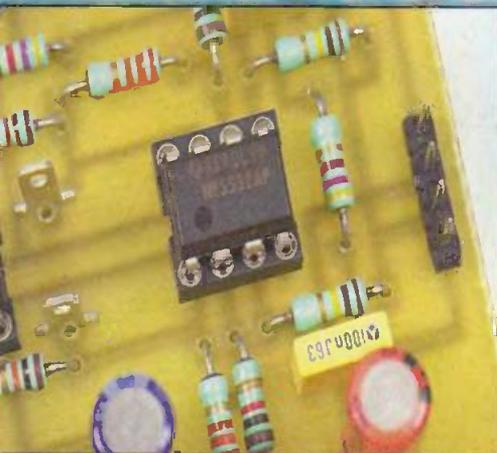
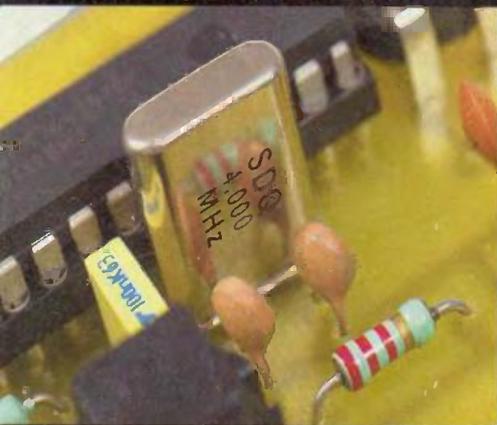


LES OUTILS DE DÉVELOPPEMENTS

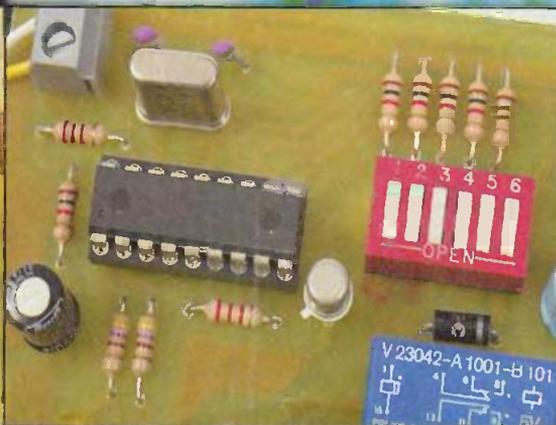
MATÉRIEL DE MISE EN ŒUVRE DES MICROCONTRÔLEURS



Casque antibruit



Thermomètre int/ext



Relais programmable

RETROUVEZ AUSSI :

- ▷ Réhausseur couleur pour DVD
- ▷ Journal lumineux
- ▷ La programmation des PIC

FRANCE : 25FF/3,81€ • DOM : 29FF/4,42€
BEL : 160FB/3,97€ • CH : 6,50FS
CAN : 5,95\$ CAN • ESP : 450PTA/2,70€
GR : 1500GRD/4,40€ • LUX : 160LUF/3,97€
MARD : 50DH • PORT : 920ESC/4,39€



SOMMAIRE

ELECTRONIQUE PRATIQUE

N° 262
DECEMBRE 2001/JANVIER 2002
I.S.S.N. 0243 4911

PUBLICATIONS GEORGES VENTILLARD

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

Président du conseil d'administration,
Directeur de la publication : Paule VENTILLARD
Vice-Président : Jean-Pierre VENTILLARD
Attaché de Direction : Georges-Antoine VENTILLARD
Directeur de la rédaction : Bernard FIGHIERA (84.65)
Directeur graphique : Jacques MATON
Maquette : Jean-Pierre RAFINI

Avec la participation de : U. Bouteville, A. Garrigou,
M. Laury, E. Lémery, P. Mayeux, Y. Mergy, P. Morin,
P. Oguic, A. Reboux, Ch. Tavernier, O. Viacava.

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

Directeur de la diffusion et promotion :

Bertrand DESROCHE

Responsable ventes :

Bénédicte MOULET Tél. : 01.44.84.84.54

N° vert réservé aux diffuseurs et dépositaires de presse :

0800.06.45.12

PGV - Département Publicité :

2 à 12 rue de Bellevue, 75019 PARIS

Tél. : 01.44.84.84.85 - CCP Paris 3793-60

Directeur commercial : Jean-Pierre REITER (84.87)

Chef de publicité : Pascal DECLERCK (84.92)

E Mail : lehp@le-hp.com

Assisté de : Karine JEUFFRAULT (84.57)

Abonnement/VPC: Voir nos tarifs en page intérieure.

Préciser sur l'enveloppe «SERVICE ABONNEMENTS»

Important : Ne pas mentionner notre numéro de compte
pour les paiements par chèque postal. Les règlements en
espèces par courrier sont strictement interdits.

ATTENTION ! Si vous êtes déjà abonné, vous faciliterez notre
tâche en joignant à votre règlement soit l'une de vos dernières
bandes-adresses, soit le relevé des indications qui y figurent.

* Pour tout changement d'adresse, joindre 3,00 F et la der-
nière bande.

Aucun règlement en timbre poste.

Forfait photocopies par article : 30 F.

Distribué par : TRANSPORTS PRESSE

Abonnements USA - Canada : Pour vous abonner à
Electronique Pratique aux USA ou au Canada, commu-
niquez avec Express Mag par téléphone :

USA : P.O. Box 2769 Plattsburgh, N.Y. 12901-0239

CANADA : 4011 boul. Robert, Montréal, Québec, H1Z4H6

Téléphone : 1 800 363-1310 ou (514) 374-9811

Télécopie : (514) 374-9684.

Le tarif d'abonnement annuel (11 numéros) pour les USA
est de 49 \$US et de 68 \$Can pour le Canada.

Electronique Pratique, ISSN number 0243 4911, is published 11

issues per year by Publications Ventillard at P.O. Box 2769

Plattsburgh, N.Y. 12901-0239 for 49 \$US per year.

POSTMASTER : Send address changes to Electronique Pratique,

c/o Express Mag, P.O. Box 2769, Plattsburgh, N.Y., 12901-0239.

Réalisez vous-même

- 48 Casque antibruit
- 52 Réhausseur de couleur pour lecteur DVD
- 56 Testeur de LED
- 60 Relais programmable
- 66 Thermomètre intérieur/extérieur avec mini et maxi
- 72 Driver de câble coaxial
- 78 Journal lumineux à persistance rétinienne pour PIC

Dossier spécial

«OUTILS de DÉVELOPPEMENTS»

- 22 Intro - 23 : REALIZER - 26 : Kit de développement pour carte à puce ASE II - 28 : Clés de protection eToken sur port USB - 32 : BASIC TIGER 34 : 5X EMUL KIT - 36 : Programmateur universel ALL-11P2 - 40 : Kit de développement pour module de reconnaissance vocale VOICE EXTREME 364 - 43 : Emulateur temps réel MR PIC

Montages FLASH

- 10 Émetteur de sécurité 16 canaux
- 13 Récepteur 16 canaux sécurisé

04 Infos OPPORTUNITÉS

DIVERS

- 18 Internet Pratique
- 88 À propos de la programmation des PIC

Un encart ELECTRONIQUE DIEFUSION de 8 pages, paginé de I à VIII, au centre de la revue sur tout le tirage



« Ce numéro
a été tiré
à 52 700
exemplaires »



Le siège d'ELECTRONIQUE Diffusion déménage et s'agrandit!

Historique d'un succès d'entreprise...

La société ELECTRONIQUE DIFFUSION voit le jour en 1982, à ROUBAIX et comprend alors 3 salariés. Le magasin, qui s'étend sur 300 m², s'adresse à une clientèle passionnée par l'électronique. Sa recette : Proposer des opportunités, un choix important, une diversité de produits en matière de composants et pièces détachées électroniques, et cela, aux meilleurs prix.

1986 : Sortie du 1er catalogue « Electronique Collège » qui marque une volonté de démocratiser les prix du composant, auprès de l'enseignement et du grand public.

-Ouverture du deuxième point de vente à LILLE.

1987 : Ouverture du magasin d'ARRAS

1989 : Ouverture des magasins de DOUAI et de LUNEL.

1990 : L'entreprise modifie son statut et passe en Société Anonyme : elle compte à cette date, 25 salariés et réalise un chiffre d'affaire de 30 MF.

Création à Roubaix d'un centre de logistique de 1000 m², destiné à traiter les commandes et approvisionner les différents magasins.

1992 : Ouverture du magasin de DUNKERQUE.

1993 : Ouverture du magasin de VALENCIENNE.

1994 : Ouverture des magasins de PARIS (MALAKOFF) et de LYON.

1995 : Agrandissement du point de vente de LILLE

Elargissement de sa gamme par l'introduction de produits finis en matière de Son, Lumières, Audio/Vidéo... Modernisation de l'image et du LOGO de l'entreprise.

1997 : Refonte complète du catalogue par l'insertion très large, des différentes familles de produits électroniques...

Mise en place d'un nouvel outil informatique plus performant.

1998 : Ouverture des magasins de ROUEN et GENTILLY.

2000 : Création du site internet et mise en place du commerce en ligne.

Le PDG fondateur, Mr Jacques MOREL, transmet l'entreprise à son fils Philippe.

2001 : Déménagement du siège social de ROUBAIX à WERVICQ, dans un nouveau local de 8500 m² sur un site de 2 hectares.



L'entreprise aujourd'hui...

ELECTRONIQUE DIFFUSION occupe actuellement une place prépondérante sur le marché national puisqu'elle se situe parmi les 200 premières entreprises françaises dont l'activité est axée sur l'électronique, et dans les cinq premières, si l'on considère les spécialités qui la caractérisent. Elle emploie, à ce jour, une cinquantaine de personnes réparties sur le siège social et les 11 points de vente nationaux.

Elle gère au total 45 000 références, dont plus de 25 000 en composants actifs, pour un chiffre d'affaire 2001 supérieur à 50 MF.

En conclusion

Après avoir visité les nouveaux locaux d'ELECTRONIQUE DIFFUSION, il était important pour nous de vous retracer le parcours à succès de cette entreprise qui nous accompagne dans le paysage électronique français, depuis 20 ans. Avec cette réorganisation structurelle, l'entreprise dotée d'un nouvel outil logistique performant et évolutif, se donne les moyens de sa réussite future, tant par ses capacités de stockage que ses services nouveaux. Elle perpétue l'esprit de conquête insufflée par une direction et une équipe motivée et passionnée par son métier; ce qui est de bonne augure pour notre profession...

ELECTRONIQUE DIFFUSION

Avenue de la Victoire - 59117 Wervicq-sud

Tél. : 03 28 04 30 60

<http://www.elecdiff.com>



Compilateur Basic BASCOM-8051©

La société OPTIMINFO annonce la vente d'un nouveau compilateur BASIC BASCOM-8051© permettant d'écrire des programmes en BASIC pour la famille des μ 8051 (AT89C1051, AT89C2051, 8031, 8032, 8051, 8052, 80552, 80535 et 80537) et fonctionnant sous les environnements Windows 95/95/2000 et NT.

Il utilise la structure BASIC compatible Microsoft VB/QB avec étiquettes, les mots clés standards (IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT CASE, Byte, Integer, Word, Long, Single) et les chaînes de caractères variables. Des bibliothèques incluses autorisent l'utilisation des afficheurs LCD, des composants I2C, du bus I2WIRE ainsi que les fonctions standards de manipulation de Bit, de variables, de chaînes de caractères, de directives de compilations.

OPTIMINFO
Route de Ménétreau
18240 BOULLERET
Tel : 0820 90 00 21
Email : Commercial@optiminfo.com



L'émulateur terminal intégré permet le dialogue avec le port série lors du fonctionnement du programme sur la carte. Un simulateur logiciel testera le fonctionnement en visualisant les variables, soit de tout le programme ligne par ligne, soit d'une ligne spécifique, ou bien des variables en les sélectionnant ainsi que l'émulation d'un afficheur LCD et des portes d'entrées sorties.

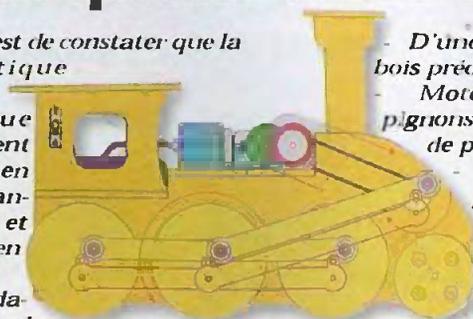
Le logiciel offre un environnement de travail couleur multi-fenêtres avec des fonctions d'éditations standards comme le retour arrière, le copier-coller...

Vendu à très bas prix, ce nouveau logiciel sera apprécié des développeurs pour sa simplicité d'utilisation et sa rapidité de programmation. La version d'évaluation est téléchargeable sur le site www.optiminfo.com

Les ROBOTS EN...BOIS ! débarquent chez SELECTRONIC

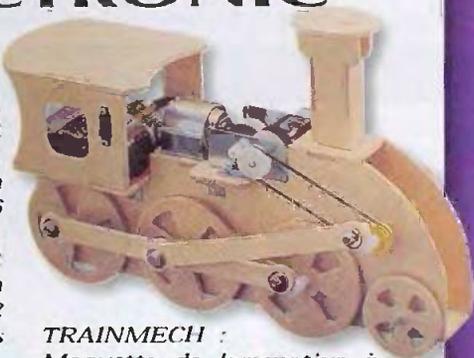
Force est de constater que la Robotique

indique à la vent en poupe en cette fin d'année 2001 et nous nous en félicitons. L'intérêt pédagogique et évolutif du sujet n'est pas étranger à ce succès. Aussi, pour rester dans l'esprit du moment et compléter sa gamme déjà riche, la société SELECTRONIC nous livre, en avant première, le détail de sa nouvelle collection de kits de Robots motorisés en bois.

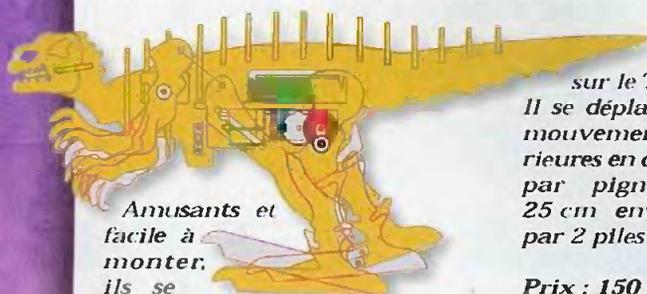


- D'une structure en bois prédécoupée,
- Moteurs, poulies, pignons, axes, support de piles, accessoires,
- Alimentation par piles R6 (non fournies).
6 modèles sont d'ores et déjà disponibles. 2 genres de robots

composent la gamme : Les robots «marcheurs» aux designs de dinosaures tels que le TYRANNOMECH, le STEGOMECH et le ROBOMECH. Puis les robots «roulants» tels que le TRAINMECH, l'AUTOMECH et le COPTERMECH.



TRAINMECH : Maquette de locomotive à vapeur mobile avec mouvement des bielles. Il se déplace en roulant. Transmission par pignonnage et poulies entraînées par élastique. Longueur : 30 cm environ, Alim par 2 piles de type R6. Prix : 150 FTTC environ (22,87 €)



Amusants et facile à monter, ils se composent en général :

Pour mieux comprendre, gros plan sur le TYRANNOMECH : Il se déplace en marchant avec mouvement des pattes antérieures en cadence. Transmission par pignon, hauteur 25 cm environ, Alim par 2 piles de type R6.

Prix : 150 FTTC environ (22,87 €)

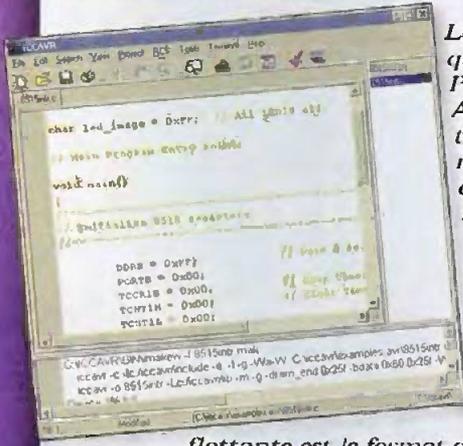


A n'en pas douter, ces petits robots séduisent et donnent l'envie de faire des cadeaux !

Distribué par : SELECTRONIC
Paris : 01 55 25 88 00 - Lille : 03 28 55 03 28
www.selectronic.fr

Compilateurs C pour 68HC11, HC12, HC16 et AVR

La société **CONTROLORD**, qui commercialise des compilateurs C et Basic pour le 68HC11 depuis longtemps, annonce la commercialisation des compilateurs C pour le 68HC12, 68HC16 de MOTOROLA ainsi que pour le AVR de ATMEL de la société ImageCraft.



Les compilateurs utilisent un langage qui est très proche de la norme ANSI. Pourtant quelques fonctionnalités du ANSI C ne sont pas adaptées à un système embarqué. Le langage est légèrement corrigé à la vitesse limitée d'un microprocesseur de 8 bits et limite l'espace du code généré.

Le compilateur connaît des types char 8 bits, short, int 16 bits, long 32 bits, pointeur 16 bits. Il supporte également le type virgule flottante float et double à 32 bits : exposant 8 bits, mantisse 24 bits. Le format utilisé pour les nombres en virgule

flottante est le format de la norme IEEE.

Chaque compilateur comprend la chaîne complète sous Windows : pré-processeur, compilateur, assembleur, link. La bibliothèque standard du C est livrée en source et en format de librairie.

CONTROLORD donne gratuitement aux professeurs qui veulent enseigner la pratique du C sur un microcontrôleur des travaux pratiques en C sur HC11.

Présent à **EDUCATEC** du 21 au 24 novembre 2001, Paris - Porte de Versailles

Email : controlord@controlord.fr

Site : <http://www.controlord.fr>

Jouets : la tendance pour 2001

L'élection "Top 10 des jouets Duracell" existe depuis 13 ans aux Etats-Unis.

L'étude est reconnue comme un excellent baromètre des futures ventes de jouets à Noël.

Cette version européenne de l'étude semble déjà convaincre les fabricants de la pertinence de ses informations et des tendances de choix pour les cadeaux de Noël.

En plus de ce classement, l'élection "Top 10 des Jouets Duracell" est l'occasion d'étudier les paramètres motivant le choix des jouets, ainsi que les activités pratiquées par les enfants durant leurs loisirs.

La télévision possède un important "rôle de prescription" dans le choix des jouets : 70% des enfants ont choisi leurs jouets

après les avoir vus dans une publicité à la télévision, alors que 34% ont suivi les conseils de leurs petits amis.

Pour répondre à l'ensemble des attentes des consommateurs,

Duracell propose 2 gammes principales de piles différenciées :

Duracell Ultra M3 : la nouvelle pile alcaline la plus puissante du monde dans les appareils photo numériques, les torches halogènes, les jouets télécommandés, etc.

Duracell Plus : la pile aux performances longue durée conçue pour répondre aux besoins des applications d'aujourd'hui (lecteurs CD portables, appareils photo, etc.)

Disponibles dans les grandes surfaces et magasins spécialisés.



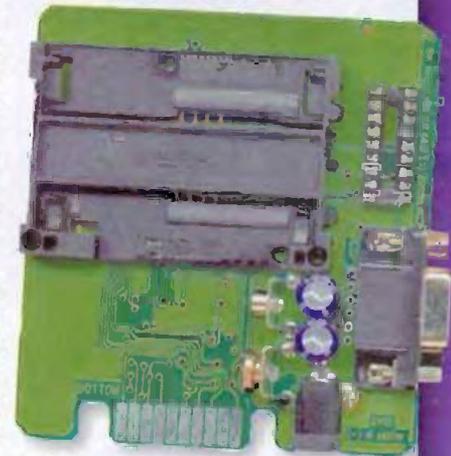
ADTEKNIK'S : multiprogrammateur version 3.4 et 3.5

Le multiprogrammateur **ADTEKNIK'S** est un programmeur qui permet notamment de programmer les microcontrôleurs PIC 16x84 et les EEPROM de MICROCHIP ainsi que les cartes GOLD, PIC-CARD2, FUNCARD.

Le programmeur est construit autour d'un PIC 16F622A de MICROCHIP qui a 2048 mots de mémoire morte et 128 bytes de mémoire vive.

Le multiprogrammateur est alimenté par une pile standard de 9V de type 6F22, mais il est possible d'utiliser un adaptateur secteur de 9V continu.

Le programmeur est capable de charger un programme dans une mémoire interne qui permet de programmer l'EEPROM sur des SMARTCARDS (ou aussi carte GOLD) qui n'est pas connectée physiquement.



Cette fonction est généralement appelée "Through-Pic-Programming" ou en français "Programmation à travers le PIC". Elle est nécessaire pour la plupart des SMARTCARDS avec EEPROM intégrée comme la carte GOLD mais aussi la GALAXY2, PLATIUM, CBMM2.

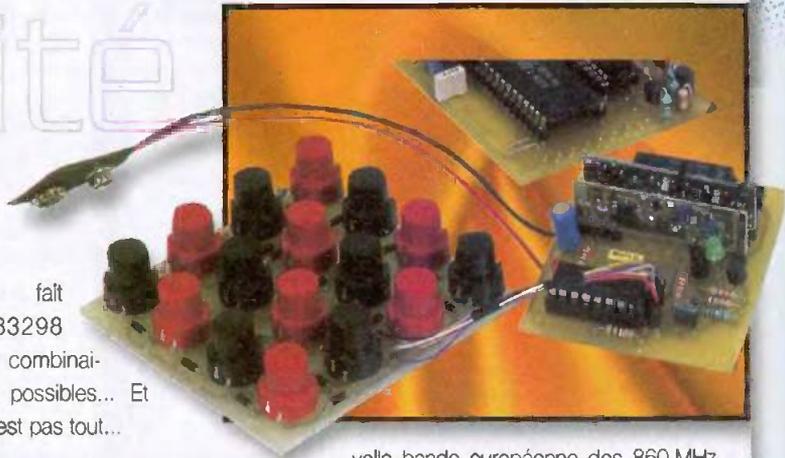
Le multiprogrammateur a un support SMARTCARD pour les cartes SMART compatibles à la norme ISO7816.

Elle est équipée d'un support DIL pour programmer toutes les puces MICROCHIP jusqu'à 18 pattes.

Prix : 699 F. TTC (106,56 €)
Distribué par : OMINFO.com

Émetteur de sécurité 16 canaux

Émetteur de sécurité



▶ A quoi ça sert ?

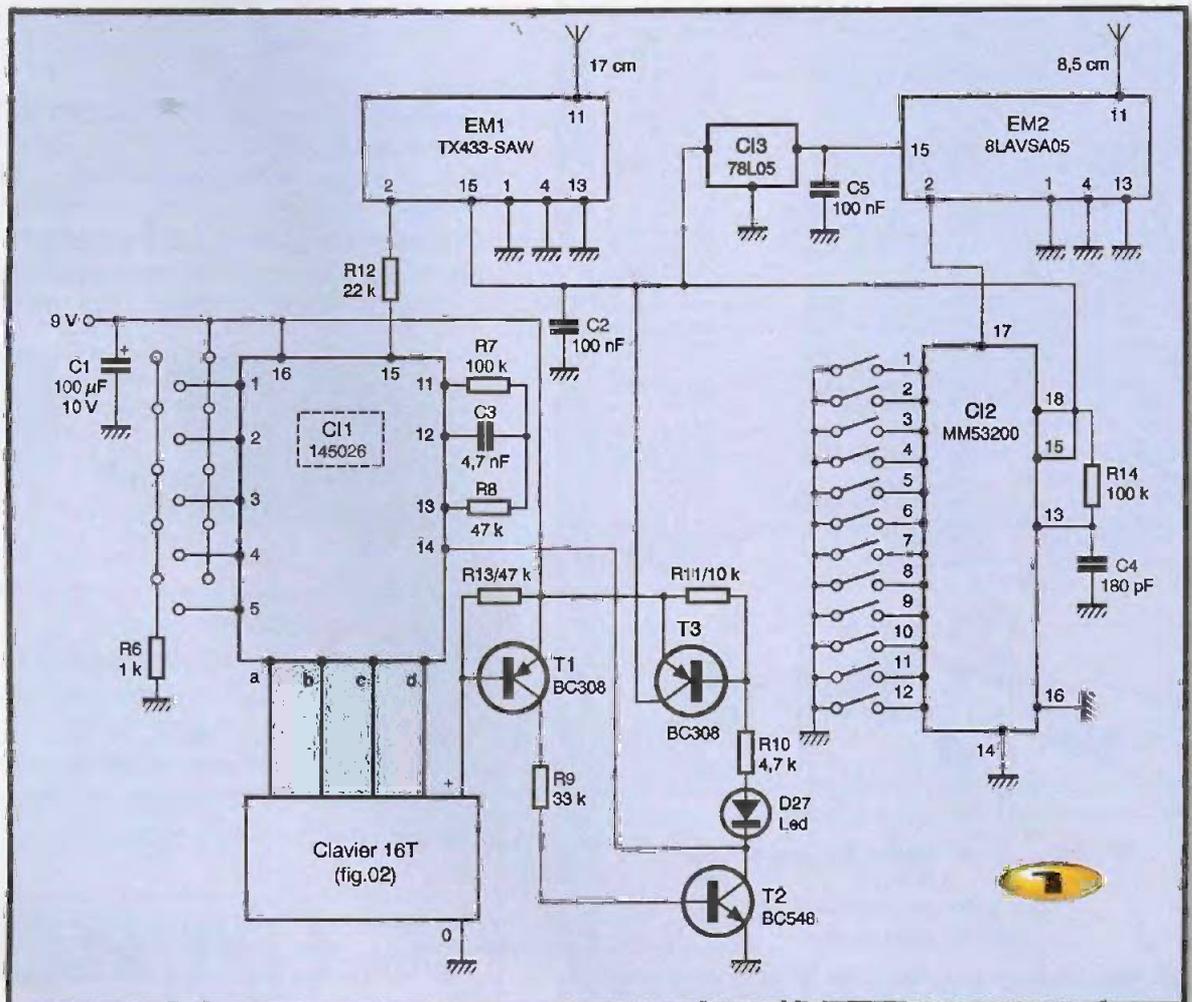
Les émissions codées bénéficient d'une certaine sécurité tant que les émetteurs ne sont pas trop répandus. Les émetteurs multi-canaux, basés sur les 145026, utilisent les bits d'adresse pour transmettre des données et, comme il en faut 4 pour les 16 canaux, on ne code plus que sur 5 bits. Notre émetteur de sécurité utilise un codage d'adresse sur 17 bits dont 5 en ternaire, ce qui

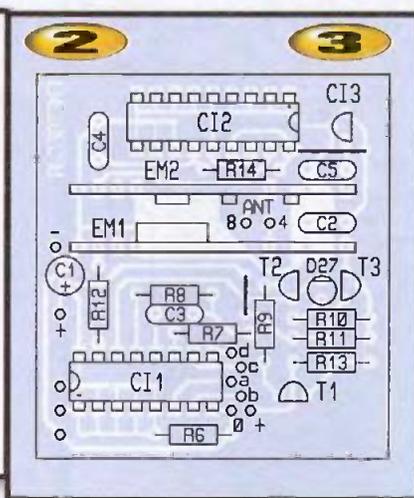
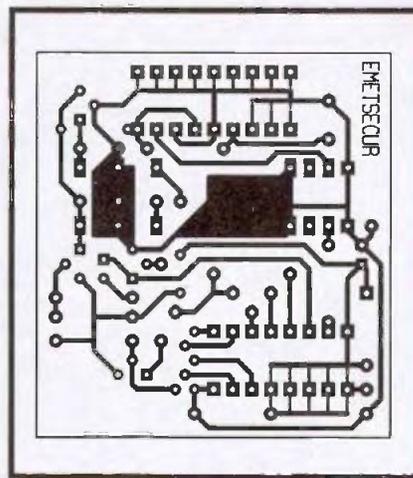
nous fait 12383298 de combinaisons possibles... Et ce n'est pas tout...

Comment ça marche ?

Ce système sécurité a été rendu possible par l'utilisation de nouveaux modules AUREL travaillant dans la nou-

velle bande européenne des 860 MHz et des poussières. Cette bande a défini tout un ensemble de fréquences autorisées avec des puissances diverses et un taux d'occupation variable. L'émission permanente a droit à une puissance inférieure à une émission de





courte durée, ce qui se comprend fort bien. Dans notre système, nous allons utiliser non seulement deux codeurs, un sur 12 bits et un sur 5 bits plus ceux des données. Le principe de l'émission des 16 canaux est celui que nous avons développé pour des émetteurs précédents. En fait, un émetteur 16

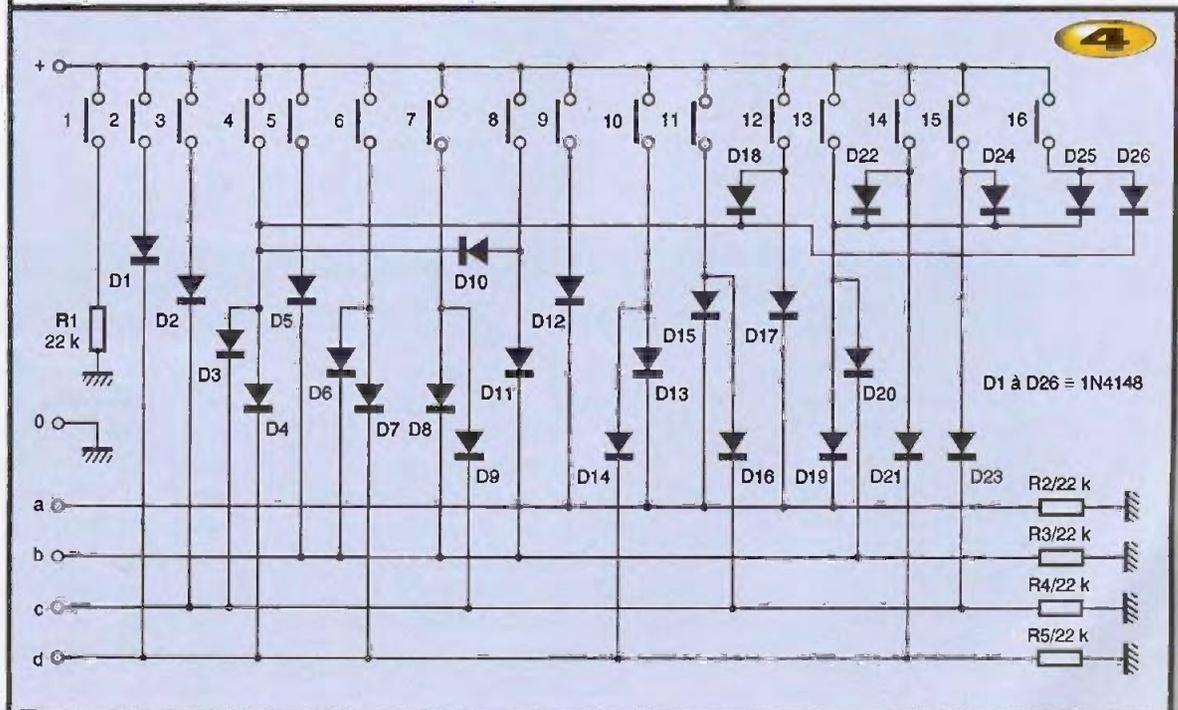
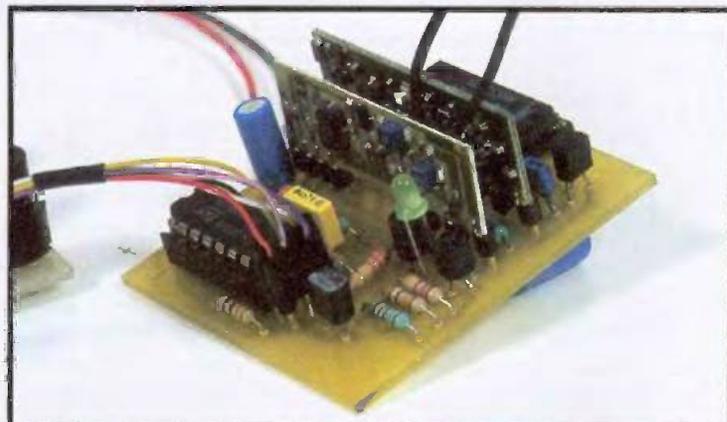
n'est rien d'autre qu'un émetteur 4 canaux dit "simultanés", la multiplication par 4 du nombre des canaux résultant d'un décodage de la situation respective des canaux "simultanés"...

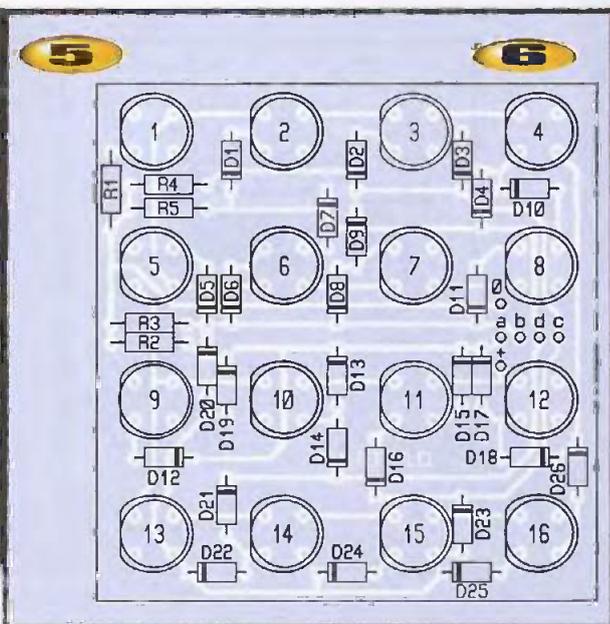
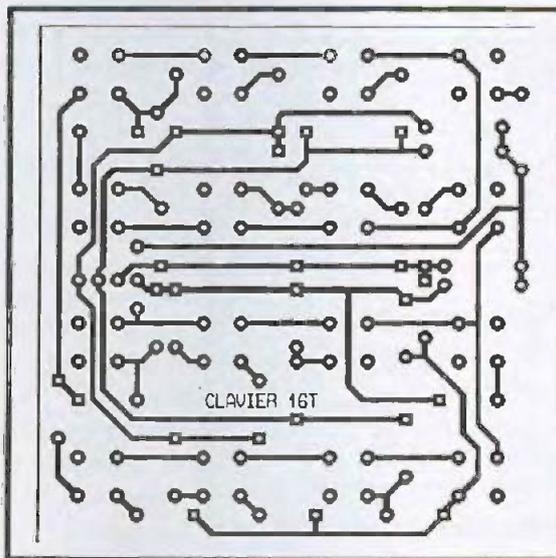
L'émetteur utilise deux modules d'émission, l'un accordé sur 433 MHz, l'autre sur 868 MHz. Chacun émet un code dif-

férent, mais ils ne sont pas interchangeable, l'un s'alimente sur 9V et l'autre 5V. L'alimentation est un problème que l'on rencontre aujourd'hui. Un fabricant comme AUREL diversifie ses productions et ne se limite plus à des émetteurs fonctionnant sous une tension de 9 à 12V. Des émetteurs très basse tension aussi puissants que les précédents, mais fonctionnant sous 3V, apparaissent. Ils utilisent la même implantation des connexions, mais ne supporteront pas d'excès de tension.

On reconnaîtra dans la partie de gauche du schéma un émetteur 16 canaux standard avec son codeur basé sur un 145026. Les 5 broches de 1 à 5 seront soit laissées en l'air, soit portées à un potentiel positif, soit mises à la masse. Ces bits sont ceux du codage des adresses. Les quatre autres bits du codeur vont déterminer la donnée à transmettre avec, cette fois, deux possibilités : soit une mise au plus, soit à la masse. Cette opération s'effectue par l'intermédiaire d'un clavier à diodes servant de codeur.

Dès que le clavier est actionné, le transistor T₁ conduit puis T₂ et enfin T₃ qui alimente l'émetteur et la seconde partie. Cette dernière utilise un autre module d'émission AUREL travaillant sur l'autre fréquence. Compte tenu des différences de tension d'alimentation, ces deux





modules ne sont pas interchangeables. Le codeur qui générera le second message de sécurité est un MM53200 de NS alimenté sous une tension de 9V. Son horloge se contente d'un simple réseau RC. Le codage se fait par une mise à la masse pure et simple à moins qu'on ne laisse l'entrée en l'air. La liaison entre la sortie du circuit et l'entrée du module d'émission est directe, cette entrée se fait sur une résistance, le circuit d'entrée devrait bien supporter cette haute tension.

Le circuit est alimenté par un régulateur de tension classique et facile à trouver...

Réalisation

Le clavier est, comme nous l'avons dit, identique à celui précédemment utilisé. Attention, si vous utilisez un codeur de MOTOROLA type MC 145026, les résistances de R₁ à R₅ devront être de 22 k Ω , une valeur plus importante peut entraîner l'absence de fonctionnement. Cette dernière n'existe pas avec les circuits ST-

MICROELECTRONICS M145026 et non MC...

Le circuit imprimé a été dessiné pour permettre également l'utilisation d'un décodeur à 48 sorties (voir EP 259, juillet/août 2001), le codage de la broche 1 sera remplacé par un inverseur à trois positions.

Le circuit imprimé comporte des pistes étroites à couper pour assurer le codage de l'émetteur, on fera attention à ne pas laisser les deux pistes sur l'une des broches, il en suivrait un court-circuit limité toutefois par la résistance R₆ qui justifie ainsi sa présence.

On fera attention au code des couleurs des résistances, il est facile, avec certaines résistances de se tromper entre un anneau rouge un peu orangé et un orange un peu rouge ! Une erreur de ce type peut interdire tout fonctionnement...

La liaison avec le clavier est un peu différente de celle de la version précédente, en effet, les pistes du décodeur associé

ne sont pas organisées tout à fait de la même façon...

Le montage ne pose pas de difficulté particulière.

Les antennes seront constituées de fil de cuivre souple ou rigide. La longueur d'antenne 433 MHz est de 17cm, celle de celui de 863 MHz de 8,5cm...

Évitez tout de même de ne pas faire de tresse avec les deux antennes, ce ne serait pas très efficace sur le plan du rayonnement.

L'émetteur sera essayé avec le récepteur et vice versa, on vérifiera tout de même que la consommation de l'émetteur au repos est nulle, sinon, un composant a sans doute un problème. La diode D₂₇ doit être éteinte en l'absence d'émission, même dans le noir.

E. LEMERY

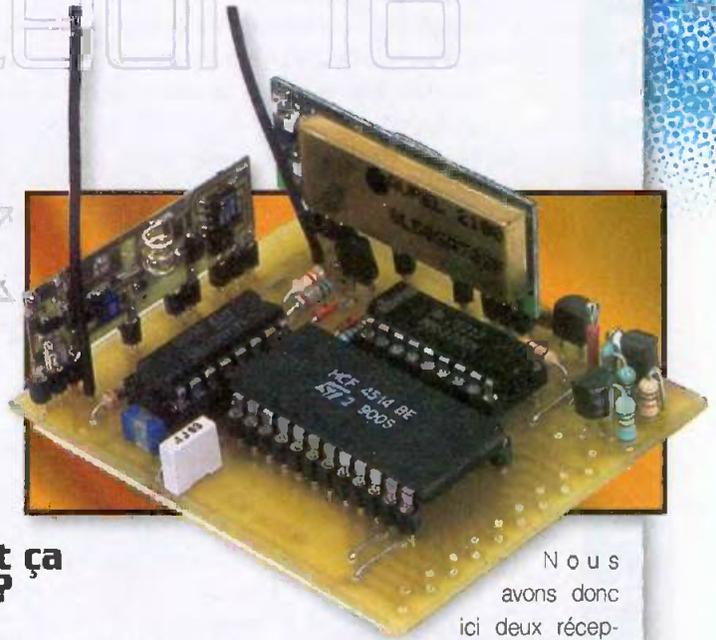
Nomenclature

R₁ à R₅ : 22 k Ω 1/4W 5%
(rouge, rouge, orange)
R₆ : 1 k Ω 1/4W 5%
(marron, noir, rouge)
R₇, R₁₄ : 100 k Ω 1/4W 5%
(marron, noir, jaune)
R₈, R₁₃ : 47 k Ω 1/4W 5%
(jaune, violet, orange)
R₉ : 33 k Ω 1/4W 5%
(orange, orange, orange)
R₁₀ : 4,7 k Ω 1/4W 5%
(jaune, violet, rouge)

R₁₁ : 10 k Ω 1/4W 5%
(marron, noir, orange)
R₁₂ : 22 k Ω 1/4W 5%
(rouge, rouge, orange)
C₁ : 100 μ F/10V chimique radlal
C₂, C₅ : 100 nF céramique
C₃ : 4,7 nF MKT 5mm
C₄ : 180 pF céramique
Cl₁ : M ou MC145026
Cl₂ : MM53200
Cl₃ : 78L05

T₁, T₃ : transistors PNP BC308
T₂ : transistor NPN BC548
D₁ à D₂₆ : diodes silicium 1N4148
D₂₇ : diode électroluminescente 3mm
EM₁ : module émetteur AUREL TX 433 SAW
EM₂ : module émetteur AUREL 8LAVSA05
Touches BP1_n : connecteur pile 9V

Récepteur 16 canaux sécurisé



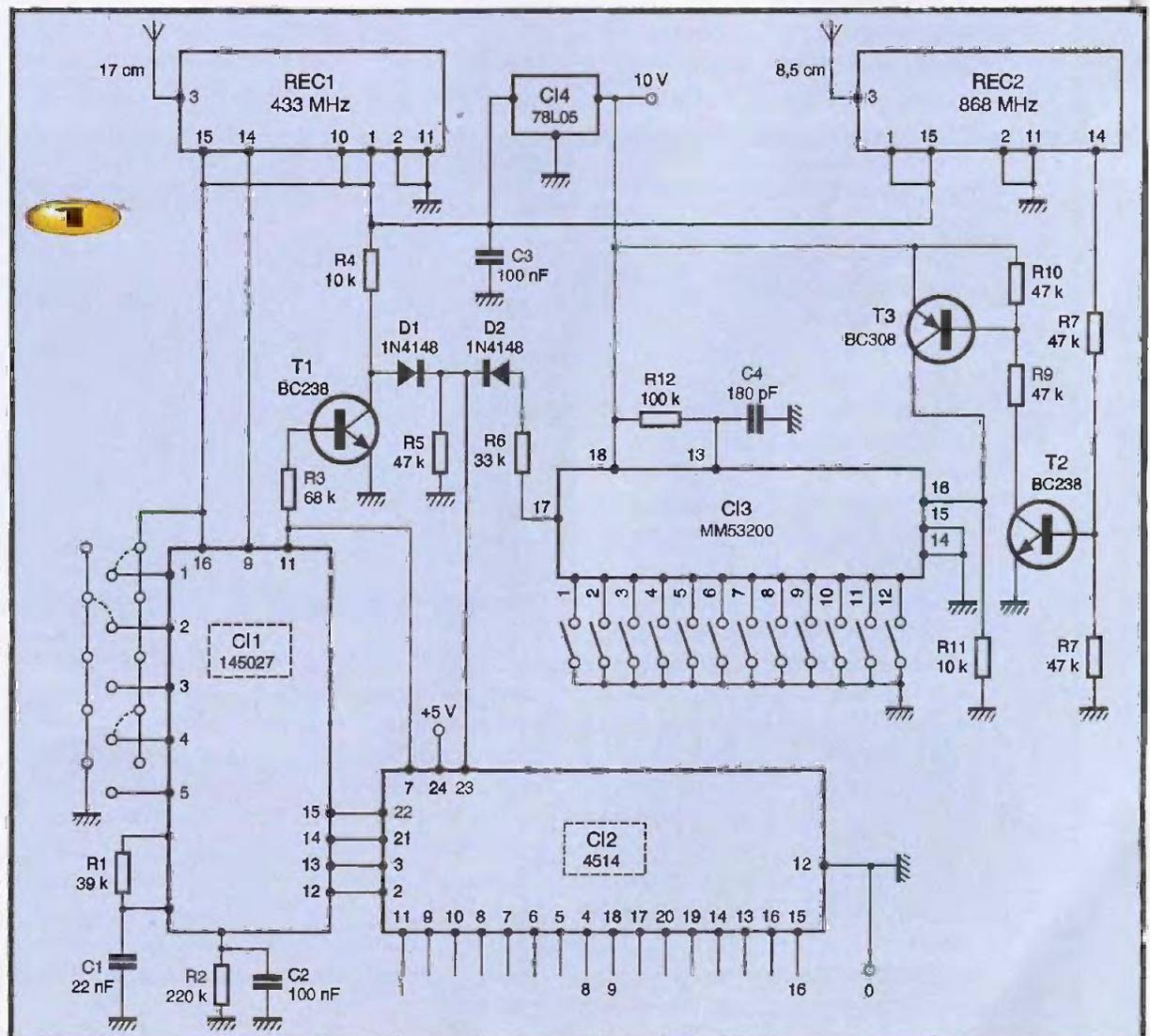
A quoi ça sert ?

