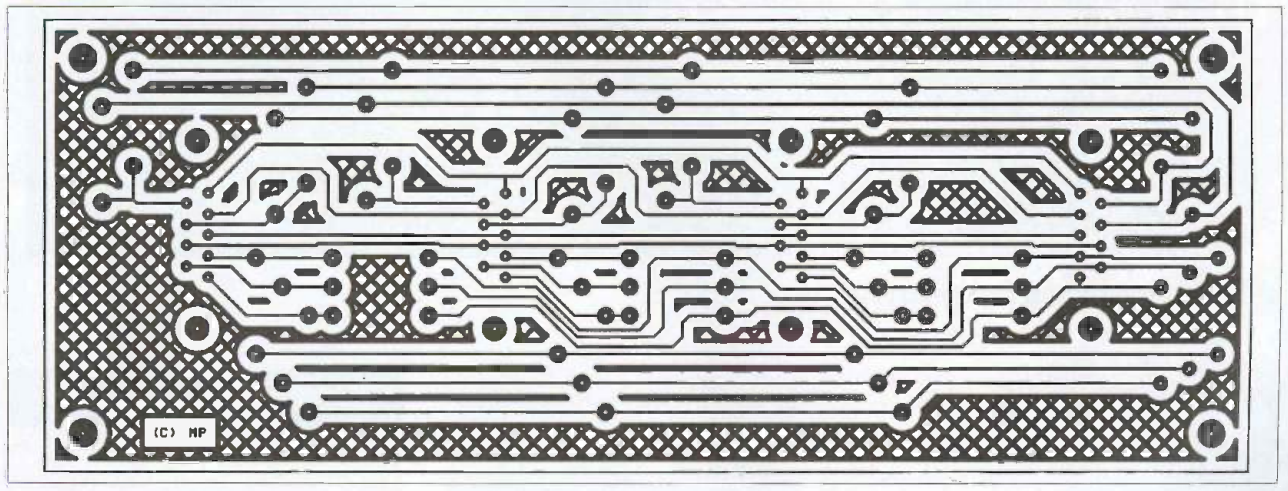
La réalisation du montage nécessite deux circuits imprimés de dimensions raisonnables. Le dessin du circuit imprimé de la carte un PC vers trois terminaux (1 DTE vers 3 DCE) est reproduit en **figure 4**. La vue d'implantation associée est reproduite en **figure 5**. Le dessin du circuit imprimé de la carte un terminal vers trois PC (1 DCE vers 3 DTE) est reproduit en **figure 6**. La vue d'implantation correspondante est reproduite en **figure 7**.

Les pastilles seront percées à l'aide d'un foret de 0,8 mm de diamètre pour la plupart. En ce qui concerne la fixation des connecteurs, n'oubliez pas de percer les trous de fixation des vis avec un foret de 3,5 mm de diamètre. Soyez très attentifs aux types des connecteurs (mâle ou femelle).

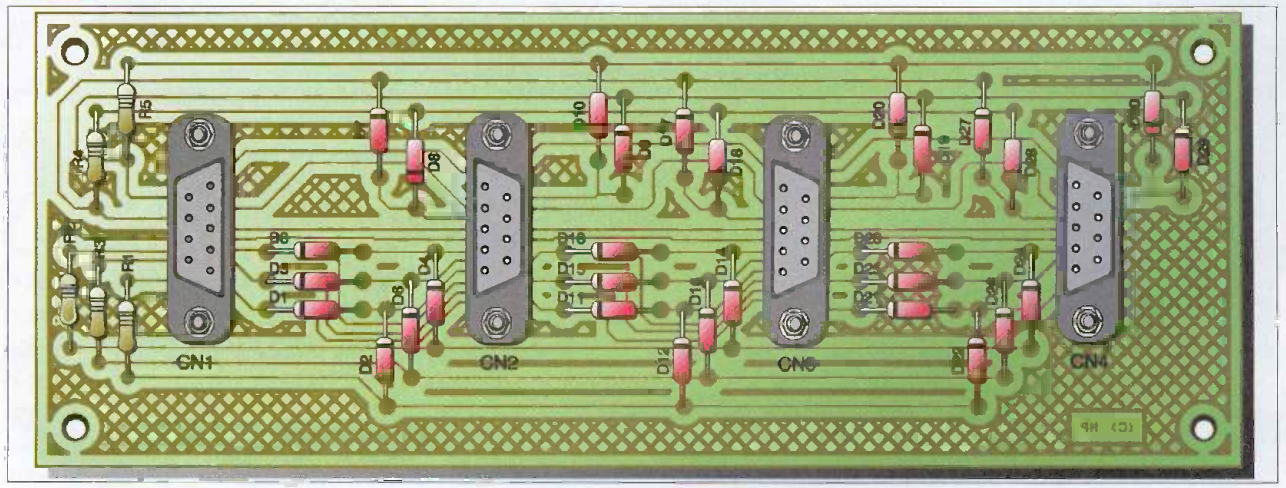
Au moment de connecter vos câbles pour la première fois, vous vous demanderez sûrement où brancher vos prises. Ce n'est pas toujours aussi simple qu'on pourrait le penser. Si votre équipement respecte les définitions des équipements DTE ou DCE ce devrait être assez facile. Sinon vous devrez déterminer vous-mêmes le rôle des signaux à l'aide d'une " jonction éclatée ". Cet appareil permet de visualiser l'état des signaux d'une liaison RS232, ce qui est utile pour trouver l'origine des conflits.

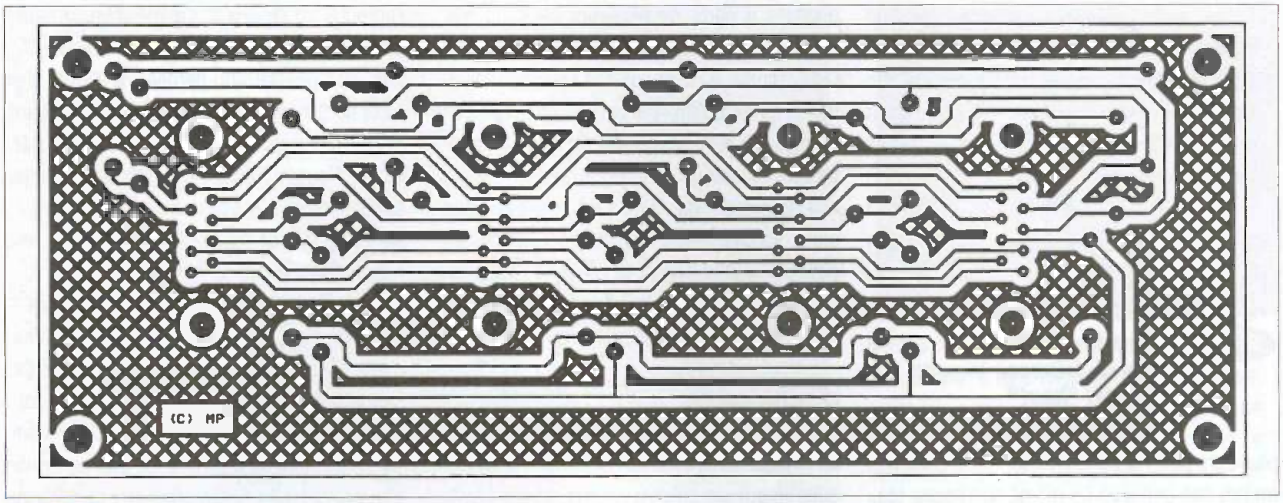
P. MORIN

4 *Circuit de la carte " un PC vers trois terminaux "*

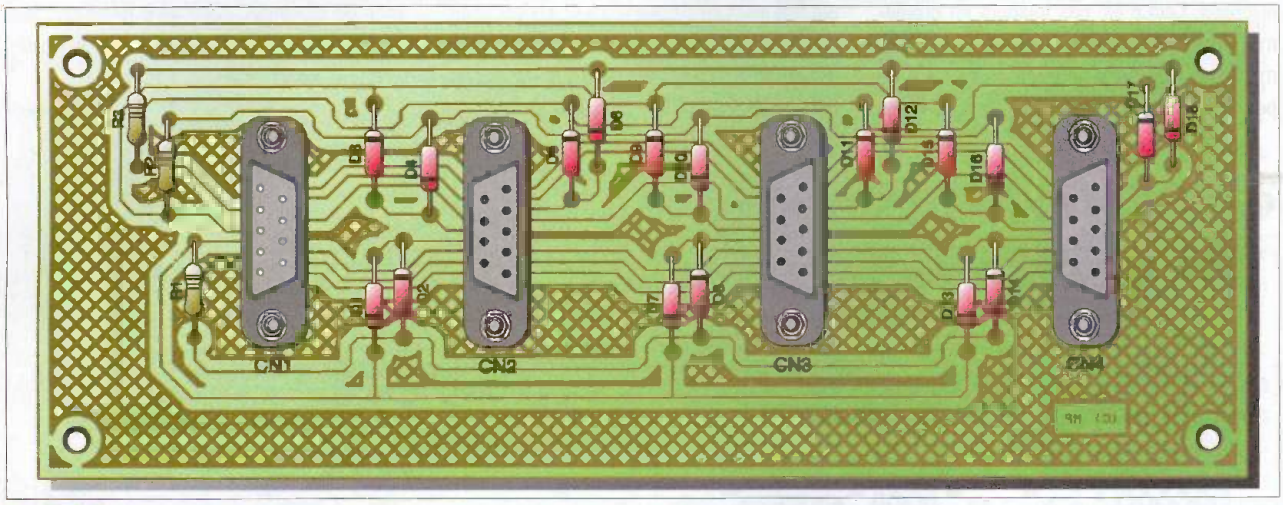


5 *Mise en place des composants. Attention à l'orientation des diodes !*

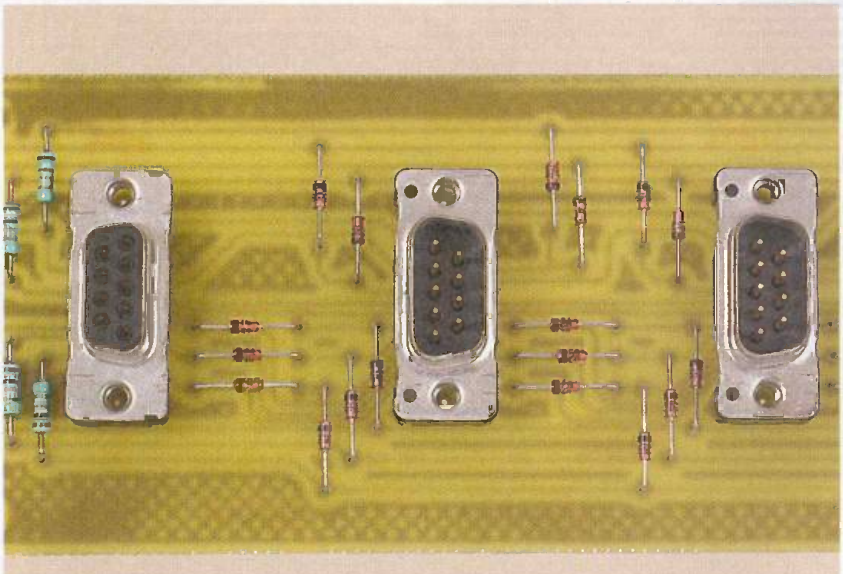




6 *Circuit de la carte " un terminal vers trois PC "*



7 *Implantation des composants*



Utilisation de connecteurs SUB-D 9 points mâle et femelle

Nomenclature

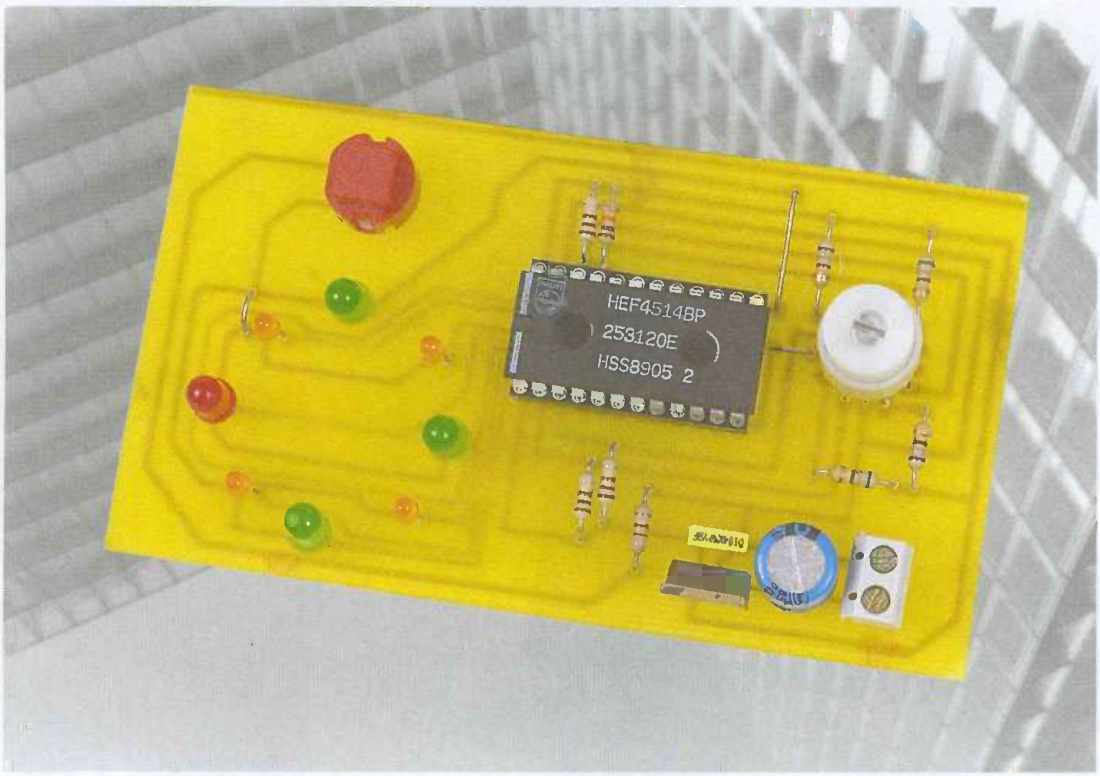
Carte 1 DTE vers 3 DCE :

CN1 : Connecteur Sub-D, 9 points, femelle, sorties droites, à souder sur circuit imprimé.
CN2, CN3, CN4 : Connecteur Sub-D, 9 points, mâle, sorties droites, à souder sur circuit imprimé.
D1 à D30 : 1N4148
R1, R2, R3, R4, R5 : 10 kΩ 1/4 W 5% (Marron, Noir, Orange)

Carte 1 DCE 3 DTE :

CN1 : Connecteur Sub-D, 9 points, mâle, sorties droites, à souder sur circuit imprimé.
CN2, CN3, CN4 : Connecteur Sub-D, 9 points, femelle, sorties droites, à souder sur circuit imprimé.
D1 à D18 : 1N4148
R1, R2, R3 : 10 kΩ 1/4 W 5% (Marron, Noir, Orange)

Un mini-compas digital



En exploitant un simple petit capteur magnétique comportant quatre détecteurs à effet Hall disposés en quadrature, il est fort aisé de visualiser les quatre points cardinaux habituels, c'est à dire NORD, EST, SUD et OUEST (N, E, S, O). À l'aide d'une poignée supplémentaire de composants, il est en outre possible de décoder les quatre positions intermédiaires, à savoir NE, SE, SO et NO.

Cette réalisation pourra prendre place à bord d'un véhicule automobile ou vous accompagner lors de vos balades. Si elle ne remplace pas le dispositif GPS, elle saura tout de même vous guider fort utilement.

Principe du montage

Le capteur utilisé est proposé à la vente, notamment par le distributeur LEXTRONIC chez lequel il porte la référence CPS 9410 (*www.lextronic.fr*).

La très petite taille du composant, approchant celle d'un dé à coudre, cache une conception fort astucieuse. Nous vous déconseillons toutefois de céder à la curiosité et de démonter la vis-pivot du dessus, d'ailleurs bloquée au vernis.

L'axe aimanté vertical est monté sur deux pivots de précision et il est entouré de quatre capteurs à effet Hall ultra sensibles.

Sa position de montage se doit d'être parfaitement verticale, avec toutefois une tolérance d'inclinaison de 12° pour un résultat acceptable.

Le déplacement en rotation de l'axe magnétique vertical ne prendra que quelques secondes, sans aucun battement nuisible au décodage.

Les sorties utiles sont de type à collecteur ouvert, avec transistor NPN. Si une sortie peut débiter jusqu'à 25 mA, il n'est guère conseillé de piloter directement la bobine d'un relais.

Bien entendu, un champ magnétique proche trop important (plus de 1000 gauss !) perturbera les indications du minuscule capteur. Une tension d'alimentation stabilisée entre 6 et 18 volts est nécessaire, avec un strict respect des polarités sur les douze broches du capteur (**figure 4**).

Analyse du schéma électronique

Il se résume à fort peu de choses et vous est proposé en **figure 1**. Nous vous signalons tout d'abord que le capteur en question ne dispose pas de repérage pour sa mise en place. Ainsi, il est probable qu'une fois soudé, il sera dans une position qui ne vous conviendra pas, par exemple pour la direction du NORD. Une led de couleur différente pourra être implantée après contrôle. Veillez toutefois à ne pas provoquer d'excès de chaleur lors des opérations de soudage... ou de dessoudage!