Le récepteur sécurisé est capable de recevoir les ordres de l'émetteur sécurisé... Les risques de sortie du décodeur d'un ordre erroné seront limités par la présence d'un double signal codé. L'absence de l'un d'eux interdisant la sortie du signal...

Comment ça marche ?

A l'émission, nous avons envoyé simultanément deux codes différents : un sur 433 MHz et l'autre sur 868.

Nous avons donc ici deux récepteurs, un modèle classique à super-réaction et très répandu pour recevoir les données



en 433 MHz et un superhétérodyne travaillant sur la plus récente fréquence de 868 MHz. Ces deux récepteurs sont alimentés par un régulateur 5V à faible courant, ce régulateur est indispensable pour assurer l'alimentation du décodeur MM53200.

Ce circuit intégré, de conception relativement ancienne, a en effet besoin d'une tension supérieure à 5V pour fonctionner correctement. Il a aussi besoin de données d'une tension supérieure à 5V crête à crête pour travailler, ce qui a imposé la présence d'un amplificateur qui conserve la phase du signal.

En effet, si on se contente d'inverser la phase du signal numérique, les données ne seront pas reconnues... Les fabricants de récepteurs ne donnent pas suffisamment de renseignements concernant les circuits de sortie de leurs modules, on ne peut donc réaliser d'adaptation de niveau simplifiée. Par exemple, une sortie à collecteur ouvert permettrait d'assurer la fonction avec une seule résistance !

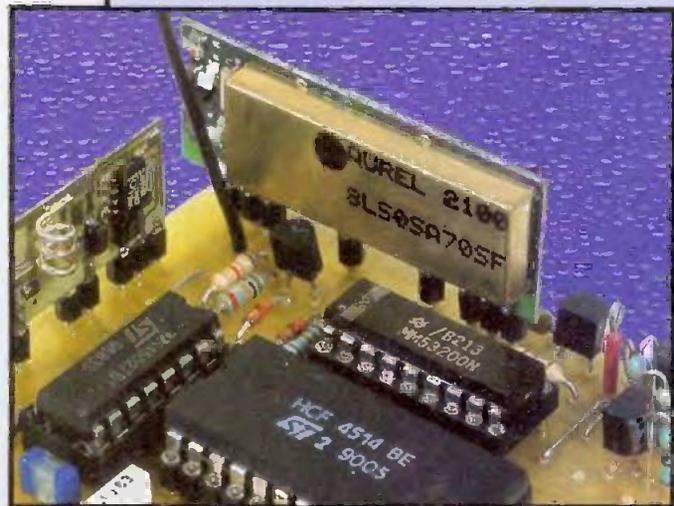
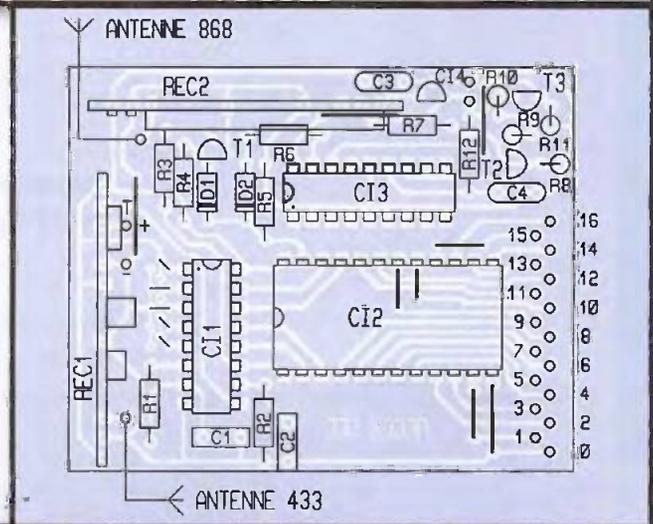
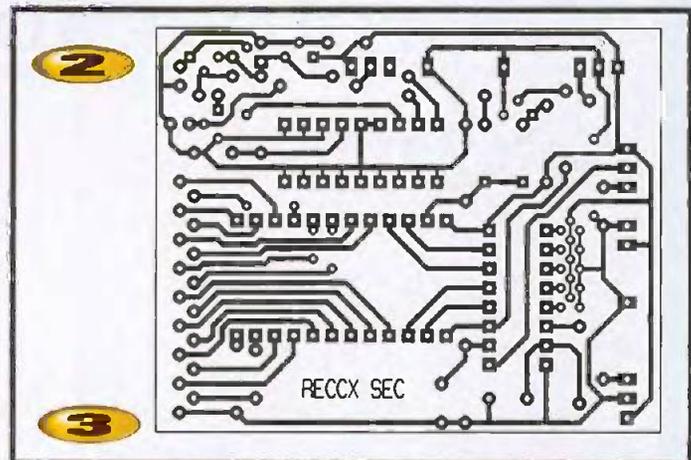
Les données et leur adresse sont envoyées sur le décodeur 145027 (ne pas utiliser de 145028). Les quatre bits de sortie du décodeur partent vers les quatre entrées d'un décodeur binaire à 16 sorties. Le 145027 conserve ses sorties en l'état de la dernière réception. On envoie donc des signaux sur des entrées auxiliaires d'autorisation du décodeur afin de rendre les sorties temporaires.

Le signal de validation est inversé par le transistor T_1 afin de permettre son exploitation avec une polarité correcte par $C1_2$.

Le circuit intégré $C1_3$ reçoit les signaux de sortie de REC_2 qui ont été amplifiés par T_2 et T_3 . La constante de temps est ici fixée par R_{12} et C_4 . Les composants sont identiques à ceux de l'émetteur. La sortie est sur la borne 17. La résistance R_6 se charge de l'adaptation du niveau et évite l'envoi d'une tension supérieure à 5V à l'entrée de $C1_2$. Les deux diodes D_1 et D_2 servent de porte ET, il faut que le collecteur de T_1 et la sortie 17 de $C1_3$ soient simultanément à zéro pour que les données de sortie de $C1_1$ soient exploitées par $C1_2$ et leur code 4 bits décodé.

Si vous désirez davantage de canaux, vous pouvez reprendre la formule utilisée dans le numéro 259 de EP. On mul-

tipliera les décodeurs $C1_1$ ainsi que les transistors T_1 , chaque cathode de D_1 et ses consœurs sera reliée à R_6 par une diode placée de la même façon que D_2 . Leur anode sera commune. Il faudra changer la valeur de R_6 : 15 k Ω pour une version à 32 canaux et 10 k Ω pour 48, la résistance alimentera en effet deux ou trois R_9 placées en parallèle... Le circuit imprimé de l'émetteur comporte l'emplacement pour un inverseur à trois positions.



Réalisation

Le circuit imprimé reçoit tous les composants y compris le petit régulateur.

Les récepteurs sont montés sur supports à tulipe, si vous désirez les souder, vous devrez plier le transistor T_1 afin qu'il n'entre pas en contact avec le blindage du récepteur. Il y a deux straps sous $C1_2$, on n'oubliera pas de les câbler, surtout si on soude les circuits intégrés...

Certaines résistances sont placées verti-

calement, on les orientera de façon à ce que leur fil nu ne puisse entrer en contact avec les autres.

REPÈRES

IMPORTATION AUREL :

P2M, 8 Allée des Châtaigniers
ZA du Buisson de la Coudre
78190 Trappes

VENTE GP :

SELECTRONIC, etc.

Le code du 145026 de l'émetteur sera reproduit par les cavaliers du 145027. Toute différence interdira la réception. On respectera aussi la valeur des composants fixant les constantes de temps internes, un trop grand écart risque de perturber le fonctionnement normal. On vérifiera la réception d'un signal valide sur la broche 11 du 145027, elle passe à l'état haut en cas de bonne réception.

Pour le MM53200, la sortie 17 passe à l'état bas lors de la réception du signal à 868 MHz correctement codé.

Lorsque les deux codes correspondent,

la borne 23 du 4514 autorise le découpage et la sortie associée à la touche de l'émetteur passe à un niveau haut.

Le circuit 4515 peut remplacer le 4514, la polarité des sorties sera opposée : un état haut permanent et un état bas fugitif.

Les sorties sont au standard CMOS, on fera suivre, si nécessaire, ce circuit d'interfaces de puissance (voir hors série Montages flash juillet/août 2000).

E. LEMERY

Nomenclature

R₁ : 39 kΩ 1/4W 5%
(orange, blanc, orange)
R₂ : 220 kΩ 1/4W 5%
(rouge, rouge, jaune)
R₃ : 68 kΩ 1/4W 5%
(bleu, gris, orange)
R₄, R₁₁ : 10 kΩ 1/4W 5%
(marron, noir, orange)
R₅, R₇ à R₁₀ : 47 kΩ 1/4W 5%
(jaune, violet, orange)

R₆ : 33 kΩ 1/4W 5%
(orange, orange, orange)
R₁₂ : 100 kΩ 1/4W 5%
(marron, noir, jaune)
C₁ : 22 nF MKT 5mm
C₂ : 100 nF MKT 5mm
C₃ : 100 nF céramique
C₄ : 180 pF céramique
D₁, D₂ : diodes silicium 1N4148
T₁, T₂ : transistors NPN BC238

T₃ : transistor PNP BC308
C₁ : M ou MC145027
C₂ : CD4514
C₃ : MM53200 (ou équivalent)
C₄ : 78L05
REC₁ : récepteur AUREL 433 MHz
super-réaction, modèles divers :
BC-NB, RF290, NB-05M.
REC₂ : récepteur AUREL
868 MHz 8L50SA70SF

N°1

DANS LE MONDE
DU CIRCUIT IMPRIMÉ,

nous mettons à votre service, notre expérience,
notre catalogue et notre réseau de distributeurs...
Ensemble, devenons partenaire de votre réussite !

Demandez notre catalogue pour tout savoir sur les circuits imprimés ; câblages électroniques ; micro-électronique ; équipements ; produits ; accessoires et services que nous vous proposons.

Gratuit pour les professionnels, les enseignants. Pour les particuliers, cher leurs revendeurs ou contre 12 frs en timbres



•ÉTUDE
•ÉDITION
•FABRICATION

11, rue Charles-Michels - 92220 Bagneux - France
Tél : 33 (0) 1 4547 4800 - Fax : 33 (0) 1 4547 1614
E-mail : cif@cif.fr - Web : http://www.cif.fr

MULTIPROGRAMMATEUR

Suprême programmeur qui peut programmer :
PIC16F84A, PIC16F84, PIC16C84, PIC12C508,
PIC12C509, PIC16C622, PIC16F628, PIC16F876
et eeproms, Funcards, Jupiter, 1 and 2 etc.

Through-pic programming utilisable pour Goldcards,
PICcard2 et Funcards etc. de manière TRANSPARENTE
pour l'utilisateur. C'est le programmeur le plus simple à
utiliser. Pas besoin de "Loader" pour les cartes Goldwafer...
tout se fait AUTOMATIQUEMENT grâce à des mémoires et
des PIC intégrés. Programme en une passe les cartes
goldwafer, les Funcard 2, les Silvercard 2, les Pic-card v1, v1.1 & v2.

15 €
Goldwafer
timbre 40g essai par quartet

30 €
Funcard 2
timbre 40g essai par quartet

27 €
Silvercard 2
timbre 40g essai par quartet

Mini Titanium Plus

34,90€

Smarcard + Phoenix

74,90€

106€

Spécial Playstation 2
PUCE
NEO4
EXCLUSIVITE

Livraison 48H

OMINFO.COM

1801
32005 CHARENTAIS CEDEX
161 03 25 31 47 23
fax : 03 25 31 69 70

RCS : CHARENTAIS 436 150 232

Achetez sur ce site en toute
sécurité

www.ominfo.com

Le mois dernier nous vous proposons de découvrir les alimentations. Ce mois-ci nous vous emmenons à la découverte des diodes électroluminescentes (Diodes LED ou DEL) et des afficheurs 7 segments.

internet PR@TIQUE

Pas de doute, les diodes LED attirent. Si la théorie du principe de fonctionnement des diodes électroluminescentes est relativement complexe, la

mise en œuvre de ces composants est d'une très grande simplicité, comme nous vous proposons de le découvrir grâce aux quelques sites Internet que nous vous proposons de visiter avec nous.

Nous profitons de cette introduction pour faire un petit aparté sur la validité des liens proposés dans ces pages. Les sites Web que l'on trouve sur Internet

Diodes LED

Base:
Description
Symboles
Unités... Formules
Valeurs
Exercices
Programmes
Liens

1.1. Descriptions
Les DEL (diode électroluminescente) ou en Anglais : LED (light emitting diode) éclairent lorsqu'elles sont parcourues par un courant de l'anode vers la cathode.

1.2. Symboles

Anode Cathode
LED

1.3. Unités : Formules

La tension de seuil dépend de la couleur et

Couleurs	Tension de seuil	Longueur d'onde
Rouge	1,6 V à 2 V	650 à 660 nm
Jaune	1,8 V à 2 V	565 à 570 nm
Vert	1,8 V à 2 V	585 à 590 nm
Bleu	2,7 V à 3,2 V	470 nm

$R1 = (V - V_{led}) / I_{led}$, donc $R1 = (12 - 1,8) / 0,02 = 510$ ohms, bien souvent on utilise une résistance de 470 ohms $I_{led} = (12 - 1,8) / 470 = 0,21$ mA

La tension de seuil dépend de la couleur et

<http://www.free.fr/dossiers/analog/analog12.htm>

Montage de l'ensemble :
Décidément, pas besoin d'être électronicien... on comprend tout de suite que si le point A est plus positif que le point B alors L2 s'allume et si contraire, si B est positif c'est L1 qui s'allume. Enfin L1 et L2 brillent simultanément lorsqu'une tension alternative sera présentée aux points de tests. Rs se calcule selon la tension à mesurer. Par ex (1,7/0,02) avec 1,7 la classe de tension aux bornes de la diode LED et 0,02 l'intensité nominale par cette dernière (100µ pour éviter la prise de tête).

Schéma :

Liste des pièces :

R	Résistance 1/4 W selon calc° pour U=12V R=560 pour U=9V R=390 etc
L1	Diode LED rouge
L2	L2 Diode LED verte
Diodes	boîtier plastique
	1 sonde rouge et 1 sonde noire

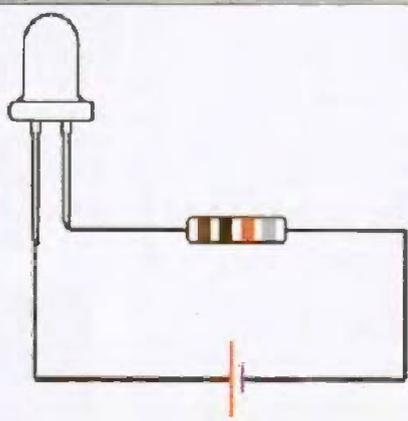
3 mm ou 5 mm comportent un plat sur la base pour repérer la cathode (ou lorsqu'elles sont... La consommation moyenne est d'environ 20 mA.

http://www.loglton.free.fr/realisations_minute/test_pol.html

proposent un contenu très dynamique et certains sites évoluent en permanence, voire de façon très soudaine. Si soudaine, même, qu'il arrive que les liens que nous vous proposons soient déjà invalides avant même la publication d'un nouveau numéro d'Électronique Pratique. Nous ne pouvons pas faire de copie des pages que nous vous proposons, pour des questions

1. LA DIODE ELECTROLUMINESCENTE - Microsoft Internet Explorer

Adresse http://users.win.be/W0005422/ladiodeled.htm



Comment calculer la résistance chutrice à associer avec une diode LED

Supposons une tension d'alimentation de 21,5 volts. On sait que pour qu'une diode LED s'éclaire normalement, elle doit être traversée par un courant de 20 mA sous une tension de 1,5 Volt (dépend du type de diode utilisée)

La tension aux bornes de R doit être de $21,5 \text{ V} - 1,5 \text{ V} = 20 \text{ V}$

Selon la loi d'ohm $R = U / I$

3 <http://www.win.be/W0005422/ladiodeled.htm>

jours mais l'arborescence du site a été modifiée. En repartant de la racine du site en question, il est souvent possible de retrouver les pages concernées.

Cet aparté étant terminé, revenons aux diodes LED et à leur mise en œuvre. Le premier site que nous vous invitons à visiter est accessible à partir de l'adresse suivante :

<http://etronics.free.fr/dossiers/analog/analog12.htm>. Le site proposé rappelle comment se comporte une diode (courbe caractéristique U en fonction de I) et explique les petites différences liées à la couleur des diodes LED. Ce site explique également comment calculer la résistance que l'on monte en série avec une diode LED, afin de limiter le courant.

Comme vous pourrez le constater sur ce site, la mise en œuvre d'une diode LED est suffisamment simple pour que cela puisse amuser nos jeunes amis (une poignée de LED, quelques résistances, une petite pile de 9V et une petite plaque de câblage rapide suffiront à captiver les plus jeunes pendant des heures).

Le deuxième site que nous vous invitons à visiter décrit comment mettre en œuvre des diodes LED dans une petite application ultra-simple : un testeur de polarité. Le

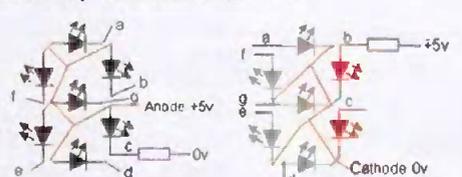
site est accessible à l'adresse suivante : http://ogloton.free.fr/realisations_minute/test_pol.html.

Comme vous pouvez le constater sur la vue d'écran reproduite en **figure 2**, il s'agit d'un petit montage vraiment très simple, dont le périmètre d'utilisation est relativement réduit, vous devez vous en douter.

offihesus a leds - Microsoft Internet Explorer

Adresse http://www.offihesus.com/leds/

1.4. Valeurs
Afficheurs 7 segments à Diodes Électroluminescentes



Afficheur à anodes communes Afficheur à cathodes communes

1.5. Variantes
Afficheurs à Diodes Électroluminescentes à matrice de points, au lieu d'avoir des barres

1.6. Utilisations
Principe de l'affichage

4 <http://www.offihesus.com/leds/1.6.diodeaff.htm>

<http://www.offihesus.com/leds/1.6.diodeaff.htm>

de copyright, vous le comprenez bien. Si les liens que nous vous proposons sont déjà invalides, nous vous invitons à contacter le 'Webmaster' du site en question pour lui demander s'il peut vous

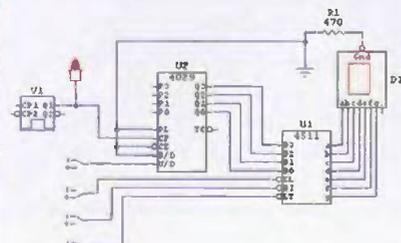
envoyer une copie des pages qui ont été supprimées de son site. Quelques fois les pages existent tou-

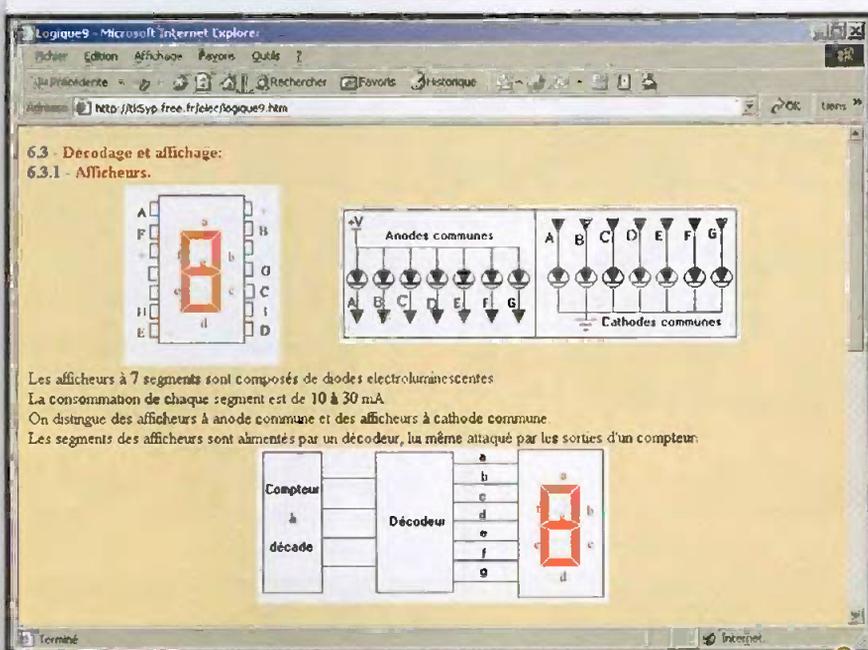
offihesus a leds - Microsoft Internet Explorer

Adresse http://etronics.free.fr/dossiers/analog/analog41/diodeaff.htm

1.6. Utilisations
Principe de l'affichage

Pour utiliser un afficheur 7 segments il est nécessaire de disposer d'un décodeur qui traduit le code BCD en code d'allumage des segments de l'afficheur





<http://www. http://tk5yp.free.fr/elec/logique9.htm>

5

Etant donnée la simplicité du sujet abordé, il n'est pas vraiment nécessaire de s'étendre, mais nous n'avons pas résisté au plaisir de vous présenter encore un site qui explique comment calculer la résistance qui limite le courant qui circule dans une diode LED. Mais ce site va nous servir d'introduction à la deuxième partie consacrée aux afficheurs 7 segments:

Comme vous l'aurez deviné, la mise en œuvre d'un afficheur 7 segments n'est guère plus compliquée que la mise en œuvre de 7 diodes LED regroupées dans un même boîtier. Reste alors à déterminer quelles sont les broches qui doivent être regroupées : L'anode ou la cathode ? C'est généralement la logique associée aux afficheurs qui détermine ce choix. La page située à l'adresse : <http://etronics.free.fr/dossiers/analog/analog41/diodeaff.htm> permet de com-

prendre de quoi il s'agit. Enfin, si vous vous demandez à quoi peut bien ressembler le contenu d'un décodeur pour afficheur 7 segments, vous êtes invités à visiter le dernier site que nous vous proposons aujourd'hui à l'adresse <http://tk5yp.free.fr/elec/logique9.htm>. Bien entendu, il y a peu de chance pour que vous choisissiez ce schéma pour piloter un afficheur 7 segments dans un montage, puisque l'on trouve des circuits intégrés très bon marché pour cette fonction. En revanche sur un plan didactique ce schéma est très intéressant.

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

P MORIN

Liste des liens de ce dossier

- <http://etronics.free.fr/dossiers/analog/analog12.htm>
- http://www. http://ogloton.free.fr/realisations_minute/test_poi.html
- <http://www. http://users.win.be/W0005422/ladiodeled.htm>
- <http://www. http://etronics.free.fr/dossiers/analog/analog41/diodeaff.htm>
- <http://tk5yp.free.fr/elec/logique9.htm>
- http://www.ensicaen.ismra.fr/~furon/_ElecNumerique/Logique/7segments/_affichage7segments.html
- <http://www.cnam.fr/hebergement/udp/gr/qrphysique/qp0164.htm>
- <http://accept.asu.edu/courses/phs110/expmts/exp13a.html>
- <http://www.multimania.com/rosemarie/logique.htm>

Electronique et programmation Apprentissage autour du 68HC11

Faire découvrir les composants programmables et leur programmation de manière progressive, telle est l'ambition de cet ouvrage.



Pour ce faire, l'auteur a choisi les microcontrôleurs de la famille 68HC11 fabriqués par MOTOROLA, en raison de leur disponibilité, leur faible coût, leur facilité d'utilisation et leur richesse en mémoire et ports d'entrée/sortie. Chaque montage est l'occasion d'apprendre et de mettre en pratique des notions de base comme la logique combinatoire, la programmation en assembleur, la conception d'un programme et l'utilisation des ports en entrées/sorties. Plus qu'un manuel technique sur le 68HC11 ou ouvrage sur la programmation, ce guide d'apprentissage est destiné aux électroniciens qui souhaitent s'initier aux composants programmables et aux informaticiens confrontés à la programmation de ces composants et désireux d'appréhender l'électronique «moderne».

A. REBOUX - E.T.S.F. DUNOD

196 pages - 158 F.

Les outils de développements pour microcontrôleurs



Les microcontrôleurs sont omniprésents dans les systèmes électroniques d'aujourd'hui. Même si les circuits logiques câblés traditionnels (et plus particulièrement les réseaux logiques programmables) sont en mesure de concurrencer, voire de surpasser, les microcontrôleurs sur le plan des performances, ils restent souvent plus contraignants à mettre en œuvre qu'un microcontrôleur. Les montages proposés ici utilisent eux aussi des microcontrôleurs.

Aussi était-il logique de vous présenter quelques - uns des outils de développement les plus courants.

Depuis toujours, le développement de systèmes à microprocesseurs et à microcontrôleurs nécessite l'utilisation d'outils bien spécifiques. Il y a encore seulement quelques dizaines d'années, ces outils étaient hors de prix et réservés aux professionnels. Aujourd'hui, les choses ont bien changé.

La panoplie des outils nécessaires à la mise en œuvre d'un microcontrôleur varie énormément d'un modèle à un autre. Malgré tout, les méthodes de développement restent à peu près les mêmes et le sacro-saint "cycle en V" est, bien entendu, applicable en toutes circonstances. Car, finalement, ce qui se dégage de ce fameux cycle de développement, c'est qu'il est utile de définir très tôt dans le développement d'un produit ce que l'on va faire et comment on va le vérifier efficacement. De cette façon, on est en mesure de se concentrer efficacement sur la tâche du moment, sans être obligé constamment de tout remettre en cause.

Dans les phases amonts du développement d'un projet, il existe de nombreux outils qui apportent une aide précieuse pour évaluer l'efficacité de l'architecture d'un logiciel. Vous trouverez dans ce dossier un article sur la suite de développement REALIZER qui permet de développer une

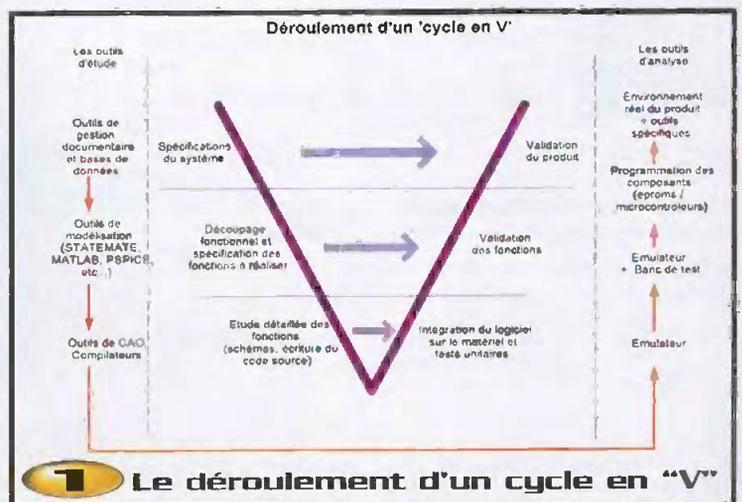
application à base de microcontrôleurs de façon entièrement visuelle et d'en simuler le fonctionnement très rapidement. Bien entendu, les outils de compilation traditionnels disposent souvent, eux aussi, de simulateurs bien pratiques.

Lorsque le développement de l'application est finalement terminé, il faut la tester méticuleusement pour éliminer les incontournables "bugs". Commence alors toute une série de tests validations (la partie montante du "V") et la liste des outils matériels nécessaires à ces travaux est parfois importante. Les émulateurs "temps réel" sont des équipements d'un grand secours pendant ces travaux, et vous

aurez l'occasion de constater dans ce dossier qu'ils ne sont pas tous hors de prix. Mais les besoins des développeurs ne s'arrêtent pas là. Bien d'autres outils sont nécessaires à l'achèvement du développement : programmeurs d'EPROM ou de microcontrôleurs, cartes d'acquisition sur PC, bancs de tests automatiques, etc.

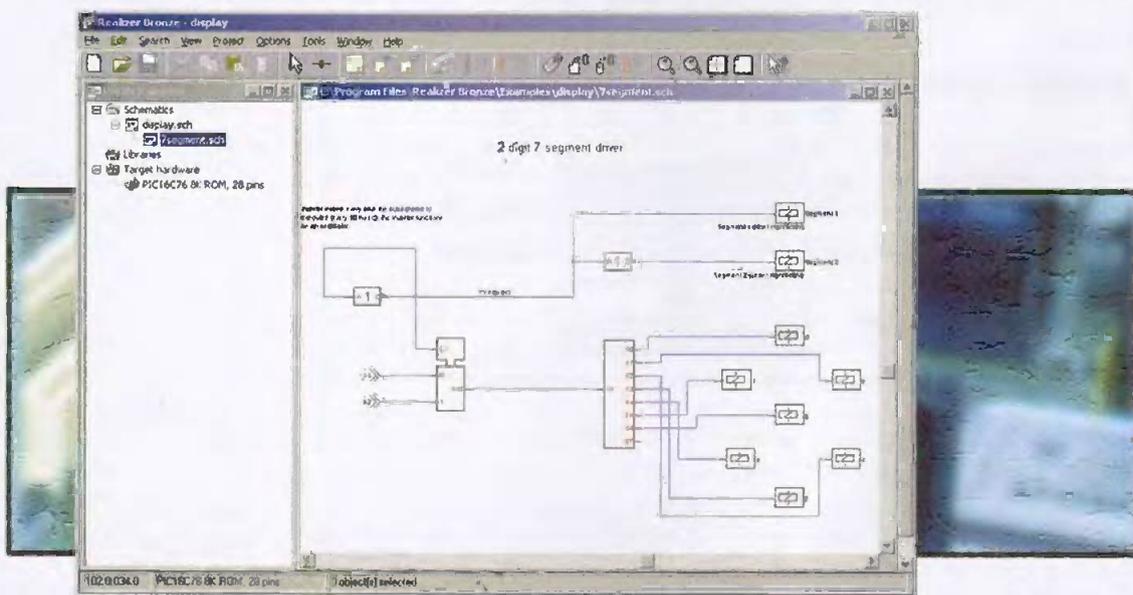
C'est donc toute cette panoplie d'outils que nous vous proposons de découvrir dans ce dossier, au travers de la présentation de quelques kits et équipements du moment.

P. MORIÛ



REALIZER

Un outil de développement graphique pour microcontrôleurs



Les outils de compilation classiques pour les microcontrôleurs reposent sur une analyse fonctionnelle préalable. En effet, il n'est pas efficace de commencer à écrire des lignes de code avant d'avoir réfléchi à la structure globale d'un logiciel et d'avoir rédigé les organigrammes associés.

Le travail de conception du logiciel se décompose donc en deux tâches (définition de la structure du logiciel et écriture du code correspondant). L'outil de développement REALIZER permet de simplifier la conception d'un logiciel en se chargeant de la génération du code et des détails d'implémentation liés à un microcontrôleur, ce qui vous permet de vous concentrer, avant tout, sur la fonction du programme.

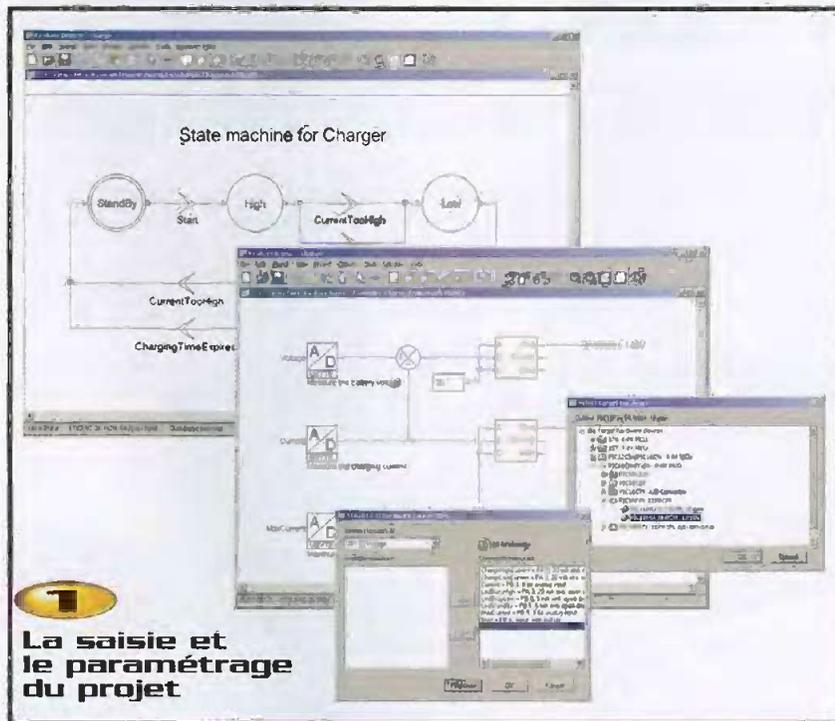
Face au développement de logiciels de plus en plus complexes, la génération de code automatique gagne de plus en plus de terrain aujourd'hui. Dans ce domaine, la suite de développement REALIZER pourrait bien intéresser de nombreux professionnels ainsi que de nombreux amateurs éclairés. Vendue à peine le prix d'une chaîne de compilation C classique pour un microcontrôleur, la suite de développement REALIZER permet de concevoir des logiciels de façon entièrement visuelle, sans même avoir à

écrire une seule ligne de code ! Bien loin d'être un gadget, un tel outil de développement permet de gagner un temps précieux lors de la conception de logiciels aux interactions complexes.

La philosophie de la suite de développement REALIZER est simple : laisser l'utilisateur se concentrer sur les phases créatives et laisser à l'outil le soin de réaliser les tâches sans réelle valeur ajoutée (le codage que l'on réalise "bêtement" à partir d'un organigramme). Si l'on compare la méthodologie de développement qui en découle par rapport aux méthodes classiques, l'avantage est évident. Avec une méthode de développement classique, on commence par analyser le travail effectué pour déterminer le flux des données et concevoir les organigrammes des traitements qu'il faudra leur appliquer. Pour aider à la formalisation de ce travail, il est possible de faire appel à des outils spécifiques (tels que STATEMATE, par

exemple) mais, au final, la traduction d'un organigramme en code machine passe tout de même par l'écriture d'un programme assimilable par un compilateur, en langage évolué ou en assembleur. Le temps nécessaire à l'écriture des lignes de codes correspondantes est souvent très long. Tous les concepteurs de logiciels reconnaissent que cette phase du développement n'est généralement pas valorisante et il n'est pas rare que cette tâche soit carrément sous-traitée (à partir du moment où les organigrammes sont suffisamment détaillés). Et c'est justement cette tâche finale, qui consiste à adapter la conception du logiciel à un microcontrôleur précis et à ses ressources internes, que se propose de réaliser pour vous la suite de développement REALIZER. Le gain de temps et de ressources qui en découle est vraiment appréciable.

Lors du développement d'un logiciel, la moindre erreur de conception ou le



1 La saisie et le paramétrage du projet

moindre changement des spécifications d'origine entraîne des "allers et retours" entre les différentes tâches de conception. Du coup, le déroulement du beau cycle en V de développement, tant adulé, prend vite des allures de double V, triple V, etc. Le développement devient alors très difficile à gérer dans ces conditions. Pour le développement de logiciels de taille conséquente, il est indispensable d'être en mesure d'évaluer l'impact des choix d'architecture le plus tôt et le plus rapidement possible dans le développement du projet. Les grands outils professionnels, tels que STATEMATE, sont, bien entendu, utilisés pour simuler le fonctionnement des logiciels en cours de développement, mais ces outils sont fort coûteux et pas toujours très simples à mettre en œuvre. Dans ce domaine, la suite de développement REALIZER apporte une solution intéressante grâce à son module de simulation intuitif et entièrement visuel.

Développé par la société ACTUM, la suite de développement REALIZER se décline en cinq versions : Bronze, Silver, Gold, Student et ST6/ST7. La version Bronze est limitée à 15 composants ce qui est suffisant pour évaluer correctement le produit. La version Silver n'a plus de limitation mais elle offre moins de souplesse pour optimiser le code généré que la version Gold. La

version Student est destinée aux étudiants, comme son nom permet de le deviner. Elle offre toutes les possibilités d'une version complète, mais le choix des microcontrôleurs supportés est un peu moins riche. Enfin, la version ST6/ST7 est équivalente à une version Silver limitée aux microcontrôleurs ST6, ST7.

Actuellement, la suite de développement REALIZER Gold permet de concevoir des projets autour des microcontrôleurs ST6, ST7, PIC12xx, PIC14xx, PIC16C5x, PIC16C6x, PIC16C7x, PIC16C8x et PIC16C87x. Pour d'autres cibles, il faudra retravailler manuellement le code C (ANSI) ou assembleur qui est généré par la suite de développement REALIZER, mais le gain de temps reste significatif.

La suite de développement REALIZER fonctionne avec les systèmes d'exploitations Windows 95/98/NT/2000. L'installation du logiciel se fait en deux temps puisqu'il faut installer, également, les fonctions de protection car le logiciel est protégé par une clé. Il faudra donc que le PC, prévu pour l'installation, soit équipé d'un port parallèle disponible. La clé peut, bien entendu, s'intercaler sur la liaison avec une imprimante. Cependant, certaines imprimantes nécessitent un accès exclusif au port parallèle. Il vaut donc mieux le savoir avant l'installation.

Pour ce dossier, nous avons testé les versions Bronze et Gold. Comme avec n'importe quelle suite de développement pour microcontrôleurs, l'utilisateur devra prévoir une phase de prise en main pour s'habituer à l'outil REALIZER. L'interface utilisateur du programme est cependant assez simple et la prise en main s'en trouve facilitée. Lors de nos essais, nous nous sommes retrouvés face à un produit qui ressemble à la fois à un logiciel de CAO électronique et à un compilateur. Pour les concepteurs, qui ont l'habitude de manipuler ces deux familles de logiciels, c'est un avantage.

La difficulté avec ce type d'outils, ce n'est pas tellement d'apprendre comment manier les éléments graphiques pour former un schéma fonctionnel et lancer la compilation du projet, mais c'est plutôt de se familiariser avec la signification des différents symboles qui sont disponibles dans les bibliothèques. Pour s'en convaincre, il suffit d'ouvrir l'un des projets disponibles dans le sous-répertoire des exemples livré avec le programme. Fort heureusement, le logiciel tire particulièrement bien profit de l'aide en ligne, ce qui permet d'accéder immédiatement à la signification d'un symbole au moyen de l'option "information" du menu contextuel disponible pour tous les objets. C'est d'ailleurs grâce à cette fonction que l'auteur a réussi à concevoir un petit projet (un décodeur pour afficheur 7 segments) en moins d'une heure tandis qu'il découvrait ce logiciel (après avoir, tout de même, consulté les fichiers d'exemple). Bien entendu, la documentation qui accompagne le produit contient également toutes ces informations.

La bibliothèque des fonctions de base fournies avec la suite de développement REALIZER est relativement conséquente. L'intérêt de ces fonctions est d'être à peu près indépendant des ressources d'un microcontrôleur donné. Pour le portage d'une application d'un microcontrôleur à un autre, c'est un avantage capital. Les fonctions disponibles dans la librairie principale couvrent la majorité des besoins les plus courants, en dehors des algorithmes de calcul. Cette librairie dispose de fonctions en rapport avec la gestion des timers, des entrées et sorties (logiques ou analogiques), des modes de fonctionnement basse consom-

mation du microcontrôleur (mode STOP/WAIT/SLOW). Cette librairie dispose également des fonctions de base pour mettre en œuvre une machine d'état, pour gérer des tables de conversion et bien d'autres choses encore. Si vous ne trouvez pas votre bonheur dans tout cela, vous avez, bien entendu, la possibilité de créer de nouvelles fonctions, mais l'auteur n'a pas eu le temps d'essayer complètement toutes les possibilités du logiciel. Comme vous pouvez le voir, les possibilités de la suite de développement REALIZER sont vraiment impressionnantes.

La facilité de navigation dans un projet complexe, grâce à une représentation hiérarchique, est également un atout important de la suite de développement REALIZER. Il est très facile d'obtenir une vue d'ensemble des flux de données et de leurs interactions, à condition d'être un tant soi peu rigoureux lors de la saisie du schéma fonctionnel. Le rapport généré par le programme apporte également une aide précieuse pour connaître rapidement le bilan d'utilisation des ressources internes du microcontrôleur (taille du code généré, espace RAM nécessaire, nombre d'entrées et sorties utilisées, etc.).

Une fois le projet compilé, il est possible de le simuler immédiatement, s'il n'y a pas d'erreur. L'interface utilisateur du simulateur est très simple et ne demande que quelques minutes de prise en main (pour quelqu'un ayant l'habitude des simulateurs utilisés en électronique). La représentation graphique des éléments affichés par le simulateur est un peu sommaire, mais l'essentiel n'est pas là. La possibilité la plus intéressante du simulateur réside dans la représentation et la visualisation des états internes du microcontrôleur. Ceci est bien plus puissant que d'observer simplement l'état des signaux à l'extérieur du microcontrôleur. Le simulateur vous permet, en effet, de connaître à tout moment la position d'une machine d'état, le contenu des variables internes ou des tableaux de données.

Cela en fait un outil d'analyse particulièrement efficace, car vous n'avez plus besoin d'interpréter les données binaires manipulées par le microcontrôleur, comme cela se fait habituellement avec un simulateur logiciel. Ici, vous manipulez directement vos entités abstraites sans avoir à en connaître la représentation interne issue de la com-

pilation. Avec la suite de développement REALIZER, l'utilisateur raisonne uniquement au niveau fonctionnel de son produit, sans avoir à se préoccuper de la représentation interne. L'utilisateur peut ainsi rester concentrer sur l'essentiel de la conception, sans que son raisonnement ne soit "pollué" par une multitude de détails liés à l'implémentation de son programme. C'est un atout important pour qui souhaite développer un logiciel vite et bien.

Notez, toutefois, une particularité du simulateur lorsque vous avez sélectionné pour cible la génération du code en langage C. Dans ce cas de figure, pour pouvoir simuler votre projet, vous devez installer un compilateur C++ dans le sous répertoire "Compiler" du programme REALIZER. La documentation du produit rappelle que la société INPRISE propose le compilateur Borland C++ V5.0 gratuitement sur son site Internet (www.borland.com).

Étant donné que la suite de développement REALIZER se charge de traduire votre projet en langage machine, les changements de spécifications ou d'architecture ne représentent donc plus une difficulté insurmontable. En effet, la compilation d'un projet ne demande que quelques secondes. Aussi, il est possible d'évaluer très rapidement les conséquences d'une modification dans un projet (par exemple pour savoir si

la taille de la RAM du microcontrôleur suffit à l'application).

Comme nous l'avons mentionné plus tôt, la suite de développement REALIZER permet actuellement de produire directement le code machine (fichier HEX) pour de nombreux microcontrôleurs. Le programme REALIZER génère également le fichier source du programme, ce qui permet de vérifier que le code produit est correctement optimisé pour une application donnée. Cela permet, également, de concevoir une partie du logiciel avec REALIZER et de reprendre le code généré automatiquement dans un autre environnement de développement. Cela fait de la suite REALIZER un environnement de développement très ouvert.

Nous concluons sur l'idée que la génération de code automatique est désormais une réalité incontournable dans le monde du développement logiciel. Dans ce contexte, la suite de développement REALIZER mérite vraiment que l'on s'y intéresse.



Distribué exclusivement par MERCURE
TÉLÉCOM

Prix :

Pack Amateur : 381 € TTC

Pack Pro : à partir de 1312 € TTC

2 La simulation fonctionnelle du projet

Kit de développement pour cartes à puces **ASE-II**



Les cartes à puces permettent de développer de nombreuses applications intéressantes touchant à de nombreux domaines. Pour le stockage de petits volumes d'informations personnalisées, elles sont idéales car elles peuvent également offrir un haut niveau de protection pour limiter l'accès aux informations confidentielles.

Le kit de développement ASE se compose d'un lecteur de cartes à puces ASEdrive accompagné de plusieurs logiciels pour le piloter ainsi que différents type de cartes à puces ASE. Le lecteur de cartes est livré installé dans un boîtier externe, mais vous pouvez le démonter pour l'installer dans un emplacement 3 1/2 sur votre ordinateur. Le lecteur de cartes à puces se raccorde au PC via un port série. Tous les câbles nécessaires aux différentes possibilités d'installation du lecteur sont fournis avec le kit. En cas de montage externe, le lecteur de cartes peut puiser son alimentation 5V directement sur l'alimentation de votre PC grâce à un cordon spécifique qui vient s'intercaler sur le port PS/2 du clavier. Si cette solution ne vous convient pas, vous avez aussi la possibilité d'utiliser une alimentation externe. Dans ce cas de figure, vous devrez prévoir une alimentation capable de fournir 300mA sous 9VDC (7V mini à 14V max.).

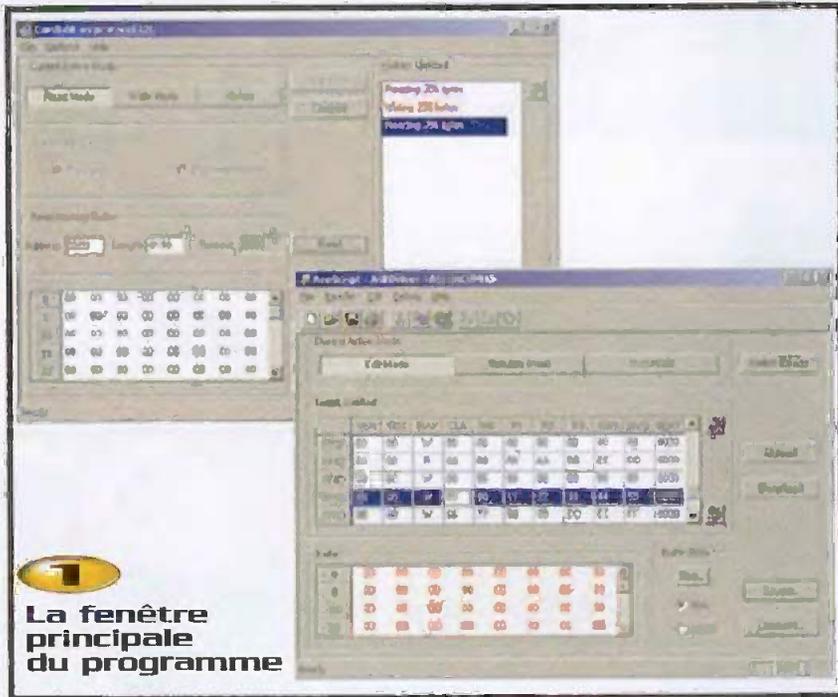
Le lecteur ASEdrive communique avec le port série de votre PC à la vitesse de 115200 bauds, ce qui

assure un transfert suffisamment rapide et qui conviendra à la plupart des applications. Si vous prévoyez d'échanger de gros volumes d'informations avec vos cartes à puces, il faudra tout de même vous armer de quelques secondes de patience. Il semblerait que le lecteur supporte également une communication par le port parallèle d'un PC, mais la documentation d'installation de notre kit

n'en faisait pas état. Pourtant la documentation principale indique qu'il est possible de brancher sur un PC jusqu'à 29 lecteurs ASEdrive reliés par des liaisons séries et 3 lecteurs ASEdrive reliés par des ports parallèles. C'est une possibilité alléchante, mais l'auteur n'a pas trouvé les indications nécessaires dans la documentation pour comprendre comment cela est possible.



Le deuxième connecteur pour les cartes à puces type "GSM"



1
La fenêtre principale du programme

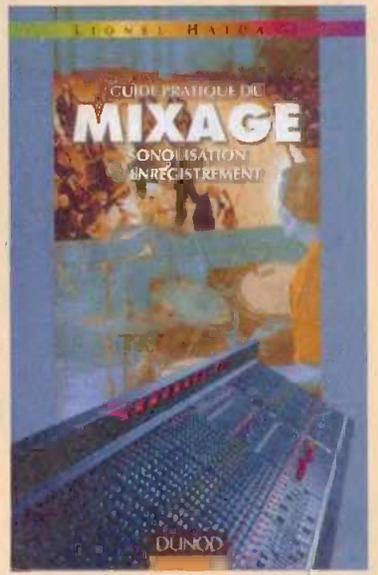
Quoi qu'il en soit, une fois correctement installé, le lecteur ASEdrive permet de lire des cartes à puces de formats très divers. L'appareil dispose d'un connecteur en façade pour les cartes à puces du format des cartes bancaires et d'un connecteur situé sous l'appareil pour les cartes à puces au format de celles des téléphones portables. L'appareil supporte les cartes avec de la mémoire I2C (adressage standard ou étendu, jusqu'à 128 Ko), les cartes à mémoire protégée et, bien entendu, les cartes à microprocesseur à partir du moment où ces dernières sont conformes au standard IS 7816-3 (T=0 et T=1). Sur un plan logiciel, le kit se décompose en trois applications, en plus des bibliothèques de développement. Le programme ASEmanager se charge de détecter vos différents lecteurs de cartes à puces tandis que le programme ASECardEdit vous permet de lire et d'enregistrer le contenu de vos cartes. Enfin, le programme ASEScript permet, comme son nom l'indique, d'écrire des scripts qui seront exécutés directement par le lecteur ASEDrive puisque ce dernier embarque un interpréteur et de la mémoire non volatile. Les possibilités de ce kit sont très étendues et elles réclament, bien évidemment, du temps pour toutes les explorer. Fort heureusement, les exemples fournis avec le kit sont nombreux et ils couvrent les développements dans les langages de programmation les plus courants. C'est ainsi que

vous trouvez des exemples d'applications écrits en Visual Basic, en PASCAL (DELPHI) et en C++. Mais vous trouverez aussi un exemple très intéressant écrit en Visual Basic Application sous Excel qui montre comment l'on peut transformer un lecteur ASEDrive en horodateur. L'intégration de l'application, au sein même d'une feuille de calcul Excel, démontre à quel point ce lecteur peut se montrer versatile. Bien entendu, pour développer votre propre application sous Windows autour du lecteur ASEDrive, vous devrez "éplucher" la documentation des différentes API (Application Programming Interface) qui offrent l'accès à toutes les fonctions du lecteur. Les fonctions disponibles sont trop nombreuses pour que nous en parlions ici, mais sachez que les fonctions de haut niveau gèrent directement des répertoires de fichiers ou la protection des données par un code PIN. Ce kit de développement pour cartes à puces contient donc tout ce qu'il faut pour réaliser des applications d'un niveau professionnel. Il exige cependant une parfaite connaissance du domaine des cartes à puces et les débutants devront être assidus s'ils souhaitent profiter du potentiel de ce kit. □

Distribué notamment par SELECTRONIC
Prix : 3650 F TTC (556,44 €)

Guide pratique du mixage Sonorisation et enregistrement

Ce livre se veut un guide pour tous ceux qui desirent apprendre les techniques de mixage.



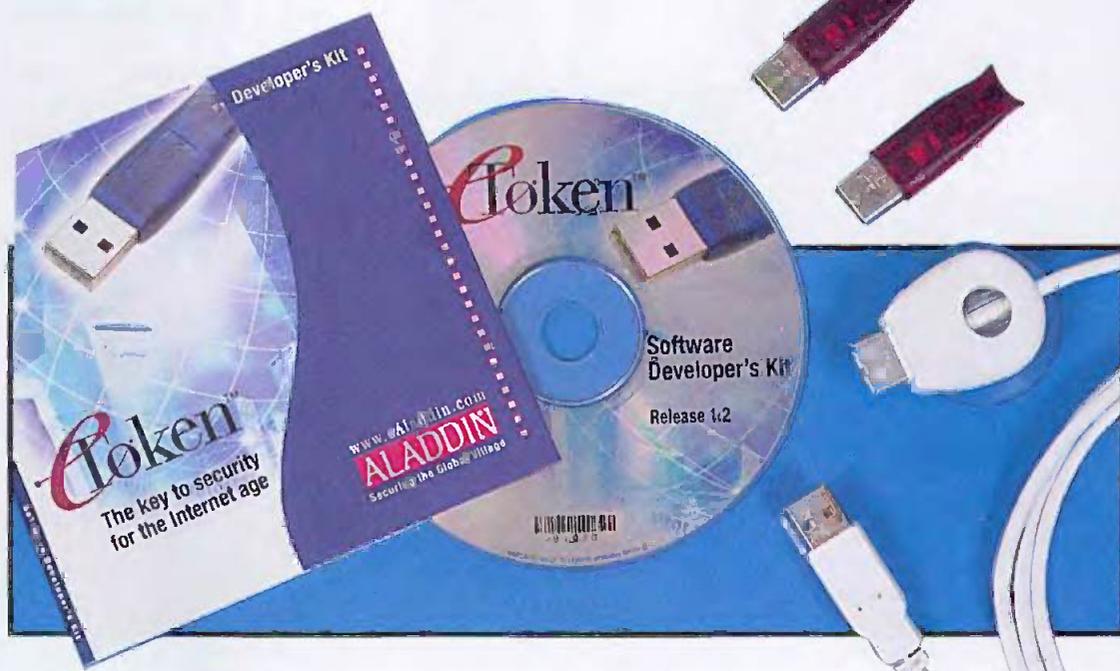
Choix, utilisation et raccords à des appareils extérieurs (processeurs, magnétophones multipistes, sonorisation) des consoles de mixage y sont explicites de manière à mener à bien le projet d'une prise de son et d'une balance orchestrale. En fin d'ouvrage, le lecteur trouvera de nombreuses suggestions de mixages faciles à réaliser en sonorisation et en enregistrement.

Sommaire :
La perception auditive. Le son. Les connexions et les connecteurs. Les consoles de mixage. Les techniques de mixage. Exemples de mixage. Glossaire. Public : Amateurs de musique et de son. Techniciens du son, musiciens et étudiants dans les métiers du son.

L. Haldant - DUNOD
136 pages - 108 FRF - 16,40 €

Clés de protection eToken

sur port USB



La société ALADDIN, déjà bien connue pour ses clés de protection très répandues qui se connectent sur un port parallèle, propose désormais des clés pour le port USB. Ce bus, qui est présent sur tous les PC récent depuis longtemps, offre bien plus de souplesse que le port parallèle. Ceci permet aux clés de protection eToken d'être beaucoup plus simples à manipuler par l'utilisateur.

Si vous développez des programmes dans un but commercial ou si vous avez besoin de prêter des logiciels pour une durée limitée en guise de démonstration, vous savez combien le piratage informatique vous est préjudiciable. En effet, la pratique malhonnête qui consiste à copier un logiciel en vue de l'utiliser sans acheter une licence engendre un manque à gagner qui pèse très lourd dans les comptes des éditeurs de logiciels qui ont parfois bien du mal à amortir leurs frais d'étude et de développement.

Certains rétorqueront que, pour un usage occasionnel ou personnel, ils n'ont pas les moyens d'acheter tel ou tel logiciel, car il est vendu bien trop cher. Mais ce qu'ils oublient c'est que si tous les utilisateurs étaient honnêtes et achetaient tous leurs logiciels, le prix des logiciels seraient bien moins élevés (bien que certains éditeurs trop gourmands ne joueraient peut être pas le jeu, mais heureusement il reste la concurrence qui pourrait jouer).

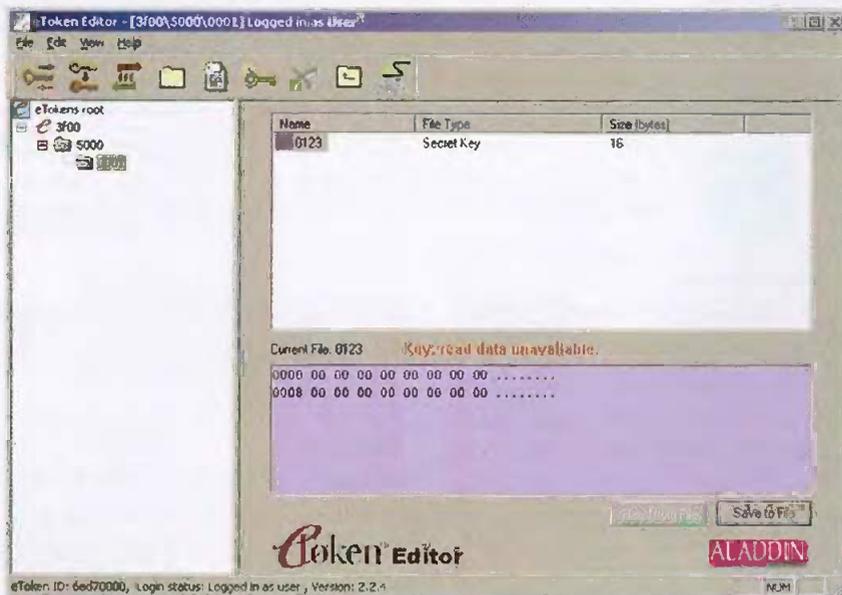
Face à l'ampleur de ce problème, certains éditeurs font appels à des clés de protections. Grâce aux clés de protection eToken, vous pourrez, vous aussi, protéger vos propres œuvres intellectuelles. Moyennant quelques euros, vous pourrez vous procurer des clés qu'il ne vous restera plus qu'à personnaliser en fonction de vos besoins, pour que vos programmes deviennent quasiment inviolables. Les possibilités offertes par les nouvelles clés eToken, par rapport aux clés des générations précédentes, augmentent en effet considérablement la difficulté de piratage, à condition d'appliquer quelques techniques de base lors de l'écriture de vos applications qu'il vaut mieux prendre en compte dès le début de la conception.

Si vous possédez des logiciels protégés par des clés de protections qui se connectent sur un port parallèle (couramment appelées "dongle"), vous avez peut-être déjà été confronté à de nombreuses difficultés d'installation. Selon la confi-

guration des périphériques d'un PC, il n'est pas toujours possible de trouver un port parallèle disponible pour y connecter la clé de protection.

Cette situation oblige couramment les utilisateurs à débrancher des équipements de leur ordinateur (et parfois même à désinstaller le logiciel associé, comme c'est le cas pour certaines imprimantes laser qui fonctionnent uniquement sous Windows grâce à un driver spécifique qui s'accapare le port parallèle). De plus, si l'utilisateur dispose d'un nombre important de logiciels ayant chacun leur propre clé de protection, il est parfois difficile d'empiler les clés.

Il faut dire que bien souvent un PC se trouve près d'un obstacle (l'arrière du bureau ou un mur) et l'empilage des clés devient vite gênant (sans parler de la torsion que subit le connecteur du port parallèle qui est soudé directement sur la carte de mère des PC récents). Enfin, chaque fois que l'utilisateur souhaite utiliser un logiciel protégé sur un



1 L'éditeur du contenu des clés eToken

autre ordinateur, il doit déplacer la clé, ce qui l'oblige à se mettre à quatre pattes sous son bureau pour accéder au port parallèle de son PC et récupérer la clé. Comme de juste, pour ce genre d'opération, l'utilisateur oublie la plupart du temps d'éteindre son ordinateur lorsqu'il installe ou démonte une clé sur le port parallèle.

Pourtant, toutes les notices mettent clairement en garde le client sur le risque de destruction des clés dans ce cas de figure. Il faut bien l'avouer, les clés de protection sur port parallèle ne sont vraiment pas pratiques.

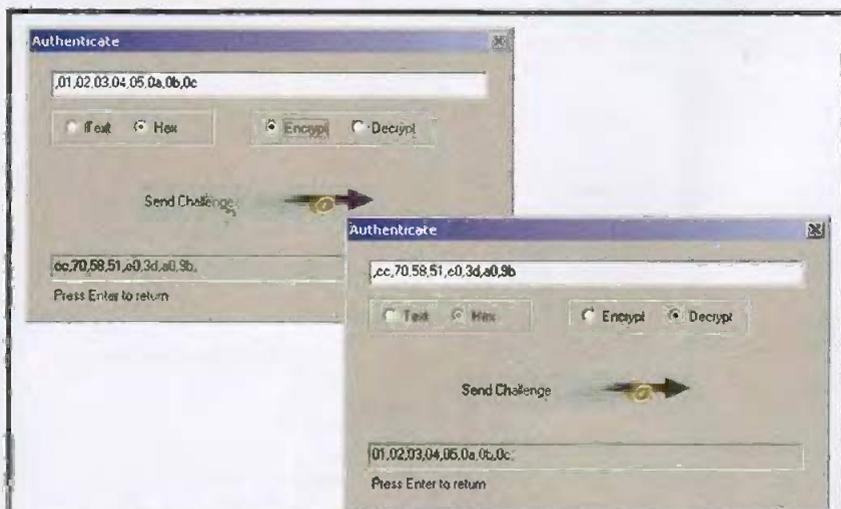
Face à toutes ces difficultés, les clés eToken sur port USB sont promises à un bel avenir, car l'utilisation du port USB élimine tous les désagréments que nous venons de mentionner. En effet, le port USB est beaucoup plus facile d'accès et il est possible d'utiliser facilement une rallonge pour rapporter le connecteur nécessaire à l'introduction d'une clé à un endroit accessible (le kit de développement pour les clés eToken fournit un tel câble). Le partage du port USB n'est plus un problème puisque, grâce à l'adjonction d'un HUB sur le bus, il est possible d'ajouter de nouveaux ports USB à votre PC (théorique-

ment un bus USB peut supporter jusqu'à 128 ports). De nombreux claviers et écrans de PC sont d'ailleurs équipés d'un HUB intégré. La multiplication des clés de protections n'est donc plus une difficulté majeure (même s'il faut tout de même acheter les HUB). Enfin, le bus USB autorise la connexion et la déconnexion à "chaud" (c'est à dire lorsque le PC est allumé) des périphériques qui lui sont raccordés sans que cela ne gêne le moins du monde les logiciels installés sur la machine. Tous ces avantages font des clés eToken une solution de protection très intéressante.

Le kit de développement que nous avons pu essayer se composait de deux clés eToken vierges, d'une rallonge pour bus USB équipée d'une ventouse pour fixer l'extrémité du connecteur (type A) où bon vous semble. Le kit s'accompagne également d'un CD-ROM contenant le logiciel, ainsi que les fichiers sources de programmes qui mettent en œuvre les nombreuses possibilités des clés eToken. Les logiciels fournis fonctionnent dans les environnements Windows 98/NT4/2000. La documentation est fournie à la fois sur support informatique et sur support papier, ce qui est appréciable (et bien trop rare).

Les clés eToken supportent un éventail impressionnant de standards. Citons PC/SC, PKCS#11 v2.01, Microsoft CAPI, RAS/Raduis/PAP/CHAP, X.509 v3, IPsec/IKE, SSLv3, S/MIME et Intel's RNG. Ouf ! Si avez tout ça et que vous ne trouvez pas votre bonheur c'est que vous le faites exprès.

Ceci étant dit, vous comprendrez que l'implémentation d'une protection contre le piratage dans un projet ne s'improvise pas. Les standards disponibles correspondent tous à une utilisation précise et il est préférable de savoir à l'avance de quelles protections l'on a besoin. Malgré tout, la mise en œuvre de ce kit reste abordable pour les programmeurs puisque les notions mises en œuvre sont exclusivement du domaine du développement des logiciels.



2 Les mécanismes de cryptage et de décryptage : un moyen simple de mettre en œuvre la technique de challenge/réponse

L'installation du kit eToken est très simple (le bus USB y est sûrement pour quelque

chose) et en quelques minutes seulement l'utilisateur est déjà en mesure de dialoguer avec les clés de protection. Sur le plan logiciel, le kit de développement se décompose en deux grandes parties : Un éditeur destiné à configurer et tester manuellement les clés, et des bibliothèques (Lib, DLL) à lier à vos applications pour appeler les fonctions de protection des clés eToken.

L'éditeur de clés est très simple à utiliser et quelques minutes suffisent pour sa prise en main. Les clés eToken disposent de 16 Ko ou 32 Ko de mémoire non volatile ce qui permet d'y enregistrer pleins d'informations de façon sécurisée. Les possibilités des clés eToken sont nombreuses et nous ne pouvons pas toutes les énumérer ici (il faut bien vous laisser quelque chose à découvrir par vous-même).

Pour tirer pleinement profit des possibilités des clés eToken, l'éditeur vous permet de tester manuellement les challenges et les réponses en fonction des clés logicielles que vous aurez créées dans la mémoire des clés physiques. Cela vous permet de connaître à l'avance la réponse à une sollicitation des clés, pour implémenter dans vos programmes les fonctions de protection. Il est à noter que pour que la protection soit efficace, il ne faut pas se contenter d'implémenter un simple test booléen pour vérifier la présence d'une clé. Une telle protection est trop simple à contourner puisqu'il est assez facile de tracer l'exécution d'un programme à rebours avec un bon "debugger".

Pour quelqu'un maîtrisant bien ce type d'outils, c'est un jeu d'enfant que d'identifier les instructions qui ont provoqué la fin d'un programme.

Pour que la protection de votre logiciel soit efficace, il est préférable d'utiliser le mécanisme de challenges/réponses pour modifier les données manipulées par le programme, à de multiples endroits stratégiques.

Par exemple, si votre programme à besoin d'un buffer de 64 Ko dont la mémoire est allouée de façon dynamique, vous pouvez utiliser le contenu de la réponse à un challenge comme para-

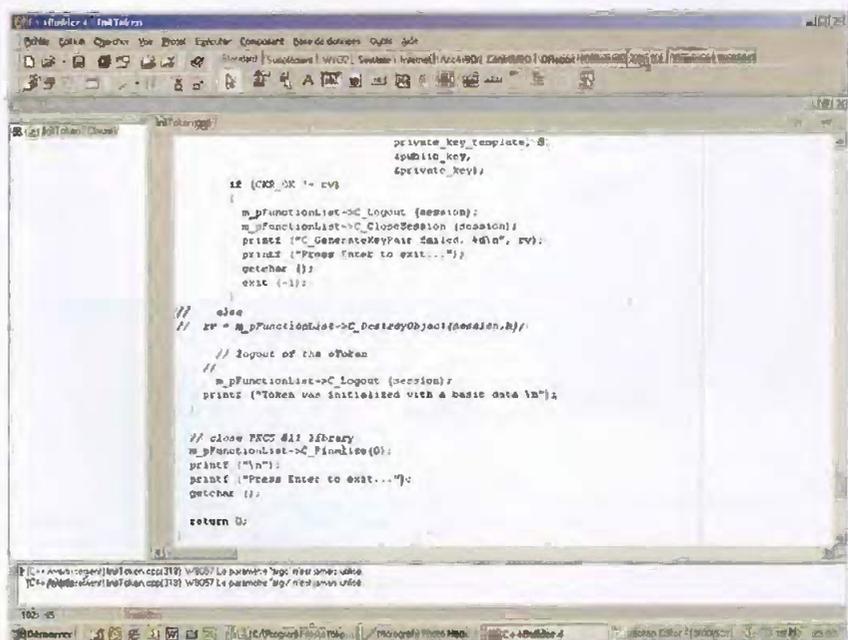
mètre lors de l'appel la fonction malloc (l'une des fonctions qui alloue de la mémoire à votre application) plutôt que de fournir directement un paramètre fixe à cette fonction. De cette façon, si la clé n'est pas reconnue, la mémoire nécessaire au fonctionnement du programme ne sera pas allouée et l'exécution du programme échouera. Et, cette fois-ci, le travail à effectuer pour détourner la protection est bien plus complexe puisqu'il ne suffit plus de détourner un saut conditionnel mais il faut trouver tous les challenges/réponses que vous avez utilisés pour protéger votre application. Bien entendu, pour utiliser ce genre d'astuces, il faut prendre le temps de gérer toutes les exceptions qui peuvent se produire au sein d'un programme, mais c'est déjà une obligation pour les applications professionnelles (n'est-ce pas !).

Il existe, bien entendu, bien d'autres possibilités pour utiliser le plus efficacement possible ce type de clés de protection. Les exemples fournis avec le kit de développement permettent d'étudier chacun des mécanismes disponibles, et ils sont nombreux ! Les fichiers sources des exemples proposés peuvent être compilés directement par un compilateur Visual C++ 6.0 de Microsoft. Mais les utilisateurs des compilateurs Builder C++

4.0 et 5.0 peuvent également exploiter directement les exemples du kit grâce à l'utilitaire de conversion des projets Visual C++ (voir le menu OUTIL du compilateur Builder C++). L'auteur, qui dispose du compilateur Builder C++ 4.0, a ainsi pu essayer tous les programmes d'exemple du kit.

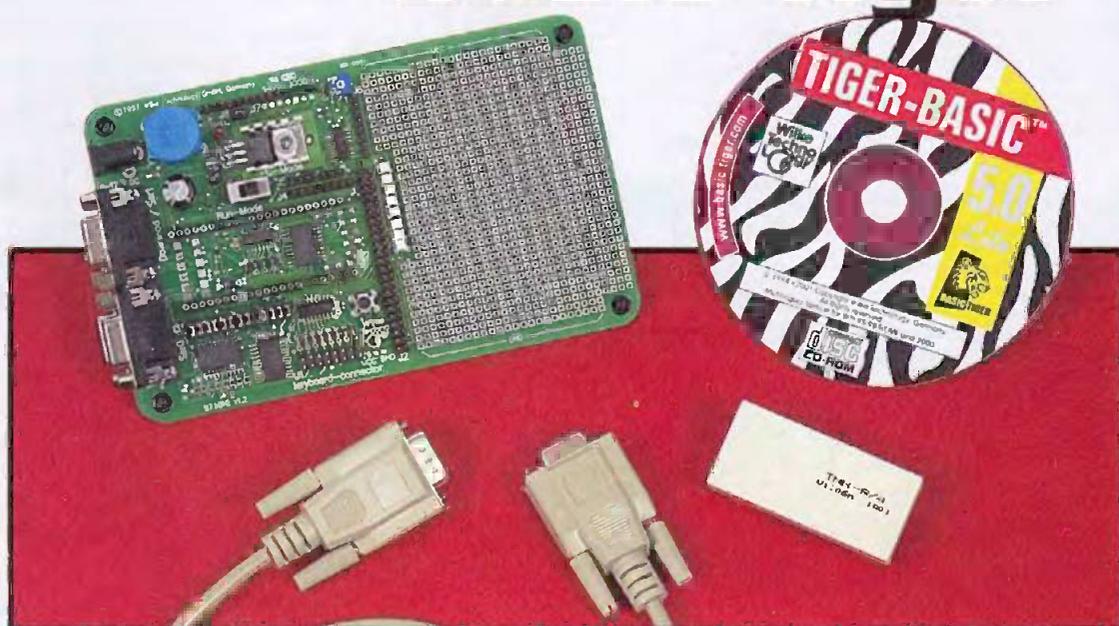
En définitive, le kit de développement pour les clés eToken est un très bon produit qui permet de mettre en œuvre rapidement et facilement la protection de vos logiciels. Grâce à la facilité d'utilisation de ce kit, vous pouvez vous concentrer sur l'essentiel : le développement de vos applications et non sur le moyen de les protéger. Si vous pensez que la protection de vos logiciels vous concerne, ou bien si vous souhaitez développer des solutions de contrôle d'accès aux données sensibles de votre entreprise, le kit de développement pour les clés "V" mérite vraiment d'être évalué.

Distribué notamment par SELECTRONIC
 Prix : 3522,50 F TTC (537 €)
 Clé supplémentaire à partir de 44,25 €.



3 Les exemples de mise en œuvre des mécanismes de protection.

Kit de développement "Basic Tiger"



La mise en œuvre d'un système à microcontrôleur rebute souvent les amateurs en raison des langages de programmations qu'il faut maîtriser (langage C ou assembleur, la plupart du temps). Pourtant on peut surmonter ce handicap en programmant certains microcontrôleurs en langage BASIC. Le kit de développement Basic Tiger fait partie des outils de développement qui permettent de développer des applications puissantes dans le langage informatique le plus populaire.

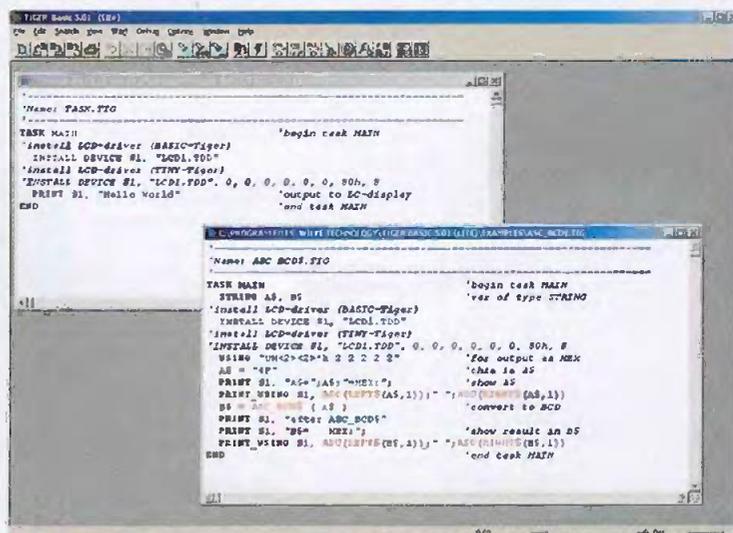
Pour autant, n'allez pas croire que le kit de développement Basic Tiger est un gadget et que les possibilités de son langage BASIC sont bridées ou limitées par rapport à des environnements de développement en langage C ou en assembleur. Il n'en est rien ! Comme vous le savez peut-être, le langage BASIC, même si on dit de lui qu'il est universel, est décliné en de nombreuses variantes. Depuis bien longtemps déjà, les points faibles du langage BASIC (comme le manque de structuration à cause des instructions GOTO et la numérotation des lignes de programme) ont été effacés et les versions du langage BASIC actuelles n'ont plus rien à envier aux langages C ou PASCAL. Pour s'en convaincre, il suffit de regarder ce que sait faire le compilateur Visual BASIC de Microsoft.

Le langage BASIC implémenté dans le kit de développement Basic Tiger est également une version très aboutie, puisqu'il s'agit d'un environnement multitâche particulièrement bien adapté aux développements d'applications embarquées exigeantes. Quelques lignes de programmes seulement suffisent pour

définir une nouvelle tâche, ce qui fait du Basic Tiger l'un des environnements multitâches les plus simples à mettre en œuvre du marché. En somme, pour qui connaît essentiellement le langage BASIC, c'est pratiquement le meilleur des mondes qui lui est proposé dans ce kit de développement. Il est toutefois à noter que le langage BASIC de ce produit n'offre aucune passerelle vers le langage assembleur. Cette contrainte est évidemment liée au fonctionnement multitâche du sys-

tème, qui ne peut pas se permettre de laisser toutes les libertés aux tâches qui s'exécutent de façon concurrente.

C'est le prix à payer pour bien garantir la stabilité du système. En contrepartie, les applications conçues avec le kit Basic Tiger sont très robustes par rapport aux applications programmées dans les langages traditionnels. Car, en effet, même si une tâche est bloquée, le reste du programme continue de fonctionner. Moyennant l'implémentation d'une



1 La fenêtre principale de l'environnement de développement

tâche dédiée à la surveillance des autres tâches du système, il est alors possible de réaliser un système particulièrement stable et capable de se sortir tout seul de la plupart des situations bloquantes.

Vous l'aurez donc compris, le langage BASIC de ce kit de développement est plein de ressources et l'exploitation de toute sa puissance réclame quelques heures pour se familiariser avec les spécificités de ce kit. Fort heureusement, le kit est livré avec une multitude de petits programmes de démonstration ainsi qu'une palanquée de drivers pour piloter de nombreux organes (afficheurs LCD, claviers matricées, etc.). Le Basic Tiger dispose même d'un bus CAN, ce qui fait de lui un allier précieux pour le développement d'applications domotiques et industrielles autour de ce bus de terrain très renommé.

Le kit de développement Basic Tiger est organisé autour d'un microcontrôleur encapsulé dans le même boîtier que l'interpréteur BASIC (à priori semi-compilé

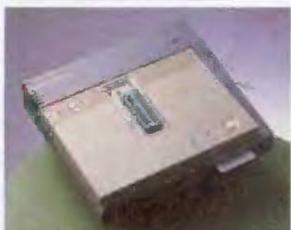
d'après ce que nous avons pu en voir, ce qui est un gage de rapidité d'exécution par rapport à un langage entièrement interprété). la mémoire flash pour stocker les programmes des utilisateurs et la mémoire RAM statique pour gérer les données. La configuration nécessaire pour supporter le logiciel qui accompagne le kit de développement est un PC sous un système d'exploitation Windows 95/98/NT/2000 possédant un processeur Pentium-II, cadencé au minimum à 233 Mhz, au moins 100 Mo d'espace libre sur le disque dur, un port série libre et un lecteur de CD-ROM pour l'installation du logiciel. Il est à noter que, sur les 100 Mo environ d'espace disque que requiert le logiciel qui accompagne ce kit de développement, 70 Mo sont consacrés à la documentation du produit (format PDF). N'essayez pas d'imprimer toute la documentation d'un coup, elle fait plus de 1700 pages au total ! Si vous n'aimez pas lire l'anglais vous risquez de trouver la documentation quelque peu rebutante. Ceci dit, il ne faut pas se plaindre, pour une fois qu'un produit est aussi bien documenté.

Le passage d'une application qui "tourne" sur la carte de développement à un produit embarqué est facilité par le fait que l'essentiel du système tient dans la puce maison encapsulée dans un boîtier plastique DIL. Le module Basic Tiger pouvant être acheté séparément du kit, il devient alors très facile de finaliser l'application dans un boîtier spécifique. De plus, la carte de développement dispose d'une zone de prototypage rapide, ce qui permet à l'utilisateur de mettre au point facilement des petits systèmes avant de finaliser le schéma de la future carte électronique.

Pour résumer, de part ses performances impressionnantes et son tarif abordable, le kit de développement Basic Tiger est un produit qui intéressera à la fois les particuliers et les professionnels ayant une préférence pour le langage BASIC.

Distributeur : OPTIMINFO
Prix : 1246,30 F TTC (190 €)

NOUVEAU



ALL-11P2 Programmeur universel

- plus de 5000 composants supportés
- port série et port parallèle
- extensible en multi-supports
- environnement windows 32-bits 95/98/2000/NT
- mise à jour gratuite et illimitée sur internet
- appareil garanti 2 ans en échange standard



Cross compilateurs, Cross-assembleurs, simulateurs, débogueurs : Philips, Intel 8051, P51XA, PIC, Motorola 68000, 68HC11, DSP, 8086, 6805, Z80/180, Siemens, Hitachi, Zilog, Rockwell, Conexant, Mitsubishi, Samsung... **CAO, DAO :** routage de circuits imprimés simulation logique et analogique...



OUTILS DE DEVELOPPEMENT POUR L'INDUSTRIE ELECTRONIQUE



Programmeur universel GALEP-III



Programmeur d'Eprom autonome



Programmeur carte WAFER



Emulateurs : Philips 8051/51XA, PIC, 68000, 68HC11/05, DSP, Eprom



Analyseurs logiques



Emulateur universel de PIC



Extension multiports RS232/422/485 (bus ISA/PCI)



Laboratoire d'apprentissage pour Intel, Philips, etc.



Lecteurs et graveurs pour cartes magnétiques

(PROGRAMMATION)

INDUSTRIE ELECTRONIQUE, EDUCATION NATIONALE, FORMATION

22, place de la République 92600 Asnières-sur-Seine - Tél. : 01 41 47 85 85 - Fax : 01 41 47 86 22

e-mail : commercial@programmation.fr - internet : www.programmation.fr

5X EMUL KIT

Un émulateur à faible coût
pour les microcontrôleurs
PIC 12C50x et 16C5x



Conçu par la société COMFILE et petit frère du MR-PIC (qui fait également l'objet d'un article dans ce dossier), l'émulateur 5X EMUL KIT est l'un des outils les plus compétitifs du marché actuellement, dans le domaine des émulateurs temps réel.

En effet, l'émulateur 5X EMUL KIT n'est pas un simple kit d'évaluation. C'est un véritable émulateur temps réel capable d'émuler les microcontrôleurs PIC des séries PIC 16C5x et PIC12C50x jusqu'à 20 MHz. Pour preuve, le logiciel qui accompagne cet émulateur est exactement le même que celui qui accompagne l'émulateur MR-PIC. Du coup, un certain nombre de remarques liées au logiciel que nous avons fait pour le MR-PIC sont également valable pour le 5X EMUL KIT.

Les utilisateurs de Windows 2000 ou Windows NT devront prendre le temps de télécharger la toute dernière version du logiciel sur le site Internet du fabricant :

(<http://www.comfile.co.kr/english/>). A l'issue de l'installation du logiciel, les utilisateurs de Windows NT/2000 devront également installer un driver supplémentaire pour autoriser l'accès au port parallèle sous ces systèmes

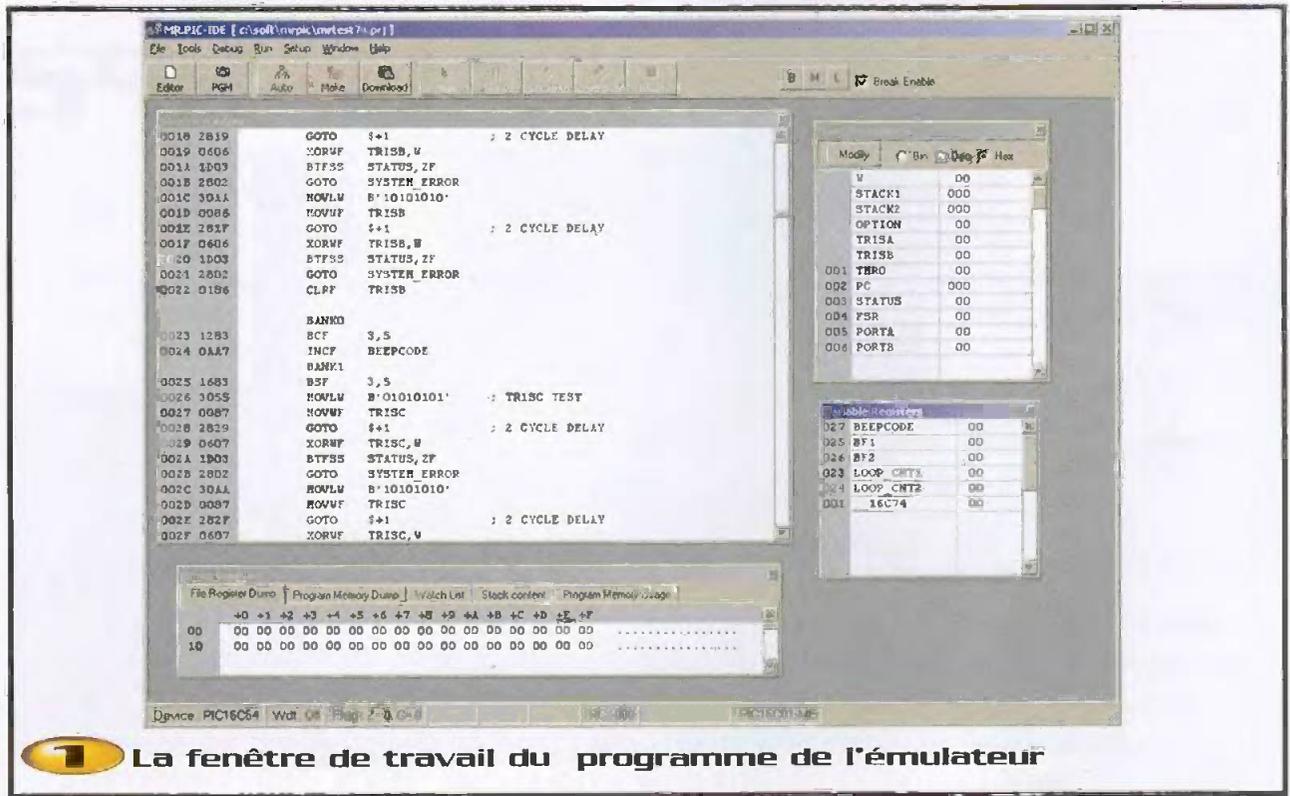
d'exploitation. Pour plus d'informations, nous vous invitons à lire l'article sur le MR-PIC publié dans ce dossier.