L'alimentation continue nécessaire est obtenue à partir d'une simple pile

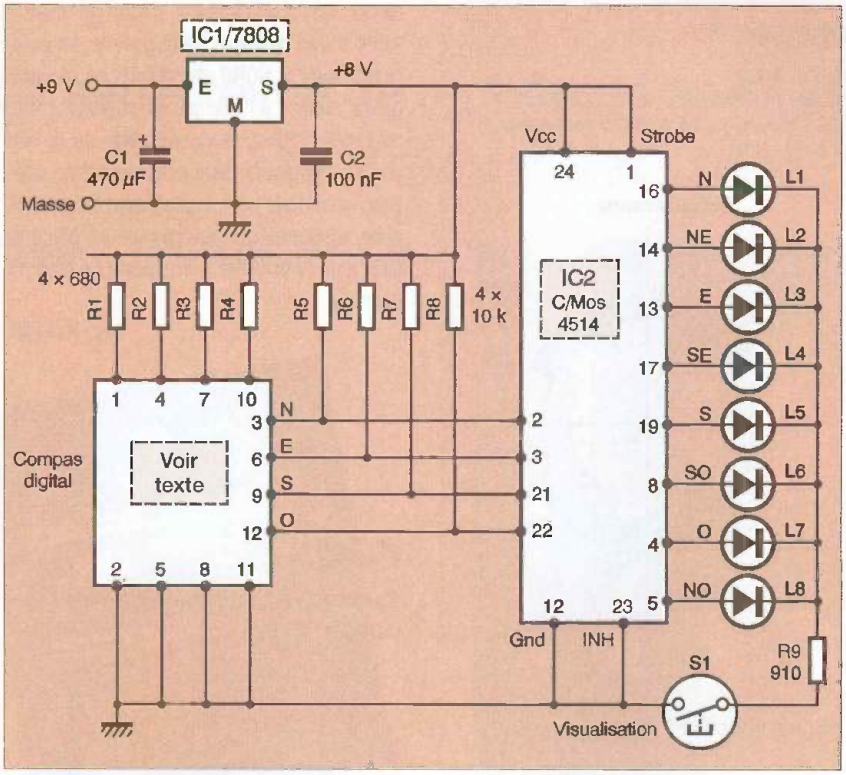


Schéma de principe

Nomenclature

Semi-conducteurs :

IC1 = régulateur intégré 8 volts positif, 7808, boîtier TO 220

IC2 = décodeur binaire 4 vers 16, C/MOS 4514, DIL 24

L1 à L8 = diodes leds 3 ou 5 mm, couleurs à définir

Capteur digital PEWATRON ou mini-compas type CPS 9410 chez LEXTRONIC

Résistances (toutes valeurs 1/4 de watt) :

R1, R2, R3, R4 = 680 Ω

R5, R6, R7, R8 = 10 kΩ

R9 = 910 Ω

Condensateurs :

C1 = chimique vertical 470 µF/25 volts

C2 = plastique 22 à 100 nF

Divers :

Bloc de 2 bornes vissé soudé, pas de 5 mm

Poussoir à fermeture pour C.I.

Support à souder DIL 2 , broches tulipe

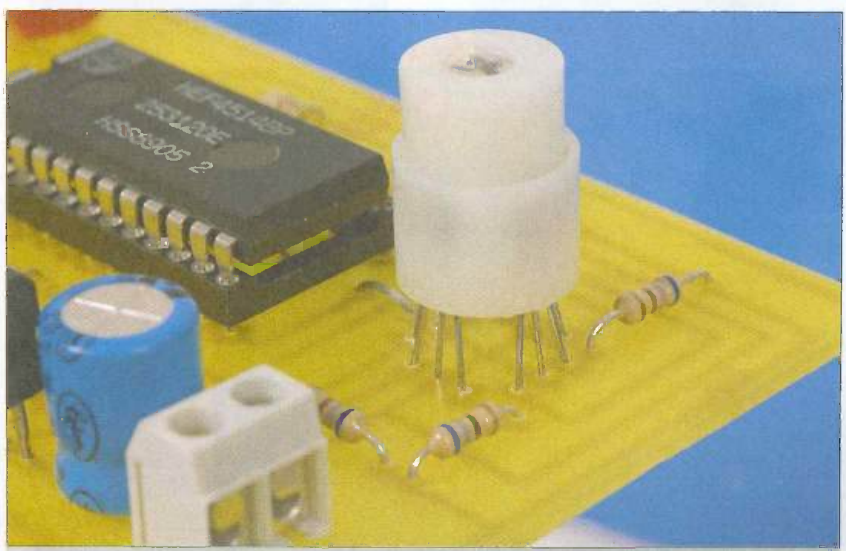
Coupleur pression pour pile 9 volts

de 9 volts, suivie d'un régulateur de type 7808 associé aux condensateurs de filtrage C1 et C2. Les broches reliées au + de l'alimentation sont dotées d'une résistance de faible valeur (R1 à R4), tandis que les broches 2, 5, 8 et 11 sont directement reliées à la masse.

Les quatre sorties utiles seront traitées par le circuit IC2, véritable décodeur binaire 4 vers 16.

En fait, huit directions seulement sont nécessaires ici, ce que détaille le tableau ci-dessus qui exploite le chevauchement des sorties du mini-compas :

DATA	O	S	E	N	Sorties	Broches
Ouest	0	1	1	1	S7	4
Est	1	1	0	1	S13	13
Nord	1	1	1	0	S14	16
Sud	1	0	1	1	S11	19
Sud-Ouest	0	0	1	1	S3	8
Sud-Est	1	0	0	1	S9	17
Nord-Est	1	1	0	0	S12	14
Nord-Ouest	0	1	1	0	S6	5
Broches >	22	21	3	2		



Le capteur digital PEWATRON ou mini-compas

Les huit leds des sorties seront disposées d'une manière pratique, avec des tailles et des couleurs à définir.

La seule résistance R9, à travers le poussoir S1 de visualisation, limite la consommation de l'affichage. Le fonctionnement est immédiat.

Réalisation

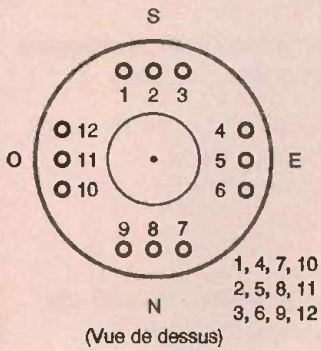
On trouvera sur la **figure 2** le tracé à l'échelle des pistes cuivrées proposées et sur la **figure 3** l'implantation des composants.

Il est bien entendu possible d'éloigner les

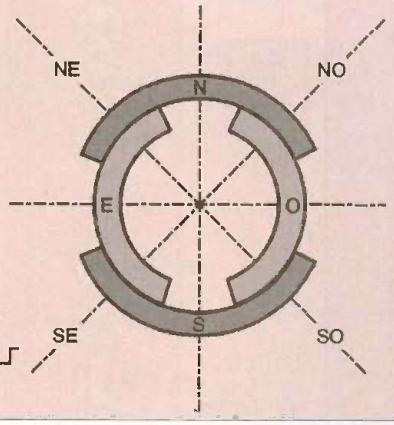
Compas digital type PW 6945-8

Constructeur : PEWATRON - AG Zurich
Distributeur : LEXTRONIC : Réf = CPS 9410
Alimentation : 5 - 20 Volts DC - 30 mA (8 à 13 Volts DC recommandé)
Sorties : 4 x collecteur ouvert NPN

Brochage : capteur vertical



Fonctionnement :

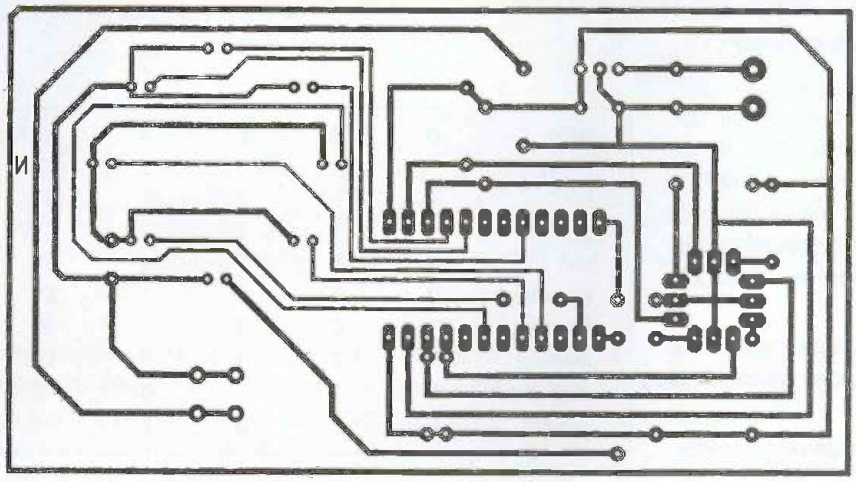


diodes leds d'affichage du circuit principal. À l'aide d'une boussole traditionnelle, on pourra retrouver le NORD et procéder au marquage du capteur à l'aide par exemple d'un trait de couleur. Ainsi, la visualisation de la rose des vents obtenue sera cohérente. Une position horizontale pour la plaquette est primordiale, plaquette qui pourra prendre place au fond d'un petit coffret translucide Heiland par exemple.

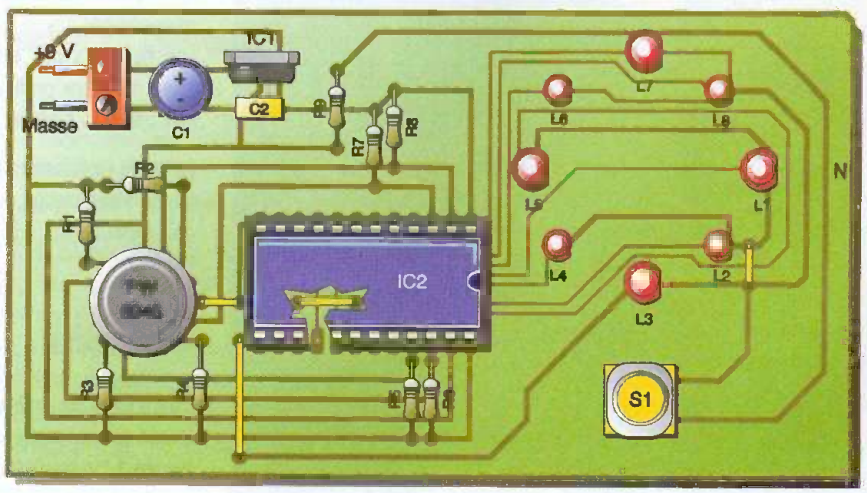
G. ISABEL

4 ***Caractéristiques principales du compas digital***

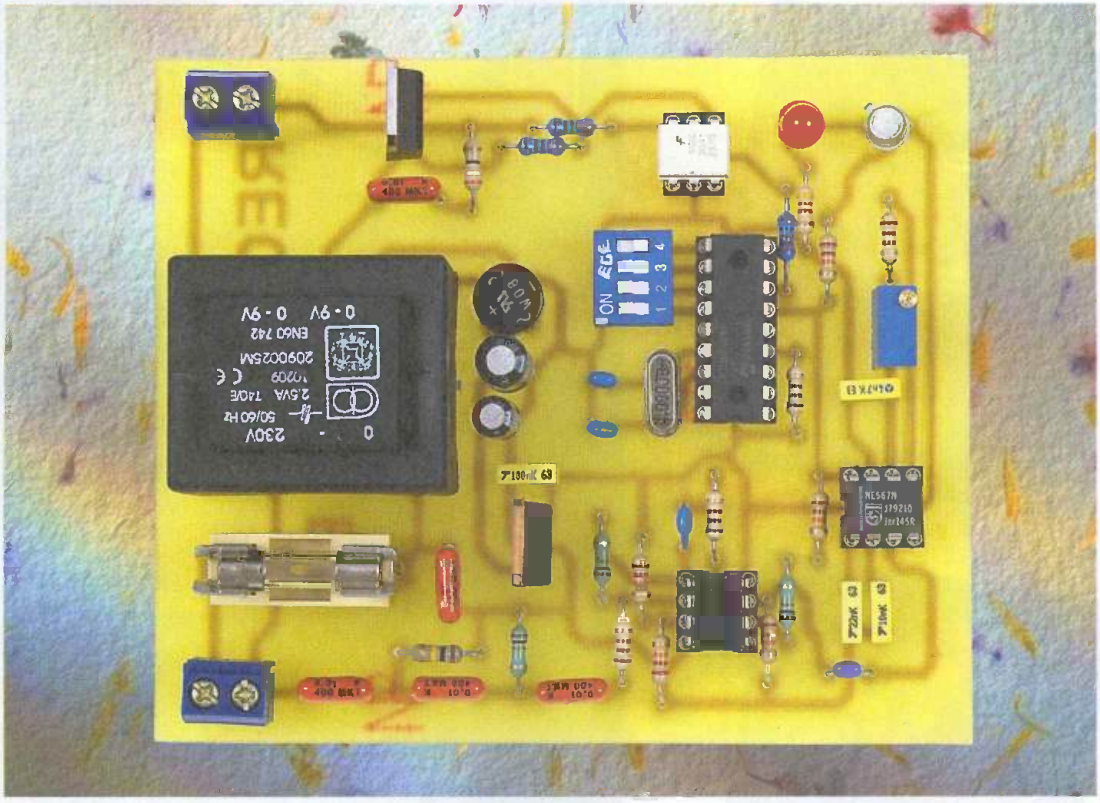
2 ***Tracé du circuit imprimé***



3 ***Implantation des éléments***



Télécommande à courants porteurs et PIC



Les microcontrôleurs sachant tout faire ou presque, de nombreux montages de télécommandes les ont déjà utilisés. Après les télécommandes par infrarouge ou UHF, nous vous proposons aujourd'hui une télécommande utilisant le réseau des fils du secteur cheminant dans chaque maison. Sans installer de câblage supplémentaire, un émetteur situé au grenier peut ainsi, par exemple, commander à la cave un récepteur sur lequel est branchée une lampe.

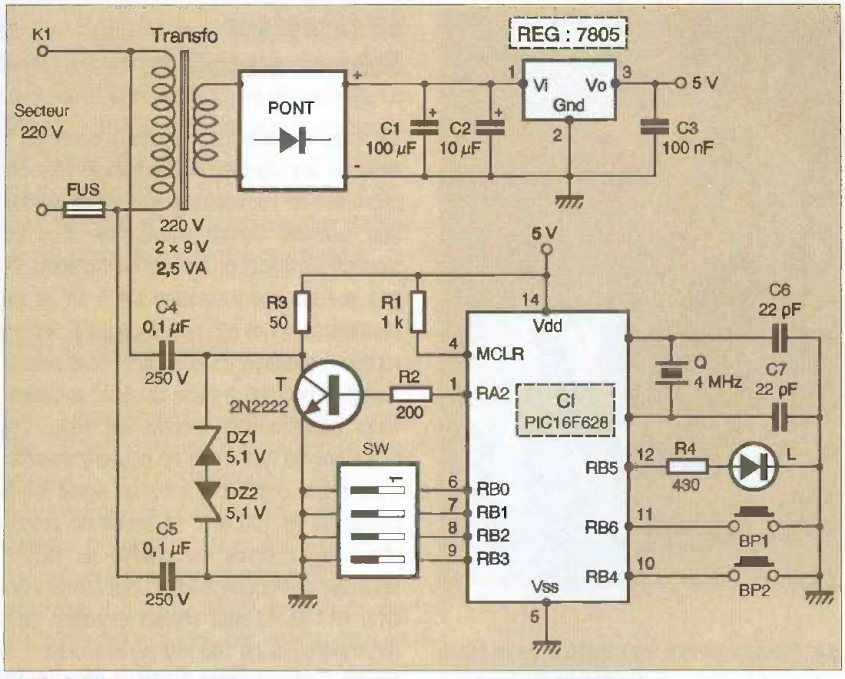
Pour ne pas changer une équipe qui gagne, les pièces maîtresses de l'émetteur et du récepteur sont encore des PIC, ici du type 16F628, cousin plus performant et moins cher que le célèbre 16F84.

Présentation générale

Avant d'aborder les schémas électriques, il reste encore quelques précisions à apporter pour terminer la présentation générale de cet ensemble de télécommande. Le principe de notre télécommande à courants porteurs consiste à superposer un courant de fréquence élevée, ici 100 kHz, au courant de fréquence 50 Hz véhiculé par les fils du secteur. Le module émetteur sera donc chargé de produire le courant HF et le module récepteur sera chargé de détecter ce courant HF. La liaison entre ces modules et le secteur est réalisée par un couplage simple capacitif qui possède la caractéristique de présenter une impédance élevée pour la basse fréquence du secteur mais une impédance faible pour le 100 kHz. Le nombre maximal de récepteurs a été fixé à 16 et chaque récepteur a une « adresse réseau » fixée par la position de 4 mini-interrupteurs. Chacun de ces récepteurs peut être commandé par un ou plusieurs émetteurs : les émetteurs sont en effet, de façon semblable, munis de 4 mini-interrupteurs permettant de désigner le récepteur à qui l'ordre est envoyé. Chaque récepteur commande un triac permettant l'alimentation de la charge à commander.

Les schémas d'un émetteur et d'un récepteur sont donnés **figures 1** et **2**. Ces schémas sont un peu frustrants car ils ne montrent pas l'essentiel, à savoir le programme contenu dans les microcontrôleurs.

Le programme mémorisé dans le PIC de l'émetteur doit en particulier générer une porteuse à 100 kHz, porteuse modulée en amplitude par les données à transmettre : adresse du récepteur et ordre d'extinction ou d'allumage. Le programme mémorisé dans le PIC du récepteur doit lui faire l'opération contraire : décoder l'adresse et l'ordre reçus afin de commander ou non le triac.