Comme pour son grand frère, l'émulateur 5X EMUL KIT requiert un PC avec une configuration minimum comprenant un processeur Pentium 166 MHz, 10 Mb d'espace libre sur le disque dur et un port parallèle. Le logiciel de l'émulateur 5X EMUL KIT est compatible avec les systèmes d'exploitation Windows 3.1, Windows 95/98/Me, Windows NT/2000 (et sûrement XP, mais cet environnement est trop récent pour que le constructeur en parle dans sa documentation).

L'émulateur se présente sous la forme d'un kit de développement (d'où son nom) dépourvu d'un boîtier. Cela peut rendre l'émulateur un peu vulnérable et il vaudra mieux l'installer sur un plan de travail bien dégagé pour éviter que des outils lui tombent dessus pendant qu'il est sous tension. A ce pro-

pos, il est important de noter que l'émulateur 5X EMUL KIT ne possède pas d'alimentation. Ne cherchez pas un connecteur pour y brancher une alimentation, il n'y en a pas. En fait, l'émulateur 5X EMUL KIT tire son alimentation 5V de la maquette cible sur laquelle la sonde est connectée en lieu et place du microcontrôleur à émuler. Cela signifie que, pour que l'émulateur fonctionne correctement, il doit être raccordé à une maquette qui est sous tension. Cela exclut d'utiliser l'émulateur comme un simple simulateur (ce qui n'est pas très gênant puisque l'environnement MPLAB, fourni gratuitement par MICROCHIP sur son site Internet, possède un simulateur).

Ce mode d'alimentation, un peu particulier, pose deux problèmes. Tout d'abord, l'émulateur présente une plus grande fragilité face aux manipulations et aux tensions excessives qui pourraient apparaître sur la cible. Par



1 La fenêtre de travail du programme de l'émulateur

exemple, il vaudra mieux éviter de raccorder la sonde de l'émulateur à une maquette qui est déjà sous tension. De plus, une erreur dans le sens de branchement de la sonde serait fatale pour l'émulateur. C'est pour cela que COMFILE propose une carte de protection vendue en option. Ensuite, le deuxième problème vient du fait que l'alimentation de la maquette cible doit être en mesure de fournir le courant supplémentaire que consomme l'émulateur. Nous avons relevé une consommation moyenne

de 120mA environ sur l'émulateur que nous avons en test. Ceci nous fait dire que le régulateur 5V de la maquette cible devra donc disposer d'une réserve d'au moins 250mA pour assurer un bon fonctionnement de l'ensemble, surtout si la cible est équipée d'un superviseur d'alimentation dont la sortie RESET pilote d'autres composants que le microcontrôleur. Sinon, cela pourrait provoquer l'initialisation intempestive de composants périphériques qui sont censés être configurés. Dans ce cas de

figure, il est fort probable que le programme en cours de validation ne fonctionne plus du tout et que vous ayez du mal à faire le lien avec un problème d'alimentation.

Concernant la fréquence d'horloge, comme nous l'avons déjà mentionné, l'émulateur 5X EMUL KIT est capable de fonctionner avec des microcontrôleurs cadencés jusqu'à 20 MHz. Mais, contrairement à l'alimentation qui est puisée sur la maquette cible, l'horloge de l'émulateur est interne. Sa fréquence peut être choisie par des cavaliers, mais les possibilités sont relativement limitées ce qui peut poser les mêmes petits soucis qu'avec le MR-PIC lorsqu'une fréquence particulière est absolument nécessaire au fonctionnement de l'application (voir l'article sur le MR-PIC). Malgré tout, l'émulateur 5X EMUL KIT est un excellent produit qui ravira les amateurs passionnés par la programmation des microcontrôleurs et qui souhaitent pouvoir enfin travailler comme des "pros". Son rapport prix/performances exceptionnel fera sans doute aussi réfléchir les labo professionnels qui ne regardent pas toujours à la dépense.



les accessoires fournis

Distribué notamment par LEXTRONIC
Prix : 1595 F TTC (243,16 €)

Programmateur universel ALL-11P2



Face à la diversité des composants programmables et des kits de programmation, l'utilisateur est vite déconcerté. S'il doit utiliser plusieurs produits différents. Pour une même famille de composants, de nouvelles références apparaissent sans cesse, rendant caducs les kits de programmations qu'il faut parfois renouveler totalement. Pour faire face à cette situation, la solution s'impose d'elle-même : il faut investir dans un programmateur universel (et évolutif).

Le programmateur ALL-11P2 est issu d'une grande lignée de programmeurs, tous plus connus les uns que les autres, et qui ont fait la renommée de la société HI-LO SYSTEMS. L'auteur, qui en était resté au modèle ALL-03, a pu constater que la qualité des produits fabriqués par cette société est restée d'un très bon niveau.

Le modèle que nous avons pu tester est un bon produit, dont le rapport qualité/prix sera particulièrement apprécié par les professionnels. Les possibilités très étendues de ce programmateur pourraient bien attirer également les particuliers qui disposent d'un budget conséquent pour assumer leur passion. En tout cas, l'achat d'un programmateur ALL-11P2 est un excellent investissement à long terme.

Pour la mise en service et le branchement du programmateur ALL-11P2, l'utilisateur a le choix des armes : l'appareil peut être connecté à un PC, soit par un port série, soit par un port parallèle bidirectionnel. L'appareil est fourni avec tous les câbles nécessaires aux deux modes

de communication avec un PC. Un adaptateur SubD9/SubD25 est même fourni, pour les PC un peu anciens qui disposent encore d'un port série équipé d'un connecteur SubD25. Cependant le PC utilisé pour communiquer avec le programmateur ALL-11P2 devra fonctionner sous une version 32 bits de l'environnement Windows (Win95/98/Me ou WinNT 4.0/2000/XP). L'auteur n'a pas trouvé sur le CD-ROM qui accompagne le produit une version MS-DOS du programme qui pilote le programmateur ALL-11P2. Ceci exclu donc l'utilisation de ce programmateur sur des PC vraiment trop vieux (les plus susceptibles par ailleurs de disposer d'un port série équipé d'un connecteur SubD25). C'est un peu dommage car, dans les labos, les vieux PC sont généralement recyclés à des postes dont l'utilisation n'est pas intensive (c'est le cas par exemple de la programmation d'une mémoire ou d'un microcontrôleur).

L'installation du logiciel nécessaire pour piloter le programmateur ALL-11P2 se déroule sans encombre en

quelques minutes seulement. Le logiciel est disponible sur CD-ROM uniquement ce qui, une fois de plus, élimine l'utilisation des très vieux PC. Le système minimum, recommandé par HI-LO SYSTEMS pour accueillir le programme, est un PC compatible IBM équipé au moins d'un processeur 486 et capable de faire fonctionner Windows 95 ou Windows 98, avec 64MB de mémoire vive et 20MB d'espace disque disponible. Une souris est indispensable pour utiliser le programme car certaines actions n'ont pas de raccourcis clavier. Si vous envisagez de connecter le programmateur via un port série, vous devrez peut-être disposer de deux ports séries sur votre PC selon votre modèle de souris.

La connexion via un port parallèle est fortement conseillée car, même si le programme sollicite le port série de votre PC à 115200 Bauds, la vitesse de communication obtenue via le port parallèle est bien plus rapide. Pour la programmation de mémoires de petite taille, le choix du port de communication n'est pas vraiment impor-

tant. En revanche, dès qu'il s'agit de programmer des composants d'une taille de plusieurs centaines de kilobits, cela devient critique.

A titre de comparaison, l'auteur s'est d'ailleurs livré à quelques essais. La lecture d'une mémoire FLASH AT49F040 avec le programmeur ALL-11P2 connecté à un PC (Pentium III 500 MHz sous Windows 2000) via un port série a nécessité plus de 4 minutes. La même opération avec le programmeur ALL-11P2 connecté au même PC mais via un port parallèle bidirectionnel, cette fois-ci, (port EPP) a nécessité seulement 30 secondes. Pour des opérations régulières avec des mémoires de fortes capacités, le choix d'une connexion par le port parallèle s'impose donc de lui-même. Notez, cependant, que le programmeur ALL-11P2 n'est pas compatible avec les anciens ports parallèles unidirectionnels. Soyez donc sûr que votre PC est équipé d'un port parallèle EPP (ou ECP) si vous choisissez ce mode de communication.

Le logiciel qui nous a été remis avec le programmeur ALL-11P2, dans l'emballage d'origine, n'était pas la toute dernière version. Nous avons eu quelques petits soucis pour faire fonctionner le programmeur ALL-11P2 via une liaison par le port parallèle sous Windows 2000 avec la version du logiciel fourni dans l'emballage. Fort heureusement, le site Internet du constructeur (<http://www.hilosystems.com/>) permet de télécharger une version plus récente du programme qui règle le problème (prévoyez tout de même de télécharger un fichier de 2,8Mo environ). Précisons également que, sous Windows NT, il faut avoir des droits d'administrateurs pour pouvoir installer le logiciel. Sous Windows Me/98/95, ces soucis vous seront épargnés (l'auteur a également testé le programmeur ALL-11P2 sous Windows 98).

Une fois l'installation du programmeur terminée, la prise en main est relativement simple. L'interface utilisateur du programme, qui accompagne le programmeur ALL-11P2, est sobre mais parfaitement fonctionnelle comme vous pouvez l'apercevoir sur la vue d'écran reproduite en **figure 1**. Ce qui frappe l'utilisateur



1 Une très importante liste des composants gérés par l'appareil

lors de la première mise en service de l'appareil, c'est la liste impressionnante des fabricants de semi-conducteurs et des références qui sont supportées. La **figure 2** vous en donne d'ailleurs un aperçu. La liste des références supportées par le programmeur ALL-11P2 ne cesse de croître au fil des mises à jours, disponibles via Internet. C'est un point très important qui mérite une attention particulière au moment de l'achat d'un tel équipement.

Le programmeur ALL-11P2 est agréé par

la plupart des fabricants de semi-conducteurs, ce qui signifie que l'algorithme de programmation utilisé pour un composant bien précis est toujours conforme aux spécifications du fabricant. Le programmeur identifie les composants qui le permettent (c'est à dire la grande majorité) afin de vérifier que l'algorithme sélectionné convient bien au composant, afin de ne pas endommager des composants inconnus ou des composants d'une autre référence qui auraient été manipulés par erreur. Certes, cette fonction est un avantage en production,



2 Programmation d'un composant AT49F040

mais en laboratoire cela peut être quelquefois bloquant lorsque l'on utilise un composant de seconde source qui n'est pas reconnu par le programmeur (ce qui ne risque pas d'arriver souvent avec le programmeur ALL-11P2, étant donné sa liste de références).

Bien souvent, pour des composants classiques (telle qu'une EPROM 27C256 par exemple), l'algorithme de programmation d'un fabricant est tellement proche de l'algorithme de ses concurrents qu'il est possible de programmer les composants avec l'un ou l'autre des algorithmes (bien entendu, cette pratique ne pose pas de problème pour une utilisation en labo, mais elle est à proscrire dans le cas d'une utilisation en grande série).

Les versions anciennes des programmeurs de la marque HI-LO SYSTEMS disposaient d'une famille de composants génériques dans leur liste de choix. Ceci permettait de programmer des composants classiques avec un algorithme générique sans avoir à se soucier de la marque du fabricant du composant.

Ceci n'est plus possible avec le programmeur ALL-11P2, ce qui n'est pas forcément gênant pour l'instant, étant donné que ce programmeur fait l'objet d'un très bon suivi actuellement. Cependant, d'ici quelques années, rien ne dit que ce sera toujours le cas et il est dommage qu'il ne soit pas possible de créer soi-même de nouvelles références, moyennant la saisie de quelques paramètres.

Ceci étant dit, l'ensemble des fonctions proposées par le programmeur ALL-11P2 est relativement complet. La manipulation des données est possible directement sous le contrôle du programme. Bien entendu, le programme dispose d'un éditeur de données en mode hexadécimal. L'éditeur travaille sur les données chargées préalablement en mémoire, soit à partir d'un fichier, soit par la lecture du contenu d'un composant programmé, ce qui est tout à fait classique. Dans le cas des PAL et autres circuits logiques programmables, l'éditeur permet de manipuler l'état des "fusibles" qui définissent la fonction logique du circuit (fuse map).

Certaines fonctions, un peu moins clas-

siques, sont également proposées avec le programmeur ALL-11P2. Bien souvent, les compilateurs pour des microprocesseurs 16 bits ou 32 bits produisent un fichier compilé, unique et linéaire, qui représente la succession des octets à placer dans le champ de vision du microprocesseur.

Généralement, le fichier produit ne tient pas compte du mode d'accès aux données qui sont externes au microprocesseur. Par exemple, pour un microprocesseur 32 bits, il est fréquent que les données soient réparties dans 4 EPROM de 8 bits. Dans ce cas de figure, l'exploitation du fichier généré par le compilateur nécessite un traitement particulier, ce que propose le programmeur ALL-11P2. Le traitement inverse est également disponible, ce qui permet de reconstituer le programme vu par le microprocesseur afin d'être désassemblé ou analysé par un "debugger".

Mais les fonctions du programmeur ALL-11P2 ne s'arrêtent pas là. L'appareil se transforme également en testeur de composants, ce qui peut s'avérer très utile pour les laboratoires de maintenance. Le programme qui accompagne le programmeur ALL-11P2 permet de tester les circuits des familles TTL et CMOS ainsi que les mémoires RAM les plus connues. En ce qui concerne les circuits logiques, le programme est capable de reconnaître automatiquement les composants les plus courants, ce qui peut rendre service lorsque l'on est face à un composant dont le marquage a été effacé. Enfin, pour tester des composants spécifiques, le programme permet, également, de créer des vecteurs de tests, ce qui le rend vraiment universel.

Bien entendu, pour programmer les composants en boîtier CMS, il faudra faire appel à l'utilisation d'un adaptateur spécifique, en fonction du boîtier du composant. Le programmeur ALL-11P2 est livré en standard avec un adaptateur pour les boîtiers DIL (jusqu'à 40 broches). L'entraxe du support à force d'insertion nulle est compatible avec des boîtiers larges (DIL 40 par exemple) et des boîtiers étroits (DIL 14 par exemple). De nombreux adap-

tateurs sont disponibles en options pour couvrir la majorité des boîtiers CMS du marché.

Pour la programmation de petites et moyennes séries de composants, il est également possible de programmer plusieurs composants en même temps à l'aide d'un adaptateur spécifique. On regrettera que l'appareil ne soit pas équipé d'une vis de mise à la masse qui permettrait de relier facilement l'appareil à l'équipement antistatique indispensable à ce type de travaux.

Rappelons que, lorsque l'on manipule des composants programmables en vue de produire des équipements destinés à une utilisation réelle, il est indispensable de le faire avec des protections antistatiques (prévoir au minimum un tapis antistatique associé à un bracelet antistatique, tous deux reliés correctement à la terre). Sans cette protection, vous exposez les composants manipulés à la destruction, dans le meilleurs des cas. Là, au moins, la sanction est nette et le problème est facile à identifier. Sinon, dans le pire des cas, les composants seront seulement fragilisés et la détection de la panne surviendra bien plus tard, ce qui peut être bien plus préjudiciable (durée de rétention des informations amoindrie pour les mémoires, pannes aléatoires et fortement conditionnées par les conditions climatiques, etc.). Il va sans dire que l'analyse de tels défauts est bien plus compliquée et qu'il est assez rare de faire le lien avec les travaux de programmation des mémoires ou des microcontrôleurs sur ce type de panne.

En conclusion, il n'est pas exagéré de dire que le programmeur ALL-11P2 soutient la comparaison avec des équipements professionnels de grande marque, sans avoir à rougir de rien.

Son excellent rapport qualité/prix et sa vocation universelle en fait sans nul doute un programmeur de choix pour les petits laboratoires et les amateurs acharnés.

Distribué notamment par la société
PROGRAMMATION
Prix : 8300 F TTC (1265,33 €)

Kit de développement pour module de reconnaissance vocale **Voice Extreme 364**



Conçu par la société SENSORY, les modules de reconnaissance vocale Voice-extreme 364 permettent de réaliser des petits systèmes autonomes avec des fonctions vocales très avancées. Pour faciliter la conception et la mise au point des programmes pour les modules Voice-extreme 364, il existe maintenant un kit de développement que nous vous proposons de découvrir dans ces pages.

Le kit Voice-extreme 364 est un outil de programmation destiné à développer très rapidement et simplement des applications de reconnaissance vocale autonomes. Car, en effet, les modules Voice-extreme 364 sont architecturés autour d'un microprocesseur capable de faire bien d'autres choses en plus de la gestion des fonctions sonores. Vous pouvez utiliser les ports d'entrée/sortie libres du microcontrôleur ainsi que ses timers ou son UART. Avec un module Voice-extreme 364, il est donc facile de piloter un afficheur LDC, contrôler un petit clavier matricé, communiquer avec un PC via une liaison série, etc.

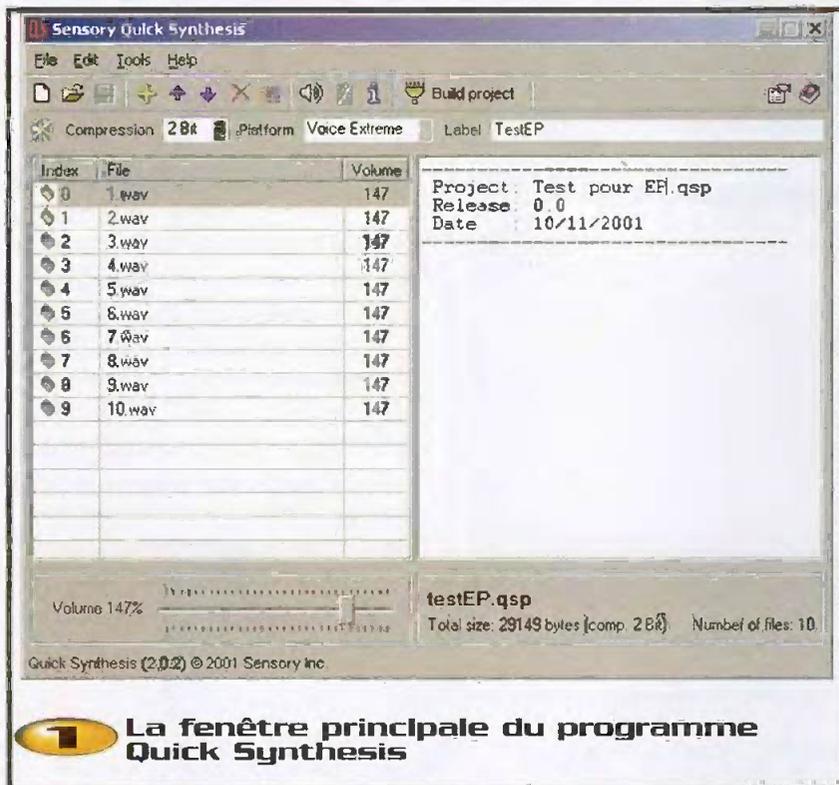
Sans aucun outil spécial, la programmation des modules Voice-extreme 364 n'est pas vraiment évidente, même si toutes les fonctions vocales sont disponibles à l'aide du simple appel d'une fonction dans la ROM du système. En revanche, avec le kit Voice-extreme 364, la programmation d'une application complète devient presque un jeu d'enfant pour qui conquiert le langage C. En

effet, le kit de développement Voice-extreme 364 se programme en langage VE-C. Il s'agit d'un langage qui est très proche du langage C ANSI, même s'il a subi quelques adaptations liées aux fonctionnalités du microprocesseur des modules Voice-extreme 364. A titre d'information, grâce au kit de développement Voice-extreme 364, il nous a fallu moins de deux heures pour mettre en place une petite application qui compte à voix haute lorsque l'utilisateur appuie sur l'un des boutons poussoir de la maquette. Ajoutons à cela que la plus grande partie des deux heures a été consacrée à la lecture de la documentation ! Tout cela pour dire que ce kit de développement se révèle vraiment très efficace.

Le kit de développement Voice-extreme 364 contient une platine équipée d'un module Voice-extreme 364 installé sur un support. La platine est également équipée d'un petit haut-parleur, un microphone, des LED, des boutons poussoirs et une liaison série RS232. Un bloc d'alimentation secteur est aussi fourni

avec le kit ainsi que le câble de liaison série RS232 pour le raccordement au PC. La carte fournie dans le kit dispose également d'une petite zone de prototypage rapide, ce qui permet de tester rapidement l'adjonction de fonctions autour du module Voice-extreme 364.

Le logiciel qui accompagne ce kit est compatible avec un PC fonctionnant sous Windows 95/98/ME ou Windows/NT/2000. Le PC devra disposer au minimum de 16 MB de RAM, 15 MB d'espace disponible sur le disque dur, ainsi qu'un port RS232 libre. Le logiciel est fourni sur CD-ROM et il est accompagné d'un manuel de démarrage rapide en anglais. La documentation complète du produit est disponible sous la forme d'un fichier d'aide mais seulement en langue anglaise. Comme vous pouvez vous en douter, les exemples des programmes qui sont fournis avec ce kit sont eux aussi en anglais. Si vous ne maîtrisez pas du tout la langue de Shakespeare, l'utili-



1 La fenêtre principale du programme Quick Synthesis

sation du kit risque donc de vous rebuter quelque peu.

Le logiciel qui accompagne le kit est constitué de deux modules principaux : Quick Synthesis et Voice-extreme IDE. Le programme Quick Synthesis est conçu pour vous aider à créer et à utiliser très facilement des messages vocaux que vous pourrez enregistrer vous-même (via le microphone de la platine) et faire prononcer au module selon les besoins de votre application. Il permet également de travailler facilement avec des fichiers au format WAV que vous aurez enregistrés avec la carte son de votre

PC. Cependant, l'utilisation des fichiers WAV produit des résultats beaucoup moins précis pour la reconnaissance vocale (ce qui est logique puisque la reconnaissance vocale a lieu via le microphone de la platine et que ses caractéristiques sont différentes de celles du micro utilisé avec votre carte son). En revanche, pour diffuser de la musique ou des messages vocaux, cette possibilité est très intéressante. Notez toutefois que le programme Quick Synthesis ne sait traiter que les fichiers WAV enregistrés dans le format PCM 22,050 kHz, mono 16 bits. Pour exploiter des fichiers enregistrés dans un autre format, vous

devez les transformer, au préalable, à l'aide du magnétophone de Windows par exemple (voir les options de la fonction "enregistrer sous").

Le programme Quick Synthesis compresse les fichiers audio pour tirer le meilleur parti de la mémoire des modules Voice-extreme 364. Vous pouvez ajuster le taux de compression selon que vous souhaitez privilégier la longueur des messages ou bien leur qualité de reproduction. Pour que votre projet puisse être compilé, la totalité des messages sonores de votre projet ne devra pas passer 64 Ko. Il est à noter que le message d'erreur affiché par le programme Quick Synthesis, en cas d'échec de compilation en raison d'un dépassement des 64 Ko, n'est pas très parlant. Pensez à utiliser le taux de compression maximum si d'aventure le programme Quick Synthesis refuse de finir la compilation de votre projet.

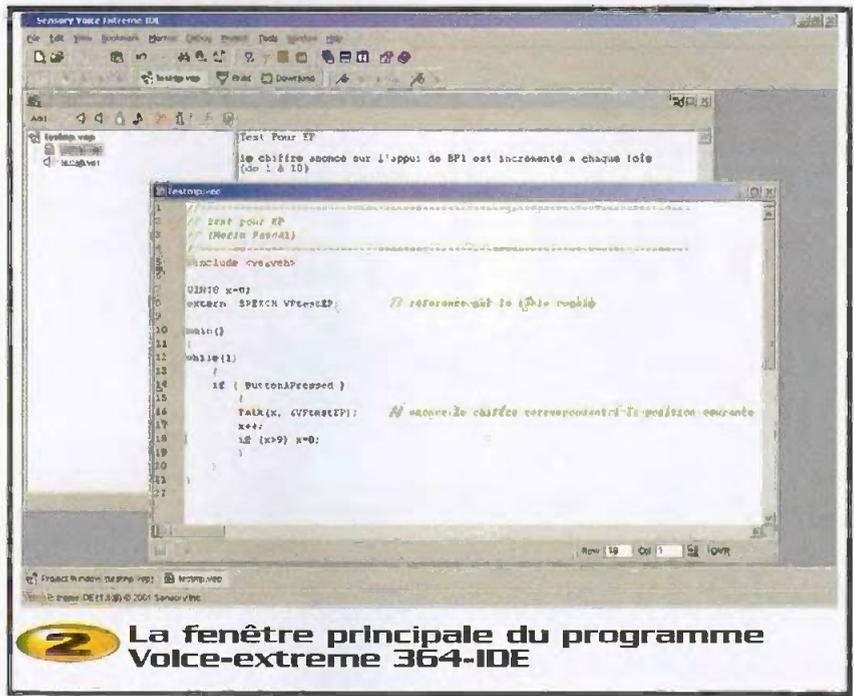
Le programme Quick Synthesis produit deux fichiers à l'issue de la compilation. Le premier fichier porte l'extension "ves" et contient les sons compilés et compressés dans un format compatible avec le programme Voice-extreme IDE. Le fichier portant l'extension "ves" devra ensuite être inclus dans votre projet, sous "Voice-extreme IDE", afin que les échantillons sonores soient disponibles pour votre application. Le second fichier porte l'extension "veh". Il contient la définition des identificateurs nécessaires pour adresser les messages que vous avez ajoutés au projet grâce à Quick Synthesis.

Pour que la création de votre projet se déroule correctement, il est important de respecter le positionnement des fichiers, selon leur extension, dans l'arborescence des sous répertoires créés lors de l'installation des logiciels du kit de développement. Si vous ne le faites pas, le compilateur n'arrivera pas à terminer l'édition des liens et le fichier binaire à télécharger dans la maquette ne sera pas créé. La documentation du produit contient toutes les indications nécessaires sur ce sujet.

Dans l'environnement du programme Voice-extreme IDE, la création d'un projet ressemble vraiment à la création d'un pro-



Gros plan sur le module Voice-extreme 364 installé sur son support



2 La fenêtre principale du programme Voice-extreme 364-IDE

programme classique en langage C. Un rapide coup d'œil aux exemples fournis avec le kit suffit à s'en convaincre. Si vous connaissez le langage C, il ne vous faudra pas longtemps pour démarrer votre projet. Bien

entendu, pour avoir accès au matériel de la platine, vous devrez prendre le temps de découvrir dans la documentation quelles sont les fonctions à appeler pour la gestion des actions sur les boutons poussoirs,

pour allumer les diodes LED et pour gérer les messages vocaux (fonction de reconnaissance ou simplement diffusion d'un message sonore).

Lorsque le projet est compilé, il ne vous reste plus qu'à le télécharger dans la mémoire FLASH du module Voice-extreme 364. Pour mettre au point le programme, Voice-extreme IDE dispose également des fonctions de mises au point d'un "débugger" traditionnel.

Ce petit kit bien sympathique est vendu à un prix tout à fait raisonnable. Étant donné la souplesse et le gain de temps qu'il procure, il serait vraiment dommage de s'en passer pour développer des applications vocales bâties autour des modules Voice-extreme 364.

Distribué notamment par Lextronic
 Prix : 1177 F TTC (179,43 €)



ModuleScope
 Oscilloscope 2 voies 20MS/s
 Entrées isolées flottantes
 CEI 1010 - EN 61010

Appareil de base NM200 avec afficheur rétro-éclairé avec le module oscilloscope 2 voies 20MS/s NM202 **3780 F HT 4520 F TTC**
 2 sondes 1:1/10:1 MZ35, logiciel compatible Windows 95/98/NT, câble d'interface PC-RS232, adaptateur secteur, kit de débouchage et notice d'utilisation en français (anglais, néerlandais, espagnol ou allemand disponibles). Livré dans une mallette de transport rigide.

Module complexe fréquencemètre 2voies 20Hz NM212 **En option**
 Caractéristiques : entrées indépendantes et isolées, déclenchement interne et externe, affichage 7 digits, base de temps compensée en température 10°, analyse statistique. Fourni avec notice et logiciel compatible Windows 95/98/NT. Disponible 2ème semestre 2001.

Module analyseur logique NM224 **En option**
 Caractéristiques : 16 voies, 40MS/s, déclenchement interne et externe variable, 32Kb par voie, compteur d'événements 16 bits. Fourni avec le démasseur pour micro contrôleur 8051 et 230 (câbles en préparation), câbles de mesure, notice, logiciel compatible Windows 95/98/NT. Disponible 2ème semestre 2001.

HAMEQ S.r.l.
 8, avenue de la République
 94800 VILLIERS-LE-BEL
 Tél. : 01 46 77 81 81
 Fax : 01 47 39 55 44
 E-mail : hameqcom@orange.fr

HAMEQ
 Instruments

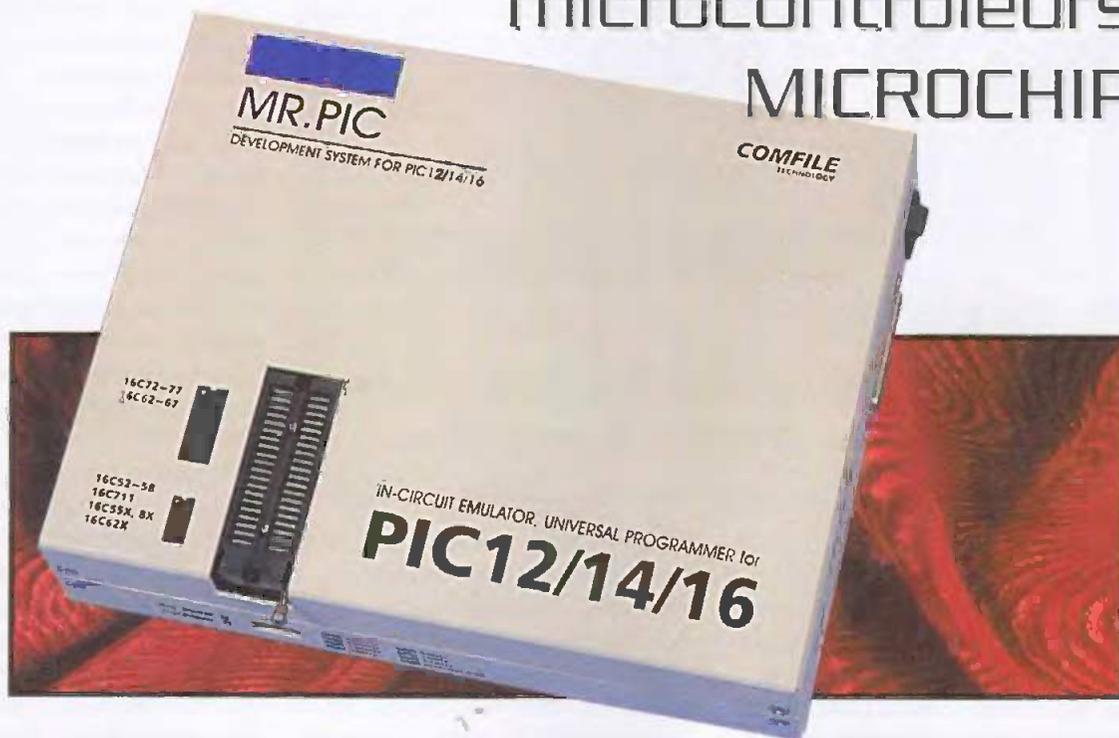
Electronique de Loisirs
Modélisme
Audio
Connectique informatique

Le Catalogue O10C en ligne

www.o10c.com

Emulateur temps réel MR-PIC

pour les
microcontrôleurs
MICROCHIP



S'il y a bien un appareil qui se révèle vite indispensable lorsque l'on développe des logiciels pour un microcontrôleur, c'est bien un émulateur de type "temps réel". Il est, en effet, pratiquement impossible de mettre au point un logiciel complexe sans avoir recours à ce type de matériel.

Le gain de temps, qu'apporte l'utilisation d'un émulateur pendant les phases d'intégration et de mise au point d'un logiciel, est tellement important que les professionnels n'envisagent vraiment pas de se passer de ce type d'appareil.

Selon les familles de microcontrôleurs à supporter, un émulateur peut se révéler extrêmement coûteux, ce qui est rarement compatible avec le budget d'un amateur (sachant que le prix d'un émulateur temps réel peut varier entre 10 000F et 300 000F). Fort heureusement, de très bons produits, vendus à des prix plus raisonnables, sont apparus sur le marché. C'est le cas, notamment, de l'émulateur MR-PIC qui a été conçu pour supporter les microcontrôleurs de la famille PIC, très utilisés par les amateurs.

Si vous avez déjà utilisé un émulateur temps réel, nous n'avons sûrement pas besoin de vous convaincre de l'utilité d'un tel équipement. Sinon, quelques explications s'imposent.

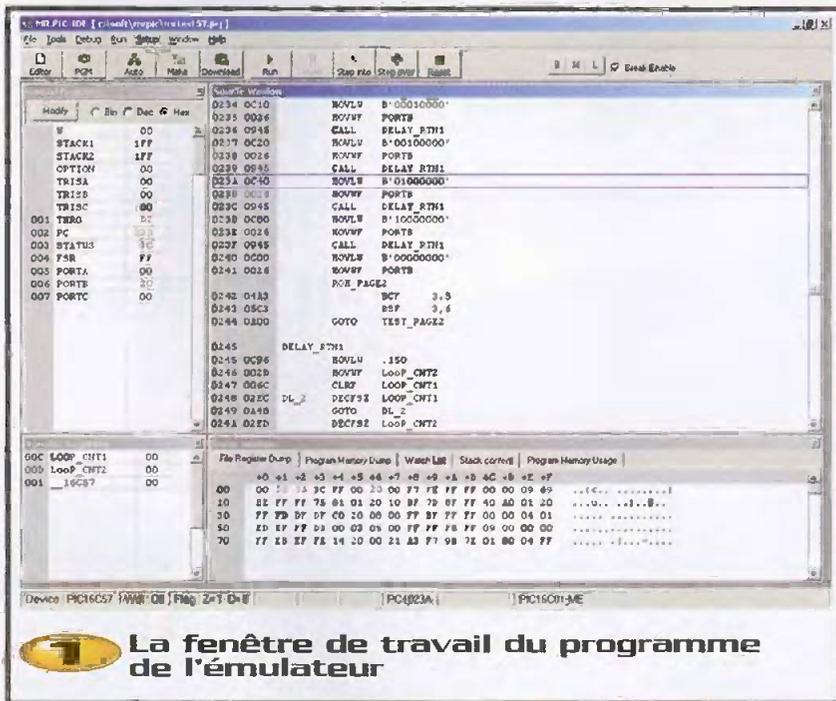
Étant donné le prix d'un émulateur temps réel, comme nous l'avons mentionné dans l'introduction, il est bien rare que les amateurs s'intéressent à ce type d'équipement.

Il est vrai que, pour ceux d'entre vous qui êtes vraiment motivés, il est possible d'apprendre à programmer un microcontrôleur sans forcément avoir besoin d'un émulateur. Mais il faut bien avouer, que dans ce cas de figure, même la mise au point d'un tout petit logiciel peut, très vite, se transformer en parcours du combattant. En effet, n'ayant aucune visibilité sur le contenu des registres et de la mémoire du microcontrôleur, l'utilisateur doit faire preuve de beaucoup d'imagination pour savoir où en est le programme. Bien entendu, il existe de nombreuses techniques et astuces qui permettent de s'en sortir plus ou moins facilement lorsque l'on souhaite mettre un logiciel au point. Lorsque le matériel, qui accueille le programme, dispose d'une liaison

série, il est possible de modifier le programme à tester pour émettre des messages qui seront visualisés à l'aide d'un terminal (ou un PC avec Hypertext, par exemple).

D'ailleurs, pour exploiter pleinement ce mode de travail, il existe des petits logiciels qui peuvent se greffer à votre application pour transformer votre maquette en mini-émulateur. Avec ce type de fonctions, il est même parfois possible de disposer des points d'arrêt sur le programme à tester et d'examiner le contenu des registres, ce qui est d'une très grande aide. Malheureusement, ces logiciels rajoutés à l'application réelle consomment des ressources (place en RAM, ajout de données sur la pile, etc.), ce qui n'est pas toujours compatible avec l'application envisagée.

Lorsque l'on ne peut pas faire appel à de tels "moniteurs", on peut parfois utiliser des broches du microcontrôleur, qui ne sont pas utilisées dans



La fenêtre de travail du programme de l'émulateur

l'application, pour signaler au monde extérieur certains événements.

Cependant, ces actions ont forcément lieu sous contrôle du logiciel, ce qui implique que le programme à mettre au point fonctionne encore, au moins en partie, puisque le code de "debug" doit pouvoir être appelé. Selon le type de "bug" auquel on est confronté, ce n'est pas toujours possible.

Même si les fonctions d'aide à la mise au point peuvent être appelées par le programme à tester, elles ne permettent pas souvent de trouver l'origine des problèmes

qui surviennent de façon dynamique, suite à un ou plusieurs événements (internes ou externes) synchrones ou non. Dans des tels cas de figures, seule l'utilisation d'un émulateur de type temps réel (c'est à dire capable d'exécuter le programme à sa vitesse d'exécution réel) permet de traquer les "bugs" de façon rationnelle. De plus, le code ajouté par un "moniteur de debug" vole des cycles d'exécution au microprocesseur. Du coup, le programme à tester ne s'exécute pas à la vitesse réelle qu'il aura ensuite. Cela peut cacher des problèmes qui finissent par apparaître lorsque l'on pense avoir fini de mettre au point le pro-

gramme et que l'on recompile ce dernier sans inclure le moniteur.

Nous arrêtons là notre démonstration. La conclusion de tout ce discours est simple : Pour faire ses premiers pas dans la mise en œuvre d'un microcontrôleur ou pour concevoir des petits projets plus ou moins simples, il est possible de "bricoler" pour s'en sortir et se passer d'investir dans un émulateur temps réel. Ceci dit, il n'est pas rare de rencontrer des "bricoleurs" de génie qui arrivent à faire des prouesses avec peu de moyens. Malgré tout, si l'on doit mettre au point des projets complexes, l'utilisation d'un émulateur temps réel devient vite une nécessité pour avoir une chance d'aboutir dans des délais raisonnables et pour travailler de manière rationnelle.

L'émulateur MR-PIC, fabriqué par la société COMFILE, permet d'accéder à ce type d'outil pour un prix tout à fait correct (aux alentours de 1000 €). Certes, pour l'amateur qui débute, cela reste encore un investissement important. C'est pourquoi la société COMFILE propose également des émulateurs un peu plus limités mais également plus adaptés au budget des débutants. Mais les amateurs avertis, comme les professionnels, trouveront dans le MR-PIC toute la puissance d'un véritable émulateur temps réel dont ils ont besoin quotidiennement et qui couvre une bonne partie de la grande famille des microcontrôleurs PIC.

L'émulateur MR-PIC permet, en effet, d'émuler les microcontrôleurs des familles PIC12Cxxx, PIC14Cxx et PIC16Cxx, pour des vitesses d'horloge allant de 32 KHz à 20 MHz. Lorsque l'on sait que des produits concurrents sont limités à une seule famille de microcontrôleurs et qu'il faut acheter un nouveau "pod" qui coûtent pratiquement le prix du MR-PIC, pour chaque nouvelle famille de microcontrôleurs que l'on souhaite ajouter, on comprend très vite que le produit proposé par COMFILE est vraiment compétitif.

L'appareil que nous avons pu tester s'est révélé un produit de bonne qualité. Sa mise en œuvre est relativement simple pour qui est habitué à la manipulation de ce type d'équipement. Sinon, il vaut mieux prévoir de se faire "la main" sur un petit projet test,



les connecteurs raccordant l'émulateur à sa cible

constitué pour l'occasion. En effet, les possibilités du logiciel qui accompagne l'émulateur temps réel sont assez nombreuses et il vaut mieux ne pas avoir la pression d'un projet à livrer rapidement pour une première prise en main du produit (le gain de temps que nous vous avons fait miroiter en introduction n'est généralement pas acquis avant un ou deux projets pour s'habituer à l'émulateur).

Le logiciel de l'émulateur, qui est fourni dans la boîte des produits qui sont en stock depuis quelques mois, est compatible seulement avec les environnements Windows 95/98/Me. Les utilisateurs de Windows 2000 (et sûrement de Windows XP, mais l'auteur n'a pas pu le vérifier) devront télécharger la dernière version du logiciel sur le site Internet du fabricant (<http://www.comfile.co.kr/english/>). A l'issue de l'installation du logiciel, les utilisateurs de Windows NT/2000 devront penser à lancer l'exécution de l'installation supplémentaire du driver qui autorise l'accès au port parallèle sous ces systèmes d'exploitation. Précisons que l'installation du driver supplémentaire requiert les droits d'administration sur le PC. Sous Windows 95/98/Me, cette étape n'est pas nécessaire.

Le PC auquel est raccordé l'émulateur doit posséder au minimum un processeur Pentium 166 MHz, disposer de 10 Mb d'espace disponible sur le disque dur et posséder un port parallèle libre. Par ailleurs, le système d'exploitation minimum requis, pour faire fonctionner le logiciel, est Windows 3.1, ce qui peut vous éviter d'avoir à acquérir un PC de dernière génération pour piloter l'émulateur.

Le logiciel de l'émulateur MR-PIC s'appuie sur l'environnement de développement MPLAB de MICROCHIP que vous pouvez télécharger gratuitement sur le site www.microchip.com. Cet environnement, pour la partie gratuite, permet de développer des projets en assembleurs. Si vous souhaitez développer des programmes en langage C, il est, bien entendu, possible d'acheter la licence d'un compilateur pour les microcontrôleurs PIC. Le logiciel de l'émulateur MR-PIC est compatible avec les compilateurs C les plus courants pour cette famille de microcontrôleurs (MPLAB-C, CCS et HITECH-C).

L'édition du code source et sa compilation est possible directement dans l'environnement de l'émulateur. A l'issue de la compilation, il ne vous reste plus qu'à télécharger

le programme dans la mémoire de l'émulateur pour tester votre programme. Les possibilités offertes par l'appareil sont nombreuses et nous ne pourrions pas les détailler dans ces pages. Bien entendu, les fonctions les plus classiques et indispensables pour un émulateur temps réels sont disponibles : utilisation des points d'arrêts, suivi du programme en mode pas à pas, en langage source ou (et) en assembleur, suivi des variables du programme, examens et modification du contenu des registres et des ports, etc.

On regrettera, cependant, que l'appareil ne soit pas équipé d'une trace temps réelle. Cette fonction est bien utile pour avoir l'historique des dernières instructions exécutées par le microprocesseur avant de passer sur un point d'arrêt. En effet, sur les appareils haut de gamme il n'est pas rare de pouvoir remonter en arrière jusqu'à 16000 instructions en assembleur, et même plus, ce qui est indispensable pour analyser le passage du programme dans des boucles qui ont généré de nombreuses itérations. De plus, lorsque cette fonction est présente, il n'est pas rare que le logiciel associé propose également une analyse du taux de couverture du code exécuté. Cette fonction est très utile pour garantir qu'un logiciel a été testé complètement. Cependant, sachant le prix auquel l'appareil est vendu, on ne peut pas reprocher au MR-PIC de ne pas disposer de telles fonctions car elles pèsent lourd sur le coût matériel d'un émulateur.