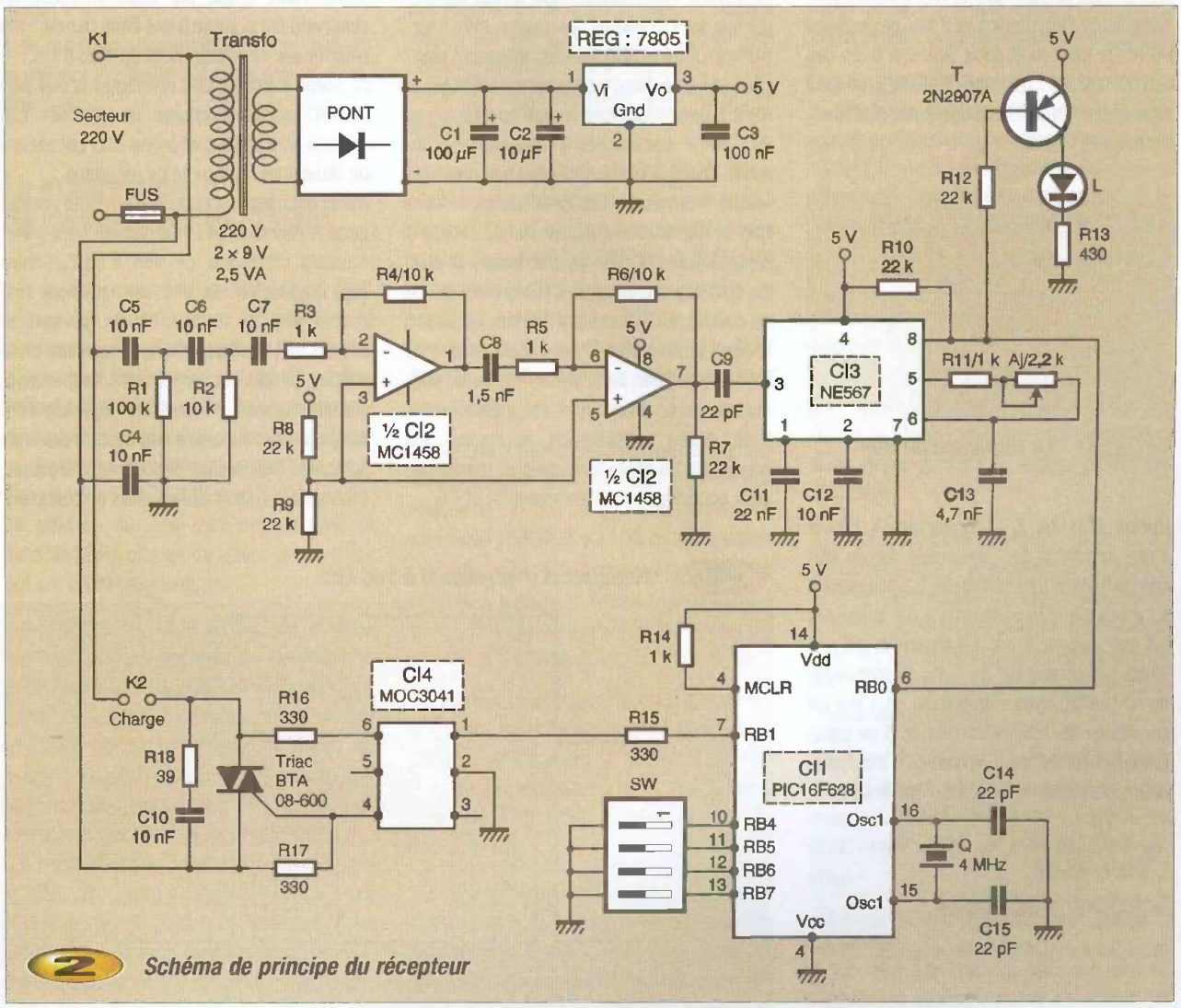
1 *Schéma de principe de l'émetteur*

Description de la trame émise sur les fils du secteur

L'appui sur un des 2 boutons-poussoirs BP1 ou BP2 (ARRET ou MARCHE) de l'émetteur entraîne une génération de données à transmettre sur les fils du secteur. Les données transmises sont semblables à celles qu'envoie n'importe quelle télécommande de matériel audiovisuel.

L'ensemble de ces données constitue une trame de bits, continuellement répétée tant qu'est maintenu l'appui sur le bouton.

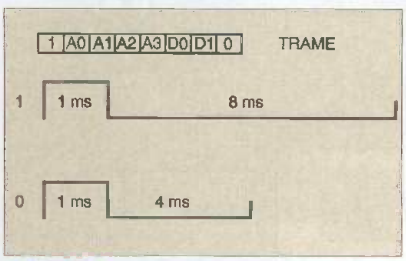
Dans l'ordre précisé sur le schéma de la **figure 3**, ces 7 bits sont d'abord le bit de démarrage toujours à 1, destiné à réveiller le récepteur et lui dire qu'une trame arrive, les 4 bits désignant l'adresse du récepteur à commander et les 2 bits de données D0 et D1 qui codent l'ordre MARCHE ou ARRET. A l'issue du dernier bit, une impulsion de 1 ms est



2 *Schéma de principe du récepteur*

Aspect d'un module émetteur

envoyée. Si on laisse son doigt appuyé sur le bouton, la même trame est envoyée environ 40 ms plus tard. Les 4 bits d'adresse A0 à A3 sont définis par la position des 4 mini-interrupteurs de l'émetteur. Les 2 bits de données D0 et D1 sont tous deux égaux à 0 en cas d'appui sur BP2 (bouton MARCHE) et tous deux égaux à 1 en cas d'appui sur BP1 (bouton ARRET).



3 *La séquence de bits*

L'envoi d'un bit à 1 correspond à l'envoi d'une impulsion de 1 ms suivie par un état bas de 8 ms et l'envoi d'un bit à 0 correspond à l'envoi d'une impulsion de 1 ms suivie par un état bas de 4 ms. La **figure 4** est une loupe sur l'impulsion de 1 ms. On remarque qu'en réalité, chaque impulsion de 1 ms est constituée de 100 impulsions de 5 µs espacées de 5 µs : notre impulsion de 1 ms module une porteuse de fréquence 100 kHz.

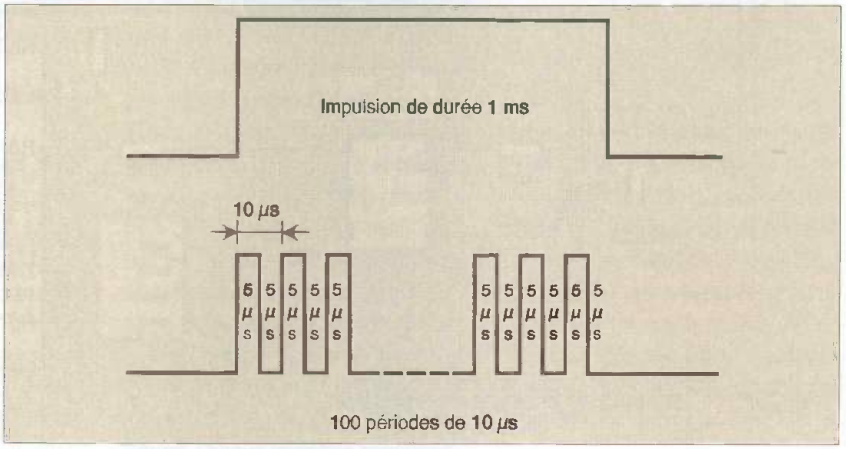
L'émetteur
Schéma électrique

Le rôle principal est bien sûr tenu par le PIC16F628. Le circuit d'horloge nécessaire à son fonctionnement est constitué du quartz à 4 MHz et de ses deux condensateurs associés C6 et C7. Les broches utilisées en entrées sont facilement identifiables : RB0 à RB3 sur lesquelles sont branchés les quatre mini-interrupteurs, RB4 et RB6 sur lesquelles sont branchés les deux boutons-poussoirs MARCHE et ARRET. Des résistances de pull-up internes au PIC "tirent" ces broches à l'état haut en l'absence d'appui ou quand elles ne sont pas reliées à la masse. Les deux autres broches RB5 et RA2 sont utilisées en sorties : RB5 est reliée à une LED témoin d'émission et c'est par RA2 que sort la trame à envoyer sur les fils du secteur. Le signal à transmettre parvient à la base du transistor T par l'intermédiaire de R2. De type NPN, ce transistor est rendu passant quand un niveau haut est présent sur sa base. Après amplification, le courant est envoyé sur les lignes du secteur au travers des deux condensateurs d'isolement C4 et C5.

Le récepteur
Schéma électrique

Le récepteur a trois missions à accomplir : détecter les signaux de fréquence 100 kHz provenant de l'émetteur, décoder ces signaux puis exécuter l'ordre commandé. Tous les signaux provenant du secteur sont d'abord filtrés par les condensateurs C4 à C7 et les résistances R1 et R2. Les signaux HF qui ont réussi le passage de ces filtres sont ensuite amplifiés par 100 à l'aide de deux amplificateurs opérationnels montés en série. Ces ampli-ops de type 1458 ne peuvent amplifier chacun que d'un facteur 10, un signal à une fréquence de 100 kHz. En sortie du second ampli-op, le signal est envoyé au NE567, décodeur de fréquence à PLL. Les composants C13, R11 et AJ sont choisis et réglés pour qu'en présence du 100 kHz sur la broche 3, la broche 8 normalement à l'état haut, passe à l'état bas. Dans notre cas, lors de la réception d'une trame émise par notre émetteur, on observera sur la sortie 8 des états bas de 1 ms séparés par des états hauts de 4 ou 8 ms.
La sortie 8 est ensuite connectée d'une part au PIC qui sera chargé de décoder ces signaux et d'autre part à une LED qui servira de témoin de réglage et de réception.
Après décodage, si l'adresse envoyée correspond à l'adresse affichée sur les mini-interrupteurs connectés de RB4 à RB7, l'ordre reçu est exécuté. Le triac est activé par l'intermédiaire du MOC3041 en portant la broche RB1 à l'état haut ou désactivée en la plaçant à l'état bas. Le MOC3041 est un petit circuit intégré d'interface, spécialement conçu pour commander des triacs à partir de systèmes fournissant des signaux logiques. Ce circuit construit autour d'un optocoupleur

4 *Modulation d'amplitude à 100 kHz*



et d'un détecteur de passage à zéro de la tension secteur offre une isolation galvanique de 7 500 V et ne génère pas de parasite. Le triac référencé dans la liste des composants est du type BTA 08-600 qui peut fonctionner sous une tension de 600 V et débiter 8 A : vous avez bien sûr, toute latitude d'en choisir un autre suivant la charge à commander.

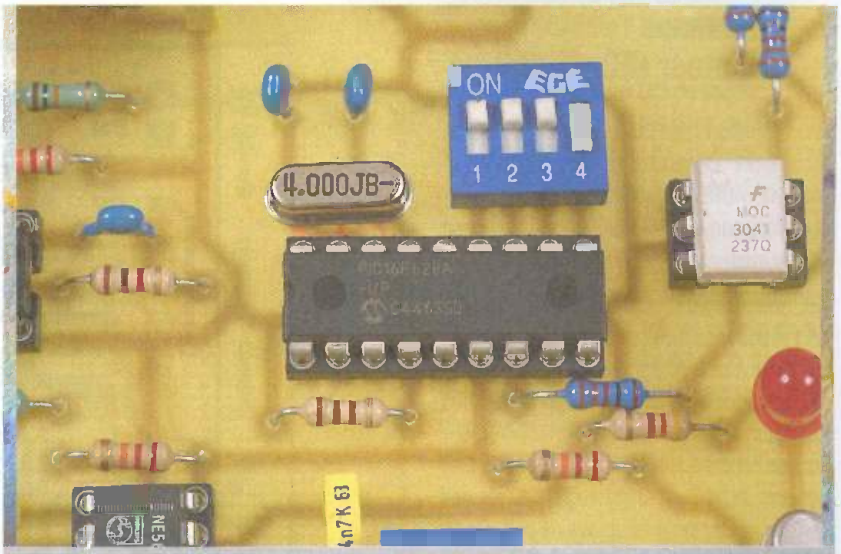
Les programmes

Les programmes EME_CP.BAS et REC_CP.BAS sont disponibles sur le site Internet d'ELECTRONIQUE PRATIQUE (*www.electroniquepratique.com*) sous deux formes : la première est le listing en BASIC F84 présenté dans cet article mais facilement adaptable à d'autres BASIC, la seconde est son fichier hexadécimal. Les lecteurs ne possédant pas le BASIC pourront ainsi charger directement le fichier hexadécimal à partir d'un des programmateurs proposés par la revue.
Les quelques commentaires qui suivent expliquent le rôle de chaque partie des programmes.

Le programme de l'émetteur EME_CP.BAS

;(1) définition des variables et tableaux. Pour utiliser une variable ou un tableau dans le programme, il faut les déclarer en tête de programme. Pour rendre plus facile la compréhension du programme, il est recommandé de baptiser les variables les plus intéressantes par un nom en rapport avec leur fonction : la variable ADRESSE représente l'adresse affichée sur les mini-interrupteurs, la variable APPUI change de valeur en cas d'appui sur un des boutons, etc....

;(2) initialisation. L'initialisation de certains registres, dont les registres de direction des ports, est primordiale. **TRISA=0 met par défaut toutes les broches du port A en sortie, et TRISB=%01011111** configure toutes les broches du port B en entrée sauf RB5 et RB7. **BCF OPTION_REG,7** permet d'activer les résistances de PULL UP avec le bit 7 (RBPU) à 0. Toutes ces opérations sont faites avec le bit RP0 du registre STATUS à 1 puisque ces registres sont en page 1. À l'issue, on remet RP0 à 0. On termine l'initialisation avec **BCF PORTA,2** qui a pour effet de bloquer le transistor T.



En situation le PIC et l'optocoupleur du récepteur

;(3) le programme principal. Le programme principal aura 2 rôles : contrôler l'appui éventuel sur un des 2 boutons et envoyer la trame correspondante si c'est le cas. Si aucune touche n'a été appuyée, pour limiter une consommation de courant inutile, le PIC est placé en mode sommeil (ou SLEEP en anglais).

Pour qu'il puisse se réveiller, quelques dispositions sont prises dont la plus évidente : lui dire qu'il doit se réveiller si on appuie sur une touche. C'est l'objet de **INTCON=%00001000** qui met le bit RBIE à 1. Afin de détecter un appui sur n'importe quel bouton, toutes les lignes du port B sont mises à 0 avec le **CLRF PORTB**. Le PIC est ensuite placé en sommeil avec **SLEEP**.

Si l'un des 2 boutons est appuyé, le changement d'état sur une des lignes RB4 ou RB6 réveille le PIC qui exécute l'instruction suivante **BSF PORTB,5**. La LED de signalisation s'illumine et le restera jusqu' au relâchement du bouton. L'appel au sous-programme BOUTON permet de déterminer le bouton appuyé et donc la donnée à envoyer.

;(4) gestion des boutons. En cas d'appui sur le bouton BP2, RB4 passe à l'état bas et DONNEE est initialisée à 0 (les 2 bits D0 et D1 à 0). En cas d'appui sur BP1, RB6 passe à l'état bas et DONNEE est initialisée à 3 (les deux bits D0 et D1) à 1.

;(5) Un des boutons a été appuyé, la trame est alors envoyée. L'émission est conforme au schéma de la figure 3 : envoi du 1, envoi des quatre bits d'adresses (ADRESSE), envoi des deux bits de données (DONNEE), envoi d'une impulsion de 1 ms et rien pendant 40 ms. Remarquez l'utilisation de **RRF** : à l'issue d'une rotation, le bit C du registre STATUS prend la valeur du bit 0 pour une rotation à droite et la valeur du bit 7 pour une rotation à gauche.

La valeur de C indique donc si ce bit était égal à 0 ou 1. Cette valeur détermine alors l'utilisation des sous-programmes ENVOI_0 ou ENVOI_1. La dernière impulsion de 1 ms et l'intervalle entre deux impulsions de 40 ms est générée par un 0 (impulsion + 4 ms) suivi d'une attente sans émission de 9 x 4 ms.

;(6) sous-programme d'envoi d'un 1. Ici, on applique le schéma de la figure 4. Le 1 correspondant à 100 impulsions (bit RA2 à 1) de 5 µs séparées par des intervalles de 5 µs (bit RA2 à 0). Puis le bit RA2 reste à 0 pendant 8 ms, durée générée par le sous-programme CYCLE8m.

;(7) sous-programme d'envoi d'un 0. Là aussi, on applique le schéma de la figure 4. Le 0 correspondant à 100 impulsions (bit RA2 à 1) de 5 µs séparées par des intervalles de 5 µs (bit RA2 à 0). Puis le bit RA2 reste à 0 pendant 4 ms, durée générée par le sous-programme CYCLE4m.

;(8) durée de 4000 cycles. On peut calculer le nombre de cycles de cette boucle avec la formule (3xVB2+4)xVB1+2, soit ici (3x32+4)x40=4000. (on n'est pas à 2 cycles près).

;(9) durée de 8000 cycles. On peut calculer le nombre de cycles de cette boucle avec la formule (3xVB2+4)xVB1+2, soit ici (3x32+4)x80=8000.

Le programme du récepteur REC_CP.BAS

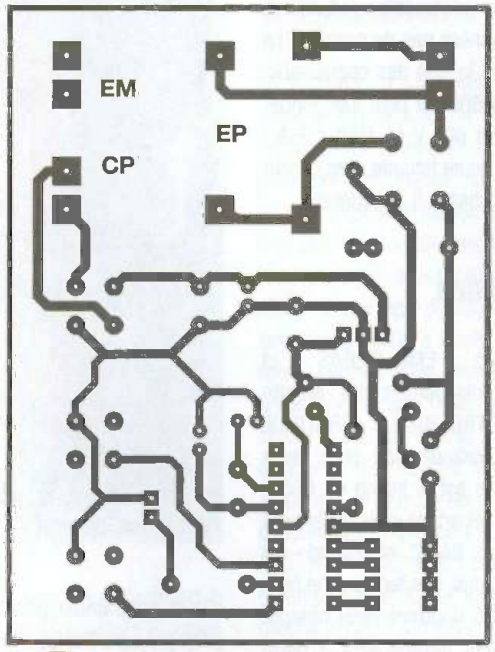
;(1) définition des variables et tableaux. Remarques identiques à EME_CP.BAS. Par exemple, ADRESSE est envoyée par l'émetteur alors que AD_REC est l'adresse affichée sur les mini-interrupteurs du récepteur.