Ceci étant dit, le rapport prix/performance du MR-PIC reste tout de même exceptionnel, d'autant plus que l'émulateur MR-PIC se transforme également en programmeur pour les microcontrôleurs qu'il supporte, ce qui est vraiment un plus.

L'appareil est livré avec tous les câbles nécessaires pour couvrir toutes les familles qu'il émule. Le MR-PIC dispose, à cet effet, de 5 connecteurs sur le côté gauche qui offrent le choix entre l'émulation de composants en boîtiers DIL18, DIL28 et DIL40. En ce qui concerne les microcontrôleurs PIC 12C508/509, l'appareil est également livré avec des adaptateurs spécifiques pour se brancher sur la cible ou pour programmer les composants. L'émulateur MR-PIC est donc livré avec tous les accessoires qui lui

2 L'émulateur est aussi un programmeur pour les microcontrôleurs PIC

sont nécessaires. Malgré tout cela, l'émulateur souffre de quelques limitations qui pourraient s'avérer gênantes dans quelques cas bien précis. Par exemple, le choix de la fréquence de travail du microcontrôleur à émuler est fixé par un jeu de microswitchs visibles sur le devant de l'appareil. L'utilisateur dispose seulement de 8

choix possibles (allant de 32 kHz à 20 MHz) et il n'est pas possible d'utiliser un oscillateur externe pour fixer la fréquence de travail. Il n'est pas possible, non plus, de faire fonctionner l'émulateur avec l'oscillateur R/C des microcontrôleurs PIC.

Si vous avez vraiment besoin d'utiliser une fréquence de travail particulière, selon les

besoins de votre application, l'émulateur ne pourra plus vous offrir un support "temps réel". Ceci étant dit, il est souvent possible d'adapter le logiciel pour s'accommoder d'une fréquence de travail approchant la valeur idéale. C'est une contrainte à prendre en compte lors de la conception du logiciel, mais peut-être également du matériel.

En conclusion, nous dirons que l'émulateur MR-PIC est un bon produit. Même si les performances de l'appareil n'égalent pas tout à fait celles des matériels haut de gamme, l'émulateur MR-PIC a l'avantage d'un excellent rapport prix/performance qui est en adéquation avec le budget des petits laboratoires et des amateurs chevronnés. Pour l'avoir essayé pendant quelques jours, l'auteur doit bien avouer que cet appareil permet de travailler efficacement. Et ne dit-on pas que les bons ouvriers ont de bons outils ?



Distribué notamment par LEXTRONIC

Prix : 6529 F TTC (995,34 €)

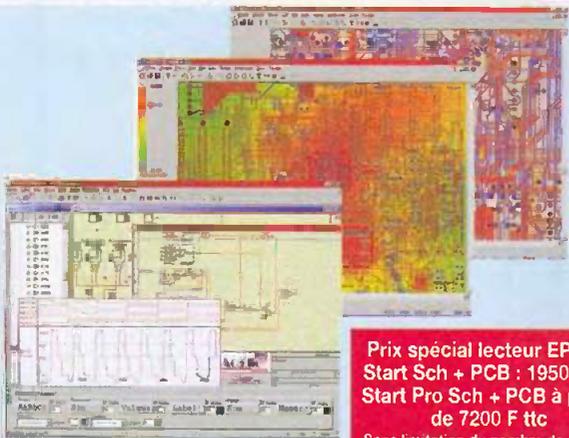


tous les accessoires sont fournis

Power Concept

La solution complète d'

Acadie Soft International



Prix spécial lecteur EP/IPC
Start Sch + PCB : 1950 F ttc
Start Pro Sch + PCB à partir
de 7200 F ttc

Sans limitation du nombre de pattes,
de connexions ni de composants

L'éditeur de **Power Concept**
enfin à la portée de tous.

Disponible en français dans ses
versions **Start** et **Start-Pro**

bureaux d'études,
enseignement, amateurs avertis.

ne payez ce que vous utilisez mais béné-
ficiez des performances et de la convivialité,
de l'éditeur d'un des plus puissants outils
professionnels du marché.

Pour plus d'informations, consultez
notre site www.acadie-soft.com

Acadie Soft International SAS au capital
de 100 000 Euros

Rue de Montfort 35380 Plélan-le-Grand

Tél. : (33) 02 99 61 81 00

Fax : (33) 02 99 61 81 01

Email info@acadie-soft.com

BON DE COMMANDE

Veuillez me faire parvenir la
version Start Start Pro
de votre logiciel

Nom :

Prénom :

Adresse :

Ci-joint mon règlement à l'ordre
de Acadie Soft International

EP 262

www.elecson.com

Composants

Câbles

Connectique

Vidéo

Outils

Alarme

Alimentations

Mesure

Haut-parleurs

Convertisseurs

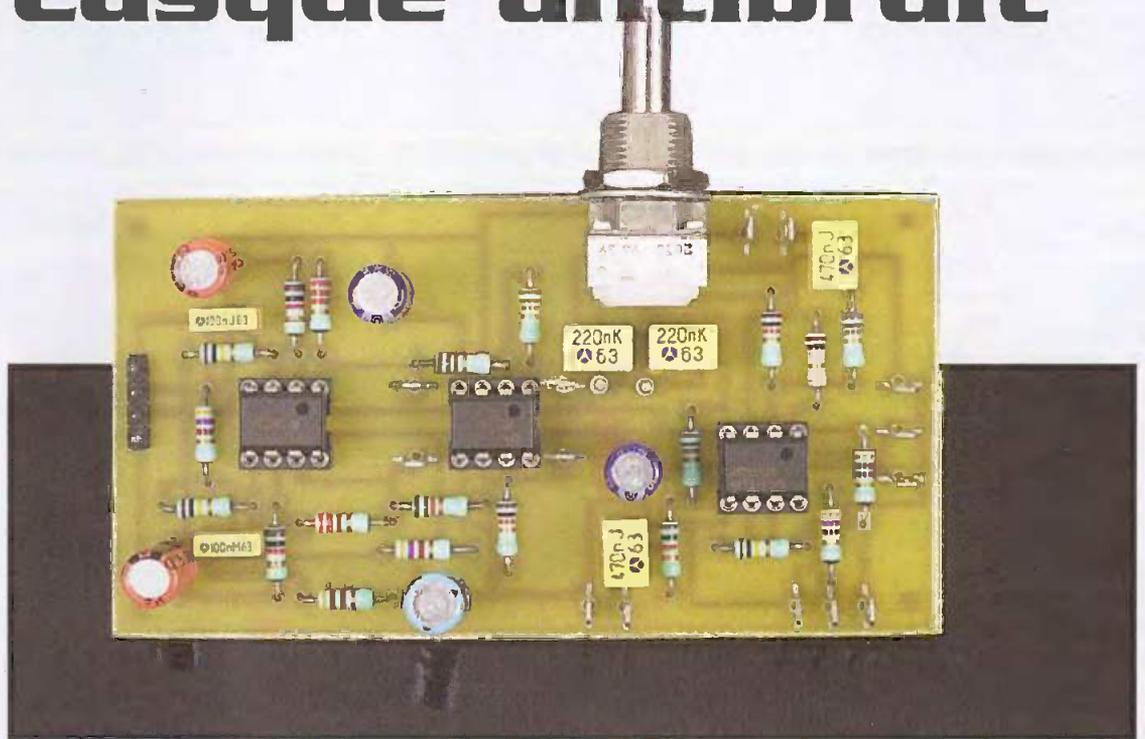
Kits (ferroviaires)

Place Henry Frenay - 4 rue Jean Bouton

75012 PARIS

Tel : 01 43 40 29 36 - Fax : 01 43 40 37 02

Casque antibruit



Le titre de cet article vous aura peut-être fait sourire en pensant qu'E.P. était devenu un catalogue de matériel de bricolage. C'est en effet dans les magasins de ce type que l'on trouve habituellement de tels accessoires. Le casque antibruit que nous vous proposons aujourd'hui, même s'il vise à réaliser la même fonction que ses homologues ne contenant aucune électronique, est cependant de nature très différente et peut recevoir deux applications distinctes.

Utilisé seul, il transforme n'importe quel casque haute fidélité en casque antibruit avec une efficacité surprenante compte tenu du principe auquel il fait appel. Vous pouvez alors vous reposer en toute quiétude toutes fenêtres ouvertes même si ces dernières donnent sur le plus bruyant des périphériques !

Il peut aussi être utilisé pour l'écoute de musique au casque, à partir d'une chaîne haute fidélité ou d'un baladeur CD par exemple, dans une ambiance bruyante qu'il se charge alors sinon d'éliminer, du moins d'atténuer très fortement.

Ces deux fonctions peuvent d'ailleurs se rejoindre car le repos peut très bien se concevoir au son d'une musique douce, débarrassée de tout bruit de fond ambiant désagréable.

Un principe très simple

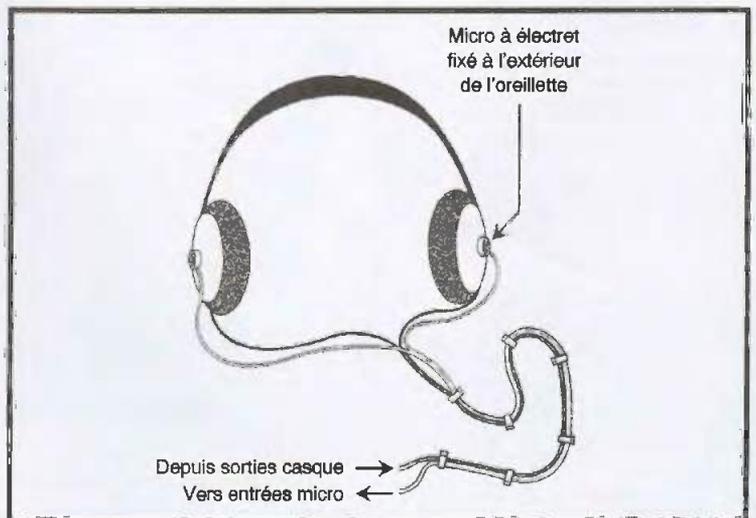
Hormis une partie électronique au demeurant relativement simple, notre casque fait appel ... à un casque bien sûr, mais à un casque légère-

ment modifié. Rassurez-vous tout de suite, nous n'allons cependant pas vous proposer de charcuter votre dernier casque hi-fi de plusieurs centaines de francs car la modification nécessaire est externe et très facilement réversible.

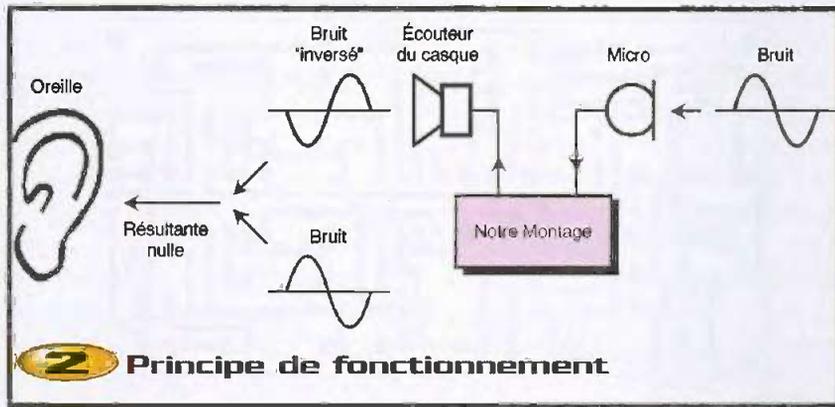
Comme le montre schématiquement la **figure 1**, elle consiste tout simplement à ajouter, sur chaque oreillette du casque mais à l'extérieur de celle-ci bien sûr, un micro à électret. Chaque micro capte ainsi le son parvenant à l'extérieur de l'oreillette de façon aussi semblable

que possible à celui que capte votre propre oreille située, elle, dans l'oreillette.

Si l'on envoie les signaux fournis par les micros aux écouteurs du casque, mais en inversant leur phase, ces derniers vont reproduire les sons externes «à l'envers», c'est à dire encore qu'ils vont se soustraire des bruits réels externes comme le montre schématiquement la **figure 2**. Le fait que le casque reproduise en outre de la musique n'a aucune influence sur ce processus, si ce n'est de nécessiter le mélange des



1
Aspect physique de notre casque antibruit



amplifié par IC_{1a} monté de manière très classique de façon à disposer d'un gain de 34 environ. La sortie de cet amplificateur aboutit directement au commutateur S_{1a} ainsi qu'à l'entrée de l'amplificateur IC_{2a}. Ce dernier est monté en étage inverseur de gain unitaire et sa sortie aboutit à son tour sur S_{1a}.

Selon la position de ce commutateur, on peut ainsi appliquer à la suite du montage le signal fourni par le micro ou son signal dont la phase a été inversée de 180° en raison de la présence de IC_{2a}. Cette sélection est rendue nécessaire par le fait qu'il nous est impossible de connaître la différence de phase réelle existant entre le signal électrique délivré par le micro et celui fourni aux écouteurs.

La sortie de S_{1a} aboutit sur le potentiomètre P_{1a} qui permet de doser l'effet produit par le montage. Il faut en effet que le niveau sonore généré en opposition de phase soit d'amplitude aussi égale que possible à celle du bruit à éliminer pour que l'annulation schématisée à la figure 2 soit la plus efficace possible. Les amplificateurs IC_{1a} et IC_{1b} sont évidemment des modèles à faible

signaux provenant des micros avec les signaux musicaux normaux, ce qui explique la raison d'être des deux applications de notre casque, présentées en introduction.

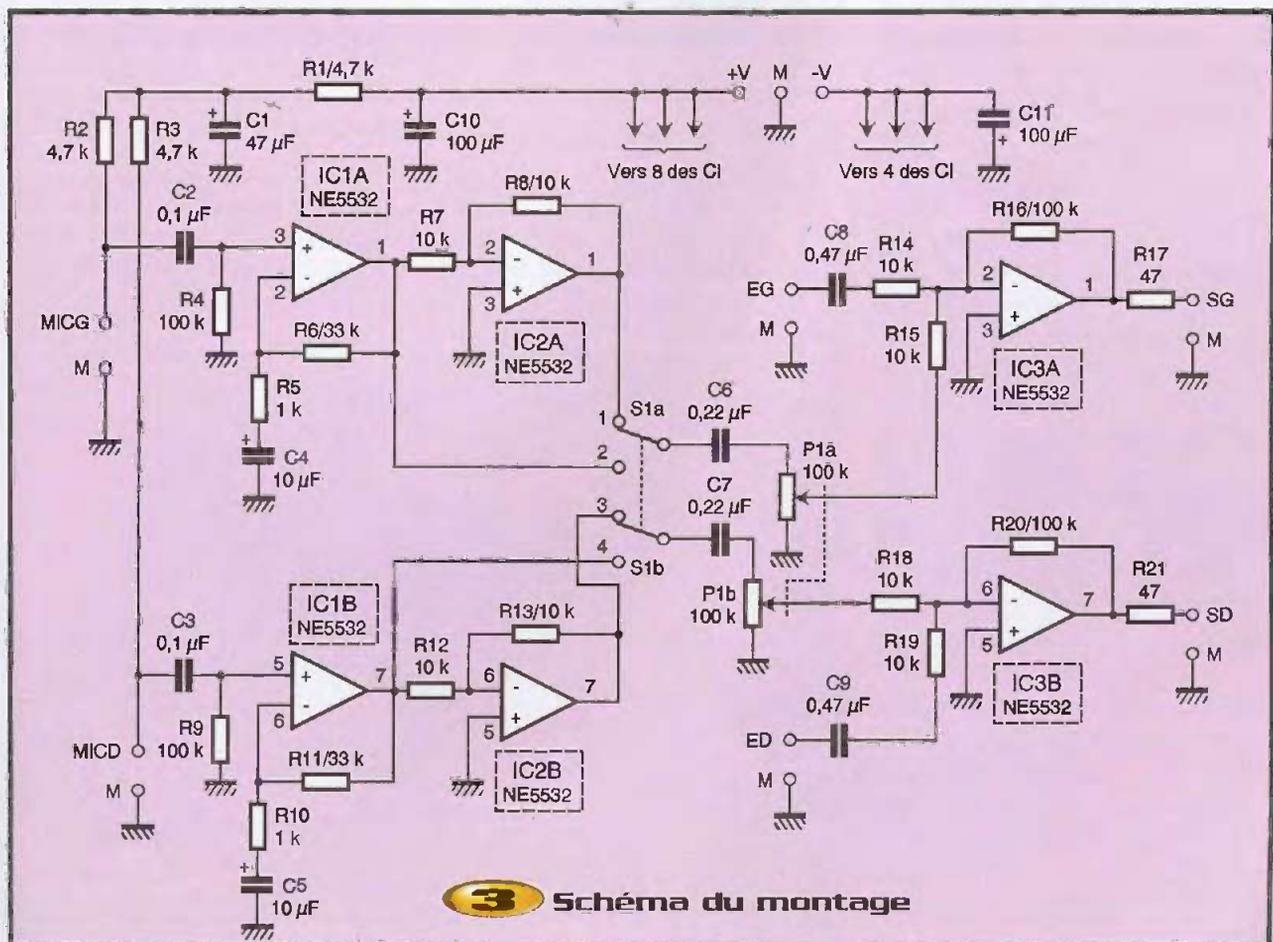
Exposé de la sorte, ce principe paraît séduisant et totalement infaillible. En pratique, ce n'est toutefois pas aussi parfait car les sons qui parviennent de l'extérieur, au niveau des micros fixés sur les écouteurs du casque, arrivent en fait sous des incidences multiples et avec des phases variables ; toutes choses que les micros utilisés, même s'ils sont de bonne qualité, ont du mal à traduire

électriquement. Néanmoins, l'effet obtenu est suffisamment spectaculaire pour justifier que nous vous proposons cette réalisation, d'autant que l'investissement nécessaire est faible.

Le schéma

Le montage étant stéréophonique, le schéma de la figure 3 est évidemment symétrique et nous n'étudierons donc que sa partie supérieure, dédiée au canal gauche.

Le signal capté par le micro gauche est



bruit afin de ne pas superposer du souffle au signal capté par le micro.

Ce sont des NE5532 très classiques, que l'on retrouve aussi dans l'amplificateur «de puissance» qui constitue le dernier étage de notre montage.

Cet amplificateur «de puissance», repéré IC_{3a}, reçoit le signal dosé par le potentiomètre P_{1a} ainsi que celui d'une éventuelle source externe, appliquée sur EG, dont il réalise le mélange. C'est sur cette entrée que vous appliquerez, par exemple, le signal provenant de votre chaîne hi-fi ou de votre baladeur laser. Il peut sembler curieux d'utiliser un NE5532 comme amplificateur de puissance pour piloter un casque, mais cela se justifie pleinement lorsque l'on regarde ses caractéristiques.

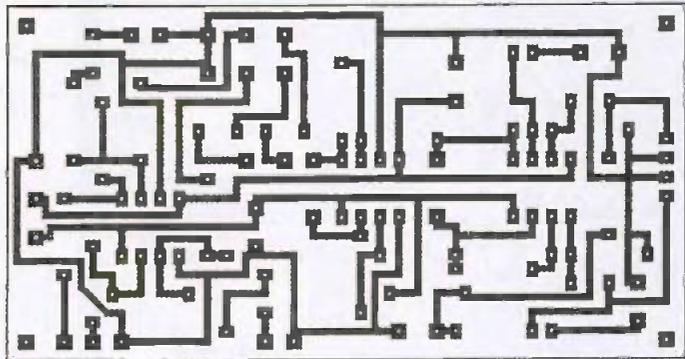
Ce circuit est en effet capable de délivrer jusqu'à 10W efficaces à une charge de 600 Ω tout en conservant une bande passante largement supérieure à 20 kHz. C'est donc plus qu'il n'en faut pour l'utilisation raisonnable d'un casque. C'est évidemment insuffisant si vous voulez vous faire éclater les tympans mais, si vous réalisez ce casque antibruit, nous pensons que cela ne devrait logiquement pas être votre cas ...

L'alimentation du montage est confiée à deux piles de 9V qui pourront être des modèles alcalines pour un usage peu fréquent et que vous remplacerez par des accus Cd-Ni pour une utilisation prolongée. Une alimentation secteur serait envisageable mais le recours à des piles ou accus rechargeables présente l'avantage d'éviter toute source d'induction parasite sur les très faibles niveaux générés par les micros et manipulés par IC_{1a} et IC₂.

La réalisation

Les composants utilisés sont d'approvisionnement facile car ce sont de grands classiques que tout revendeur tient en stock. Le circuit imprimé, dont le tracé vous est présenté **figure 4**, ne présente pas non plus de difficulté. Il supporte tous les composants du montage, potentiomètre P₁ compris.

Seul l'interrupteur S₁ est externe afin de ne pas vous imposer le choix d'un modèle précis. Le montage des composants est à faire en suivant les indications de la



4 Circuit imprimé, vu côté cuivre, échelle 1

5 Implantation des composants

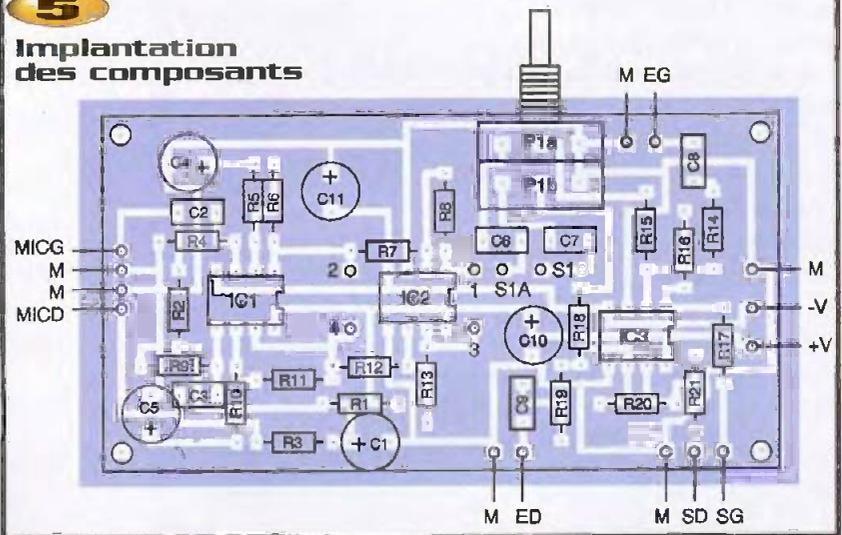
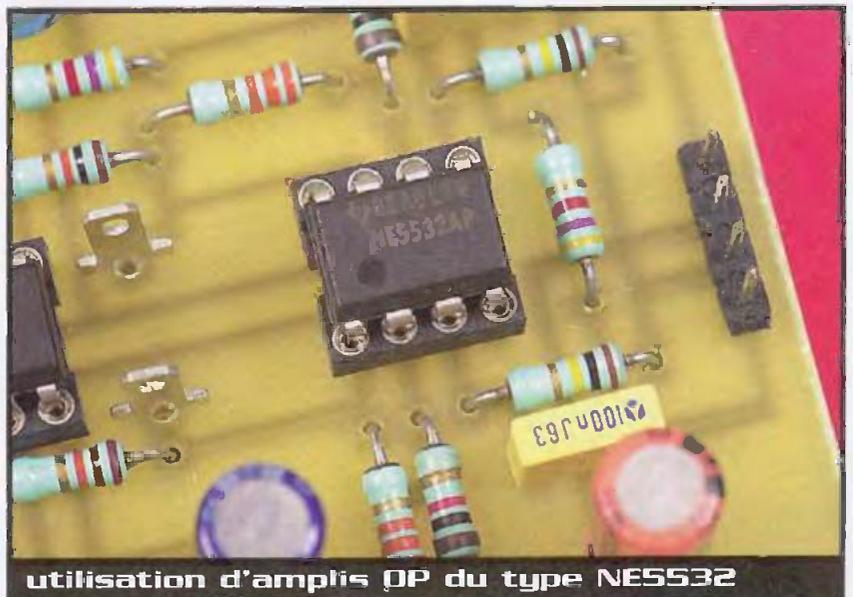


Figure 5. Vous travaillerez dans l'ordre classique : supports de circuits intégrés, résistances et condensateurs. Veillez à bien respecter le sens des composants polarisés que sont les condensateurs chimiques et à insérer correcte-

ment les circuits intégrés dans leurs supports.

La liaison du montage aux écouteurs du casque utilise le câble normal de ce dernier puisque aucune modification n'est faite à ce niveau. Par contre, les micros



utilisation d'amplis OP du type NE5532

fixés sur les oreillettes du casque doivent être reliés au montage avec du fil blindé basse fréquence. Utilisez du fil blindé double qui se présente comme du scintex lumière et dont on peut facilement séparer les deux fils. Ce fil pourra être fixé sur celui des écouteurs du casque, soit au moyen de petits colliers en plastique comme cela est schématisé figure 2, soit en les faisant passer tous les deux dans une gaine commune.

Pour ce qui est de la fixation des micros à électrets sur les oreillettes du casque, nous laissons libre cours à votre imagination. La légèreté des capsules de micros à électret permet en effet toutes les solutions (colles, Velcro, etc.). Leur position exacte n'est pas très critique mais, dans la mesure du possible, il faut que la partie sensible au son de chaque micro se trouve à peu près à la vertical du trou de votre oreille.

Utilisation

Nous vous conseillons de commencer les essais sans source sonore externe. Pla-

cez-vous dans un environnement raisonnablement bruyant et, en manœuvrant le potentiomètre P_1 et l'interrupteur S_1 , déterminez les positions les plus efficaces de ces deux réglages. Vous devez constater, sinon une disparition totale du bruit ambiant, du moins une très nette atténuation de ce dernier.

Si vous n'y parvenez pas, vérifiez si vous n'avez pas interverti les canaux droite et gauche au niveau des liaisons aux micros et aux écouteurs du casque. Si tel est le

cas en effet, le système peut devenir totalement inopérant.

Une fois cette efficacité vérifiée avec le casque utilisé seul, vous pouvez, si vous le désirez, raccorder une source sonore externe aux entrées prévues à cet effet et savourer son écoute dans le silence le plus total !

C. TAVERNIER

Nomenclature

IC₁ à IC₃ : NE5532

R₁ à R₃ : 4,7 kΩ 1/4W 5%
(jaune, violet, rouge)

R₄, R₉, R₁₆, R₂₀ : 100 kΩ 1/4W 5%
(marron, noir, jaune)

R₅, R₁₀ : 1 kΩ 1/4W 5%
(marron, noir, rouge)

R₆, R₁₁ : 33 kΩ 1/4W 5%
(orange, orange, orange)

R₇, R₈, R₁₂ à R₁₅, R₁₈, R₁₉ : 10 kΩ 1/4W 5%
(marron, noir, orange)

R₁₇, R₂₁ : 47 Ω 1/4W 5%
(jaune, violet, noir)

C₁ : 47 µF/25V chimique radial

C₂, C₃ : 0,1 µF mylar

C₄, C₅ : 10 µF/25V chimique radial

C₆, C₇ : 0,22 µF mylar

C₈, C₉ : 0,47 µF mylar

C₁₀, C₁₁ : 100 µF/25V chimique radial

P₁ : potentiomètre linéaire rotatif double de 100 kΩ

S₁ : interrupteur 2 circuits 2 positions
Micg et Micd : micros à électret
2 fils

3 supports de CI 8 pattes

LECTEUR/ENCODEUR DE CARTE A PUCE

Le système de développement BasicCard comprend :

- 1 Lecteur/Encodeur CyBermouse
- (Série ou USB)
- 1 BasicCard 1 Ko EEprom
- 2 BasicCard 8 Ko EEprom
- 1 Lecteur avec afficheur LCD (Balance Reader)
- 1 CD avec logiciel de développement
- 1 Manuel



LECTEUR/ENCODEUR DE CARTE MAGNÉTIQUE

- MCR/MSR : Lecteur simple avec interface Série/TTL/Keyboard
- MSE-6xx :
- Lecteur/encodeur avec interface série



PROGRAMMATEUR ET MULTICOPIEUR UNIVERSEL, AUTONOME, PORTABLE



ANALYSEUR LOGIQUE



EMULATEUR D'EPROM ET DE MICROCONTROLEUR



SYSTEME DE DÉVELOPPEMENT VHDL



CARTES D'ÉVALUATION, D'ACQUISITION, SIMULATION



68HC 11/12/16
68 332
80C 552
80C 31/51
80C 535

COMPILATEUR C & ASSEMBLEUR
68HC 11/12/16
68/332
80C 31/51/552
MICROCHIP PIC

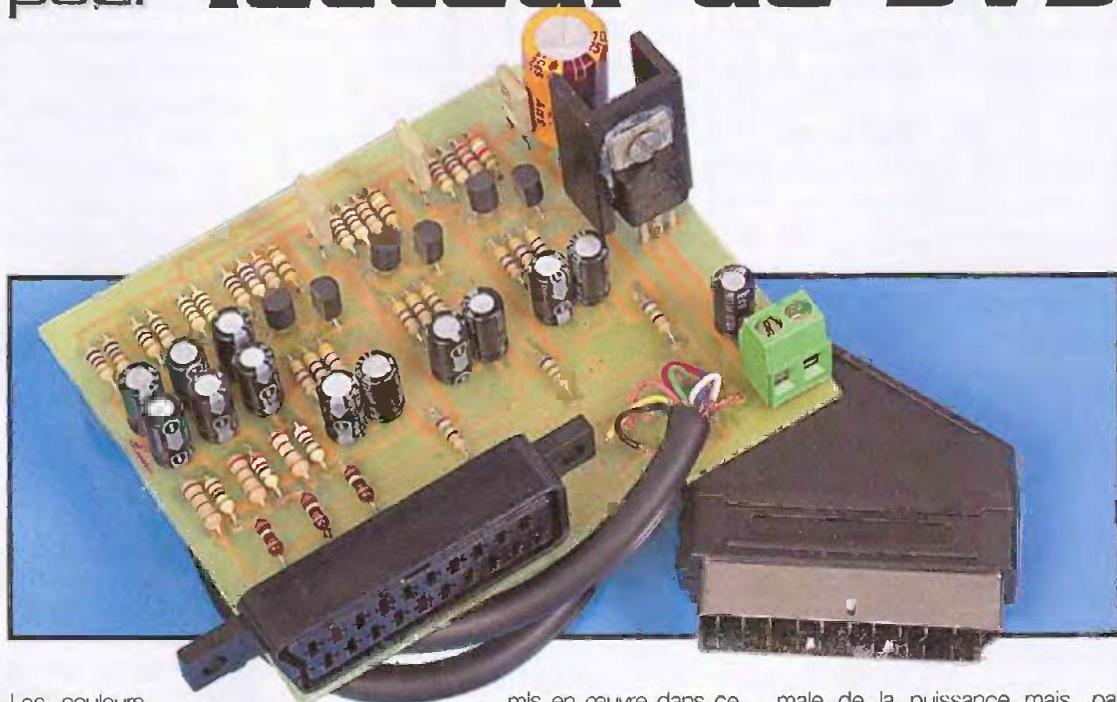
HI TECH TOOLS (H.T.T.)

27, rue Voltaire
72000 LE MANS

Tél : 02 43 28 15 04
Fax : 02 43 28 59 61

<http://www.hitechtools.com>
E-mail : info@hitechtools.com

Un réhausseur de couleur pour lecteur de DVD



Si la grande majorité des productions en DVD se caractérise par une image d'une très grande finesse, certaines souffrent d'un rendu plutôt terne des couleurs auquel va tenter de remédier ce montage très simple.

Les couleurs ternes sont des couleurs peu saturées, c'est à dire mélangées avec une forte proportion de blanc. Pour illustrer cela avec un exemple numérique, nous prendrons les niveaux de sorties suivant des signaux Rouge/Vert/Bleu du lecteur DVD à un instant donné (correspondant en fait à un point de l'image).

Valeurs crêtes

$R = 0,5V$ - $B = 0,7V$ - $V = 0,5V$: la couleur résultante est bleue.

Nous la décomposerons alors de la façon suivante :

$R = 0,5V$

$B = 0,2 + 0,5V$

$V = 0,5V$

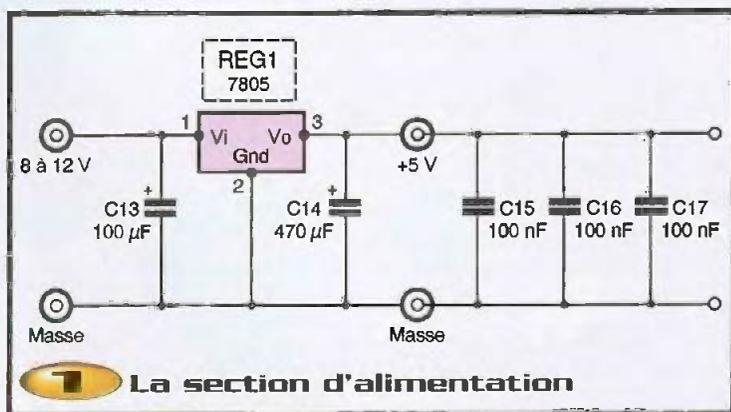
Nous voyons bien alors que cette couleur est le résultat du mélange d'un niveau de blanc de 0,5V et d'un niveau additionnel de bleu de 0,2V, ce qui en fait une couleur bleue, fort peu saturée et assez grisâtre. Pour augmenter le taux de saturation de cette couleur, nous devons diminuer les niveaux des signaux vert et rouge et, plus généralement, pour renforcer la saturation de la couleur dominante d'un point de l'image, il nous faudra diminuer le niveau des deux autres couleurs de base : c'est le principe

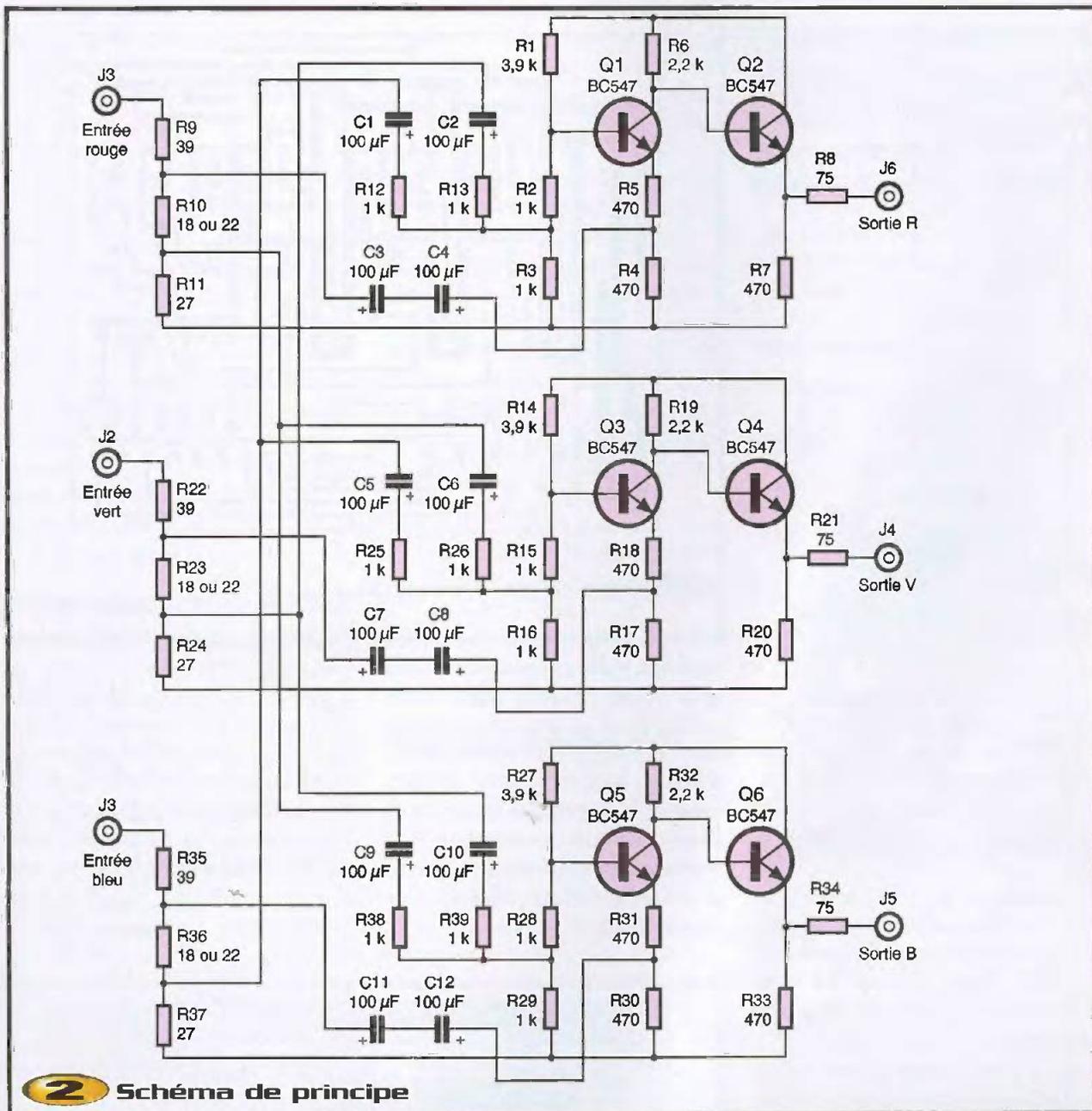
mis en œuvre dans ce montage : chaque signal Rouge, Vert ou Bleu aura tendance à diminuer les deux autres et cela d'autant plus que son niveau d'entrée sera élevé. Les différences relatives de niveau seront donc accentuées.

Ce montage est bâti pour chaque voie RVB autour d'un amplificateur vidéo classique, formé pour la voie ROUGE par le transistor Q_1 monté en base commune et le transistor Q_2 monté en collecteur commun. Le signal de sortie est fourni à la charge de 75Ω (l'entrée ROUGE du téléviseur) sous une impédance de 75Ω grâce à R_3 . Cette égalité d'impédance permet une transmission opti-

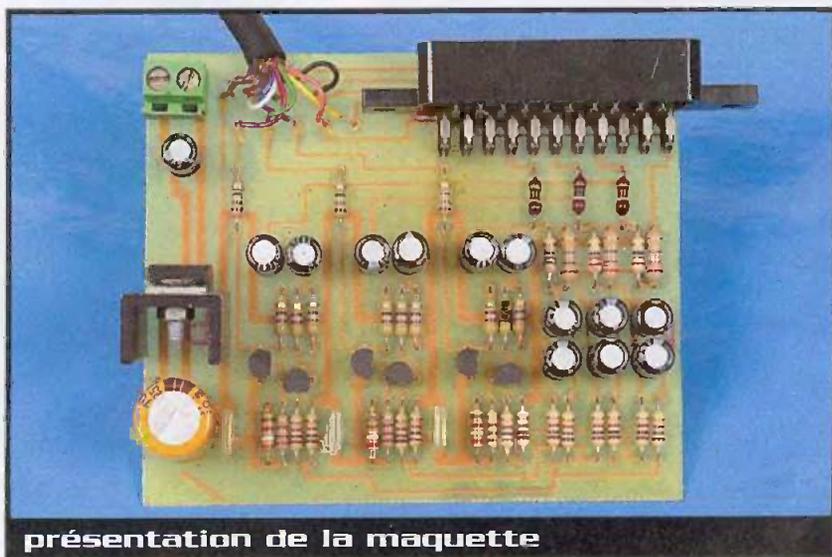
male de la puissance mais, par contre, divise par deux la tension du signal effectivement disponible à l'entrée du téléviseur par rapport à celle présente à l'émetteur de Q_2 : Si l'on doit fournir 1V crête à l'entrée du téléviseur (valeur normalisée), on devra donc avoir 2V crête à l'émetteur de Q_2 et, comme le lecteur de DVD ne fournit lui-même que 1V crête à l'entrée du montage, il nous faudra donc amplifier par deux ce signal d'entrée : c'est le rôle de Q_1 .