;(2) initialisation. **TRISA=0 et TRISB=%11110001** configurent toutes les broches des ports A et B en sortie sauf les 4 broches du port B sur lesquelles sont connectées les mini-interrupteurs et la broche RB0/INT, sur laquelle est connecté le NE567. Comme tout à l'heure, le bit 7 du registre OPTION est mis à 0 pour utiliser les résistances de pull-up. Le bit 6 (INTEDG) est lui aussi mis à 0. Après sa mise en sommeil, le réveil du PIC est prévu sur un front descendant de RB0. Toutes ces opérations sont faites avec le bit RP0 du registre STATUS à 1 puisque ces registres sont en page 1. À l'issue, on remet RP0 à 0. L'initialisation se termine par la lecture du port B et la détermination de AD_REC.

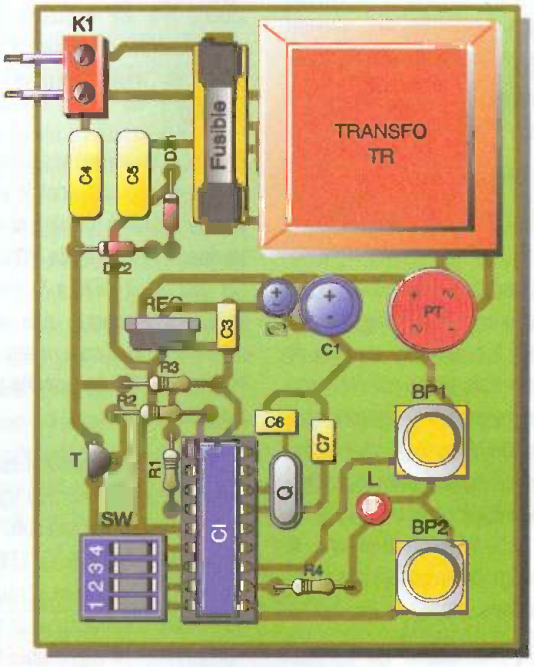
;(3) le programme principal. Le programme principal commence par le placement du PIC en mode SOMMEIL! Mais comme tout à l'heure, on a pris des dispositions pour qu'il se réveille si une trame est détectée par le NE567 et donc par un changement d'état (front descendant) du bit RB0. Ces dispositions sont écrites dans la ligne INTCON=%00010000 (INTE=1) et précisées plus haut par INTEDG à 0.

;(4) décodage des cinq premiers bits. Le PIC est réveillé car il a perçu un changement d'état sur sa ligne RB0, il va donc analyser les bits qui arrivent. Pour savoir si le bit envoyé est un 0 ou un 1, on mesure la durée de l'état haut, qui est aussi la durée intermédiaire entre deux états bas. La variable ADRESSE qui contiendra la valeur des cinq premiers bits reçus est mise à 0. Comme le PIC vient de se réveiller, c'est que sa broche RB0 vient de passer à l'état bas (il vient de recevoir le début du premier bit à 1 et les signaux sont inversés par le NE567). Comme cet état à 0 ne nous intéresse pas, on attend qu'il se termine avec la boucle TEST_A-GOTO TEST_A. Dès que le signal passe à l'état haut, on démarre un compteur qui s'incrémente toutes les 100 ms et qui s'arrêtera dès que le signal repassera à l'état bas. Pour un 0, cet état haut doit durer 8000 ms : le compteur doit donc marquer 80 dans ce cas. Pour un 1, cet état haut doit durer 4000 ms : le compteur doit donc marquer 40 dans ce cas. Si le compteur est supérieur à 80, c'est donc que l'état 1 dure plus longtemps que prévu et c'est le cas quand il n'y a plus d'émission de trame. Dans notre programme de réception, où tout est prévu large, on indique au récepteur que s'il n'a reçu aucun signal depuis 20 ms, il doit retourner au début du programme principal pour se mettre en sommeil. Avec la même approximation (mais qui marche très bien), on indique au récepteur que si le compteur est inférieur à 60, c'est que c'est un 0 et que si la durée est supérieur à 60, c'est que c'est un 1.

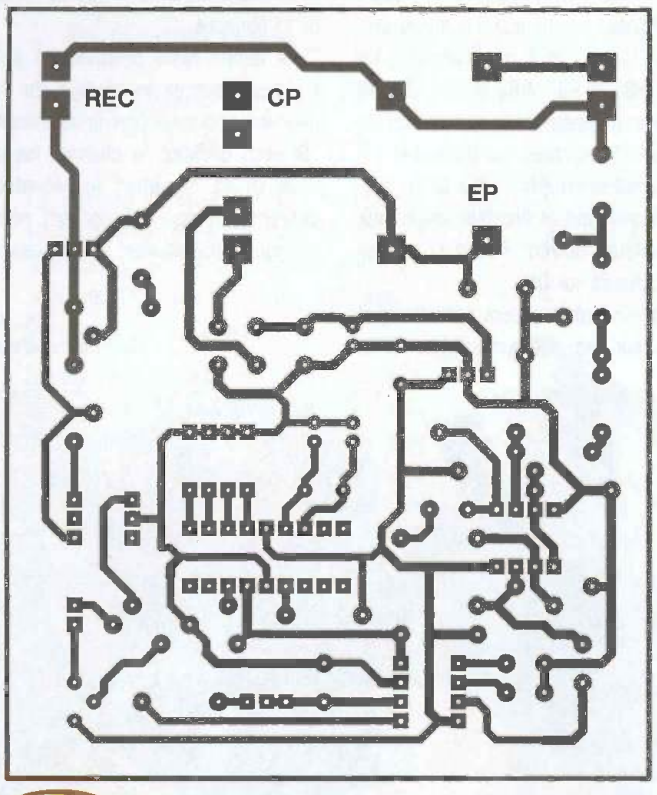
On aurait pu faire des tests plus précis mais ça ne sert à rien. Une fois la valeur du bit déterminée, on place cette valeur dans le bit C du STATUS, on fait une rotation à droite de ADRESSE. Cette opération est répétée cinq fois pour les cinq premiers bits reçus. Le bit



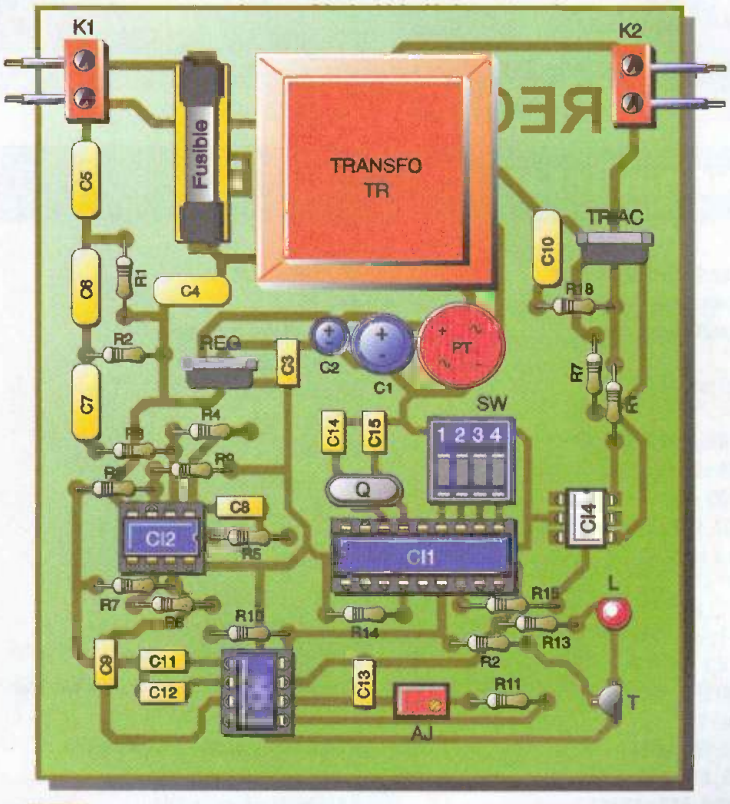
5 *Tracé du circuit imprimé de l'émetteur*



6 *Implantation des éléments*



7 *Tracé du circuit imprimé du récepteur*



8 *Implantation des éléments*

3 de ADRESSE contient donc le 1 du début d'émission et les bits 7 à 4 la valeur des mini-interrupteurs.

;(5) décodage des deux bits de données. De la même façon, on traite les deux derniers bits reçus qui correspondent à la donnée envoyée et qui sont stockés dans la variable DONNEE.

;(6) extraction des quatre bits d'adresse. Après le SWAPF, la valeur des quatre mini-interrupteurs de l'émetteur correspondant maintenant aux bits 3 à 0 de ADRESSE est extraite en faisant un **AND** entre ADRESSE et.15 (la valeur décimale de 15 correspond à %00001111 en notation binaire, le AND ne conserve que les quatre bits de poids faible de ADRESSE)

;(7) Comparaison de l'adresse reçue et de l'adresse du récepteur: si l'adresse ne correspond pas, on ne regarde même pas la donnée envoyée, on retourne directement au début de programme.

;(8) bonne adresse : On active ou désactive le triac suivant l'ordre reçu. Dans le premier cas, DONNEE=0, on a donc appuyé sur le BP2 (MARCHE), on demande au PIC de mettre RB1 à 1. Dans le cas contraire, on a reçu DONNEE=3 et RB1 est mis à 0.

;(9) données traitées, on repart au début. C'est la fin de la boucle principale. On retourne au début.

;(10) pause de 100ms. sous-programme utilisé pour incrémenter le compteur de durée toutes les 100 ms.

Réalisation et réglage

Le circuit imprimé de l'émetteur est présenté **figure 5**. Les composants seront implantés en respectant le dessin de la **figure 6**. A priori, il n'y a pas de difficultés particulières. Bien sûr, on veillera comme d'habitude à respecter la bonne orientation du support, du transistor, du régulateur, des condensateurs polarisés et de la LED.

Le circuit imprimé de la carte récepteur est présenté **figure 7** et son schéma d'implantation est présenté **figure 8**. Là aussi, on veillera à respecter la bonne orientation des composants.

Avant la mise en coffret définitive, les montages seront provisoirement terminés pour entamer la phase de réglage: on procédera donc à la mise en place d'un câble deux conducteurs muni d'une prise mâle sur chacun des connecteurs K1 de l'émetteur et du récepteur et d'une lampe test sur le connec-

teur K2 du récepteur. Le réglage consiste à positionner précisément l'ajustable AJ du récepteur pour que le NE567 détecte les émissions à la fréquence de 100 kHz. Avant réglage, l'ajustable pourra être pré-positionné aux environs de 1 kΩ. Il n'est pas nécessaire d'afficher les mêmes adresses sur l'émetteur et le récepteur mais celles-ci pourront être par exemple initialisées à 0 : pour cela, tous les interrupteurs de l'émetteur et du récepteur seront placés sur ON (donc reliés à la masse). Connectez les deux montages sur deux des prises secteurs de votre maison.

Ces deux montages étant reliés au réseau 220 V, ce réglage devra être effectué avec beaucoup de précautions. Si l'émetteur fonctionne, sa LED témoin d'émission doit s'allumer en cas d'appui sur un des deux boutons BP1 ou BP2. Si c'est le cas, tout en maintenant l'appui sur un des boutons de l'émetteur, tourner délicatement et très lentement avec un petit tournevis la vis de réglage de l'ajustable du récepteur jusqu'à ce que la LED témoin de réception s'allume. Une fois la zone d'allumage déterminée, l'ajustable sera réglé en position de luminosité maximale.

Mise en œuvre et utilisation

Une fois réglés, la dernière chose à faire pour utiliser nos montages est de faire correspondre les valeurs des adresses affichées sur l'émetteur et le récepteur : il faut manipuler les huit mini-interrupteurs pour que l'état de la broche RB0 de l'émetteur soit le même que l'état de la broche RB4 du récepteur, de même pour RB1 et RB5, RB2 et RB6, RB3 et RB7. Par exemple, pour l'adresse 1, RB0 de l'émetteur et RB4 du récepteur doivent être à l'état haut (mini-interrupteurs sur OFF) tandis que toutes les autres broches reliées aux mini-interrupteurs doivent être à l'état bas (mini-interrupteurs sur ON).

Une fois les mini-interrupteurs correctement positionnés sur les deux montages et le réglage de l'ajustable effectué, un appui sur BP2 doit allumer la lampe et un appui sur BP1 l'éteindre.

Si la lampe reste obstinément allumée ou éteinte, reprenez le réglage de l'ajustable pour le positionner plus précisément.

Si vous décidez de changer les adresses, vous devez "reseter" le récepteur en le débranchant puis rebranchant pour que le changement d'adresse soit pris en compte.

A. REBOUX
alain.reboux@wanadoo.fr

Le régulateur ne dispose pas de dissipateur

Nomenclature

L'émetteur :

R1 : 1 kΩ
R2 : 200 Ω
R3 : 50 Ω/0,5 W
R4 : 430 Ω
C1 : 100 µF
C2 : 10 µF
C3 : 100 nF
C4, C5 : 100 nF/250 V MKT
C6, C7 : 22 pF
T : 2N2222
CI : PIC16F628
Q : quartz 4 MHz
L : LED
REG : régulateur 7805
Pont : Pont de diodes moulé 1 A
DZ1, DZ2 : diodes Zeners 5,1 V
BP1, BP2 : boutons-poussoirs
SW : interrupteur DIL 4 pôles
TR : transformateur 220 V / 2 x 9 V/2,5 VA
FUS : porte-fusible+fusible
K1: connecteur d'alimentation
1 support 18 broches

Le récepteur :

C1 : 100 µF
C2 : 10 µF
C3 : 100 nF
C4 à C7, C10 : 10 nF/400 V MKT
C8 : 1,5 nF
C9 : 22 pF
C11 : 22 nF
C12 : 10 nF
C13 : 4,7 nF
C14, C15: 22 pF
R1: 100 kΩ
R2, R4, R6 : 10 kΩ
R3, R5, R11, R14 : 1 kΩ
R7 à R10, R12 : 22 kΩ
R13 : 430 Ω
R15 à R17 : 330 Ω
R18: 39 Ω
AJ : ajustable multitours 2,2 kΩ
T : 2N2907A
CI1 : PIC16F628
CI2 : MC1458
CI3 : NE567
CI4 : MOC3041
Q : quartz 4 MHz
L : LED
REG : régulateur 7805
TRIAC : BTA08-600
Pont : Pont de diodes moulé 1 A
SW : interrupteur DIL 4 pôles
TR : transformateur 220 V / 2 x 9 V/2,5 VA
FUS : porte-fusible+fusible
K1, K2 : connecteurs d'alimentation
1 support 18 broches
2 supports 8 broches
1 support 6 broches

Opti-machines

La Qualité Professionnelle au service des Particuliers

SCIE QUANTUM S121 G D'OPTI-MACHINES

La société OPTI-MACHINES représente en France et en exclusivité la marque allemande OPTIMUM réputée pour ses produits de qualité (tours, fraiseuses, perceuses, scies, tourets,...).

Si vous recherchez une scie à ruban performante et de précision pour le travail des métaux, OPTI-MACHINES propose aujourd'hui la Quantum S 121 G au prix de 249 € HT offrant un exceptionnel rapport qualité/prix.

La Quantum S 121 G est une scie à ruban, conçue pour le travail des métaux. Son arc de scie pivotant permet la coupe en équerre. Elle possède un arrêt de coupe automatique. Le guidage de la lame se fait par roulements 3 points. Le réglage est simple. Elle bénéficie de trois vitesses de coupe (21/31/54 m/min). Elle est proposée avec un équipement complet : une lame bi-métal, un étau à serrage rapide et un socle de série.

Sa haute précision de coupe lui offre de multiples possibilités. D'une capacité moteur de 370 W, cette scie est idéale pour les passionnés du travail du métal. Ses capacités de coupe (115 mm dans le rond ; 100 X 150 mm dans le plat) en font une machine très aboutie dans sa catégorie. Ainsi, la scie Quantum S 121 G d'Opti-machines saura répondre à l'attente de tous ceux qui veulent par exemple réaliser leurs propres châssis ou coffrets spécifiques.

Vendue dans toute la France, cette machine aux normes C.E., est extrêmement performante et se révélera très vite indispensable.

Cette scie est distribuée en exclusivité par :

OPTI-MACHINES – Parc d'Activités du Vert Bois – Rue Jean-Baptiste Lebas - 59910 BONDUES.
Tél. : 03.20.03.69.17
Fax : 03.20.03.77.08

A signaler que les catalogues Machines-Outils (Optimum-Quantum 2004/2005) et Outillage-Accessoires (196 pages couleurs et tarifs) sont disponibles contre 10 timbres à 0.5 euros ou un chèque de 5 euros remboursés au premier achat

Site Internet : *www.optimachines.com*

L'INTERFACE PARALLÈLE DU PC

Par Patrice Oguic

L'Interface parallèle du pc décrit de manière fort complète la constitution des différents ports parallèles (ou ports imprimantes) qui équipent les ordinateurs de type pc.

Il propose aussi la réalisation de plusieurs interfaces afin d'effectuer la commande de divers processus du plus simple au plus complexe tels que : commandes de réseaux ferroviaires miniatures, commandes de rotation de moteurs électriques, commandes de l'alimentation de circuits divers par relais électromécanique ou par transistors, commande spécifiques en fonction d'évenements extérieurs par cartes d'entrées/sorties.

Une très grande place est accordée aux convertisseurs, analogiques/numériques et numériques/analogiques dont le fonctionnement est détaillé.

Plusieurs montages les mettent en application. On pourra ainsi réaliser divers systèmes de mesures et de commandes de moteurs.
Tous les montages proposés peuvent aussi fonctionner avec un pc « d'ancienne génération », il suffit qu'il soit équipé d'un port parallèle bi-directionnel. L'auteur, Monsieur Patrice Oguic, connaît parfaitement ce type de montage (voir ses réalisations dans Electronique Pratique), gage de sérieux s'il en est.