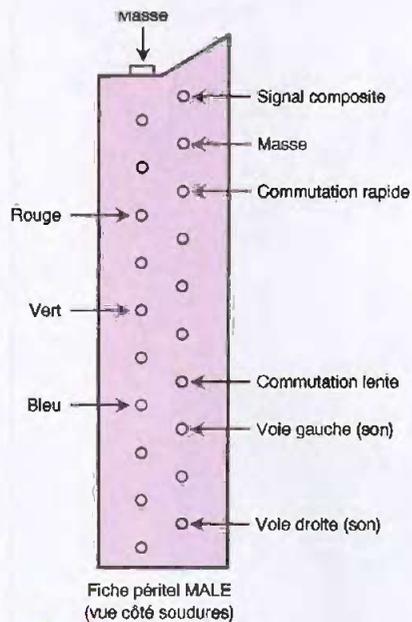
Celui-ci, monté en base commune, voit donc sa base portée à un potentiel fixe par le pont de résistance formé par R_1 et R_2+R_3 (nous ne nous intéressons pas, pour l'instant, à la présence des deux résistances R_{12} et





R_9). L'attaque de l'étage base commune se fait alors au point de jonction entre les résistances R_4 et R_5 , par une fraction du signal d'entrée prélevée sur le pont diviseur formé par R_9 et $R_{10}+R_{11}$ transmise par les condensateurs C_3 et C_4 assemblés de façon à se comporter comme un condensateur unique non polarisé. En effet, une tension continue de 0 à 2V peut être superposée aux composantes RVB et il est de fait impossible de savoir lequel des points de jonction, de R_9 et R_{10} d'une part et de R_5 et R_4 d'autre part, possèdera le potentiel le plus élevé (cependant, si l'on peut mesurer cette tension continue superposée au signal vidéo, on pourra ne conserver que le condensateur correctement polarisé et





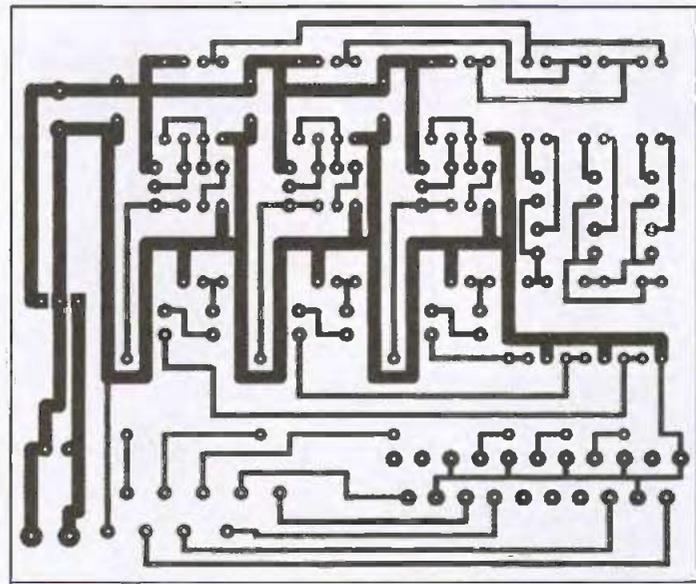
3

Brochage de la prise PERITEL

remplacer le second par un simple fil). Une fois le signal parvenu à la jonction entre R_4 et R_5 , nous observons qu'une variation positive de ce signal tend à diminuer la tension aux bornes de R_5 (car la tension émetteur/base de Q_1 varie très peu en comparaison et que le potentiel de sa base est fixe). Le courant diminue donc dans R_5 et, par-là même, dans R_6 puisque ces deux résistances sont parcourues par le même courant. Les valeurs des résistances R_9 , $R_{10}+R_{11}$, R_{12} , R_5 et R_6 font que l'amplitude du signal obtenu à la base de Q_2 est d'environ deux fois celui disponible à l'entrée du montage. On notera aussi que la liaison entre Q_1 et Q_2 est directe et que l'impédance d'entrée du montage en alternatif est proche de 75Ω car R_4 et R_6 viennent alors se mettre en parallèle avec $R_{10}+R_{11}$. Cet amplificateur vidéo, que nous venons de décrire, ne modifie pour l'instant en rien la composante de couleur fournie à son entrée, il nous faut à présent considérer le rôle des résistances R_{12} et R_{13} . R_{12} reçoit une fraction de la composante Bleu, issue du pont diviseur formé par $R_{35}+R_{36}$ et R_{37} , par l'intermédiaire de C_1 . De même, R_{13} reçoit une fraction de la composante Vert, issue du pont diviseur formé par $R_{22}+R_{23}$ et R_{24} , par l'intermédiaire de C_2 . Ces fractions injectées en ce point du circuit vont servir à DIMINUER le niveau de la composante ROUGE lorsque les composantes VERT ou BLEU sont présentes et

4

Tracé du circuit imprimé

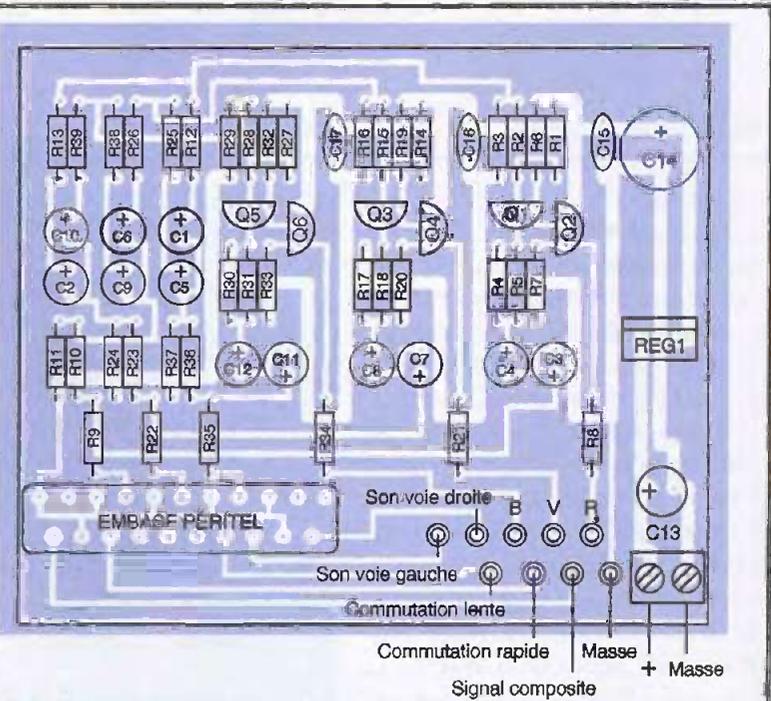


cela d'autant plus que leur niveau est élevé. Par exemple, une variation positive de la composante BLEU entraînera, par le biais de R_{12} , une augmentation de la tension aux bornes de R_3 et par la même, une augmentation du potentiel de la base de Q_1 (l'amplitude des fractions injectées a été déterminée expérimentalement et correspond environ à 10% de la valeur des composantes RVB).

Lorsque le potentiel de la base de Q_1 s'élève, cette variation est presque entièrement reportée aux bornes de R_5 ; sa tension ainsi que le courant qui la traverse augmentent, ce qui conduit à une diminution du potentiel de la base de Q_2 : le niveau de la composante rouge baisse donc. Les résistances qui forment les ponts diviseurs $R_{22}+R_{23}$, R_{24} et $R_{35}+R_{36}$, R_{37} ont de faibles valeurs en comparaison des résis-

5

Implantation des éléments



tances R_{12} et R_{13} , cela pour permettre une attaque en tension et, donc, minimiser par la même l'interaction possible entre les voies Bleu et Vert par le biais du point de jonction entre R_{12} et R_{13} .

Le potentiel du point commun entre R_3 et R_2 étant plus élevé que celui du point commun entre R_1 et R_5 , il n'y a aucune incertitude sur le sens de branchement des condensateurs polarisés C_1 et C_2 .

Chaque amplificateur vidéo de composante de couleur est donc soumis à l'action des deux autres couleurs complémentaires ; ainsi, si par exemple le niveau du signal Bleu augmente, il diminuera le niveau de Rouge, comme nous venons de le voir, mais également le niveau de Vert, et la couleur dominante diminuera beaucoup plus les deux autres que celles-ci ne la

diminuent (d'autant qu'elles se diminuent aussi entre elles), car leur action réciproque est proportionnelle à leur niveau ; il s'ensuivra alors un renforcement de la saturation de cette couleur.

On remarquera que le montage ne modifie pas la balance des blancs, puisque des niveaux identiques des trois signaux R V B les fera diminuer de la même valeur par le montage ; on notera cependant que le niveau global du blanc sera de ce fait abaissé, ce qui adoucira le contraste : c'est une limitation de ce montage.

Les résistances R_{10} , R_{23} , R_{36} déterminent l'efficacité du montage ; si l'on souhaite une image juste un peu plus colorée, on pourra prendre :

$$R_{10} = R_{23} = R_{36} = 22 \Omega.$$

Si l'on apprécie les images très colorées (mais moins contrastées...), on prendra :

$$R_{10} = R_{23} = R_{36} = 18 \Omega.$$

L'alimentation est construite autour d'un simple régulateur 5V et de quelques condensateurs de découplage : C_{13} à C_{17} .

Nomenclature

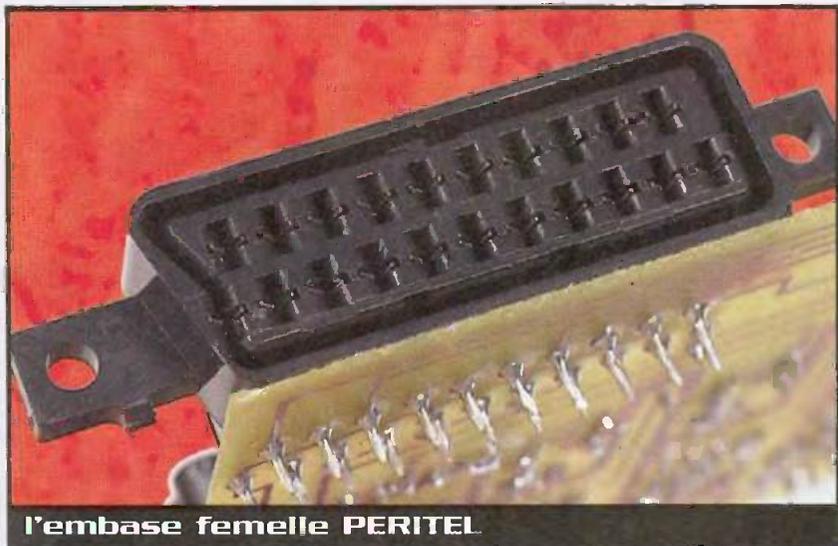
- $R_1, R_{14}, R_{27} : 3,9 \text{ k}\Omega$
- $R_2, R_3, R_{12}, R_{13}, R_{15}, R_{16}, R_{25}, R_{26}, R_{28}, R_{29}, R_{38}, R_{39} : 1 \text{ k}\Omega$
- $R_6, R_{19}, R_{32} : 2,2 \text{ k}\Omega$
- $R_4, R_5, R_7, R_{17}, R_{18}, R_{20}, R_{30}, R_{31}, R_{33} : 470 \Omega$
- $R_8, R_{21}, R_{34} : 75 \Omega$
- $R_9, R_{22}, R_{35} : 39 \Omega$
- $R_{10}, R_{23}, R_{36} : 18 \Omega \text{ ou } 22 \Omega \text{ (voir texte)}$
- $R_{11}, R_{24}, R_{37} : 27 \Omega$
- $Q_1 \text{ à } Q_6 : \text{BC546, BC547}$
- $C_1 \text{ à } C_6 : 100 \mu\text{F}$
- $C_{14} : 470 \mu\text{F}$
- $C_{15} \text{ à } C_{17} : \text{de } 27 \text{ nF à } 100 \text{ nF}$
- $\text{REG}_1 : 7805 \text{ avec dissipateur}$
- Embasse femelle Péritel
- Fiche Péritel mâle
- Bornier de deux contacts à vis
- Câble 10 conducteurs blindé

Réalisation (figure 4 et 5)

La réalisation du circuit ne pose pas de problèmes particuliers. On soudera en premier les résistances et les transistors, puis les composants plus volumineux comme les condensateurs, le régulateur et l'embasse Péritel.

On se reportera aux plans pour déterminer l'emplacement des fils à souder sur la fiche Péritel.

O. VIACAVA



l'embasse femelle PERITEL

Electronique et programmation Apprentissage autour du 68HC11

Faire découvrir les composants programmables et leur programmation de manière progressive, telle est l'ambition de cet ouvrage.

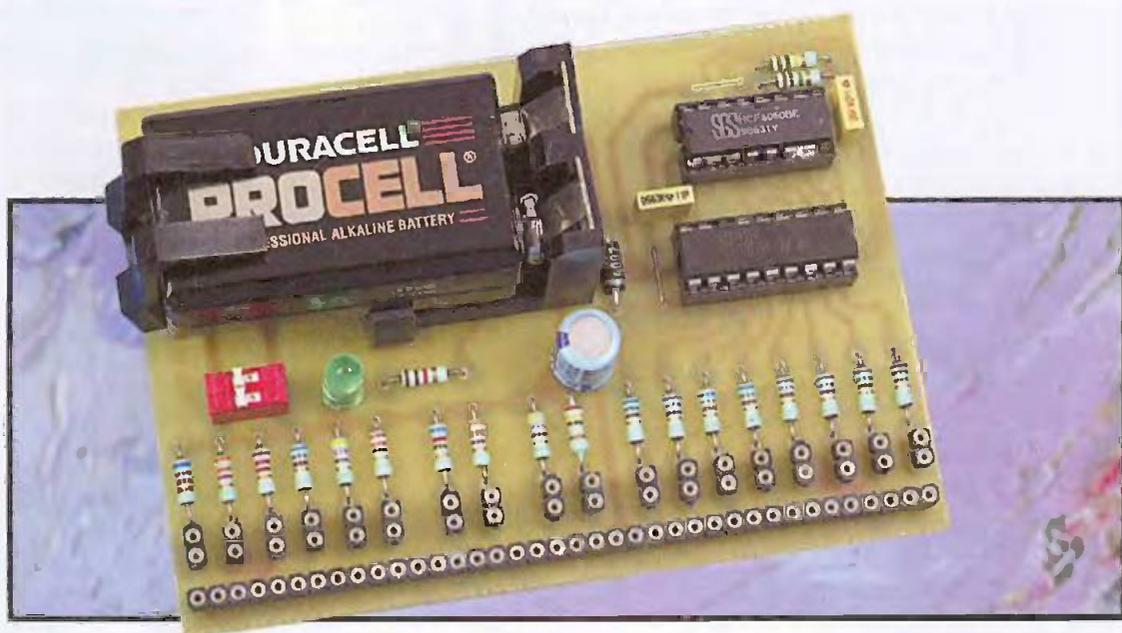


Pour ce faire, l'auteur a choisi les microcontrôleurs de la famille 68HC11 fabriqués par MOTOROLA, en raison de leur disponibilité, leur faible coût, leur facilité d'utilisation et leur richesse en mémoire et ports d'entrée/sortie. Chaque montage est l'occasion d'apprendre et de mettre en pratique des notions de base comme la logique combinatoire, la programmation en assembleur, la conception d'un programme et l'utilisation des ports en entrées/sorties. Plus qu'un manuel technique sur le 68HC11 ou ouvrage sur la programmation, ce guide d'apprentissage est destiné aux électroniciens qui souhaitent s'initier aux composants programmables et aux informaticiens confrontés à la programmation de ces composants et désireux d'appréhender l'électronique « moderne ».

A. REBOUX - E.T.S.F./DUNOD

196 pages - 158 F.

Testeur de LED



Parmi les composants les plus utiles et les plus agréables à utiliser, les LED, ou diodes électroluminescentes, se placent en bonne position. Ces petites pièces ont envahi nos circuits électroniques. Du simple voyant à l'écran multicolore, en passant par le journal lumineux défilant, elles servent partout où une indication lumineuse est nécessaire. Il existe une grande variété de LED qui diffèrent, bien sûr, par leur couleur mais aussi par leur aspect et, surtout, par leur tension et leur intensité nominale.

Même en consultant le plus beau des catalogues, il est difficile de s'imaginer le degré d'éclairement ou la couleur exacte que rendra une LED lorsqu'elle animera la face avant d'une réalisation. L'appareil que nous vous proposons aujourd'hui pallie toutes ces lacunes. Il permet de voir la LED fonctionner sous sa tension de service, mais aussi d'adapter au mieux son intensité et de retrouver sans risque sa polarité.

Jusqu'à présent, nous n'avons parlé que de l'utilisation statique d'une LED, mais certaines applications nécessitent un éclairage multiplexé où le scintillement ne doit pas être visible à l'œil.

Le testeur de LED vous permettra également de choisir la bonne fréquence pour parvenir à vos fins.

Schéma de principe

Le schéma de la **figure 1** révèle une grande simplicité. L'alimentation est fournie par une pile de 9V assurant ainsi l'autonomie indispensable à l'appareil.

Faisant suite à l'interrupteur, la LED L_1 , protégée par sa résistance de limitation R_{21} , indique la mise sous

tension correcte du testeur.

La diode D_1 évite la destruction des circuits intégrés en cas d'inversion de la pile. Les condensateurs C_2 et C_3 filtrent et découplent la tension au plus près de CI_1 . Afin de tester dynamiquement les LED pour une utilisation clignotante ou multiplexée, le circuit CI_1 est muni d'un oscillateur dont la fréquence de base est fixée par le condensateur C_1 et les résistances R_1 et R_2 .

Cette fréquence est divisée par une cascade de diviseurs internes pour en fournir huit autres s'échelonnant de 271 à 1 Hz.

Le circuit CI_2 intègre huit amplificateurs inverseurs destinés à soulager les sorties de CI_1 .

Les LED testées à ces différentes fréquences sont soumises à une tension de 1,6V et limitées à une intensité de 10mA environ par les résistances R_3 à R_{10} .

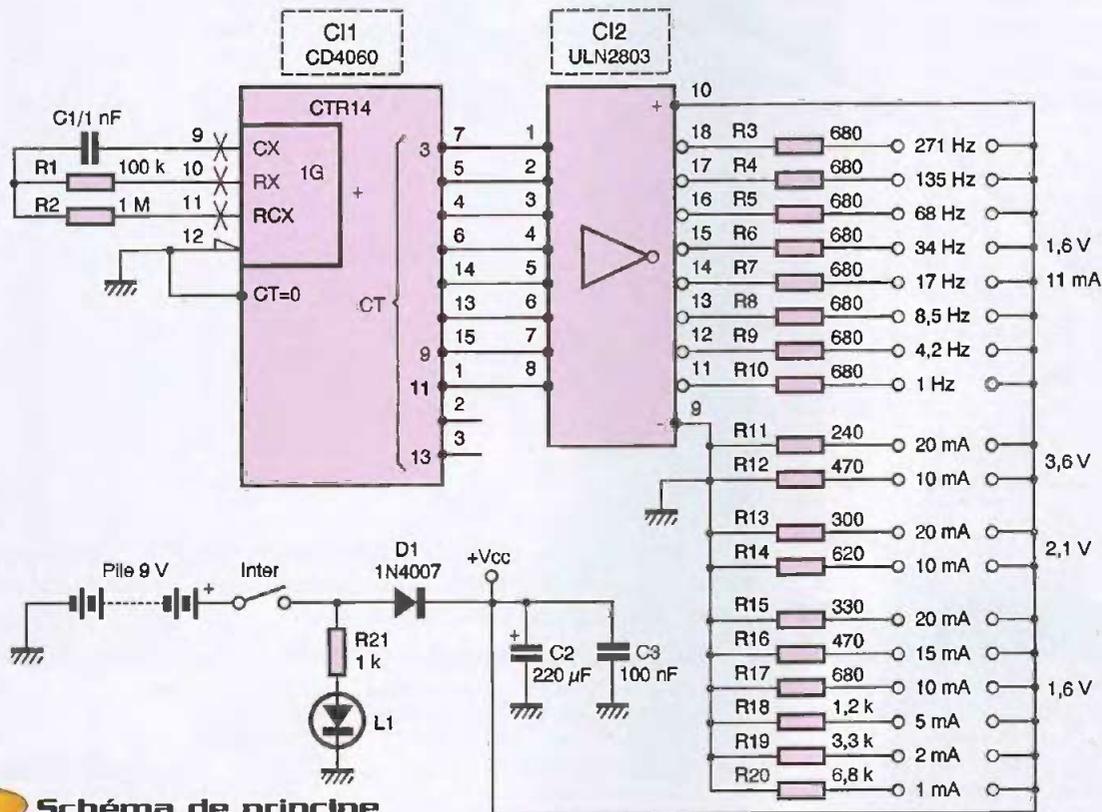
Les prises de tests statiques sont simplement limitées en courant par les résistances R_{11} à R_{20} afin de fournir l'intensité voulue pour les tensions nominales des différentes LED (1,6, 2,1 et 3,6V).

Réalisation

Le dessin du circuit imprimé simple face est donné à la **figure 2**. Sa réalisation par la méthode photographique est vivement recommandée compte tenu de la finesse de certaines pistes.

Après révélation et rinçage, la plaque est ensuite gravée dans un bain de perchlore de fer puis, de nouveau, abondamment rincée. Vous devez maintenant percer les pastilles à l'aide d'un foret de 0,8 mm de diamètre. Les trous de la diode et du support de pile doivent être alésés à des diamètres supérieurs.

La **figure 3** donne le plan d'implantation. La première opération consiste à souder les 2 points de câblage. Poursuivez le travail en soudant dans l'ordre les résistances, la diode, les supports de circuits intégrés, les condensateurs au mylar, les connecteurs constitués de barrettes sécables de type tulipe, la LED, le condensateur chimique, l'interrupteur et, pour terminer, le support de pile de 9V. Veillez à ne pas inverser les quelques composants polarisés (circuits intégrés, diode, LED, condensateur chimique).



1 Schéma de principe

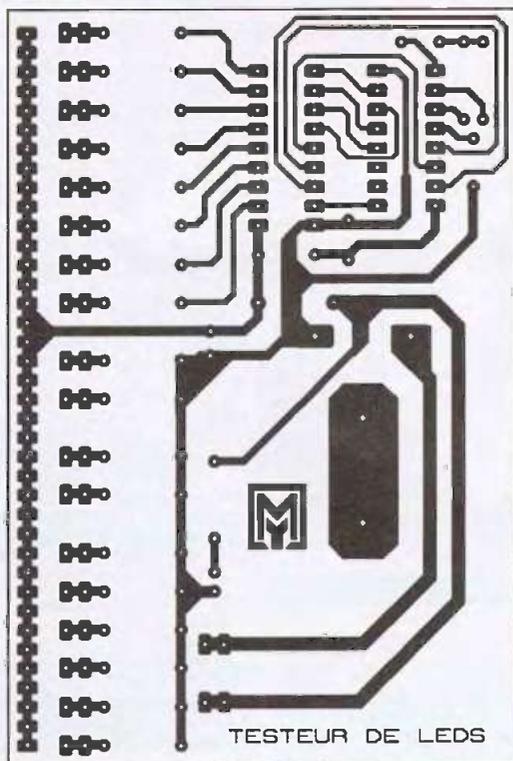
Utilisation

Avant de mettre votre testeur sous tension, vérifiez une dernière fois l'isolation entre les pistes, l'emplacement et la valeur des com-

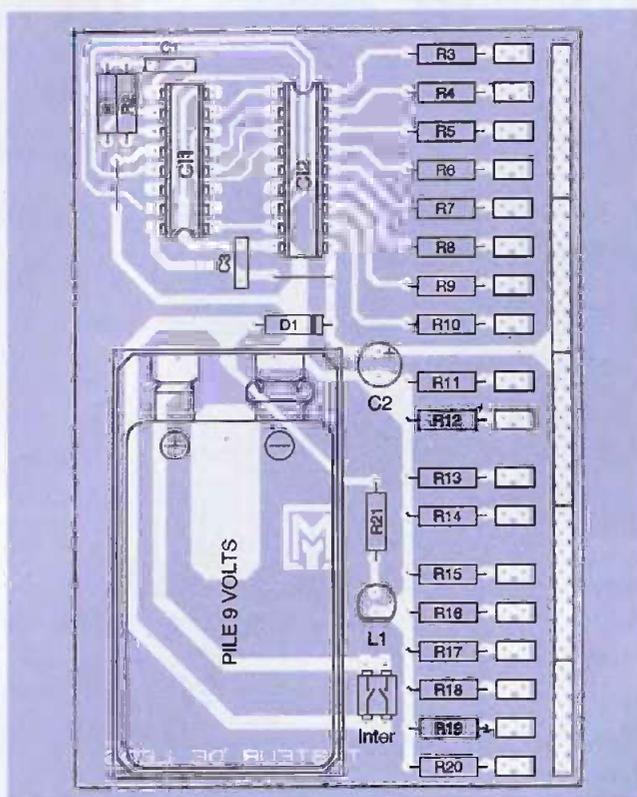
posants. Il suffit, bien sûr, de mettre en contact les deux pattes de la LED à tester sur la prise de test choisie, le positif étant commun sur toute la longueur du connecteur. L'emploi de cet appareil, bien que très

simple, nécessite quelques précautions. Si vous ne connaissez pas la tension de service d'une LED, commencez par la tester sous 1,6V, puis remontez progressivement les intensités et la tension. Ne poussez pas

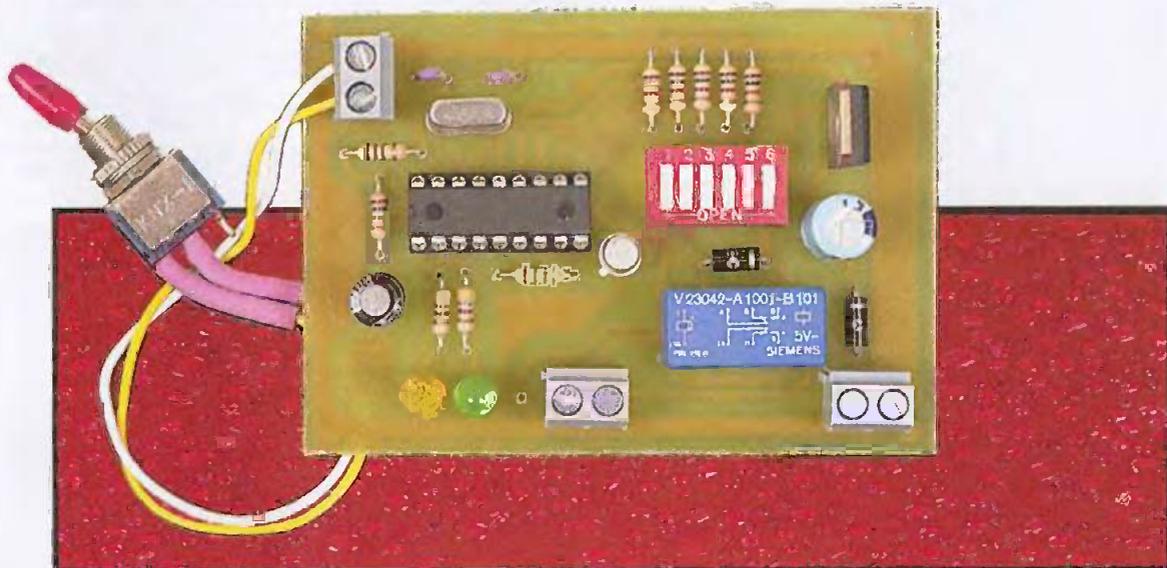
2 Tracé du circuit Imprimé



3 Implantation des éléments



Relais programmable



Le relais programmable décrit dans cet article pourra vous être très utile. En effet, il est capable de réaliser les fonctions de minuterie, cadenceur, télérupteur et retardateur. Nous vous proposons donc de réaliser un montage capable de piloter un relais selon plusieurs modes programmables par l'utilisateur à l'aide de switches. Ce montage est réalisé, encore une fois, à partir du célèbre PIC 16F84.

Vous pourrez commander un relais selon quatre modes de fonctionnement qui sont :

- mode minuterie,
- mode télérupteur,
- mode retardateur,
- mode cadenceur.

Chaque mode fonctionne avec des durées de temporisation également programmables, le tout à partir d'un schéma extrêmement simplifié.

Il est à noter que ce montage peut être portable, en effet, il peut être alimenté à partir d'une pile 9V.

Schéma de principe (figure 1)

L'alimentation de la platine est issue, dans notre exemple, d'une pile 9V, un régulateur 7805 fournit le 5V nécessaire au microcontrôleur.

Le montage peut être alimenté avec un bloc secteur positionné sur 9 ou 12V. La diode D_2 vient protéger le montage en cas d'inversion de polarité de l'alimentation.

La résistance R_7 de 1 k Ω et le condensateur C_2 de 10 μ F permettent de fournir à la broche MCLR (4) un signal de reset à la mise sous tension.

Toute la partie traitement, comme vous le voyez, est confiée au microcontrôleur IC₁ (PIC 16F84).

Cinq switches vont permettre d'adapter le montage à vos besoins, chaque switch arrive sur une entrée du PIC (RB3, RB4, RB5, RB6, RB7), une résistance de rappel de 1 k, reliée au 0V, polarise correctement chaque entrée.

L'horloge du PIC est confiée à un quartz de 4 MHz.

Le principe de fonctionnement est le suivant :

Le switch 1, positionné sur «ON», permet de programmer la platine selon un des quatre modes et selon une des quatre temporisations disponibles (**tableaux 1 et 2**).

Cette programmation est enregistrée dans la mémoire EEPROM du PIC 16F84 et restera mémorisée, même hors tension, tant qu'une nouvelle programmation n'est pas effectuée.

Une fois la platine programmée, une action sur l'interrupteur d'entrée provoquera la mise en service du mode de fonctionnement enregistré.

La réalisation

La **figure 2** donne le dessin du circuit imprimé, celui-ci sera réalisé sans difficultés.

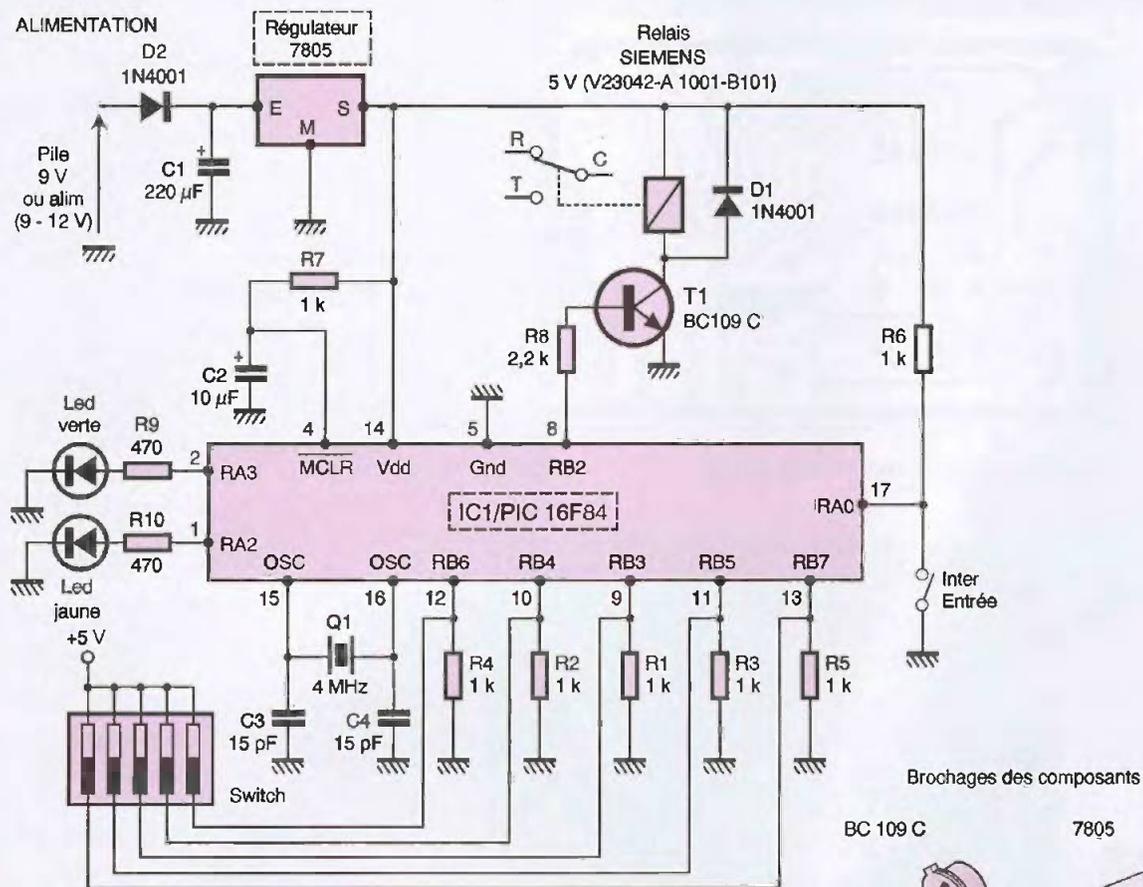
Le perçage des trous se fera en 0,8 mm et 1 mm pour le passage des pattes de composants plus larges.

La **figure 3** présente l'implantation des composants. Souder, dans un premier temps, les résistances, les diodes, le support de circuit, les borniers, les condensateurs et terminer par le transistor, le régulateur, les LED, les switches et le relais.

Mise en service

Après avoir vérifié qu'aucun court-circuit éventuel n'est présent, que les valeurs de composants ont été respectées, on peut mettre sous tension le montage à l'aide d'une pile 9V ou bien d'un bloc secteur sur une position 9 ou 12V.

Positionner tous les switches à 0. Positionner les switches 2, 3, 4 et 5 selon les tableaux 1 et 2 puis positionner le switch 1 sur la position «ON», la LED jaune (LED₁) s'allume pour signaler le mode programmation. Au bout de quelques secondes,



1 Schéma de principe

la LED jaune s'éteint indiquant la programmation enregistrée en EEPROM. Repositionner le switch 1 sur «OFF».

Pour essayer le montage selon un des quatre modes de fonctionnement, actionner l'interrupteur connecté à l'entrée de la platine et vérifier le fonctionnement du montage selon le mode programmé avec les temporisations décrites ci-après.

Option du montage

Le switch 1 en position «ON» passe la platine en mode programmation,
Les switches 2 et 3 en position «ON» permettent de sélectionner un des quatre modes de fonctionnement :

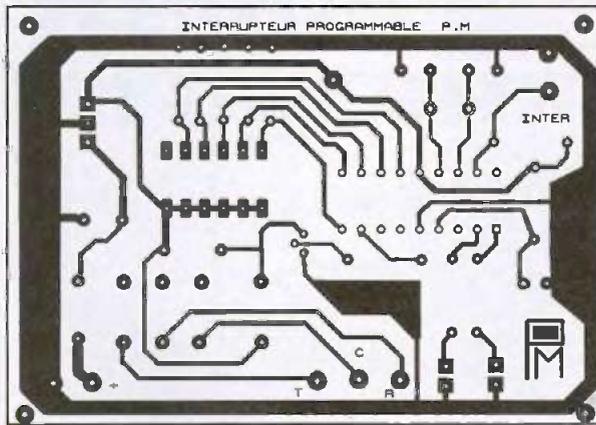
1) mode minuterie : le relais «colle» dès l'appui sur l'interrupteur pendant la temporisation déterminée avec les switches 4 et 5. Le relais chute une fois la temporisation échu. Un nouveau cycle redémarre si l'interrupteur est repositionné à l'état 0 puis à l'état 1.

2) mode retardateur : le relais «colle» dès que la temporisation programmée est échu suite à l'appui sur l'interrupteur. Une fois le relais «collé», il reste monté tant que l'interrupteur n'est pas repositionné à 0.
3) mode télérupteur : le relais «colle» dès

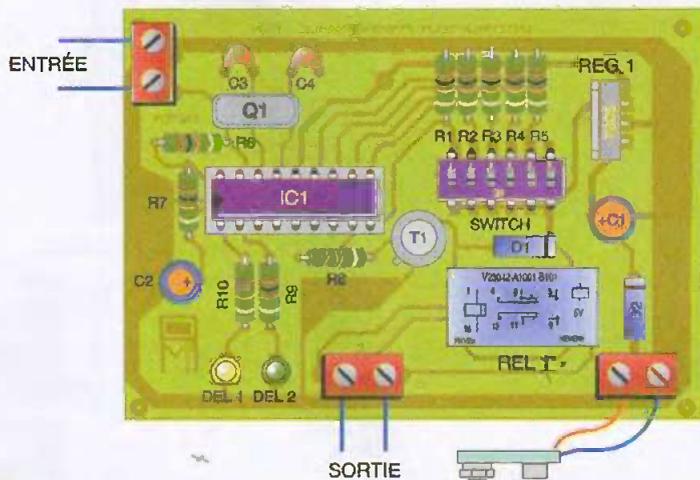
l'appui sur l'interrupteur ou le bouton poussoir, chaque nouvel appui fait basculer à nouveau le relais dans l'état opposé.
4) mode cadenceur : le relais «colle» dès l'appui sur l'interrupteur pendant la temporisation déterminée avec les switches 4 et 5.



l'interrupteur mini-dil à 6 positions



2 Tracé du circuit imprimé



3 Implantation des éléments

Messagerie (pm.ep@libertysurf.fr)
site : <http://perso.libertysurf.fr/p.may>



Le relais, une fois la temporisation échuë, chute pendant la même durée que pour le collage, puis il se «recolle». Le cycle reste ainsi tant que l'interrupteur n'est pas repositionné à l'état 0.

Les switches 4 et 5, ces deux switches déterminent la durée de la temporisation dans chaque mode décrit selon le tableau 2.

Le programme

Le programme au format HEX (compilé) est disponible comme d'habitude sur notre site www.eprat.com. Il vous suffit donc de le télécharger et de relancer ce programme vers la mémoire du PIC à l'aide d'un programmeur. De nombreux programmeurs de PIC sont disponibles en kit auprès des annonceurs de notre revue.

Conclusion

Cette platine peut commander un relais selon plusieurs modes, le fait d'être alimentée avec une pile 9V et un relais en sortie, vous permettra de l'utiliser rapidement et avec simplicité. Ce montage est reprogrammable aisément grâce à l'emploi du PIC 16F84.

P. MAYEUX

Nomenclature

- IC₁ : PIC 16F84
- T₁ : transistor NPN BC109C ou équivalent
- D₁, D₂ : 1N4001
- REG₁ : régulateur 7805 [5V]
- DEL₁, DEL₂ : diodes électroluminescentes 5 mm (jaune, verte)
- Q₁ : quartz 4 MHz
- R₁ à R₇ : 1 kΩ 5% (marron, noir, rouge)
- R₈ : 2,2 kΩ 5% (rouge, rouge, rouge)
- R₉, R₁₀ : 470 Ω 5% (jaune, violet, marron)
- C₁ : 220 pF/25V électrochimique sortie radiale
- C₂ : 10 pF/63V électrochimique sortie radiale
- C₃, C₄ : 15 pF céramique
- 3 borniers à vis pour circuit imprimé
- 1 relais 2RT/5V type SIEMENS (V23042-A1001-B101)
- 1 mini inter DIL 6 interrupteurs
- 1 connecteur pour pile 9V
- 1 support tulipe 18 broches

Switch 2	Switch 3	Mode de fonctionnement
0	0	Minuterie
1	0	Retardateur
0	1	Télérupteur
1	1	Cadenceur

T1 Tableau 1

Switch 4	Switch 5	Temporisation
0	0	5 secondes
1	0	30 secondes
0	1	1 minute
1	1	30 minutes

T2 Tableau 2

Thermomètre intérieur/ extérieur

avec mini et maxi



Le titre de cet article parle de lui-même puisque nous allons vous proposer de réaliser un double thermomètre, c'est à dire encore un thermomètre capable de mesurer simultanément la température de deux lieux différents tels que l'intérieur et l'extérieur de votre habitation par exemple. En outre, chaque mesure est assortie d'une fonction mini/maxi analogue à celle que l'on rencontre sur les bons vieux thermomètres à alcool.

Afin que la manipulation de notre thermomètre soit la plus simple possible, l'affichage est conversationnel et en français, bien sûr, et seuls deux poussoirs sont utilisés : un pour la lecture des minima et maxima, l'autre pour leur remise à zéro.

Malgré ces nombreuses fonctions, le schéma utilisé est fort simple, comme vous avez peut être déjà pu le constater si vous avez feuilleté votre EP avant de le lire, grâce à l'utilisation de circuits intégrés particulièrement bien choisis.

Les circuits DALLAS à interface "1 fil"

Même s'il existe aujourd'hui de nom-

breux capteurs de température, tous ne sont pas nécessairement faciles à interfacer avec un système d'affichage ou bien encore avec un microcontrôleur. En ce qui nous concerne, nous avons choisi les DS18B20 de DALLAS en raison de plusieurs particularités remarquables qui les caractérisent.

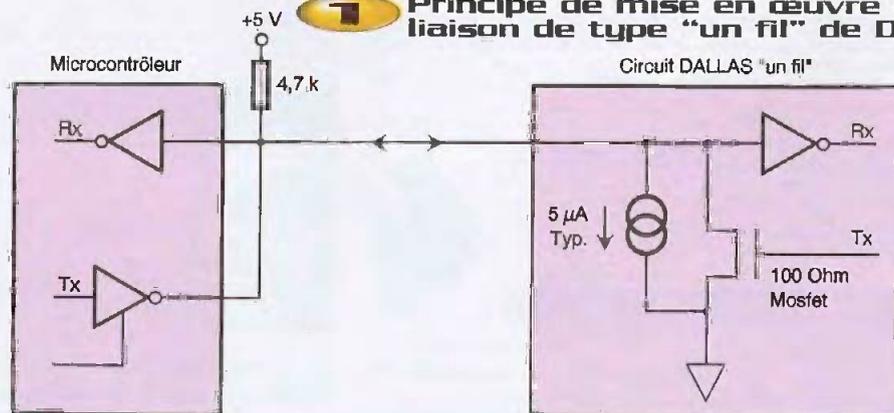
Ces circuits, qui se présentent sous forme d'un boîtier analogue au TO92 bien connu des transistors, disposent en effet d'une large plage de mesure de température puisqu'elle s'étend de -55°C à $+125^{\circ}\text{C}$ avec une précision d'un demi-degré, ce qui est déjà une belle performance.

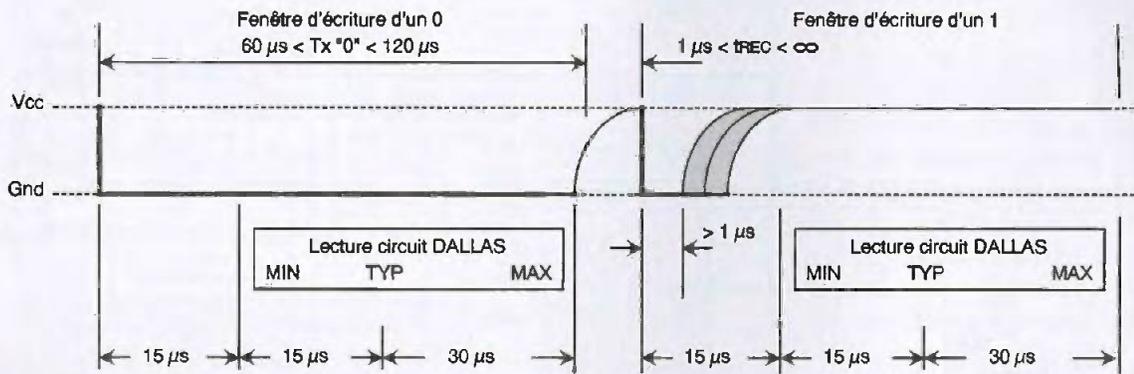
Ils fournissent directement le résultat de leur mesure sous forme numé-

rique, codée sur 9 bits, il est vrai, ce qui réclame quelques petites manipulations mais, et c'est là une de leurs particularités notables, ils utilisent pour cela l'interface ou bus "un fil" ("one wire" en bon anglais) développée par DALLAS pour de nombreux circuits intégrés de sa fabrication. Nos capteurs DS18B20 n'utilisent donc en tout et pour tout que trois fils de liaison : deux fils d'alimentation et un fil de signal, bidirectionnel de surcroît.

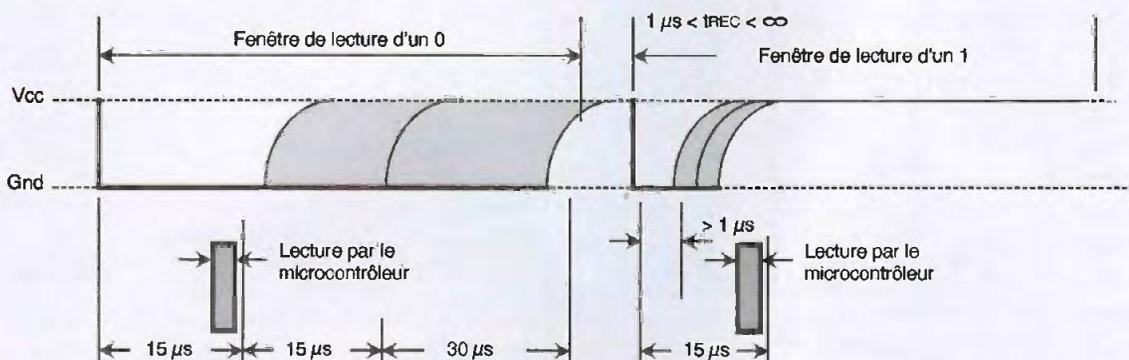
Cette interface "un fil" se retrouvant sur de nombreux circuits DALLAS et étant particulièrement intéressante pour de nombreuses applications à base de microcontrôleurs, voici en quelques mots comment elle fonctionne.