Edité par ETSF - 23 €

Carte 8 sorties et analyseur logique 5 voies

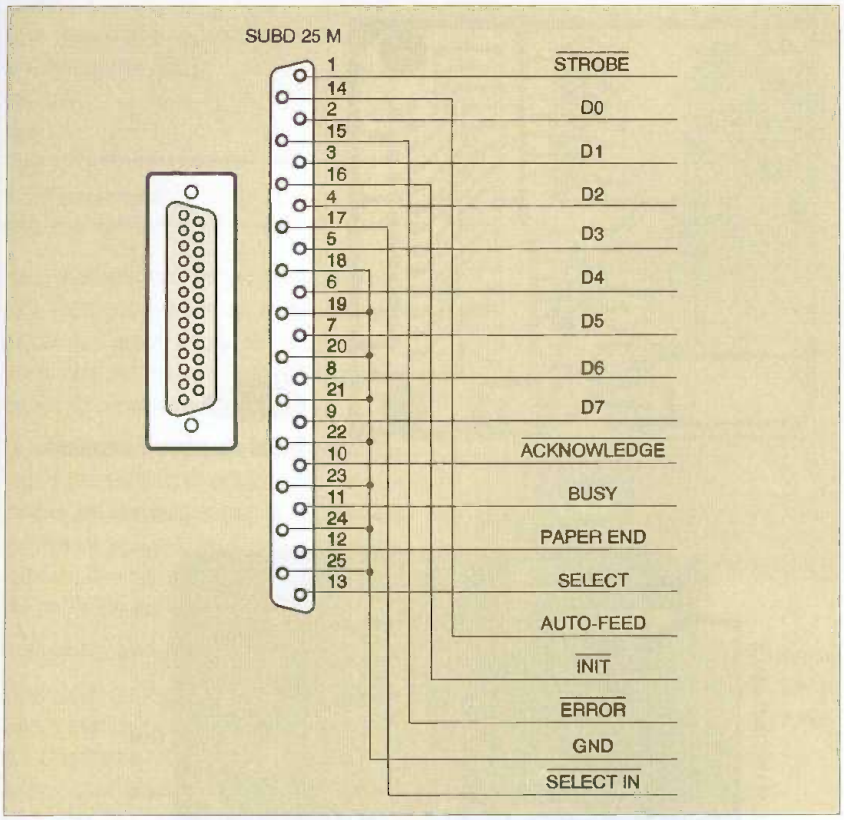
Faisant suite aux articles concernant les ports du PC, nous vous proposons aujourd'hui la description et la réalisation d'une carte d'interface permettant le contrôle de huit sorties et de cinq entrées, ces dernières pouvant être utilisées afin de se constituer un petit analyseur logique. C'est, nous le pensons, une bonne mise en application des descriptions théoriques vues précédemment.

Dans un numéro précédent d'Électronique Pratique, nous avons consacré un article sur les ports parallèles de l'ordinateur PC, décrivant leurs différentes adresses, leurs différentes lignes ainsi que la façon de les utiliser. Nous pensons que la théorie doit être connue, bien évidemment, mais que rien ne vaut la mise en pratique. C'est pourquoi nous vous proposons une application.

Celle-ci, très simple puisque n'étant constituée que de trois circuits intégrés courants, permet néanmoins de disposer de huit lignes de commandes à fort courant de sortie et de cinq lignes d'entrées utilisées pour la réalisation d'un petit analyseur logique. Néanmoins, ce dernier disposant de cinq voies, permet différents réglages et sa vitesse de lecture dépend de celle du microprocesseur équipant l'ordinateur qui le pilote. Il peut donc être utilisé dans la plupart des cas, ne serait-ce que comme sonde logique à plusieurs entrées.

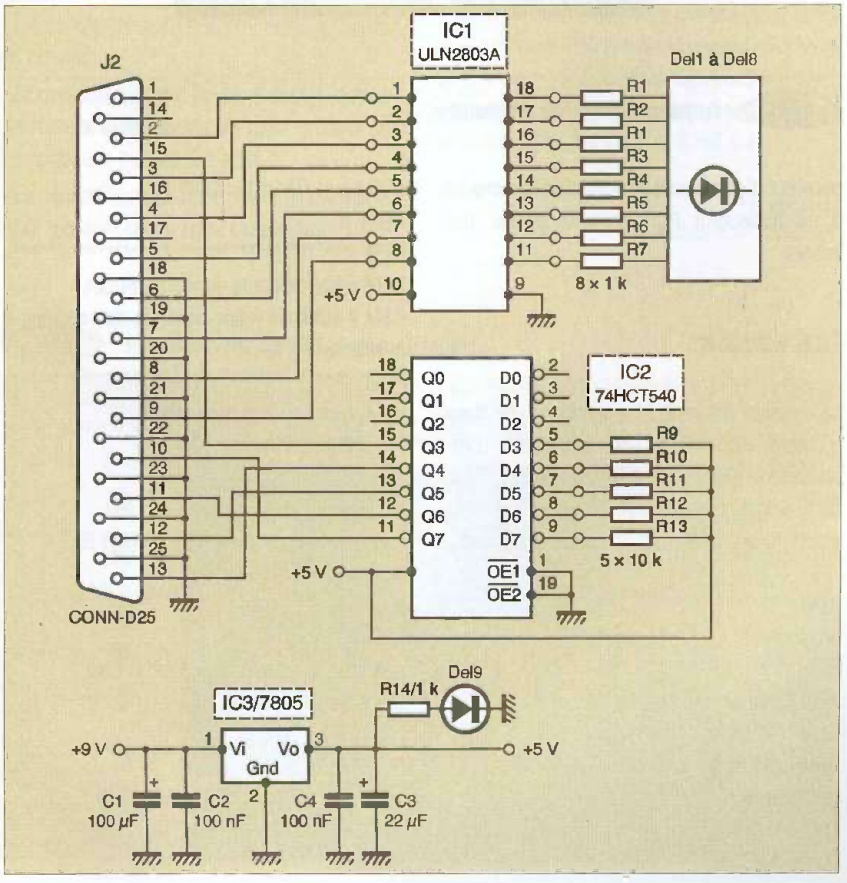
Le schéma de principe

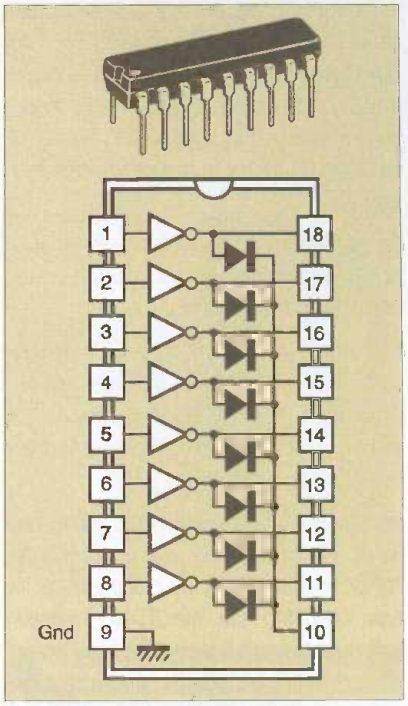
Pour rappel, nous proposons en **figure 1** le câblage de l'interface parallèle du PC. Elle dispose de huit lignes de données en entrées ou en sorties, de quatre lignes de contrôle et de cinq lignes de lecture d'état. Les huit lignes de données ne sont utilisées ici que comme lignes de sorties alors qu'elles peuvent l'être également en entrées, si le " setup " de l'ordinateur a été configuré. Les cinq lignes de lecture d'état, quant à elles, constituent les cinq voies de l'analyseur logique. Les quatre lignes de contrôle ne sont pas prises en compte. Le schéma de principe de notre réalisation est donné en **figure 2**. Un connecteur SUBD 25 permet de connecter la platine au port parallèle du PC. Ce connecteur étant de type mâle, le circuit peut être relié au PC sans utiliser un câble. Les huit lignes de données commandent un circuit de puissance de type ULN2803A (ou ULN2804A) constitué de huit transistors Darlington. Le brochage du circuit est donné en **figure 3**, tandis que la **figure 4** donne, pour information, le schéma interne de chacun des huit amplificateurs. Le ULN2803A est spécialement conçu pour s'adapter aux signaux de commande de type TTL. Le ULN2804A est réservé, en principe, à



1 *Câblage de l'interface parallèle du PC*

2 *Schéma de principe de la réalisation*



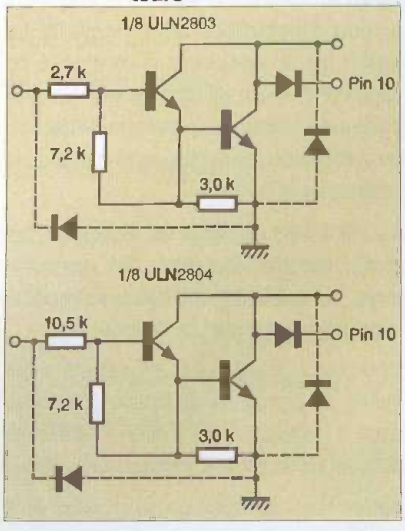


3 *Brochage du circuit ULN2803A*

des signaux aux normes CMOS et PMOS, c'est-à-dire des signaux d'amplitude comprise entre 6 et 15 V. Des essais prouvent malgré cela que ce dernier fonctionne également correctement avec la logique TTL.

Chacun des amplificateurs peut fonctionner sous une tension maximale de 50 V et débiter un courant de 500 mA. Cependant, la dissipation de puissance de chacun des buffers est de 1 W tandis que la dissipation totale du boîtier est de 2,25 W. Il est donc inutile d'espérer alimenter huit relais électromécaniques sous 12 V et consommant un courant de

4 *Schéma interne de chacun des huit amplificateurs*



100 mA. Par contre, des relais alimentés sous 9 V et demandant un courant de 20 ou 30 mA pourront être connectés aux sorties de la platine. On a cependant la possibilité de mettre en parallèle deux à deux les sorties de l'ULN2803A, ce qui augmente les possibilités.

On peut également, au moyen des sorties du circuit, commander une interface présentant une puissance plus importante.

Les sorties du circuit intégré de puissance sont connectées à des LED indicatrices afin de vérifier le niveau présent sur ses entrées, et donc sur les huit lignes du bus de données de l'interface parallèle.

Les entrées, au nombre de cinq, utilisent les lignes ACKNOWLEDGE, BUSY, SELECT IN, PAPER END et ERROR. Un octuple buffer de type 74HCT541 (ou 74HCT540, inverseur) tamponne les cinq lignes d'entrées. Ceci est une précaution à respecter absolument afin de ne pas endommager, en cas de fausse manœuvre, l'interface parallèle du PC.
Trois lignes du circuit sont inutilisées.

La platine peut être alimentée sous une tension variant entre 9 V et 12 V, tension qui est choisie en fonction des organes à commander. En effet, si le buffer est alimenté sous 5 V au moyen d'un régulateur de tension, l'ULN2803A est directement soumis à la tension primaire, ainsi, bien sûr, que les relais. Il faudra donc choisir la tension d'alimentation en fonction de la tension des bobines des relais (ou de la tension maximale acceptée par les systèmes commandés).

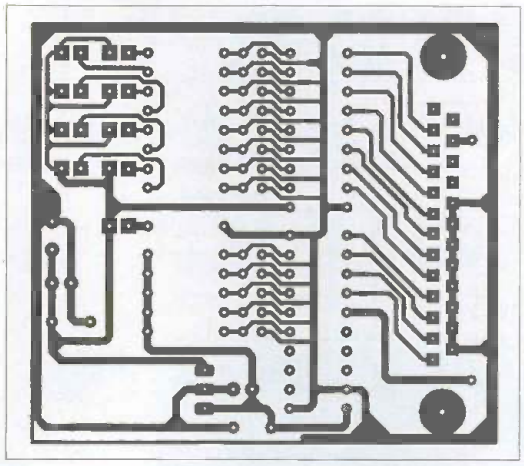
La réalisation

Le tracé du circuit imprimé est donné en **figure 5**, tandis que la **figure 6** représente le schéma d'implantation des composants. La platine est de dimensions restreintes et ne supporte que peu de composants. Un seul strap est à implanter, ce qui sera à effectuer en premier lieu, ce dernier se situant sous le connecteur SUBD25.

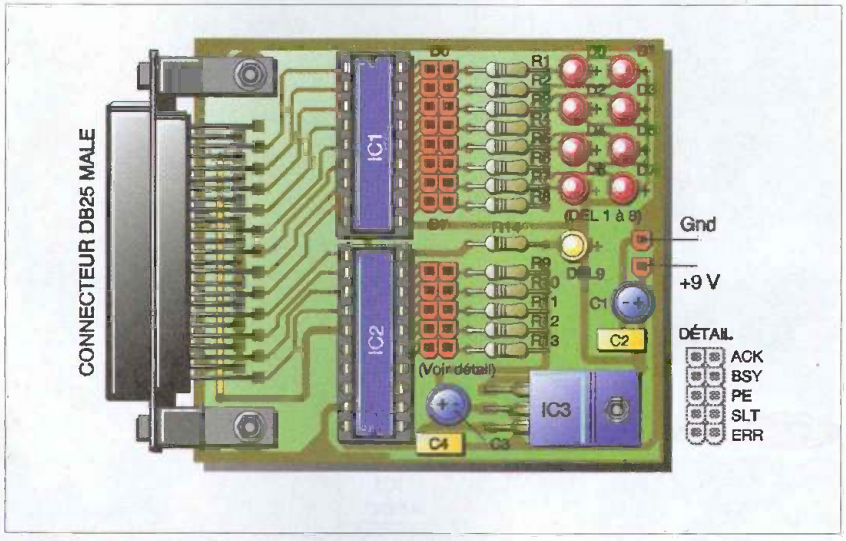
Il suffit ensuite de câbler les résistances, les diodes électroluminescentes, les condensateurs, le régulateur de tension et les deux supports pour les circuits intégrés.

Le connecteur est soudé en dernier. Il est inutile de prévoir un dissipateur thermique pour le régulateur de tension, celui-ci ne débitant qu'un courant infime.

Après une vérification minutieuse qu'il



5 *Tracé du circuit imprimé*



6 *Implantation des éléments*

convient d'effectuer correctement (connexion à un ordinateur PC), on peut passer aux essais.

Les essais

Les essais peuvent être effectués de deux manières différentes : sous le système d'exploitation DOS ou sous WINDOWSTM. Sous DOS, ce qui peut s'avérer une bonne solution, il suffit d'entrer le petit programme suivant :

```
REM ****************************************
programme:
REM ****************************************
REM Essai des huit lignes de sorties
REM ****************************************
REM Les LED s'illuminent selon une progression binaire
FOR d = 0 TO 255
OUT &H378, d
FOR t = 0 TO 3000: NEXT t
NEXT d
REM ****************************************
REM Essai des cinq lignes d'entrées
REM Il suffit de relier chacune des entrées à la masse
REM ****************************************
REM Lecture de la ligne BUSY
A = INP(&H379) AND 128
PRINT A
SLEEP 1
REM Lecture de la ligne ACKNOWLEDE
A = INP(&H379) AND 64
PRINT A
SLEEP 1
REM Lecture de la ligne PAPER END
A = INP(&H379) AND 32
PRINT A
SLEEP 1
REM Lecture de la ligne SELECT IN
A = INP(&H379) AND 16
PRINT A
```

```
SLEEP 1
REM Lecture de la ligne ERROR
A = INP(&H379) AND 8
PRINT A
SLEEP 1
REM ****************************************
GOTO programme
REM ****************************************
```

Avec ce programme, après l'illumination des LED, l'état des entrées est continuellement affiché sur l'écran. Il suffit de relier chacune des entrées à la masse afin de vérifier si elles répondent correctement.

Le second moyen de test est l'utilisation du logiciel fonctionnant sous WINDOWS™ et disponible en téléchargement. Ce logiciel est fonctionnel puisqu'il permet la commande manuelle des huit sorties ainsi que l'utilisation de l'analyseur logique.

Une vue d'écran est donnée en **figure 7**.

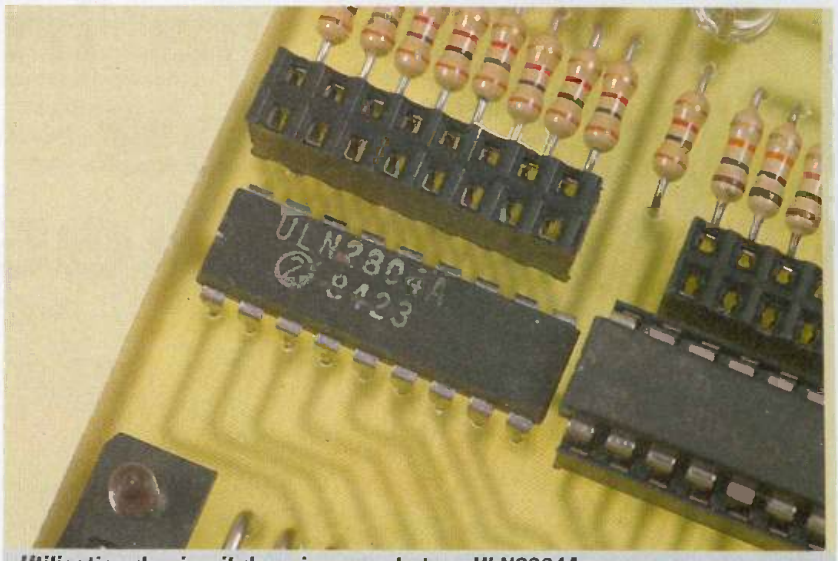
L'analyseur, bien que de conception simple, permet d'afficher des signaux en provenance des cinq lignes de l'interface parallèle (huit canaux sont affichés, seuls cinq fonctionnent). Ses principales caractéristiques sont les suivantes :

- Déclenchement manuel : démarrage de l'enregistrement en cliquant sur le bouton rouge.

- Déclenchement en boucle : enregistrement en continu.

- Déclenchement sur data 0 à data 4 : l'enregistrement démarre sur un front présent sur l'un des cinq canaux D0 à D4.

- Le déclenchement peut être effectué sur un front montant ou un front descendant



Utilisation du circuit de puissance de type ULN2804A

Le bouton " imprimante " ouvre une seconde fenêtre qui permet une prévisualisation et une modification de la position et de l'échelle du graphique avant son impression.

La base de temps peut être modifiée :

- Option " horloge système " qui offre la vitesse maximale.

Le taux d'échantillonnage peut être réglé entre 1/1000 et le maximum,

- Option " heure " pour les phénomènes lents et qui permet l'échantillonnage entre 0,05 s et 5 s. Une centaine d'enregistrements sera mémorisée. La durée de cet enregistrement sera comprise entre 5 s et 8 mn 3 s.

P. OGUIC
Patrice.oguic@tiscali.fr

Nomenclature

Résistances :

R1, R2, R3, R4, R5, R6, R7, R8, R14 : 1 kΩ (marron, noir, rouge)

R9, R10, R11, R12, R13 : 10 kΩ (marron, noir, orange)

Condensateurs :

C1 : 100 µF 16 V

C2, C4 : 100 nF

C3 : 22 µF 16 V

Semi-conducteurs :

DEL1 à DEL9 : diodes électroluminescentes rouges

Circuits intégrés :

IC1 : ULN2803A, ULN2804A

IC2 : 74HCT541, 74HCT540

IC3 : régulateur de tension 7805

Divers :

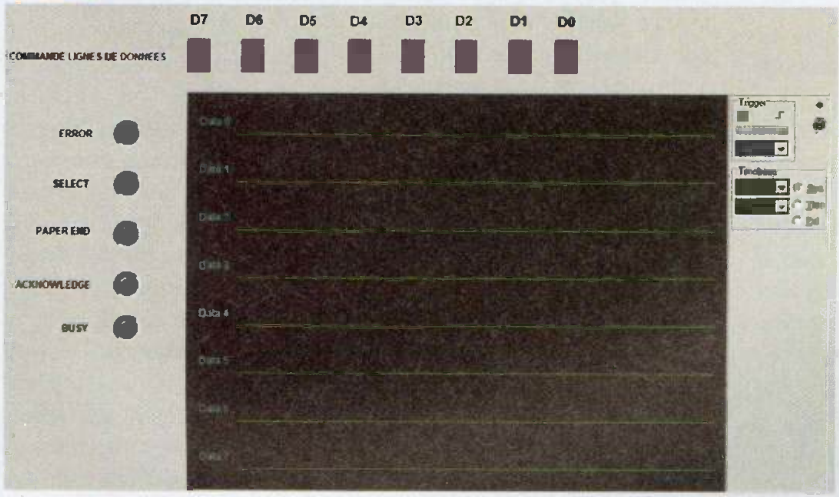
1 support pour circuit intégré 20 broches

1 support pour circuit intégré 18 broches

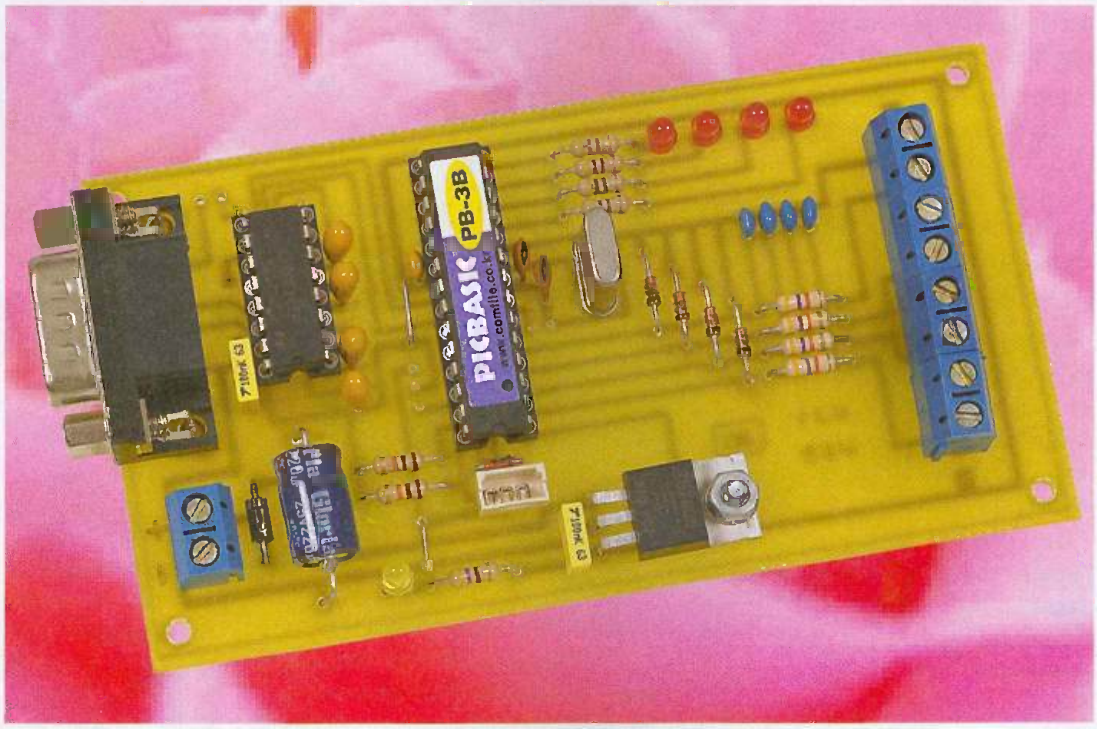
1 connecteur SUBD à 25 broches coudées mâle pour circuit imprimé

2 picots à souder

7 *Vue d'écran de l'état des entrées*



Mesurer via internet : 4 entrées analogiques



Le nombre d'utilisateurs sans cesse croissant d'Internet permet aux fournisseurs d'accès de proposer des abonnements ADSL à des prix très abordables. Cette technologie autorise une connexion rapide et surtout sans limite de temps. Electronique Pratique saisit cette opportunité pour vous proposer une série de montages qui ouvrent de nouvelles perspectives dans le domaine de la télémesure.

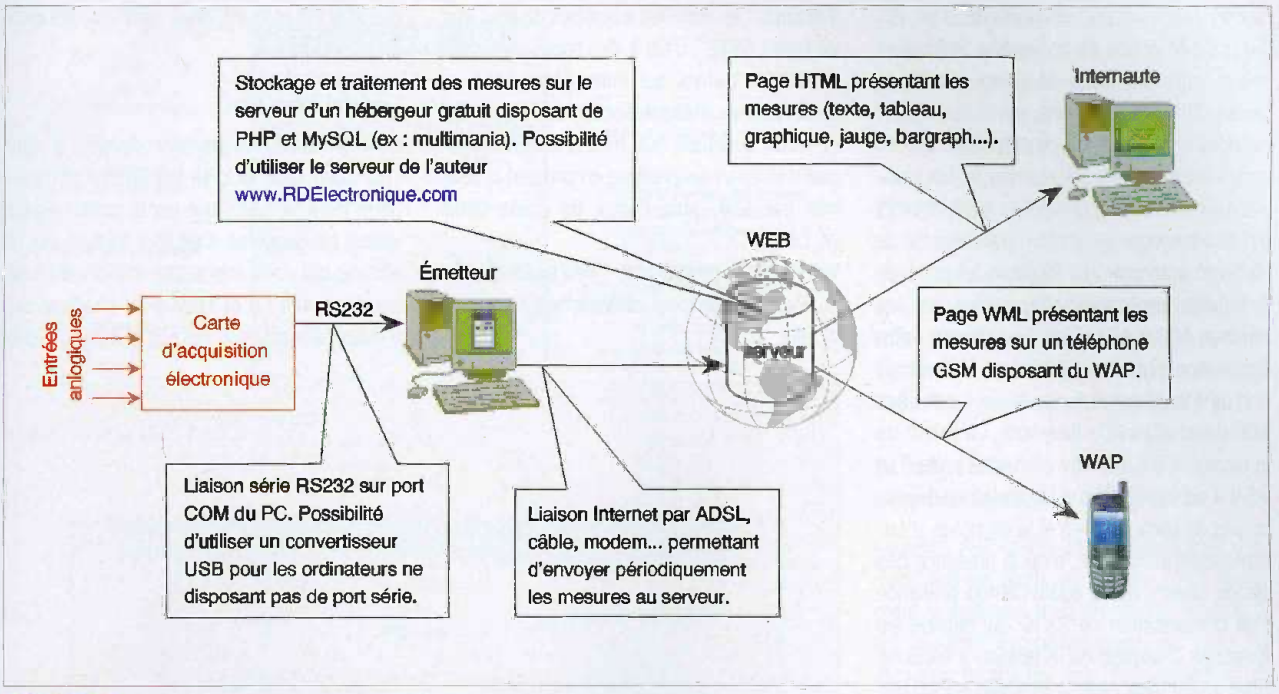
Le principe (figure 1)

Un montage électronique très simple connecté sur le port série de votre ordinateur (nommé "Émetteur") réalise la lecture de 4 entrées analogiques. Un logiciel tournant sur votre PC envoie périodiquement chacune des mesures effectuées sur Internet à destination d'un serveur chargé, dans un premier temps, de les sauvegarder dans une base de données. Ces mesures peuvent être à tout moment consultées n'importe où dans le monde comme n'importe quelle autre page Internet. Concernant le serveur chargé de faire le lien entre l'émetteur et l'internaute, il peut s'agir d'un hébergeur gratuit tel que Multimania qui dispose du langage PHP, ou du serveur de l'auteur sur lequel se trouvent déjà installés tous les scripts de traitement. Le PHP est un langage de programmation adapté à Internet, il permet de créer dynamiquement des pages HTML. Ainsi, il est facile de présenter à l'internaute l'historique des mesures sous différentes formes, texte, tableau, graphique...etc. Il est même possible de générer des pages en WML visionnables sur un téléphone GSM disposant du WAP !

Schéma électrique (figure2)

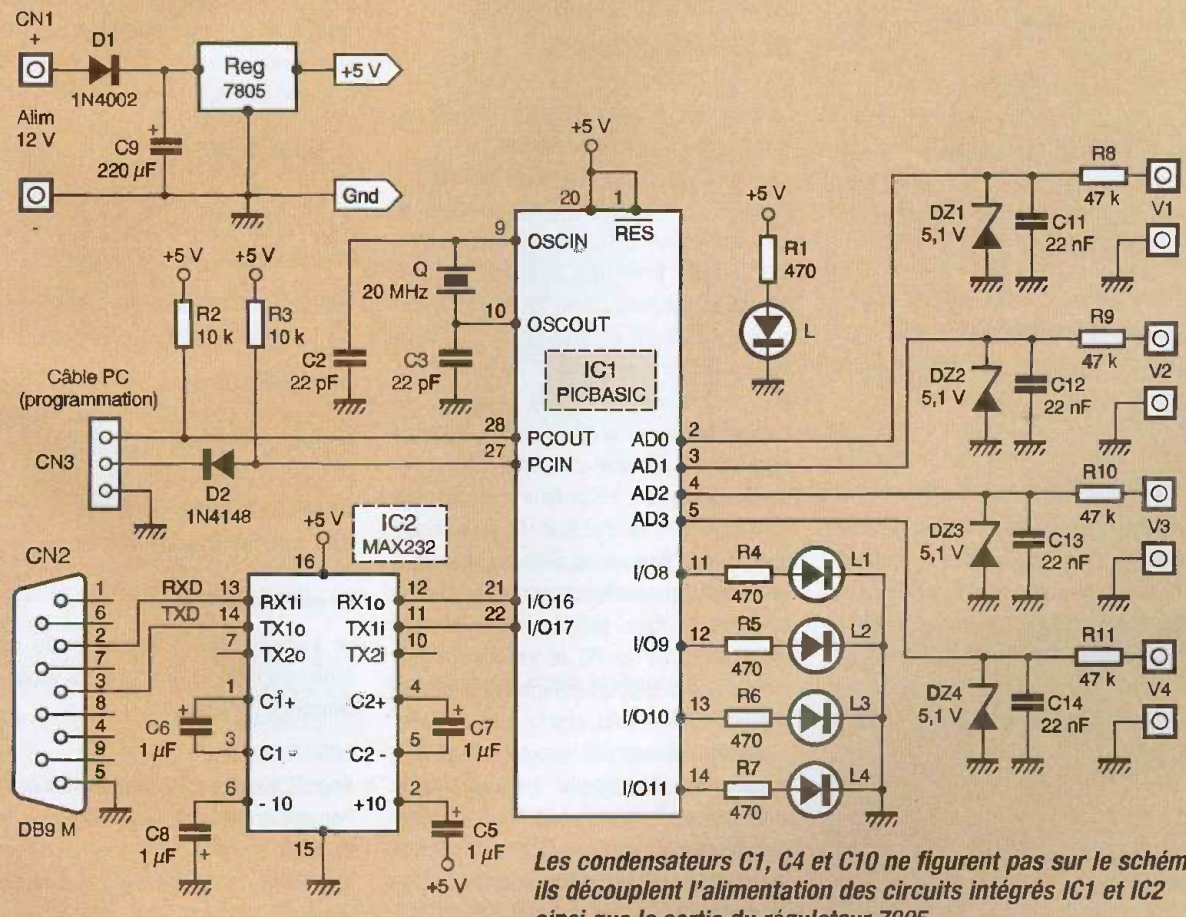
Le microcontrôleur utilisé pour cette réalisation est un PicBasic du constructeur Coréen COMFILE TECHNOLOGY. Il existe 3 familles de PicBasic. Celui que nous avons choisi, le PICBASIC-3B, appartient à la deuxième famille, il est un bon compromis entre le coût et les possibilités offertes. Disponible en boîtier au format DIP 18 broches, il intègre un PIC 16C74A-04, un quartz de 4,19 MHz et une mémoire EEPROM série d'une capacité de 4 Ko.

Que tous ceux qui sont allergiques au langage assembleur se rassurent, comme son nom le laisse présager, le PicBasic se programme en basic. Le logiciel PICBASIC-LAB fourni par le fabricant permet, à l'aide d'un PC, une conception vraiment très aisée du programme. Le puissant compilateur intégré permet de traduire les lignes basic en instructions spécifiques compréhensibles par le microcontrôleur. Le programme compilé peut ensuite être implanté dans la mémoire EEPROM du PicBasic par le biais d'un



1 *Principe du fonctionnement*

2 *Schéma théorique bâti autour du PICBASIC-3B*



Les condensateurs C1, C4 et C10 ne figurent pas sur le schéma, ils découplent l'alimentation des circuits intégrés IC1 et IC2 ainsi que la sortie du régulateur 7805

cordon relié au port imprimante d'un PC. Notez qu'en phase de conception, la fonction debug vous permettra de suivre pas à pas l'exécution du programme par le PicBasic. Il est notamment possible de consulter l'état de toutes les variables utilisées par le programme. Une fois le programme au point, il suffit de déconnecter le cordon pour rendre le PicBasic autonome. Le PicBasic-3B possède 5 entrées analogiques disponibles sur les broches AD0 à AD4. Dans le cadre de notre application, nous nous limiterons à l'acquisition de 4 tensions analogiques sur les entrées AD0 (broche2) à AD3 (broche5). La valeur de la tension à lire doit être comprise entre 0 et +5 V. Il est impératif que la tension ne dépasse pas la barre des +5 V sous peine d'endommager le PicBasic, d'où la présence des diodes zeners. Notez également la présence d'un condensateur de 22 nF qui élimine les variations brusques de la tension à mesurer (filtre passe bas). Avec la version du PicBasic que nous utilisons, la résolution du convertisseur est de 10 bits. Cela signifie que le circuit est capable de convertir une tension en un nombre binaire composé de 10 bits. La précision de la mesure est donc égale à $5/2^{10}$ = 0,005 V. Rappelons la relation qui permet à partir de la valeur binaire de calculer la tension : $V = (Dx5)/2^{10}$.

V est la tension mesurée exprimée en volts et D est la donnée exprimée en décimal calculée par le PicBasic. Par exemple, si D=512, cela signifie que la tension mesurée est égale à 2,5 V. L'instruction basic qui permet de réaliser une conversion est :

```
ADIN(port)
```

Avec port compris entre 0 et 3.

L'instruction `SEROUT` envoie ensuite le résultat sur la liaison série :

```
SEROUT Port, Paraml, Mode,
Param2, [DATA]
```

Cette instruction permet de transmettre des données sous forme série selon le protocole RS232. Une fois exécutée, la broche **Port** du PICBASIC transmettra la ou les données **DATA** à une vitesse définie par **Param1**, selon la correspondance du tableau ci-dessous. Le paramètre **Mode** permet d'instaurer

Vitesse (bits/s)	Param1
4800	207
9600	103
19200	51

une temporisation entre chaque caractère émis dont la durée en ms est fonction de **Param2**. Les données envoyées doivent être de type " BYTE " c'est à dire comprises entre 0 et 255. L'envoi est donc décomposé en deux octets, d'abord l'octet de poids fort (V_MSB) constitué que de 2 chiffres significatifs obtenus simplement en divisant la donnée par 256, puis l'octet de poids faible (V_LSB).

Voici donc le programme " 4ea.bas " destiné au PicBasic, on peut difficilement faire plus simple.

```
DIM V      AS INTEGER              'Tension mesurée
DIM V_MSB AS BYTE                  'MSB de la tension mesurée
DIM V_LSB AS BYTE                  'LSB de la tension mesurée
DIM n      AS BYTE                 'numéro de voie
DIM voie   AS BYTE                 'code ASCII voie
DIM c      AS BYTE                 'Variable pour clignotement LEDS

DEBUT:                             'Etiquette DEBUT
 FOR n=0 TO 3                      'Balayage des voies 0 à 3
  V=ADIN(n)                        'Lecture de la tension
  V_MSB=V/256                      'Calcul du MSB
  V_LSB=V                          'Calcul du LSB
  voie=n+49
  SEROUT 16, 103, 0, 1, ["V", voie,"=",
V_MSB, V_LSB, ";"]                 'Envoie la donnée sur la liaison série
  NEXT n
 IF c=0 THEN c=15 ELSE c=0         'Alternance LEDS ON ou OFF
 BYTEOUT 1,c                       'Leds ON ou OFF
 DELAY 1000                        'Tempo 1s entre chaque série de mesures
GOTO DEBUT                         'Branche sur l'étiquette DEBUT
```

Réalisation du montage et programmation

Le tracé du circuit vous est présenté **figure 3**, le schéma d'implantation **figure 4**.