Principe de mise en œuvre d'une liaison de type "un fil" de Dallas





2 Chronogrammes d'émission de données par le maître d'un bus "un fil"



3 Chronogrammes de réception de données par le maître d'un bus "un fil"

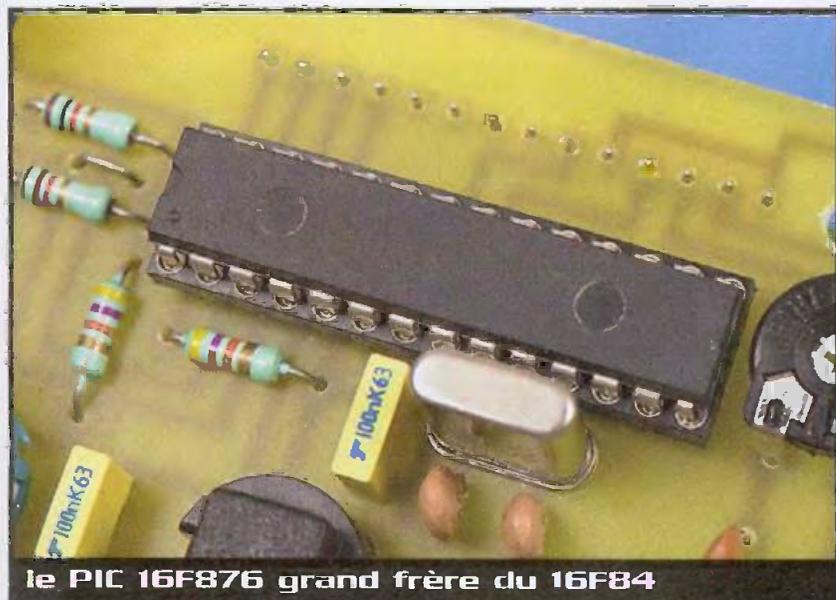
Tout d'abord, cette interface étant bidirectionnelle alors qu'elle n'utilise qu'un fil, il est évident qu'elle doit respecter, au niveau de tous les circuits qui y sont connectés, un schéma un peu particulier présenté **figure 1**.

La partie interne des circuits DALLAS ne nous concerne évidemment pas puisque nous ne pouvons agir dessus, par contre

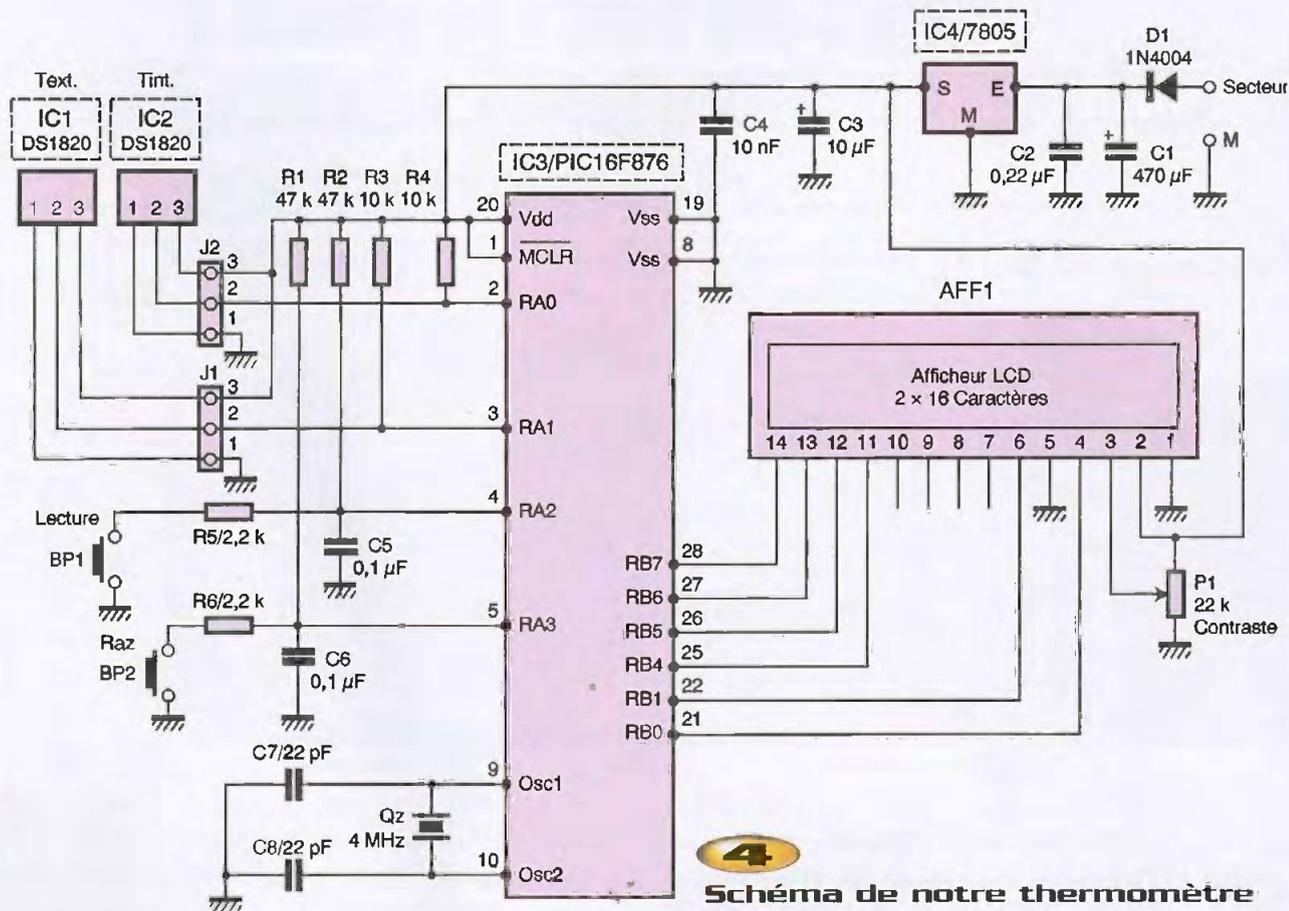
elle conditionne une partie du schéma d'utilisation du bus "un fil". En effet, comme vous pouvez le constater, la sortie du circuit DALLAS à destination de ce bus a lieu au moyen d'un transistor MOS à drain ouvert. Il faut donc impérativement ramener ce bus au positif de l'alimentation par une résistance de tirage de 4,7 à 10 kΩ de valeur typique.

Côté microcontrôleur, puisque c'est généralement un circuit de ce type qui est associé à ce genre de composant, il faut que l'émetteur de données à destination du bus "un fil" puisse être validé seulement sur commande, ce qui ne présente généralement aucune difficulté. Précisons encore que, la majorité des circuits "un fil" étant adressable, il est possible d'en connecter plusieurs sur le même bus sous le contrôle d'un seul et même microcontrôleur. Ce dernier est alors le maître du bus et les circuits DALLAS sont les esclaves selon une terminologie désormais classique.

Ceci étant vu, la **figure 2** montre comment le microcontrôleur émet les deux niveaux logiques possibles à destination du circuit DALLAS. Pour générer un zéro logique, il force tout simplement le bus "un fil" au niveau logique bas pendant au moins 60 μs, avec une durée maximum qui ne doit pas excéder 120 μs, sachant que le circuit DALLAS lit alors le bus entre 15 et 60 μs après sa descente comme schématisé figure 2. Pour générer un niveau logique haut, par contre, il force le bus "un fil" au niveau bas (oui, bas, vous avez bien lu) mais pendant une durée comprise entre



le PIC 16F876 grand frère du 16F84



4 Schéma de notre thermomètre

1 et 15 μ s au maximum. Comme le circuit DALLAS lit toujours le bus au même moment, il voit bien alors un niveau logique haut puisque celui-ci est alors assuré par la résistance de tirage au positif de l'alimentation.

Pour lire les données émises par un circuit DALLAS, le procédé est similaire mais repose sur le fait que la sortie des circuits à interface "un fil" est à drain ouvert. En effet, comme le montre la **figure 3**, le microcontrôleur force alors le bus un fil au niveau bas pendant au moins 1 μ s. Si le circuit DALLAS veut générer un 0 logique, il maintient alors le bus au niveau bas pendant au moins 15 μ s alors que, s'il veut émettre un 1 logique, il laisse le bus libre et ce dernier remonte donc, sous l'effet de la résistance de tirage au niveau haut, dès que le microcontrôleur relâche le niveau bas qu'il imposait. Comme le microcontrôleur doit lire le bus à la fin de la fenêtre de 15 μ s comme schématisé figure 3, il lit bien alors le niveau logique voulu par le circuit DALLAS.

Comme vous pouvez le constater, ce protocole est relativement simple à mettre en œuvre. Pour pouvoir utiliser avec succès

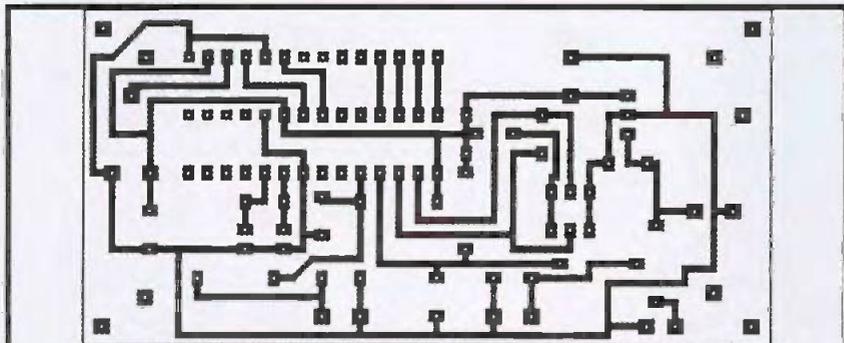
les circuits DALLAS, il doit évidemment être complété par d'autres informations constituées par les ordres reconnus par tel ou tel circuit ainsi que par le codage des données qu'il fournit en réponse. Ces informations dépendent bien évidemment du circuit utilisé et ne peuvent donc être détaillées ici dans le cadre de cette présentation générale.

Schéma de notre thermomètre

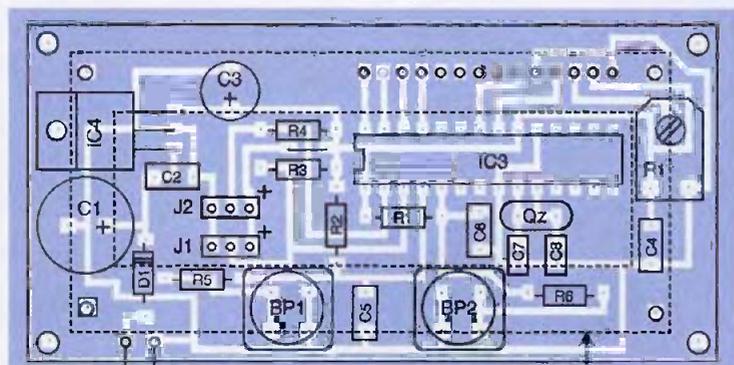
Comme le montre la **figure 4**, le cœur du montage est constitué par IC₃ qui est un microcontrôleur PIC mais, contrairement à ce qui commence à devenir une habitude, ce n'est pas un 16F84 mais son "grand frère" le 16F876. Ce circuit présente les mêmes caractéristiques générales que le



le très classique module afficheur 2x16 caractères



5 Circuit imprimé, vu côté cuivre, échelle 1



Afficheur monté côté cuivre (vu par transparence)
6 Implantation des composants
 +V M Gnd Vcc
 Lect. RAZ
 Circui DS 1820 vu côté fils

16F84 mais dispose de ressources internes plus nombreuses et surtout, et c'est ce qui a motivé son choix ici, d'une mémoire de programme de taille beaucoup plus importante.

Il s'alimente, ainsi que tout le reste du montage d'ailleurs, sous une tension unique de 5V régulée par IC₄ à partir de la tension de

9V délivrée par un bloc secteur externe style "prise de courant".

Deux lignes de ses ports parallèles, RA0 et RA1, sont affectées à la connexion avec les deux capteurs de température que sont IC₁ et IC₂. Ils sont baptisés Text et Tint car, en usage courant, ce thermomètre a été prévu pour une utilisation domestique, mesurant

ainsi la température intérieure et extérieure d'une habitation.

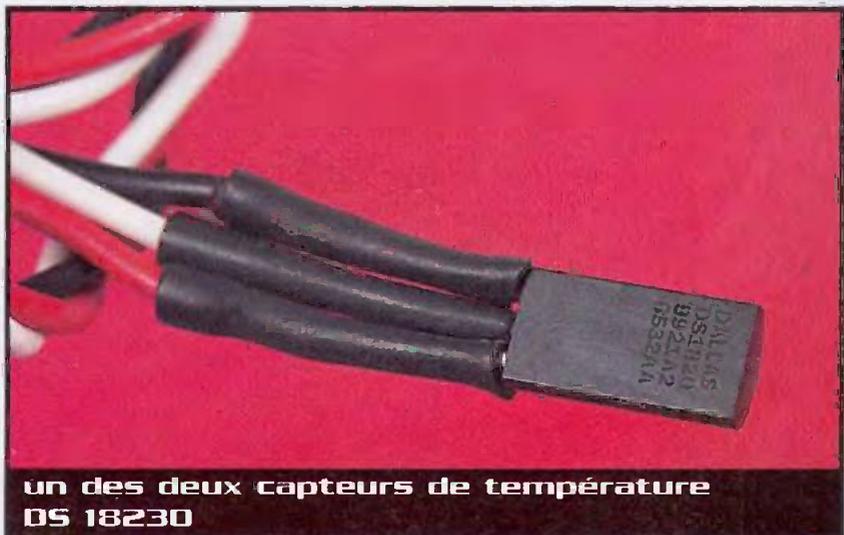
Les deux autres lignes de port parallèle RA2 et RA3 servent, quant à elles, à connecter deux poussoirs destinés, l'un à la remise à zéro des mémoires des minima et maxima, l'autre à provoquer leur affichage.

Cet affichage justement est confié à un afficheur à cristaux liquides standard à deux lignes de seize caractères à interface parallèle. Cette interface est utilisée ici en mode quatre bits et fait appel aux lignes de port parallèle RB7 à RB4 ainsi qu'à RB1 et RB0. Le classique quartz d'horloge associé à ses deux condensateurs de faible valeur complète ce schéma, comme il est de coutume pour un microcontrôleur. Notez également la liaison directe de l'entrée de reset (MCLR) du 16F876 au positif de l'alimentation puisque ce circuit dispose d'une circuiterie de reset automatique à la mise sous tension, comme le 16F84, mais également d'un "brown out reset" ou reset automatique en cas de baisse anormale de la tension d'alimentation. Pour plus de précisions à ce sujet, nous ne saurions trop vous recommander la lecture de notre ouvrage "Les microcontrôleurs PIC - Description et mise en œuvre" dans sa dernière édition (celle accompagnée des deux CDROM de MICROCHIP) publié chez DUNOD.

La réalisation

La nomenclature des composants ne devrait pas vous poser trop de problème. Si vous avez du mal à trouver les DS1820, sachez qu'il y en a chez FARNELL, ainsi d'ailleurs que des PIC 16F876. Pour ce qui est de l'afficheur, n'importe quel modèle standard à deux lignes de seize caractères convient. Revoyez à ce propos, si nécessaire, notre article du n°257 de mai 2001. Le dessin du circuit imprimé vous est présenté **figure 5**. Il supporte tous les composants du montage, l'afficheur étant câblé côté cuivre du circuit comme vous pouvez le voir sur notre maquette.

Afin que cet afficheur soit démontable, son circuit imprimé est équipé de picots à souder mâles/mâles. Le circuit imprimé du montage, quant à lui, est équipé de contacts tulipes femelles en bande, soudés côté cuivre à l'emplacement qui fait face au connecteur de l'afficheur. Ces contacts tulipes femelles sont à souder en



un des deux capteurs de température DS 18230

premier car c'est plus facile tant que la face composants du circuit imprimé est vierge et peut donc être posée parfaitement à plat sur une table.

Lorsque c'est fait, le câblage du côté composants est à faire en suivant les indications de la **figure 6** dans l'ordre habituel : straps, support de circuit intégré, résistances, condensateurs et semi-conducteurs. Veillez à la bonne orientation des condensateurs chimiques et de la diode. Le régulateur IC₄ est monté à plat sur le circuit imprimé mais, vu la faible consommation du thermomètre, tout radiateur est inutile. Les poussoirs peuvent être montés directement sur le circuit, comme c'est le cas sur notre maquette, mais c'est assez peu pratique si vous ne trouvez pas un boîtier spécialement adapté. Vous pouvez aussi les déporter en face avant du boîtier au moyen de fils souples isolés soudés sur les grosses pastilles prévues à cet effet. Les capteurs sont reliés au circuit par tout moyen à votre convenance. Nous avons utilisé sur notre maquette des picots mâles/mâles au pas de 2,54 mm, soudés sur le circuit imprimé, tandis que les fils souples provenant des capteurs étaient raccordés sur des contacts femelles en bande au pas de 2,54 mm mais toute autre connectique, voire même la soudure directe des fils de liaison, convient. Dans tous les cas, veillez à bien respecter le sens de branchement des connecteurs afin de ne pas inverser la polarité de l'alimentation des capteurs.

Ces derniers sont soudés à l'extrémité de leurs fils souples de liaison et les soudures sont isolées avec de la gaine thermorétractable. Il est également possible de rigidifier ces points de liaison en y déposant une goutte de colle époxy.

Le microcontrôleur 16F876 doit évidemment être programmé correctement pour que le montage puisse fonctionner. Le programme correspondant est à votre disposition sur le serveur d'E.P. et se nomme dbt-therm.hex. Son format est compatible de tous les logiciels de programmeurs de PIC courants (voir si nécessaire le numéro 238 d'E.P. ou le site Internet de l'auteur : www.tavernier-c.com).

A ce propos, et comme le 16F876 dispose de nombreux "fusibles" de configuration dont certains n'existent pas sur les 16F84, précisons qu'il faut programmer ceux-ci de

la façon suivante :

- Oscillateur en mode XT
- Timer chien de garde (WDT) non validé
- Timer à la mise sous tension (PWRT) validé
- Reset "brown out" (BODEN) validé
- Programmation sous faible tension (LVP) non validé
- Protection du code de l'EEPROM (CPD) non validé
- Autorisation d'écriture en mémoire flash (WRT) non validé
- Mise au point en circuit (DEBUG) non validé
- Protection du code de la mémoire flash (CPO et CP1) non validé si vous voulez pouvoir relire votre circuit après programmation

Utilisation

Si aucune erreur n'a été commise, le montage fonctionne dès sa mise sous tension mais n'oubliez pas de commencer par ajuster le potentiomètre P₁ afin de rendre l'affichage visible. Ce réglage est en effet très (trop !) efficace avec les afficheurs actuels et peut effacer totalement tout affichage ou, au contraire, transformer celui-ci en une suite de pavés noirs.

Après un message d'accueil constitué du texte "Double thermomètre", notre montage affiche les températures intérieures et extérieures, selon la dénomination donnée aux capteurs en figure 4. Cette mesure est renouvelée toutes les deux secondes. Si un capteur est mal branché ou est absent, le montage le signale en indiquant "X absent" où X est égal à intérieur ou extérieur selon le cas. Par contre, la mesure continue normalement sur l'autre capteur.

Si vous pressez le poussoir "Lecture", les minima et maxima de chaque capteur sont affichés à tour de rôle et restent visible deux secondes chacun, puis le montage retourne seul à l'affichage de la température courante. Si vous maintenez le doigt appuyé ou si vous remplacez le poussoir par un interrupteur, l'affichage "tourne en rond" et affiche tour à tour les minima et maxima puis la température courante et ainsi de suite. C'est également un mode d'utilisation possible car toutes les données affichées sont actualisées.

Si vous pressez le poussoir "RAZ", les minima et maxima des deux capteurs sont

effacés ce qui est confirmé par le message "RAZ réalisée". Ils recommencent à être actualisés dès que le poussoir est relâché. Notez, à propos de ces poussoirs, qu'il faut parfois maintenir la pression jusqu'à deux secondes afin que l'opération demandée soit prise en compte. Cela résulte du mode de travail du programme qui ne peut pas les scruter lorsqu'il est en train de lire la température transmise par les capteurs. Il ne vous reste plus qu'à disposer vos capteurs en des points judicieusement choisis pour disposer d'une mesure permanente de la température, assortie d'un relevé automatique des minima et maxima, le tout avec une précision d'un demi-degré.

C. TAVERNIER

Nomenclature

IC₁, IC₂ : DS1820 de DALLAS chez FARNELL

IC₃ : PIC 16F876

IC₄ : 7805 (régulateur +5V/1 A, boîtier T0220)

D₁ : 1N4004

AFF₁ : afficheur alphanumérique LCD standard 2 lignes de 16 caractères

R₁, R₂ : 47 kΩ 1/4W 5%

(jaune, violet, orange)

R₃, R₄ : 10 kΩ 1/4W 5%

(marron, noir, orange)

R₅, R₆ : 2,2 kΩ 1/4W 5%

(rouge, rouge, rouge)

C₁ : 470 µF/25V chimique radial

C₂ : 0,22 µF mylar

C₃ : 10 µF/25V chimique radial

C₄ : 10 nF céramique

C₅, C₆ : 0,1 µF mylar

C₇, C₈ : 22 pF céramique

P₁ : potentiomètre ajustable horizontal de 22 kΩ

BP₁, BP₂ : poussoirs 1 contact travail (contact en appuyant) par exemple ITT type D6 carré

1 support de CI 28 pattes étroit

Picots à souder mâles/mâles au pas de 2,54mm : 1 x 14 contacts

Contacts tulipes en bande au pas de 2,54mm : 1 x 14 contacts

Connectique pour les capteurs (voir texte)

Driver de câble coaxial



Cet article se propose de décrire un driver de câble coaxial en utilisant un amplificateur opérationnel à contre-réaction en courant. En effet, ce type de composant est très bien adapté pour ce genre d'application. L'amplificateur utilisé dans notre montage est le LM6181 de NATIONAL Semiconductor.

Description du montage

Le schéma de notre montage est représenté à la **figure 1**. L'amplificateur à contre-réaction en courant LM6181 est une combinaison incomparable de bande-passante, de fréquence de balayage et de courant de sortie. Cet amplificateur peut directement commander jusqu'à des charges capacitives de 100 pF sans oscillation et un signal de 10V dans un système à câble coaxial d'impédance de charge de valeur 50 ou 75 Ω sur la plage totale de température industrielle. Ceci représente une amélioration radicale dans sa capacité de commande de sortie pour un circuit intégré avec seulement 8 broches contenant un amplificateur à haute vitesse qui le rend idéal pour des applications vidéo.

Construit à partir d'un procédé amélioré de NATIONAL Semiconductor (VIP II à haute vitesse 'Vertically Integrated PNP' ou encore intégration verticale de transistor PNP), le LM6181 utilise une contre-réaction en courant qui fournit une bande-passante qui ne varie pas dramatiquement avec le gain : 100 MHz pour un gain de -1 et 60 MHz pour un gain de -10. Avec une vitesse de balayage de 2000V/ μ s, une distorsion de la seconde harmonique de -50 dB à

10 MHz et un temps d'établissement de 50 ns à 1%, les performances dynamiques du LM6181 le rendent idéal pour l'acquisition des données et les applications pour les amplificateurs d'impulsions à haute précision. La **figure 2** représente le schéma interne du LM6181. Pour un amplificateur à contre-réaction en tension traditionnelle, la bande-passante résultante pour petits signaux est inversement proportionnelle au gain désiré pour une approximation du premier ordre basé sur le concept gain/bande-passante. En contraste, la topologie de l'amplificateur à contre-réaction en courant, tel que le LM6181, transcende sa limitation pour offrir une bande-passante du signal qui est relativement indépendante du gain en boucle ouverte. La bande-passante en boucle fermée du LM6181 dépend de sa résistance de contre-réaction. Par conséquent, cette résistance doit être ajustée pour obtenir le gain en boucle fermée désiré.

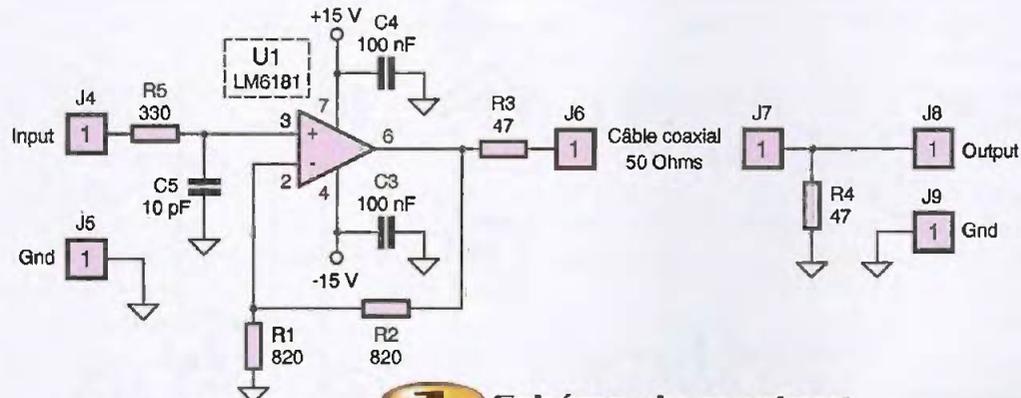
Une condition fondamentale pour la conception des amplificateurs à haute vitesse est un découplage adéquat des tensions d'alimentations. Il est critique de maintenir une faible impédance par rapport à la masse sur une large bande sur les broches des tensions d'alimentation de l'amplifica-

teur afin d'assurer la fidélité des signaux transitoires sur l'amplificateur à haute vitesse.

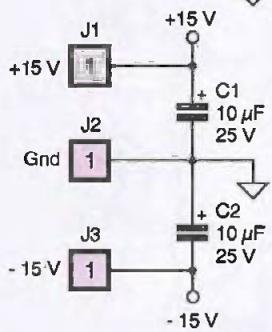
Des capacités au tantale de 10 μ F et en céramique de 0,1 μ F sont recommandées pour chacune des broches de la tension d'alimentation. Ces capacités de découplage doivent être placées aussi près que possible des broches de l'amplificateur.

La sélection de la résistance de contre-réaction est un facteur dominant pour la compensation du LM6181. Pour des applications générales, ce composant maintient les performances spécifiées avec une résistance de contre-réaction de valeur 820 Ω . Bien que cette valeur fournisse de bons résultats pour beaucoup d'applications, il peut être avantageux d'ajuster légèrement sa valeur.

Considérons, par exemple, l'effet d'une réponse en impulsion avec deux configurations différentes dans lesquelles les gains en boucle fermée sont dans les deux cas égaux à 2 et les résistances de contre-réaction ont pour valeur respectivement 820 et 1640 Ω . Le fait d'augmenter la résistance de contre-réaction fait diminuer la bande-passante de l'amplificateur à contre-réaction en courant. En conséquence, des résistances de contre-réaction avec des valeurs plus faibles que 820 Ω peuvent être utili-



1 Schéma du montage



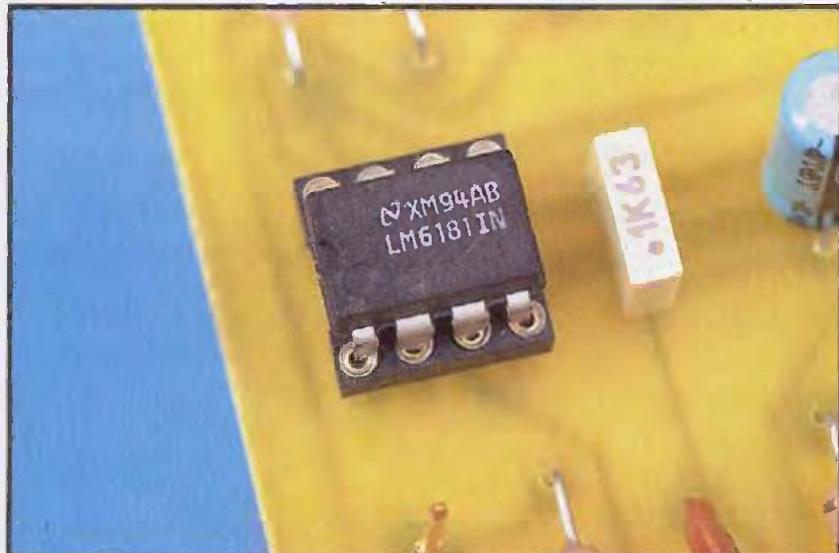
sées pour compenser la réduction de la bande-passante à des gains en boucle fermée élevés, à cause des effets du second ordre.

Les caractéristiques de la fréquence de balayage des amplificateurs à contre-réaction en courant sont différentes de celles des amplificateurs traditionnels à contre-réaction en tension. Dans ces derniers, la limitation de la vitesse de balayage ou le comportement non linéaire de l'amplificateur est dominé par le courant du premier étage

qui charge la capacité de compensation. La vitesse de balayage des amplificateurs à contre-réaction en courant, par contre, est non constante. Le courant de transition sur l'entrée inverseuse détermine cette vitesse de balayage à la fois pour les gains inverseur et non-inverseur. Pour la configuration non inverseuse, la vitesse de balayage est aussi déterminée par les limitations de l'étage d'entrée. En conséquence, des variations de la vitesse de balayage se produisent pour différentes topologies de circuit. Le LM6181 peut commander de manière significative de plus forte valeur de capacité comme charge que la plupart des autres amplificateurs en courant. Bien que le LM6181 puisse commander directement des charges aussi élevées que 100 pF sans oscillation, la réponse qui en résulte est une fonction de la valeur de la résistance de contre-réaction. Une sur-modulation

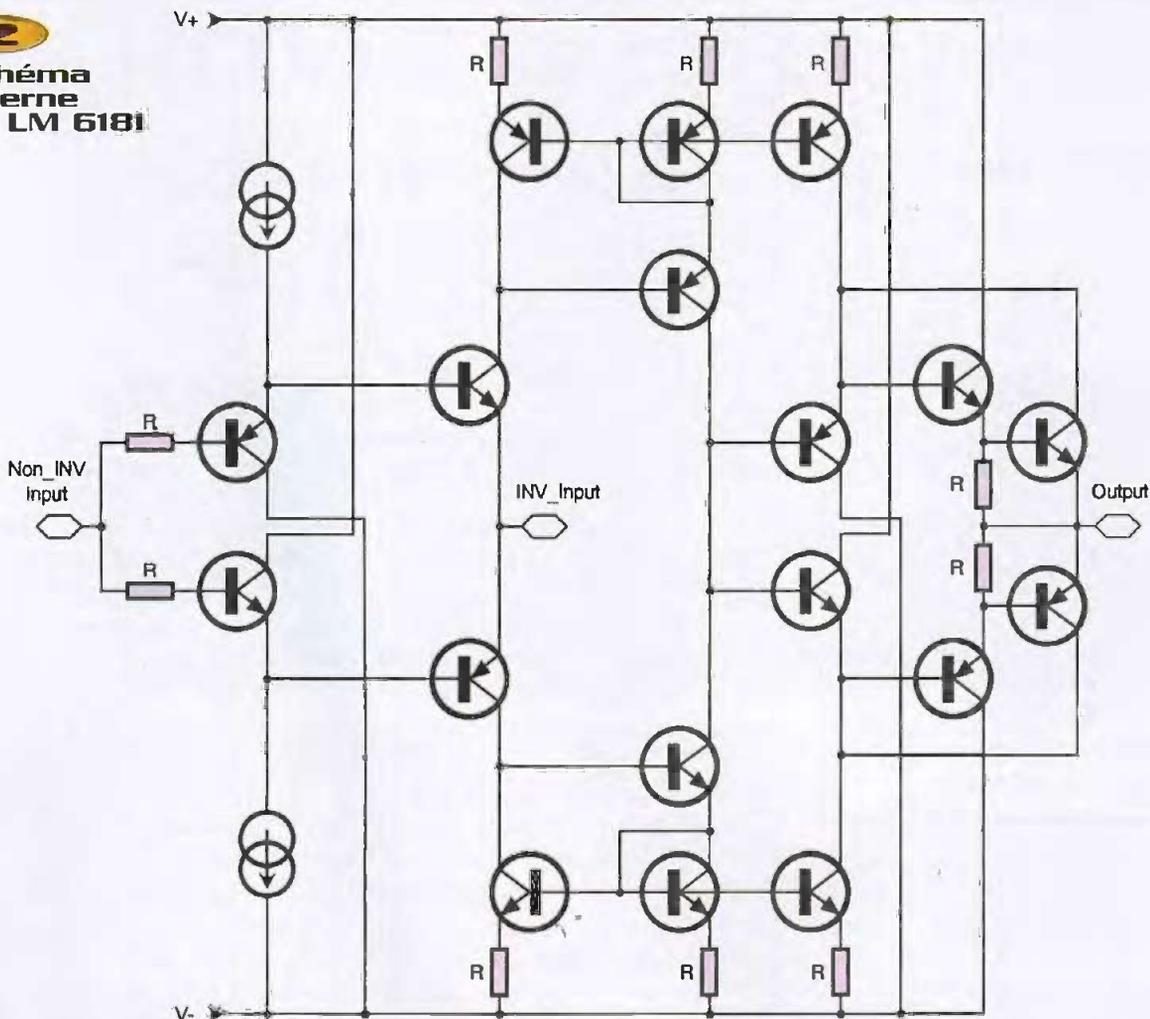
persiste pendant environ 70 ns. Afin d'obtenir des réponses en impulsion avec moins de sur-modulation, soit la valeur de la résistance de contre-réaction peut être augmentée, soit une résistance d'isolement peut être utilisée (une valeur entre 10 et 51 Ω fonctionne assez bien). L'une de ces deux techniques, cependant, a pour conséquence de réduire la bande-passante. Pour des amplificateurs à contre-réaction en tension, il est tout à fait commun de placer une petite capacité de compensation en parallèle avec la résistance de contre-réaction. Cette compensation sert à réduire les crêtes de l'amplificateur dans le domaine de fréquence qui apprivoise de façon équivalente la réponse transitoire. Afin de limiter la bande-passante des amplificateurs à contre-réaction en courant, il ne faut pas utiliser une capacité aux bornes de la résistance de contre-réaction. L'impédance dynamique des capacités dans la boucle de contre-réaction réduit la stabilité de l'amplificateur. Par contre, la réduction des crêtes dans la réponse en fréquence et la limitation de la bande-passante peuvent être obtenues en ajoutant un circuit résistance/capacité. Lorsque la plage des tensions d'entrée ou de sortie des amplificateurs à haute vitesse est dépassée, l'amplificateur doit se récupérer d'une condition de surexcitation. La conception de systèmes analogiques à haute vitesse peut souvent être une tâche ardue.

Typiquement, après que la définition initiale du système et l'approche de la conception soient établies, la tâche de la sélection des composants commence. Malheureusement, une simple confiance sur les paramètres de la documentation fournit seule-



théorie et pratique du LM6181 de NATIONAL

2

Schéma
Interne
du LM 6181

ment une idée partielle des nuances du fonctionnement réel du composant. Ceci est vrai même si la documentation de l'amplificateur à haute vitesse est la plus complète possible. Seulement l'expérience requise en passant du temps sur une table de laboratoire d'électronique avec le composant permet d'utiliser de façon fiable les amplificateurs à haute vitesse.

Le développement d'une application à base de LM6181 doit se concentrer sur les points suivants : la conception d'un produit plus indulgent, comme par exemple pouvoir commander directement des câbles non appariés (avec une charge continue très élevée) ou encore des charges capacitives de fortes valeurs ; développer une carte de démonstration à haute vitesse qui est facilement reconfigurable pour un fonctionnement de l'amplificateur en mode inverseur ou non-inverseur ; incorporer un modèle SPICE très précis du LM6181 (ce modèle peut être utilisé en conjonction avec les résultats des mesures afin de converger plus rapidement sur un schéma à haute

vitesse plus fiable). Bien qu'il puisse être perçu que l'évaluation du fonctionnement d'un circuit à haute vitesse puisse être obtenue plus rapidement avec une simulation sur ordinateur, une évaluation avec des mesures ne peut pas être évitée. En intégrant ces deux outils complémentaires, le temps de cycle à partir de la sélection du composant jusqu'au schéma final peut ainsi être réduit.

La carte de démonstration haute vitesse peut être utilisée pour examiner soit le domaine temporel, soit le domaine fréquentiel. Cependant, nous allons surtout nous intéresser à la compensation de la réponse dans le domaine temporel du LM6181 pour la majorité des applications rencontrées.

Les exemples les plus fréquemment rencontrés incluent le driver de câble coaxial (comme dans notre application), avec distribution de charges capacitives et pour lesquelles l'utilisateur obtient généralement une fidélité de la spécification en ce qui concerne la réponse impulsionnelle.

Grâce aux exigences en ce qui concerne la charge et la bande-passante typique, cette application particulière impose des contraintes importantes sur un amplificateur (**figure 2**).

La sortie du LM6181 incorpore un étage de sortie avec un gain en courant important qui fournit une plus faible impédance de sortie vers des charges importantes, de l'ordre de 100 à 150 Ω . Ceci améliore la capacité de l'amplificateur pour commander des câbles coaxiaux ($\pm 10V$ dans 100 Ω par exemple) puisque le courant interne qui commande l'étage de sortie de l'amplificateur est utilisé de façon plus efficace. De plus, les bénéfices de la topologie de la contre-réaction en courant du LM6181 permettent un fonctionnement sur une large bande-passante de 100 MHz, même lorsque celui-ci est configuré en boucle fermée avec un gain de +2.

Certaines considérations doivent être évoquées en ce qui concerne le signal de test choisi pour l'évaluation de la réponse transitoire du système linéaire. Grâce aux pro-

propriétés des transformées de Laplace, si une entrée impulsionnelle égale à l'unité est utilisée et si la mesure de la réponse en sortie est intégrée, le résultat de l'application d'une transformée de Laplace inverse établit alors la réponse fréquentielle du système. Cette approche n'est pas utile typiquement puisque les générateurs d'impulsions ne génèrent pas de telles impulsions et que l'intégration devient trop complexe. De plus, cette technique ne sert pas à établir une sensation intuitive. Comme autre alternative, si un signal d'entrée qui varie lentement avec le temps est utilisé comme signal de test à l'entrée, les composants haute fréquence dans le système ne sont pas excités de façon significative. La réponse à l'échelon unité fournit souvent une évaluation significative des performances de l'amplificateur et représente un signal beaucoup plus pratique.

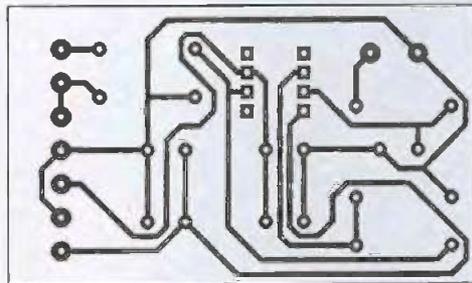
D'autres avantages à utiliser une réponse à l'échelon est que cette dernière fournit directement le gain en régime continu et que la nature haute fréquence de l'échelon excite les pôles haute fréquence de la fonction de transfert du système amplificateur. Lors de l'évaluation des performances de la réponse à un échelon pour la bande-passante de l'amplificateur, il est important d'utiliser un générateur d'impulsions qui fournit un temps de montée suffisamment rapide. L'évaluation de la réponse à un échelon pour l'amplificateur LM6181, dans lequel la bande-passante typique pour un gain de +2 est de 100 MHz, nécessite un signal à échelon en entrée avec un maximum de temps de montée de 3,5 ns. Puisqu'il y a toujours un certain taux de dégradation du temps de montée dû à la sonde de l'oscilloscope, il faut utiliser le même équipement de mesure pour évaluer à la fois l'intégrité du signal d'entrée et la mesure de la réponse en sortie du système. Probablement, la plus grande zone de difficulté dans la conception d'applications avec des amplificateurs à haute vitesse est lorsque ce dernier commande des charges capacitatives. Malheureusement, beaucoup d'amplificateurs sur le marché sont spécifiés pour conduire un maximum de charge capacitive de l'ordre de 20 pF avant que l'oscillation se produise. La limitation maximale équivalente implique que la réponse impulsionnelle des amplificateurs soit sensible à la capacité typique de l'oscilloscope

(la sonde devient alors une partie intégrante du circuit dans son ensemble), ce qui rend le jugement plein de sens mais beaucoup plus difficile à évaluer au niveau de la mesure.

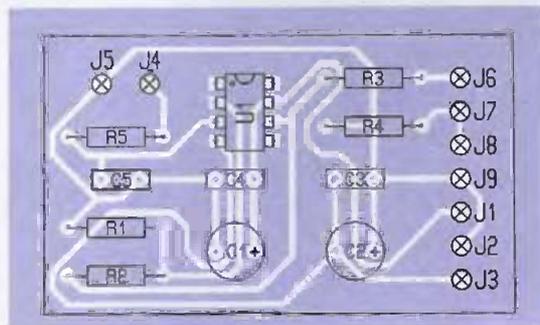
Bien que la charge capacitive directe doive être typiquement minimisée dans la pratique générale, le LM6181 se comporte encore très bien dans l'ensemble. Le modèle SPICE, qui existe pour le LM6181, possède de superbes caractéristiques en régime alternatif et en réponse transitoire. La dégradation de la marge de phase due à la charge capacitive directe des amplificateurs à haute vitesse peut potentiellement induire de l'oscillation,

L'impédance de sortie de l'amplificateur, couplée avec la charge capacitive, forme un réseau à retard dans la transmission de la boucle de l'amplificateur. Puisque ce réseau retarde la boucle de contre-réaction, la marge de phase est réduite de telle manière que même lorsqu'un système n'oscille pas de manière excessive, une sur-modulation peut se produire. Une solution directe pour réduire la sur-modulation afin de commander des charges capacitatives est de se relier indirectement à cette charge, c'est-à-dire isoler la charge avec une impédance réelle telle qu'une valeur modérée de résistance. Un exemple peut

être une résistance de 47 Ω utilisée pour isoler une impédance complexe capacitive de la sortie de l'amplificateur, préservant par conséquent la marge de phase de l'amplificateur. Un compromis évident existe entre l'approvisionnement de la réponse dans le domaine temporel et le maintien de la bande-passante de l'amplificateur, puisque cette forme de compensation ralentit directement la réponse de l'amplificateur, pour des applications générales du LM6181, la résistance de contre-réaction suggérée est de 820 Ω . Cependant, une caractéristique unique des amplificateurs à contre-réaction en courant est que ces derniers possèdent des bandes-passantes différentes suivant la résistance de contre-réaction. Le résultat est que les amplificateurs à contre-réaction en courant maintiennent une bande-passante en boucle fermée qui demeure la même pour des variations modérées du gain en boucle fermée (ceci bien sûr est une approximation, car les effets de second ordre interviennent aussi). Cette caractéristique rend actuellement les amplificateurs à contre-réaction en courant relativement aisés à compenser. En graduant simplement l'établissement du gain et l'impédance de contre-réaction, la bande-passante appropriée peut être obtenue à la valeur désirée du gain en boucle fermée. Une



3 Tracé du circuit Imprimé



4 Implantation des éléments

application pratique est l'utilisation d'une valeur modifiée de la résistance de contre-réaction pour la compensation du LM6181 lorsque ce dernier conduit une charge capacitive de 100 pF. En réduisant la bande-passante en boucle ouverte de l'amplificateur, la dégradation résultante de la marge de phase est réduite, ce qui améliore par conséquent la fidélité de la réponse impulsionnelle. Un facteur à surveiller souvent, pour comprendre le comportement dynamique des amplificateurs haute vitesse, est l'effet que la charge continue constitue sur la vitesse de l'amplificateur.