La réalisation du circuit reste relativement simple.

Lors de l'implantation des éléments veillez à la bonne orientation des composants polarisés, condensateurs C5 à C9 et des diodes zeners. Le PB-3B est mis en place sur un support DIL 28 broches étroit. Le connecteur, avec détrompeur, permettant de recevoir le câble utilisé pour la programmation in situ est notamment disponible chez Lextronic.

Il reste maintenant à transférer le programme dans l'EEPROM du PicBasic. Pour raccorder le PC et le PicBasic via le câble imprimante, il vous faudra impérativement couper l'alimentation du montage, puis connecter le câble avant d'allumer le PC et en dernier lieu mettre le montage sous tension. Lancez le logiciel PICBASIC-LAB, ouvrez alors le fichier " 4ea.bas ", cliquez sur le bouton " RUN ". Le programme est compilé en instructions assembleurs qui sont ensuite implantées dans la mémoire du PicBasic. Pour rendre le montage autonome, coupez toujours l'alimentation du montage puis celle du PC. De même, ne déconnectez le cordon de liaison que si le PC et le montage sont tous les deux hors tension.

Logiciel : 4ea.exe

Un logiciel spécifiquement développé pour l'occasion sous DELPHI est chargé de récupérer les 4 tensions lues par la carte. Chaque valeur se compose d'un mot binaire de 10 bits, ce qui nous donne une valeur décimale comprise entre 0 et 1024 pour chacune des 4 voies de mesure. C'est ces 4 valeurs que le logiciel doit envoyer périodiquement au serveur. Il existe deux méthodes très simples pour envoyer des données à un serveur : la méthode GET et la méthode POST.

La *méthode* GET

Le logiciel envoie le résultat d'une mesure en le concaténant à l'adresse (URL) qu'il demande au serveur.

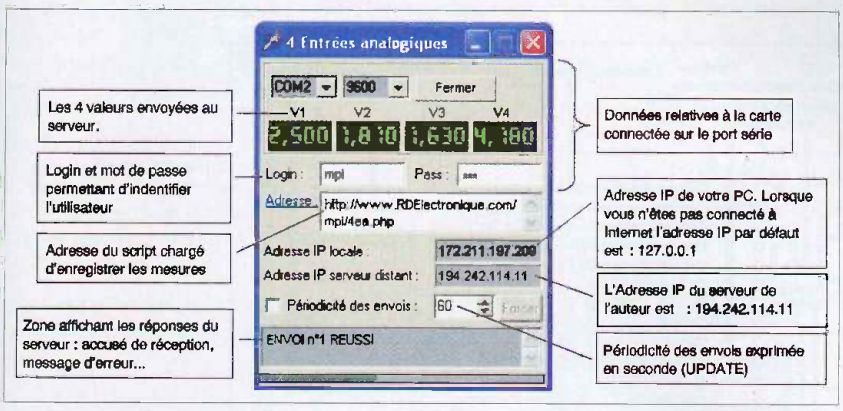
http://*www.RDElectronique.com/mpi/4ea.php?*V1=239&V2=512&V3=5&V4=1000

www.RDElectronique.com/mpi/4ea.php est l'adresse du script PHP, celui-ci récupère le résultat de la mesure en lisant tout ce qui suit le point d'interrogation. V1 à V4 sont des variables récupérées par le script et mémorisées dans une base de données.

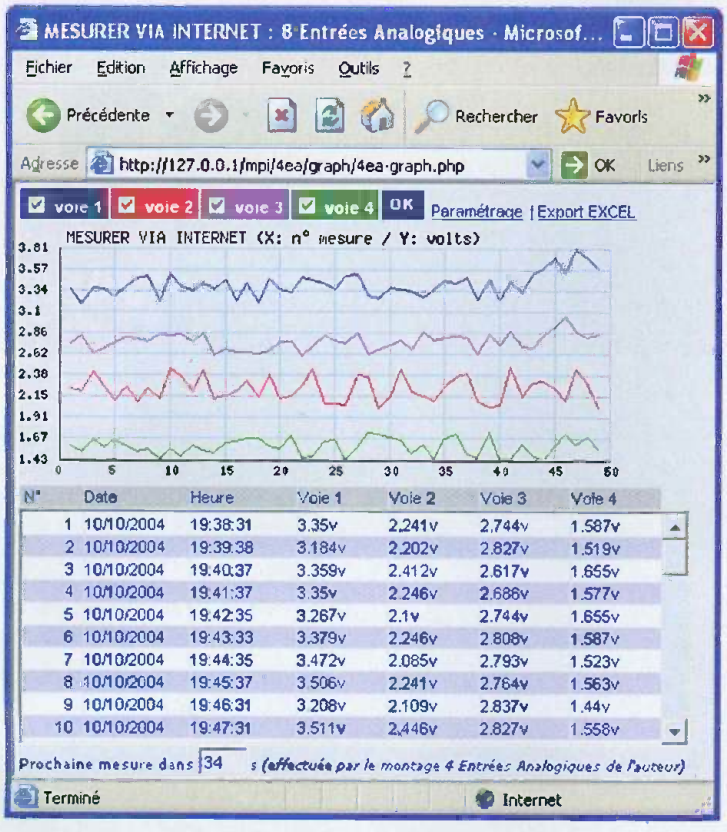
La *méthode* POST

Le logiciel envoie une requête au serveur contenant l'URL, suivie dans des lignes additionnelles, les données.

POST " http://
www.RDElectronique.com/mpi/4ea.php "
Content-length=20
V1=239
V2=512
V3=5
V4=1000



Présentation de l'écran



Présentation des 10 derniers relevés des 4 voies

Le script lit ces données et s'arrête lorsqu'il a reçu le nombre de caractères spécifié par Content-length.

En réalité le logiciel 4EA utilise les deux méthodes. La méthode GET pour envoyer les données LOGIN et PASS et la méthode POST pour envoyer la DATE et l'HEURE à laquelle la mesure a été réalisée et V1 à V4. Pour chaque mesure réceptionnée, le serveur envoie un accusé de réception qui s'affiche dans la zone de texte au bas de l'écran. Si tout va bien, vous devriez voir le message suivant :" Envoi n°x réussi ". En cas d'échec, un message vous indiquera la nature de l'erreur (pas de connexion internet, mauvais mot de passe...)

Le script PHP d'enregistrement : 4ea.php

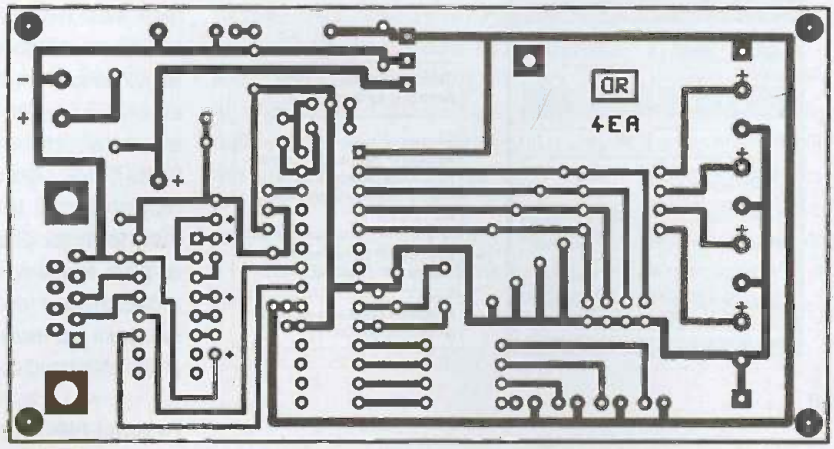
Le script d'enregistrement sollicité par l'ordinateur " Émetteur " est situé sur le serveur de l'auteur accessible à l'adresse suivante : *http://www.RDElectronique/mpi/4ea.php*. Il récupère les variables passées en POST (Vx, DATE et HEURE). Par exemple, l'instruction $_POST[Vx] récupère la donnée Vx qui est ensuite enregistrée dans une base de données, avec x compris entre 1 et 4. Chaque mesure réceptionnée est associée à un utilisateur unique identifié par les variables passées en GET (LOGIN et PASS) récupérées par les instructions $_GET[LOGIN] et $_GET[PASS]. Ainsi, plusieurs utilisateurs peuvent déposer leurs mesures sans risque d'interférences. Ceux qui souhaitent mettre en place leur propre serveur et profiter de l'occasion pour découvrir le langage PHP trouveront un exemple de script simplifié (mono utilisateur) dans le répertoire php.

Le script PHP de présentation des résultats

Le script PHP utilisé ici présente les 50 derniers relevés dans un tableau mais également sous forme d'un graphique à points. Il est possible de sélectionner les voies de mesures à afficher par l'intermédiaire de cases à cocher. Le graphique s'adapte automatiquement en fonction des valeurs mesurées min et max. Notez en bas de page le compte à rebours qui débute au nombre de secondes indiqué dans le logiciel 4ea (constante UPDATE exprimée en secondes). Lorsque la valeur zéro est atteinte, la page HTML est mise automatiquement à jour. Le fait de double-cliquer sur une ligne du tableau va déclencher la mise à jour de la page, une série de pointillés va alors mettre en évidence la mesure correspondante sur le graphique. Pour peu que l'ordinateur utilisé pour visualiser la page soit équipé du logiciel Excel, un simple clic de souris sur le lien "Export EXCEL" suffit pour exporter l'ensemble des mesures dans un fichier au format xls. Le lien "Paramétrage" donne accès à une page personnalisée permettant de paramétrer le graphique : type, taille, couleurs, quadrillage, résolution, titre...

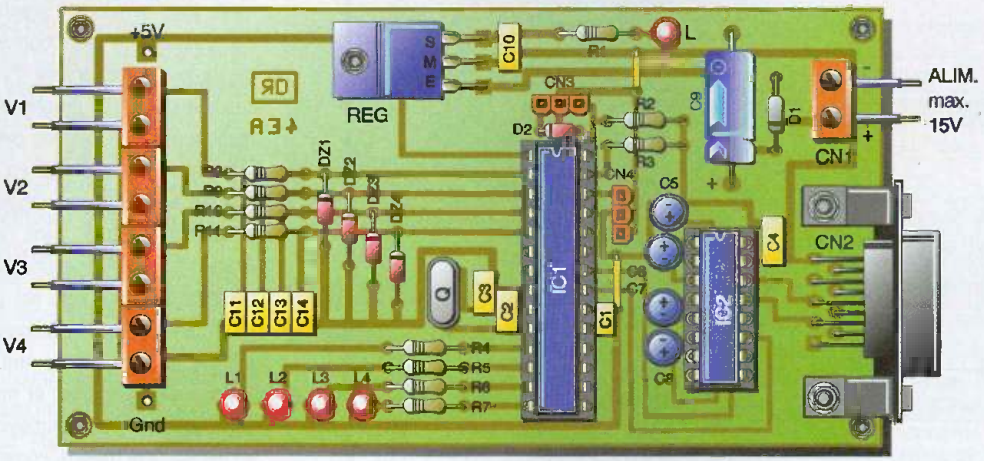
Pour accéder à vos mesures à partir de n'importe quel ordinateur dans le monde disposant d'une connexion Internet, il

Affichage d'une image sur l'écran d'un téléphone portable



3 *Tracé du circuit imprimé*

4 *Implantation des éléments*



suffit de vous rendre à la page *http://www.RDElectronique.com/?page=MpiAccueil.php*, de rentrer votre login et votre mot de passe puis de cliquer sur le bouton OK.

Consulter dès maintenant un exemple sur le site de l'auteur :

http://www.RDElectronique.com?page=4ea-exemple.php

Il est également possible d'afficher une image sur l'écran d'un téléphone portable. L'affichage du graphique se fera en noir et blanc et sous un format d'image spécifique nommé wbmp. Une seule voie de mesure pourra donc être affichée simultanément. La sélection de la voie se fait par l'envoi d'un paramètre noté v passé en GET. Lorsque ce paramètre est omis, c'est la voie n°1 qui est affichée par défaut. Par exemple, pour visualiser la voie 2, vous devrez saisir l'adresse suivante :

http://www.rdelectronique.com/w/4ea-g.php?l=x&p=y&v=2 (x=login et y=mot de passe)

Consulter dès maintenant un exemple sur votre téléphone portable :

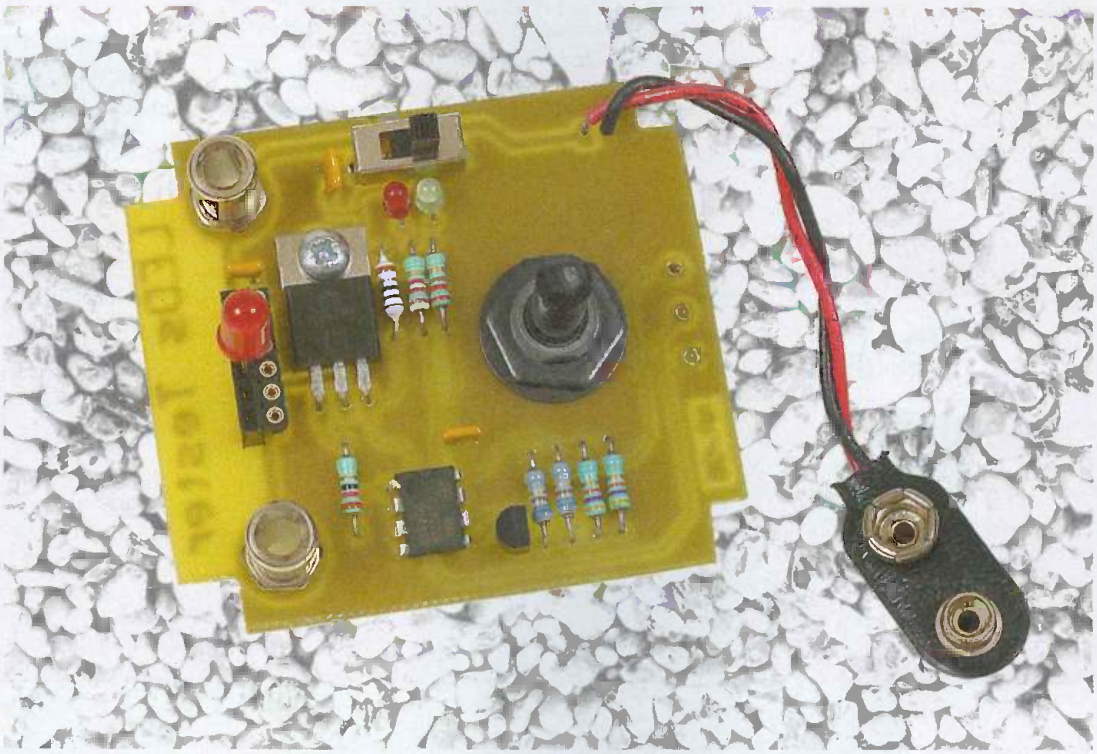
http://www.RDElectronique.com/w/4ea-g-ex.php

D. REY
www.RDElectronique.com

Nomenclature

R1, R4 à R7 : 470 Ω
R2, R3 : 10 kΩ
R8 à R11 : 47 kΩ
DZ1 à DZ4 : diode zener 5,1 V
C1 : 100 nF (pas de 2,54 mm)
C2, C3 : 22 pF / céramique
C4, C10 : 100 nF / LCC jaune
C5, C6, C7, C8 : 1 µF / tantale / 15 V
C9 : 220 µF / électrolytique / 15 V
C11 à C14 : 22 nF / céramique
D1 : diode 1N4002
D2 : diode 1N4148
L, L1 à L4 : led standard
Q : quartz 20 MHz
REG : régulateur 7805
CN1 : bornier à vis 2 plots
CN2 : connecteur DB9 mâle pour CI/coudé à 90°
CN3 : connecteur pour câble de programmation (LEXTRONIC)
CN4 : connecteur pour écran LCD série (facultatif)
IC1 : PICBASIC PB-3B (LEXTRONIC) + support DIL 28 broches (étroit)
IC2 : MAX232 + support DIL 16 broches

Testeur de LEDs



Les diodes électroluminescentes sont des composants très ludiques. Elles sont depuis quelques années revenues dans la « lumière » grâce à l'apparition de nouvelles technologies de semi-conducteurs permettant d'obtenir des couleurs spécifiques (bleu, blanc ou multicolore). Dans ce cadre, nous vous proposons de réaliser un testeur qui permettra d'évaluer facilement quasiment toutes les LEDs disponibles sur le marché.

Avant de décrire la réalisation de ce testeur, commençons par quelques rappels théoriques concernant les LEDs. Leurs principales caractéristiques sont : la chute de tension à leurs bornes, le courant maximum admissible et la tension inverse de claquage. Cette dernière correspond à la tension maximum avant destruction que l'on peut appliquer à la LED lorsqu'elle est bloquée. Cette tension est dénommée VR (Voltage Reverse) dans les datasheet. Ensuite, la chute de tension dénommée VF (Voltage Forward) correspond à la différence de potentiel aux bornes de la diode lorsque celle-ci est passante (quand elle s'allume). Cette tension est donnée pour une valeur de courant dans une datasheet, car lorsque le courant change, cette tension change également. Enfin, le courant maximum admissible est le courant maximum que l'on peut injecter dans la LED sans risque de destruction. Ce courant est directement corrélé à la luminosité de la LED. Autrement dit, plus le courant circulant dans la LED augmente, plus la lumière émise par cette dernière augmente. Ce courant est noté IF (Current Forward) dans une datasheet.

Considérons maintenant les besoins d'un utilisateur : il désire obtenir un certain niveau de lumière tout en consommant un minimum d'énergie. Pour ce faire, il a deux solutions. La première n'est possible que s'il sait donner la luminosité souhaitée et que cette dernière soit convertible en courant grâce à la datasheet de la LED. La seconde solution consiste à réaliser des essais au préalable sur la LED afin de déterminer le courant en observant la luminosité, et enfin à mesurer la chute de tension à ses bornes. Techniquement, cet essai n'est pas très compliqué à réaliser mais reste assez délicat et fastidieux si l'on répète cette opération trop souvent.

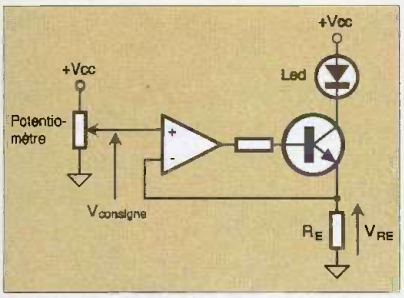
Afin de faciliter cette opération, nous vous proposons un petit instrument parfaitement adapté. C'est une réalisation portable, simple à réaliser et peu onéreuse.

Générateur de courant

Si on se rappelle ce qui a été dit dans la description technique ci-dessus, la principale fonction que doit remplir cet instrument est de pouvoir générer un courant constant dont la valeur est fixée par l'opérateur afin de maîtriser la luminosité.

La première étape consiste donc à construire un générateur de courant pour alimenter la LED. Pour cette fonction, nous avons choisi d'utiliser un transistor contrôlé par un amplificateur opérationnel (**figure1**).

Ce générateur de courant fonctionne de la manière suivante : le courant, dans la diode, circule également dans la résistance R_E. Aux bornes de R_E, nous avons donc une tension qui est quasiment l'image du courant dans la LED (au courant de base du transistor



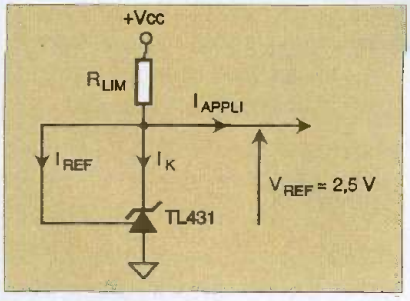
1 *Générateur de courant*

près). Ensuite, l'amplificateur opérationnel commande la base du transistor jusqu'à obtenir une tension V_{RE} égale à $V_{CONSIGNE}$. En choisissant, astucieusement la valeur de R_E et la tension de consigne $V_{CONSIGNE}$, il est donc possible de choisir avec précision le courant dans la LED.

Référence de tension

La fonction principale étant définie, il faut vérifier que cela fonctionne dans toutes les situations. En effet, il est intéressant que cet instrument soit portable ce qui signifie qu'il sera alimenté par une pile (ou une batterie) dont la tension varie de manière importante en fonction de son niveau de décharge.
Maintenant, si on reporte la tension de la pile comme tension d'alimentation dans le générateur de courant, on s'aperçoit que la consigne variera de la même manière que la tension d'alimentation. Il faut donc fixer un potentiel de référence afin que la valeur du courant soit correcte quel que soit l'état de la pile. On peut réaliser cette référence avec un régulateur de tension ou mieux avec une référence de tension intégrée. C'est cette dernière solution que nous avons retenue en utilisant un TL431. Ce composant, très courant, a de nombreux avantages : il est peu onéreux, d'un faible encombrement, d'une bonne précision et d'une grande facilité d'utilisation.

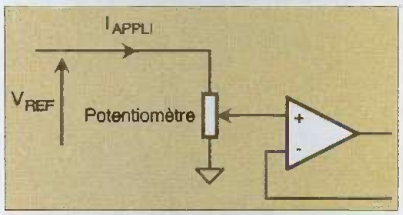
2 *Référence de tension*



Le câblage de la **figure 2** montre que ce TL431 garantit une tension de référence de 2,5 V.
Il faut uniquement calculer la résistance de limitation R_{LIM} afin que le circuit fonctionne correctement.

$$R_{LIM} = \frac{V_{CC(MIN)} - V_{REF}}{I_{K(MIN)} + I_{REF} + I_{APPLI}}$$

Par construction, $I_{K(MIN)}$ vaut 1 mA. Cette valeur est notée dans la datasheet du TL431. De la même manière, dans ce document, on trouve I_{REF}. La valeur de ce dernier ne dépasse pas 6,5 µA. On pourra donc sans problème négliger ce courant dans le calcul car il est très petit devant I_K.
Ensuite, il suffit d'avoir la tension $V_{CC(MIN)}$ de la pile. Pour une pile 6LR61(communément dénommée : pile 9 V), on choisira 5 V, cette tension minimum conviendra aussi bien pour une pile, qu'une batterie CdNi ou qu'une batterie NiMH. Il ne reste plus qu'à déterminer I_{APPLI} pour calculer la valeur de la résistance R_{LIM}.



3 *Etage d'entrée du générateur de courant*

On peut calculer I_{APPLI} grâce à la valeur du potentiomètre et à la valeur du courant de polarisation de l'entrée positive de l'amplificateur opérationnel.

$$I_{APPLI} = \frac{V_{REF}}{R_{Potentiomètre}} + I_{AOP}$$

Nous avons désormais tous les éléments permettant de calculer la résistance R_{LIM}.

Choix du potentiomètre

Comme nous l'avons vu, le potentiomètre permet à l'utilisateur de choisir le courant injecté dans la LED. Sachant que les courants actuellement utilisés dans les LEDs sont répartis par groupes en fonction des technologies, des couleurs et de la luminosité (de 1 à 3 mA, de 15 à 30 mA et de 50 mA à 100 mA), il semble beaucoup plus judicieux d'utiliser un potentiomètre logarithmique (**figure 3**). En effet, la variation de résistance de ce potentiomètre est telle que nous aurons une meilleure précision pour régler les faibles valeurs de courant.

Détection de niveau de batterie

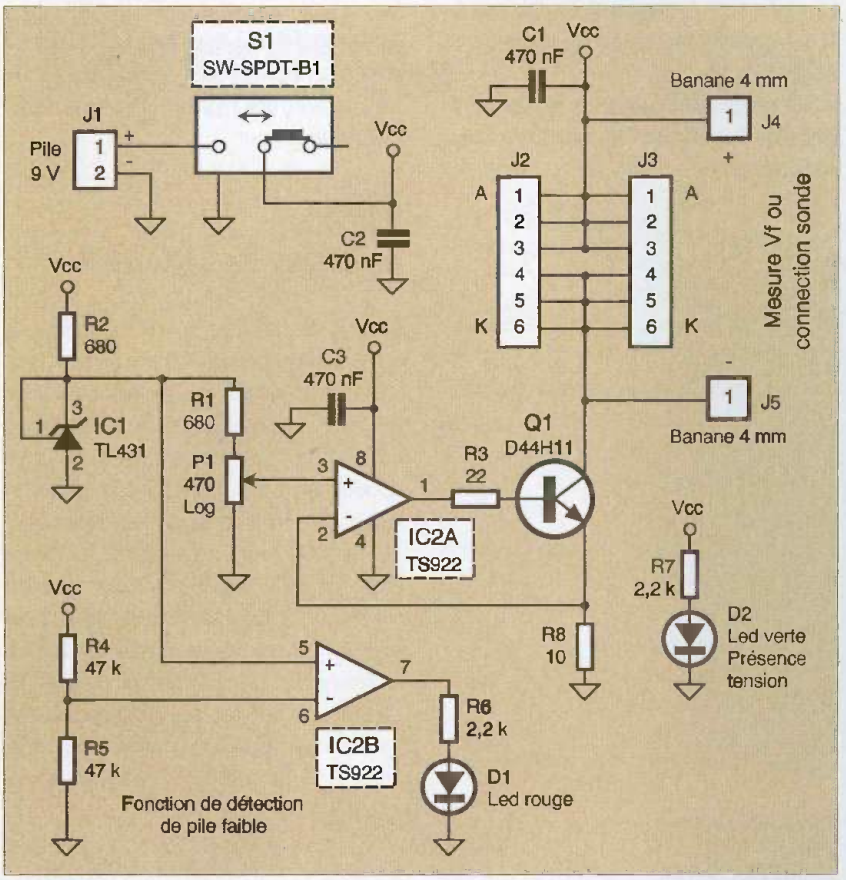
Comme nous l'avons spécifié ci-dessus, la référence de tension fonctionne à partir d'une tension d'alimentation minimum de 5 V. Ce qui signifie qu'en dessous de cette tension, nous ne pouvons pas garantir la valeur du courant circulant dans la LED. Afin de faciliter l'utilisation de notre testeur, nous avons ajouté une petite fonction permettant de vérifier que la batterie fournit la tension minimum nécessaire. En effet, un second amplificateur opérationnel compare la moitié de la tension d'alimentation (grâce au pont diviseur R4/R5) et la tension de référence. Cette fonction allume alors une LED rouge si la tension est inférieure à 5 V (si $V_{CC}/2 < V_{REF}$, ou encore si $V_{CC}/2 < 2,5$ V).

Calcul des composants

Désormais, nous allons nous attacher à calculer la valeur de chaque composant. Commençons par le générateur de courant. Ce générateur doit pouvoir fonctionner entre 0 et 100 mA. La résistance R_E est nommée R8 sur le schéma de principe de la **figure 4**. Nous fixons la valeur de cette résistance à 10 Ω afin de faciliter les calculs et de ne pas avoir une chute de tension trop importante. Nous aurons donc une tension à ses bornes allant respectivement de 0 à 1 volt quand le courant variera de 0 à 100 mA.
La consigne doit donc également varier entre 0 et 1 V. Par conséquent, il faut ajouter une résistance « talon » en série avec le potentiomètre afin de limiter la tension de référence. En négligeant le courant de polarisation de l'entrée de l'amplificateur opérationnel, le calcul est le suivant :

$$V_{CONSIGNE} = V_{REF} \frac{P_1}{P_1 + R_1}$$

Il faut alors choisir la valeur du potentiomètre logarithmique. N'ayant qu'un seul potentiomètre logarithmique en stock, le choix est assez rapide. Dans la mesure du possible, il sera préférable d'avoir une valeur comprise



4 *Schéma de principe*

entre 10 kΩ et 100 kΩ afin de diminuer la consommation de la pile. Néanmoins, quel que soit la valeur de ce potentiomètre, il suffira de calculer les composants en suivant notre exemple.

$$R_1 = P_1 \times \left(\frac{V_{REF}}{V_{CONSIGNE}} - 1 \right) = 470 \times \left(\frac{2,5}{1} - 1 \right) = 705\Omega$$

Connaissant donc P1 = 470 Ω, par le calcul, nous trouvons R1 =705 Ω.

On prendra donc R1 = 680 Ω comme valeur normalisée ce qui permettra d'obtenir une tension de consigne théorique variant entre 0 et 1,02 V.

On peut alors calculer la résistance de limitation de la référence de tension car on connaît à présent le courant I_{APPLI} :

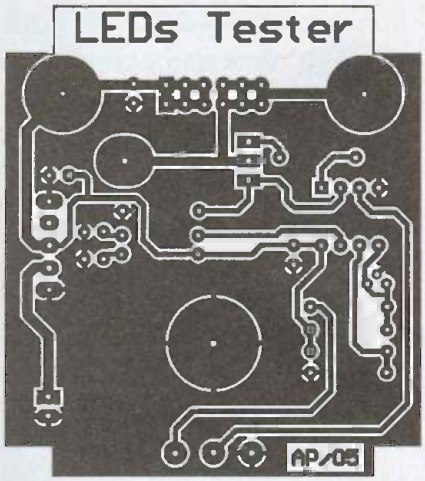
On prendra donc comme valeur normalisée R_{LIM} = R2 = 680 Ω qui correspond à un I_K de 1,5 mA (ce qui constitue une marge confortable par rapport à $I_{K(MIN)}$). Par ailleurs, nous avons ajouté une résistance en série avec la base du transistor du générateur de courant. Cette résistance n'est pas nécessaire dans un usage courant mais elle permet de protéger la base du transistor si on alimente le montage sans connecter de LED dans le testeur.

$$I_{APPLI} = \frac{V_{REF}}{R_1 + P_1} + I_{AOP} = \frac{2,5}{680 + 470} + 100.10^{-9} = 2,17mA$$

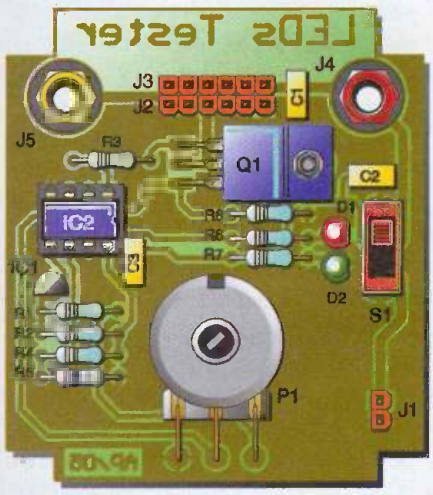
$$R_{LIM} = \frac{V_{CC(MIN)} - V_{REF}}{I_{K(MIN)} + I_{REF} + I_{APPLI}} = \frac{5 - 2,5}{1.10^{-3} + 6,5.10^{-6} + 2,17.10^{-3}} \approx 787\Omega$$

Circuit imprimé

La réalisation et le câblage du circuit imprimé ne présentent aucune difficulté particulière. C'est un circuit simple face de petites dimensions comme on peut le voir sur les **figures 5** et **6**.



5 *Tracé du circuit imprimé*



6 *Implantation des éléments*

Mise en boîte

Le circuit imprimé ci-dessus est prévu pour prendre place dans un boîtier possédant une trappe pour une pile 6LR61 comme on peut le voir sur la photographie de la **figure 7**. Nous vous proposons également le typon de la face avant de ce boîtier (**figure 8**).

Utilisation

L'utilisation de cet appareil, bien que très simple, nécessite quelques commentaires.

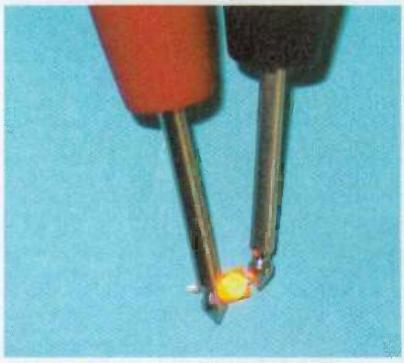
7 *Exemple de mise en boîtier*

J5 est de pouvoir mesurer la chute de tension V_F aux bornes de la LED pendant le test à l'aide d'un voltmètre ou d'un multimètre. Cette mesure est représentée sur la **figure 10**. La valeur de cette tension associée à la valeur du courant permet ensuite de calculer la résistance nécessaire pour alimenter la LED dans les mêmes conditions.

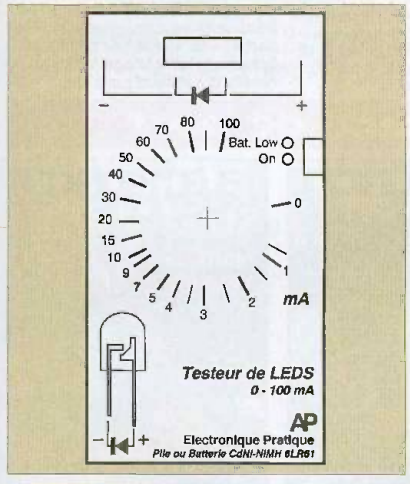
Sur le schéma classique de la **figure 11**, on connaît I_F et V_F, il est donc très facile de calculer la résistance R :

$$R = \frac{V_{CC} - V_F}{I_F}$$

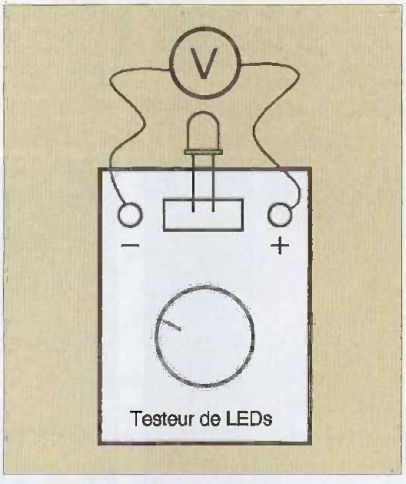
Ph. ANDRÉ



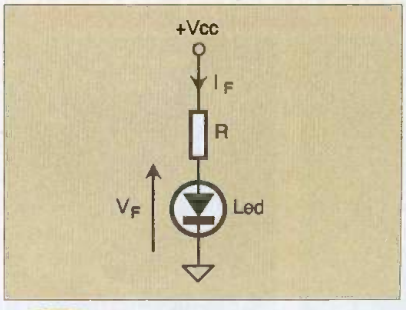
9 *Test d'une petite LED*



8 *Proposition de face avant*



10 *Mesure de VF*



11 *Calcul de R*

Nomenclature

R1, R2 : 680 Ω 1/2 W
R3 : 22 Ω 1/2W
R4, R5 : 47k Ω 1/2 W
R6,R7 : 2,2 kΩ 1/2 W
R8 : 10 Ω 1/2W
P1 : 470 Ω Log (voir texte)
C1, C2, C3 : 470nF
J1 : Connecteur pile 9V
J2 : Connecteur Tulipe (6 points)
J3 : Connecteur Picot Carré (6 points)
J4, J5 : Banane 4mm
Q1 : D44H11, TIP31 ou autre
D1 : LED Rouge
D2- : LED Verte
IC1 : TL431
IC2 : TS922 ou autre
S1 : Inter ON/OFF
B1 : Boîtier MultiComp type BC2 dispo chez Farnell ref: 223-554

(IC2 : doit être un amplificateur dont les VIL et VOL sont les plus faibles possible (noté : « negative rail input/output » dans les datasheets)).

Avant toute utilisation, ramenez le potentiomètre sur la valeur 0 mA évitant ainsi la destruction d'une LED qui ne supporterait pas la consigne de courant affiché.

Ensuite, connectez la LED dans le bon sens (voir la proposition de face avant comportant le schéma de la LED que l'on peut observer par transparence).

Puis alimentez le testeur avec l'interrupteur S1. Enfin augmentez le courant « raisonnablement » jusqu'à la valeur maximum admissible par cette LED. Les connecteurs bananes de 4 mm J4 et J5 peuvent avoir deux fonctions : la première permet de réaliser des essais sur une LED CMS ou de petite taille ne pouvant pas être insérée dans les connecteurs J2 ou J3 comme on peut le voir sur la **figure 9**.

La seconde utilisation des connecteurs J4 et

Le générateur de courant