Lors de la commande de câbles coaxiaux, par exemple, la charge équivalente de Thévenin est habituellement soit de 100, soit de 150 Ω. De plus, puisque les caractéristiques dynamiques d'un amplificateur en général sont aussi affectées par les tensions d'alimentation de ce dernier, le LM6181 est totalement caractérisé à la fois pour fonctionner sous ±5 et ±15V. Souvent, pour la configuration en amplificateur inverseur, une simple modification de la résistance de contre-réaction qui établit le gain est la façon la plus facile pour compenser les crêtes et la sur-modulation dans la réponse à un échelon. La configuration non-inverseuse, cependant, peut être alternativement compensée en ajoutant une résistance en série en entrée. Cette résistance, en combinaison avec l'entrée et les capacités parasites d'entrée limitent la réponse à un échelon en entrée et, en conséquence, réduit la crête de la réponse de sortie. Cet effet est équivalent à l'augmentation du temps de montée du front avant de l'impulsion d'entrée (certains générateurs d'impulsions possèdent d'ailleurs ce réglage).

Réalisation pratique

Le câblage de notre circuit ne pose aucune difficulté particulière. Il est bien sûr recommandé de mettre le circuit intégré LM6181 sur un support au cas où ce dernier devrait être changé si une mauvaise manipulation survenait. La **figure 3** représente le circuit côté pistes et la **figure 4** côté composants.

Conclusion

Composant souvent méconnu et peu utilisé, l'amplificateur à contre-réaction en courant permet de réaliser des applications qu'un amplificateur à contre-réaction en tension ne peut faire ou, du moins, avec des performances très inférieures. Les domaines privilégiés d'un amplificateur à contre-réaction en courant sont, entre autres, la commande de câbles coaxiaux, l'amplification vidéo, le filtrage à hautes fréquences ou encore les systèmes d'imagerie et de scanner.

M. LAURY

Nomenclature

- U₁ : LM6181 + support DIL 8 broches
- C₁, C₂ : 10 µF/25V radial
- C₃, C₄ : 100 nF
- C₅ : 10 pF
- R₁, R₂ : 820 Ω (gris, rouge, marron)
- R₃, R₄ : 47 Ω (jaune, violet, noir)
- R₅ : 330 Ω (orange, orange, marron)
- J₁ à J₉ : 9 picots

Environnement de Développement

Basic Tiger :

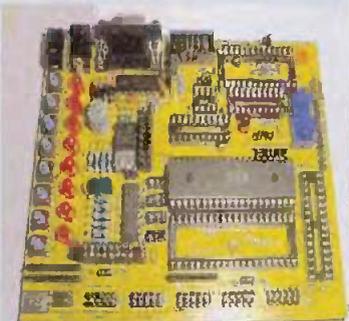
- * Basic Multitâches
- * Mise au point sur carte
- * Drivers pour périphériques
- * Jusqu'à 4 MB de Flash
- * Jusqu'à 1920 E/S Num ou Ana
- Starter kit 1 : 1247 F TTC



AVR :

- * Carte de développement AVR
- STK200 : 635 F TTC

- * Compilateur Basic avec simulateur intégré, gestion du bus I2C, 1 Wire, SPI, Icd, Bus Can : 773 F TTC



Carte d'application montée format barrette mémoire avec AVR 2313 : 316 F TTC, avec AVR 8535 : 427 F TTC

PIC : Compilateurs C, Basic disponibles.



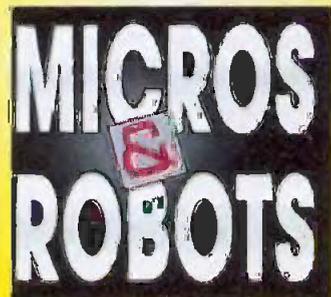
www.optiminfo.com

Route de Ménétreau
18240 Boullieret
Tel : 0820 900 021
Fax : 0820 900 126

INTERFACES ET DEVELOPPEMENTS PC

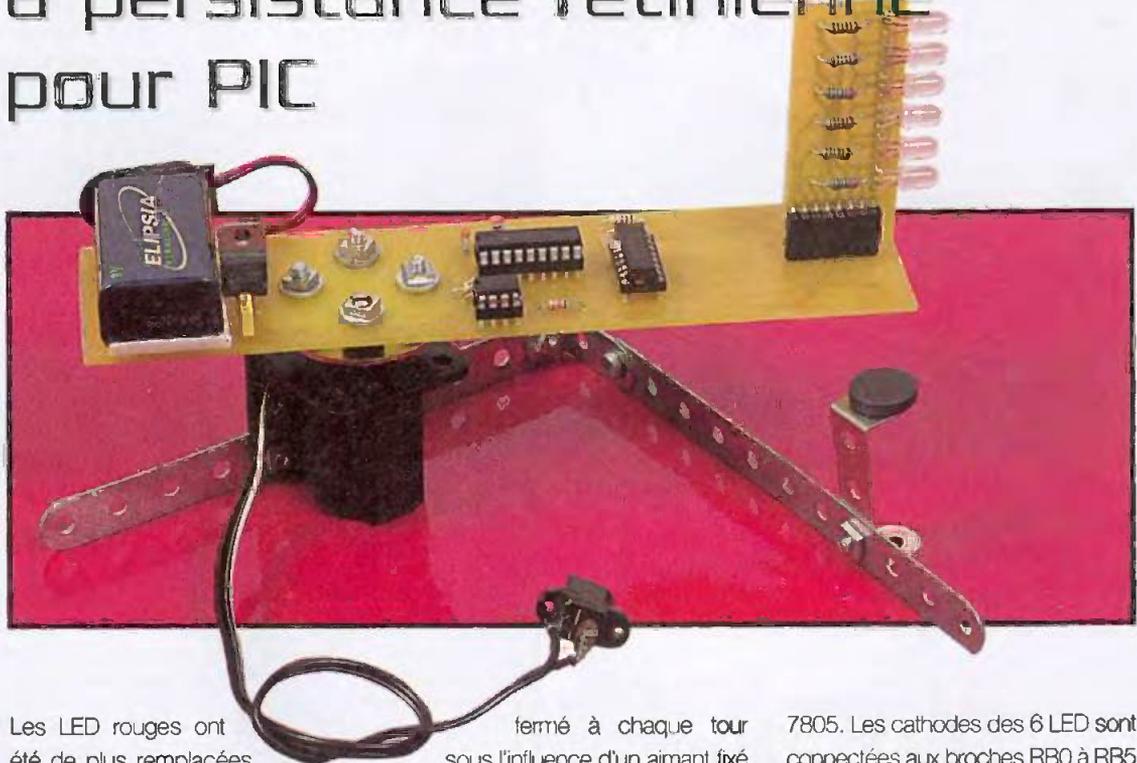
**N°11 prochain numéro
sortie en kiosque le 7 décembre**

**en vente chez tous les marchands
de journaux 5 €**



**N°3 prochain numéro
sortie en kiosque mi-mars
en vente chez tous les marchands
de journaux 4,57 €**

Journal lumineux à persistance rétinienne pour PIC



Dans le n°238 de juillet 1999, un tel système utilisant la persistance rétinienne avait déjà été présenté. C'est un montage idéal, avec son aspect magique, pour s'initier agréablement à l'utilisation des microcontrôleurs. La gestion des LED était confiée à un microcontrôleur 68HC11 et le contenu du message lumineux était stocké dans la mémoire programme de celui-ci. Ici, la gestion des LED est confiée à un microcontrôleur PIC16F84 et les données de l'affichage sont stockées soit dans la mémoire programme du PIC soit dans une EEPROM externe.

Les LED rouges ont été de plus remplacées par des LED bicolores, ce qui permet en fait l'affichage des symboles en 3 couleurs : rouge, vert et orange. Pour votre plus grand confort, l'élaboration des données est en plus facilitée par le programme TOURNIQUET pour PIC, disponible sur le site d'ELECTRONIQUE PRATIQUE.

Principe de l'affichage

Le principe physique de cette technique d'affichage est présentée **figure 1**. Une colonne de 6 LED se déplace dans un plan horizontal sur la circonférence d'un cercle à une vitesse de rotation donnée. L'état de chaque LED, allumé ou éteint, dépend de la position angulaire de la colonne. A chaque tour, plusieurs fois par seconde, les LED sont allumées ou éteintes au même endroit : la rétine de l'œil d'un observateur va avoir l'impression d'un affichage fixe permanent. Le signal déclenchant l'allumage des colonnes de LED est produit par un interrupteur ILS fixé sur le circuit et

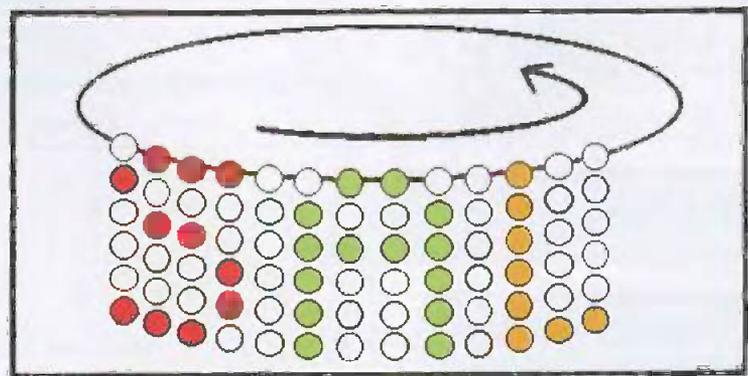
fermé à chaque tour sous l'influence d'un aimant fixé sur le bâti moteur. Sans trop s'éloigner de ce principe, le PIC sera programmé de telle sorte que le texte ne soit pas totalement fixe devant les yeux de l'observateur, mais défile doucement de droite à gauche.

Schéma de principe

Le schéma électrique du montage est présenté **figure 2**. Il est assez simple comme c'est souvent le cas dans les montages à base de microcontrôleurs. Au cœur, se trouve le PIC16F84. Ce circuit, cadencé par un quartz à 4 MHz, est alimenté par une tension continue de 5V fournie par une pile de 9V et un régulateur

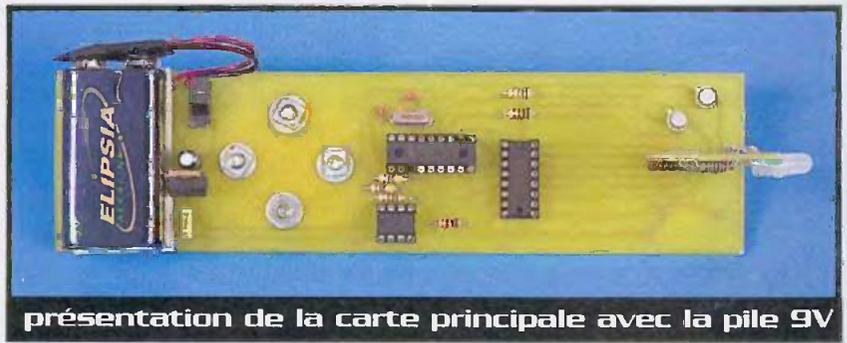
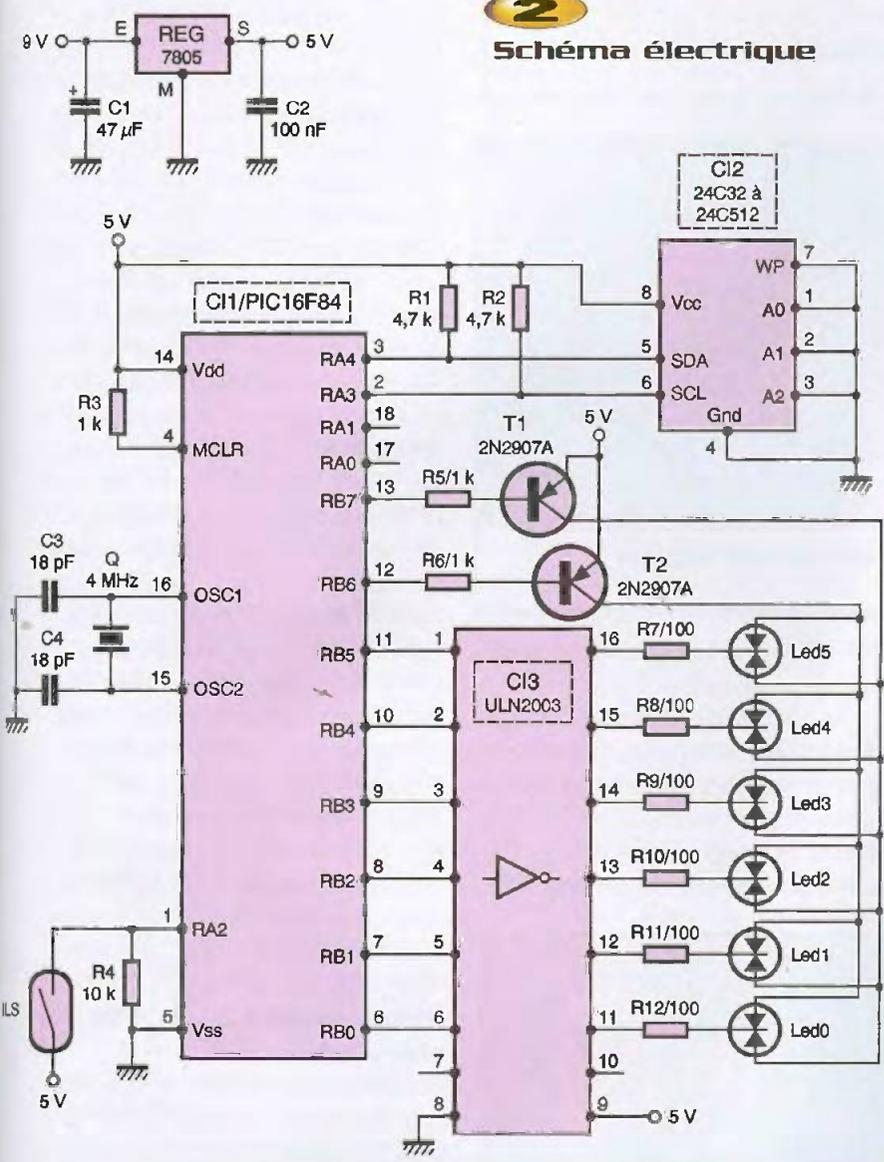
7805. Les cathodes des 6 LED sont connectées aux broches RB0 à RB5 du PIC par l'intermédiaire de l'amplificateur ULN2003 et des résistances R_1 à R_{12} de 100 Ω limitant l'intensité du courant. Les 2 anodes des LED sont, quant à elles, connectées à l'un des 2 transistors T_1 ou T_2 , dont les bases sont alimentées par RB6 ou RB7. Ces transistors de type PNP sont passants si la base est à l'état bas : par exemple, pour allumer en rouge la LED 2, il faudra donc un état bas sur RB6 et un état haut sur RB2, pour l'allumer en vert, il faudra toujours un état haut sur RB2, mais un état bas sur RB7. Enfin, si les broches RB6 et RB7 sont toutes deux à l'état bas, la LED sera orangée. La broche RA2 sera programmée en entrée :

1 Principe physique





2 Schéma électrique



présentation de la carte principale avec la pile 9V

quand l'interrupteur ILS est ouvert, RA2 est à l'état bas grâce à R₄ reliée à la masse. Quand l'ILS se ferme en passant devant l'aimant, RA2 passe momentanément à l'état haut.

Pour les lecteurs qui désireraient une large capacité mémoire pour stocker de longs messages, le schéma décrit la connexion d'une EEPROM externe I2C. Le programme du PIC décrit plus loin JOURI2C.bas est capable de lire toutes les EEPROM de type 24C32 (4 ko) à 24C512 (64 ko). Si vous avez une mémoire de plus petite capacité comme une 24C16, il faudra adapter le programme en conséquence. La ligne SDA est reliée à RA4, la ligne SCL à RA3, ces 2 lignes étant maintenues à l'état haut en l'absence de signal par les résistances R₁ et R₂. L'adresse de cette EEPROM sur le bus I2C qui ne supporte qu'elle, est tout simplement 0, les broches A0, A1 et A2 étant reliées à la masse. Cet article n'abordant qu'une utilisation particulière des EEPROM I2C, j'invite les lecteurs désirant de plus amples connaissances à faire un tour sur le site de Pierre COL, d'adresse Internet : http://col2000.free.fr/i2c/protocol/i2c_pro.htm. Vous y trouverez des explications sur leur fonctionnement, des schémas de programmeurs et des programmes d'applications. Les sous-programmes de lecture de l'EEPROM utilisés dans JOURI2C.bas sont en fait les versions en BASIC des sous-programmes de Pierre COL en assembleur.

Programmation du PIC

Élaboration des données

Avant de rentrer dans le vif du sujet, il faut avant tout comprendre comment sont constituées les données transmises sur le port B pour l'allumage de la colonne de LED. C'est l'objet du schéma présenté **figure 3**.

Utilisation du logiciel TOURNIQUET pour PIC

Les plus courageux d'entre vous pourront s'amuser à calculer chaque donnée une par une, les autres préféreront utiliser le logiciel créé spécialement pour ce montage. La **figure 4** montre une copie d'écran du logiciel en cours d'utilisation. Après avoir ouvert l'application en double-cliquant sur l'icône, il suffit de colorier les ronds figurant

258 à 318, etc. La difficulté du programme résidera dans le fait que les données sont ici mémorisées dans 2 zones de la mémoire programme du PIC : avant l'instruction ADDWF PCL, 1, suivant la zone utilisée, il conviendra donc de préciser la valeur de PCLATH : 1 ou 2.

Remarques post liminaires : Durée d'affichage d'une donnée : avec un moteur de MECCANO et une petite alimentation en 4,5V, le système tournera à une vitesse V d'environ 10 tours par secondes (si votre moteur tourne plus vite, c'est mieux). Le centre de chaque LED étant espacé de 8 mm, les colonnes seront aussi espacées de 8 mm. Le rayon R du cercle de rotation étant de 120 mm, l'intervalle de temps t séparant l'affichage de chaque colonne sur le port B devra être de $1 \times 8 / (2\pi \times R \times V)$, soit 1 ms. Les 50 colonnes seront donc affichées sur un angle de $360 \times 10 \times 50 \times t$, soit 180° .

Ces remarques faites, le programme écrit en BASIC F84 se comprend facilement. Les quelques commentaires qui suivent expliquent le rôle de chaque partie du programme.

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 : par exemple, la variable VAL contient la valeur à afficher sur le port B, COL1 représente le premier tableau de 255 valeurs. DEP représente l'indice dans son tableau de la première valeur à envoyer (première colonne affichée), PCLATHDEP identifie le tableau d'où est extrait la première colonne affichée.

2) initialisation

L'initialisation des registres de direction des ports, est primordiale : ici toutes les broches sont en sortie sauf RA2 sur laquelle est connectée l'ILS. On va commencer l'affichage des colonnes par la première du premier tableau, donc PCLATHDEP=0 et DEP=0.

3) programme principal

Le programme principal commence par le test de l'ILS car l'affichage des colonnes est déclenché par le passage de l'ILS près de l'aimant. On boucle tant que RA2 est bas. Si l'ILS se déclenche, la carte principale a

fait un tour, donc TOUR=TOUR+1. Pour afficher la valeur de la colonne de départ, on donne à PCLATH la valeur de PCLATH-DEP. Pour afficher la valeur de la colonne J, il faut déterminer dans quel tableau elle est. Si $DEP + J > 255$, on a donc une retenue et on passe au tableau suivant. Par exemple, si $DEP = 240$ et $J = 20$, $DEP + J = 260 = 4$. Pour sauter l'instruction ADDWF PCL, 1 on ajoute 1. La valeur lue sera donc la 5ème du tableau suivant. Avant d'afficher cette colonne (avec $J = 20$), on a eu lorsque $J = 15$, $DEP + J = 255$. A cet instant, on est passé d'un tableau à l'autre : la colonne de départ était dans le tableau 1 et la colonne à afficher dans le tableau 2 ou vice-versa. Il faut donc passer de PCLATH=1 à PCLATH=2 ou de PCLATH=2 à PCLATH=1 : c'est le but de la formule $PCLATH = 3 - PCLATH$. Il suffit maintenant de donner à VAL, la valeur lue dans le bon tableau et de l'afficher pendant la durée fixée par TEMPO. Ces opérations sont effectuées pour chacune des 50 colonnes à afficher successivement. Pour avoir un défilement du texte, DEP est augmenté de 1 tous les 2 tours. De la même manière que précédemment, si $DEP = 255$, la colonne de départ est la première de l'autre tableau d'où $DEP = 0$ et $PCLATHDEP = 3 - PCLATHDEP$.

4) temporisation de 1 ms

On peut calculer le nombre de cycles de cette boucle avec la formule $(3 \times VB2 + 4) \times VB1 + 2$.

5) les 510 colonnes

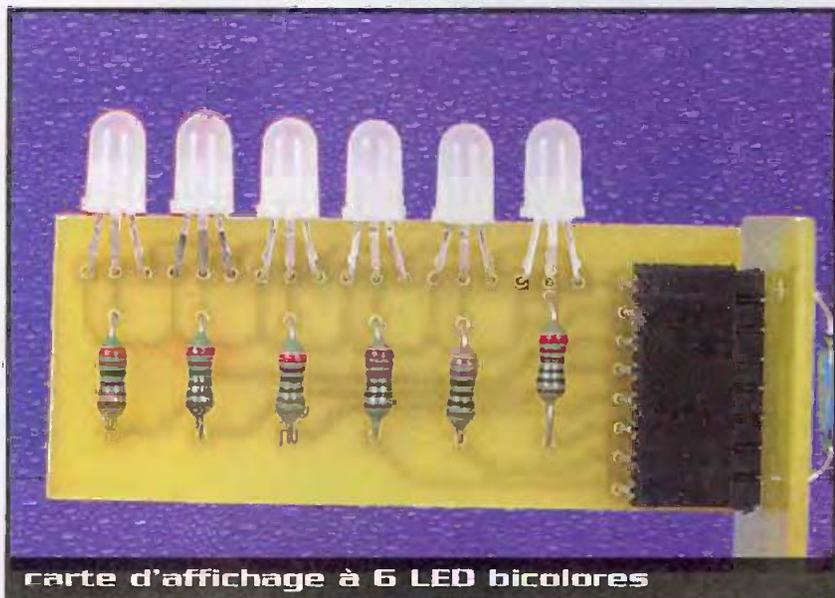
Les données sont réparties dans 2 tableaux COL1 et COL2 de 255 valeurs.

ORG H'100'
COL1 67,76,112,76,67,64,64,127
.....

Programme JOURI2C.bas pour EEPROM 24C32 à 24C512

Les données des colonnes ont été préalablement chargées dans une EEPROM I2C. Un des sous-programmes du programme JOURI2C.bas est chargé de lire le contenu de chaque mémoire de l'EEPROM, cette mémoire étant précisée par son adresse. Ce sous-programme est écrit pour les EEPROM 24C32 à 24C256. Si vous voulez utiliser une 24C16 par exemple, il faudra le modifier légèrement (prendre contact avec l'auteur en cas de difficulté).

Remarques préliminaires : N'ayant pas de tableaux différents à gérer, ce programme qui fait appel à moins d'astuces est plus simple à comprendre que le précédent. Sur le site d'Électronique Pratique, vous trouverez un fichier nommé 24C256000.hex et qui représente le fichier hexadécimal d'une EEPROM 24C256. C'est dans ce fichier qu'est mémorisé un exemple de message à afficher. En fait, sur les 32 ko, seuls les 2 premiers ont servi, ce qui représente déjà un texte d'une bonne longueur comme vous pourrez le constater. L'adresse de chaque mémoire est codée sur 2 octets : l'octet de poids fort ADD_HAUT et l'octet de poids faible ADD_BAS, avec $adresse = ADD_HAUT \times 256 + ADD_BAS$. Notre message occupant 2 ko, la dernière colonne est située à l'adresse H'07FF' : $ADD_HAUT = 7$ et $ADD_BAS = 256$.



carte d'affichage à 6 LED bicolores

La **figure 5** schématise le protocole de lecture d'une EEPROM, mémoire par mémoire, utilisé dans notre programme.

Détails particuliers du programme

1) déclaration des étiquettes et variables

Les étiquettes SCL, SDA et I2CACK sont utilisées pour rendre plus compréhensible le programme (ainsi BSF PORTA,SDA se comprend mieux que BSF PORTA,4). De même que l'adresse de chaque mémoire est précisée par ADD_HAUT et ADD_BAS, l'adresse de la colonne de départ est précisée par DEP_ADD_HAUT et DEP_ADD_BAS. ADD_I2C est l'adresse de l'EEPROM sur le bus I2C.

2) initialisation

Les ports sont configurés comme précédemment. Les 2 lignes SDA et SCL sont mises à l'état haut. ADD_I2C est initialisée à la valeur %10100000. les 4 premiers bits 1010 indiquent que le périphérique I2C est une EEPROM, les 3 zéros suivant représentent l'adresse de l'EEPROM (A2,A1,A0), le dernier zéro est le bit de lecture/écriture ici en écriture pendant l'initialisation.

3) le programme principal

Il commence comme le précédent programme avec le test de l'ILS et l'incréméntation de TOUR.

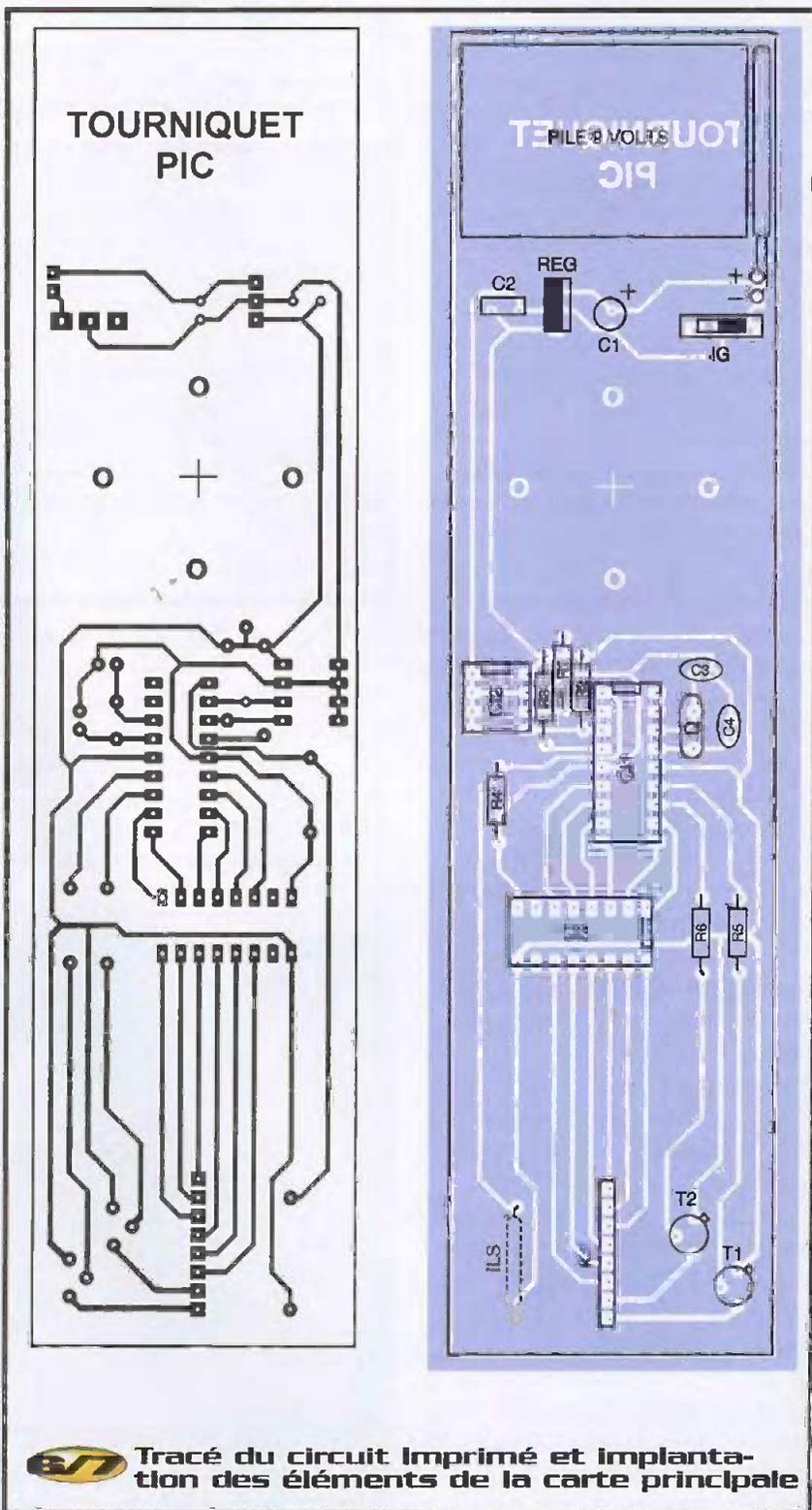
La première colonne affichée étant l'adresse définie par DEP_ADD_HAUT et DEP_ADD_BAS, ADD_BAS de la jème colonne est DEP_ADD_BAS + J. ADD_HAUT=DEP_ADD_HAUT ou DEP_ADD_HAUT +1 si DEP_ADD_BAS+J >255.

Une fois cette adresse calculée, on appelle le sous-programme LIRE_EEP qui renvoie dans OCTET, la valeur de la colonne. Cette valeur est ensuite affichée avec OUT PORTB, OCTET. Puis on passe à la lecture de la colonne suivante. On remarquera qu'il n'y a pas de temporisation de 1 ms entre chaque lecture ! C'est parce que la lecture de l'EEPROM dure elle-même un peu plus de 1 ms : ce n'est pas la peine d'en rajouter. Par contre, on rajoute cette TEMPO pour l'affichage de la dernière colonne avant de l'effacer.

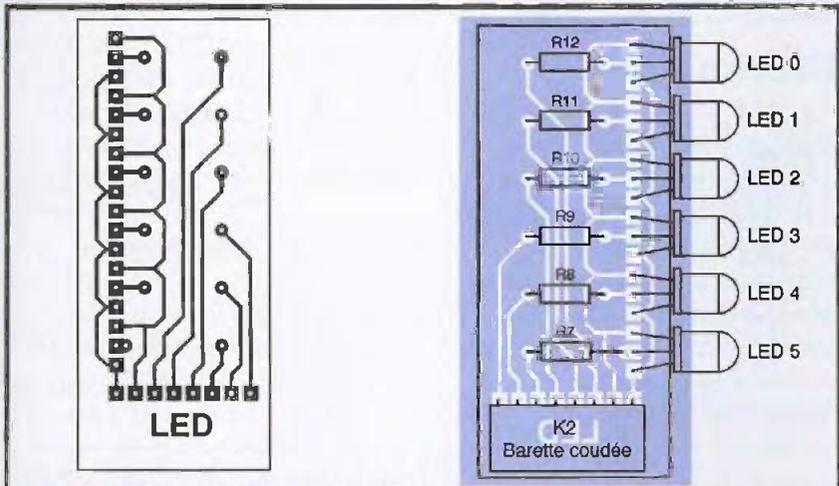
Comme tout à l'heure, le texte est décalé tous les 2 tours par incréméntation de DEP_ADD_BAS. Si DEP_ADD_BAS >255, on incrémente DEP_ADD_HAUT. Si DEP_ADD_HAUT égale 8, on est arrivé à la



5 Lecture d'une mémoire



Tracé du circuit Imprimé et Implantation des éléments de la carte principale



8/9 Tracé du circuit imprimé et implantation des éléments de la carte LED

fin du message mémorisé, il faut repartir en début d'EEPROM.

4) temporisation de 1000 cycles

idem programme précédent.

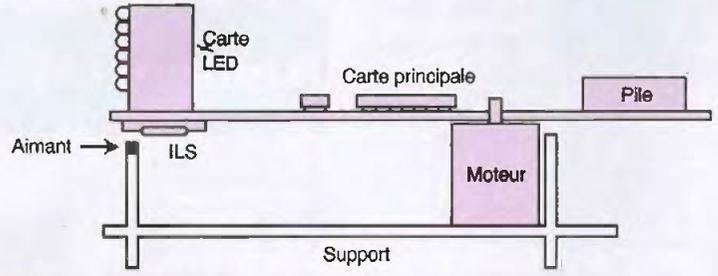
5) tous les sous-programmes de gestion de l'EEPROM

Les sous-programmes qui suivent sont la stricte application du schéma de lecture d'une EEPROM présenté figure 5. Pour plus de détails, faites un tour sur le site de

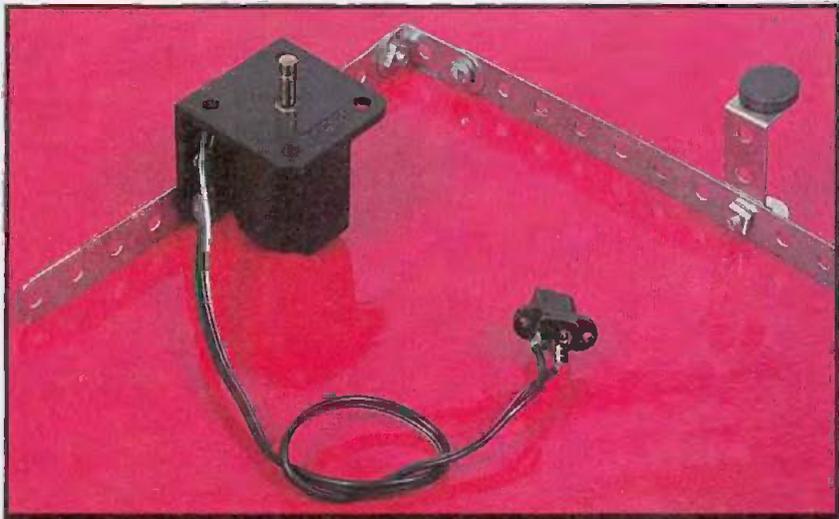
Pierre Col indiqué précédemment,

Chargement du programme dans le PIC

Les 2 programmes sont disponibles sur notre site Internet (www.eprat.com) sous 3 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 assembleur et la



10 Structure minimale



le moteur 6V et la mise en place de l'aimant

troisième son fichier hexadécimal. Lors de la programmation, il ne faudra pas oublier de préciser le type d'horloge utilisée : ainsi, les utilisateurs de PP.exe devront mettre le préfixe -x dans leur ligne de commande puisque les horloges sont à quartz.

Réalisation pratique

Carte principale

Le circuit imprimé du montage est présenté **figure 6**. Les composants seront implantés en respectant le dessin de la **figure 7**. On veillera à respecter la bonne orientation des supports et composants polarisés : condensateur C₁, régulateur 7805, transistors T₁ et T₂, circuits intégrés Cl₁, Cl₂ et Cl₃. On terminera le circuit par la mise en place de l'interrupteur ILS en faisant très attention à la capsule de verre. On pourra même enrober cet interrupteur dans une grosse goutte de colle pour bien le protéger des chocs.

Carte LED

Le circuit imprimé de cette petite carte est présenté **figure 8**. Les composants seront implantés en respectant le dessin de la **figure 9**. Pour ne pas risquer d'inverser les couleurs par rapport au logiciel TOURNIQUET, il faudra positionner le méplat des LED en bas. Si comme l'auteur, vous n'arrivez plus à trouver chez votre revendeur des barrettes coudées, il faudra utiliser une barrette femelle droite qui sera ensuite coudée puis collée avec une goutte de colle cyanolyte à la carte LED.

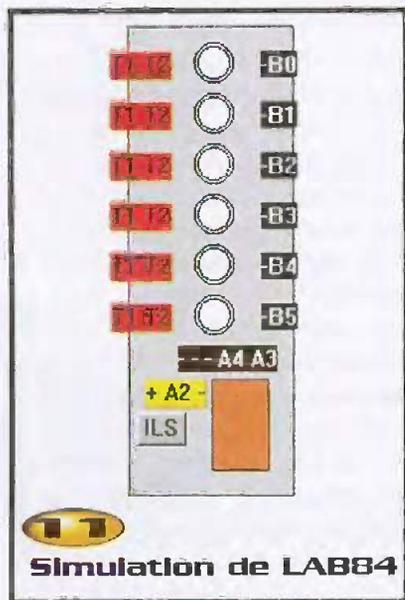
Réalisation mécanique

Chacun est libre d'imaginer le support qu'il veut. La platine tournante est mise en rotation par un moteur de MECCANO fixé sur un petit support. Dans la réalisation de l'auteur de l'article, le support est tout simplement constitué d'une armature de pièces de MECCANO. L'alimentation du moteur est confiée à un bloc secteur classique fournissant des tensions de 3V à 6V. Cette réalisation, schématisée **figure 10**, représente un minimum mais respecte quelques besoins indispensables : le système doit être stable pour s'affranchir d'un éventuel défaut d'équilibrage de la carte principale. Il faudra toutefois essayer de rechercher l'équilibre parfait de cette carte autour de

son axe de rotation : dans un premier temps, on testera cet équilibre après avoir monté la carte sur un axe horizontal et en déplaçant petit à petit la pile maintenue par un scotch double face. Si vous n'arrivez pas à atteindre l'équilibre par cette méthode, il faudra avoir recours à un contrepoids, mais ce serait étonnant, la carte présentée dans cet article ayant été conçue pour être équilibrée sans avoir besoin de contrepoids supplémentaire. L'aimant fermant l'ILS doit être convenablement positionné, un peu à l'extérieur de la trajectoire de l'ILS. Enfin, avant la première mise en œuvre, on testera le branchement du moteur : l'ensemble des pièces en rotation, vu de haut, doit tourner dans le sens trigonométrique.

Mise en œuvre du journal lumineux

- Mettez votre montage dans un endroit dégagé,
- Alimentez la carte principale,
- A la main, faites défiler l'ILS au-dessus de l'aimant permanent : les LED doivent s'illuminer brièvement,
- Alimentez le moteur, votre message lumineux doit défiler comme par magie devant vos yeux. Vous pourrez jouer sur l'affichage des mots en modifiant la vitesse de rotation par le choix de la tension d'alimentation entre 3 et 6 V. Si votre moteur va trop vite ou au contraire pas assez vite, vous pourrez aussi modifier dans les programmes la durée de la temporisation entre l'affichage



des colonnes.

Simulation avec LAB84

Réalisation du montage virtuel

Le microcontrôleur PIC16F84, l'alimentation et les autres composants nécessaires à son fonctionnement comme la circuiterie d'horloge étant par définition déjà intégrée dans LAB84, créer le montage virtuel du journal lumineux consiste à énumérer dans un fichier texte, les autres composants du montage selon leurs types et leurs connexions.

```

circ 100 250
PNP1 T1 B7 +
PNP1 T2 B6 +
LEDB 25 20 T1 T2 -B0
LEDB 25 50 T1 T2 -B1
LEDB 25 80 T1 T2 -B2
LEDB 25 110 T1 T2 -B3
LEDB 25 140 T1 T2 -B4
LEDB 25 170 T1 T2 -B5
POUS ILS 25 200 20 20 + A2 -
24C256 50 210 - - - A4 A3
    
```

Même sans être très familier de LAB84, le fichier est créé rapidement sans difficulté d'après le schéma électrique de la figure 2. La ligne CIRC 100 250 crée un rectangle gris de 100x250 pixels représentant le circuit imprimé. La ligne PNP1 T1 B7 + indique que la base d'un transistor PNP de nom T₁ est reliée à RB7 et que son émetteur est relié au +. La ligne PNP1 T2 B6 + indique que la base d'un transistor PNP de nom T₂ est reliée à RB6 et que son émetteur est relié au +. La ligne LEDB 25 20 T1 T2 -B0 indique qu'une LED bicolor est placée aux coordonnées x=25 et Y=20 sur le circuit, son anode verte est reliée au collecteur de T₁, son anode rouge est reliée au collecteur de T₂, sa cathode à RB0, par l'intermédiaire d'un circuit inverseur (l'ULN2003). Idem pour les 5 autres LED. La ligne POUS ILS 25 200 20 20 + A2 - indique qu'un bouton-poussoir de nom ILS est placé aux coordonnées x=25 et Y=200. Ses dimensions horizontale et verticale sont de 20x20 pixels. Quand il est appuyé, le + est connecté à RA2, sinon c'est le - qui est connecté à RA2. Idem pour les 3 autres boutons. Enfin, la ligne 24C256 50 210 - - - A4 A3 indique le branchement d'une EEPROM I2C de type 24C32 à

24C512, d'adresse chip 0 (A2,A1,A0 connecté au -), avec SDA connecté à RA et SCL connecté à RA. Une fois ce fichier texte écrit, enregistrez-le sous format texte (.txt) sous le nom, par exemple, de circ-JOUR.txt.

Déroulement de la simulation

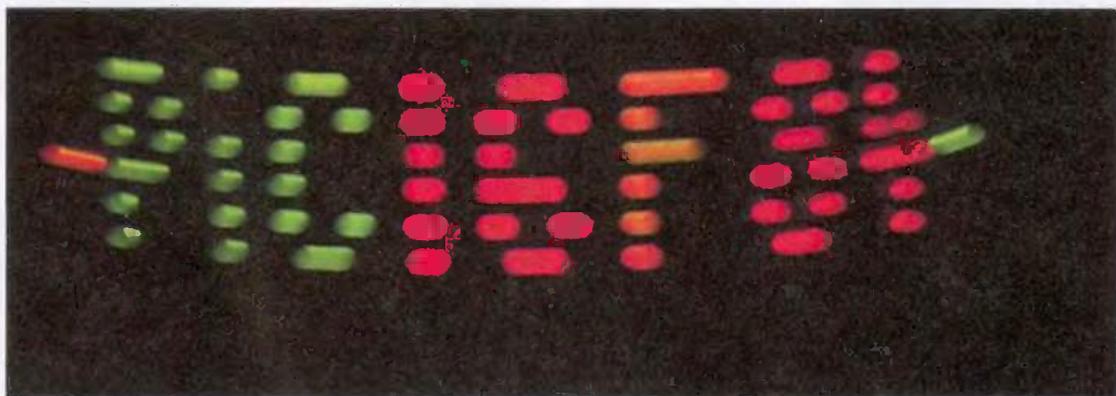
Ouvrez l'application LAB84. Dans menu fichier, item Ouvrir Circuit, sélectionnez le fichier créé précédemment circJOUR.txt. Le dessin du circuit apparaît alors à l'écran comme présenté **figure 11**. Dans menu fichier, item Ouvrir Programme, sélectionnez le fichier assembleur créé par BASIC F84, JOUR.asm ou JOURI2C.asm. Si vous désirez tester le programme JOURI2C, placez le fichier hexadécimal 24C256000.hex dans votre dossier LAB84. Tout est prêt : le montage est construit, le programme voulu est chargé. Démarrez la simulation avec le menu Simulation/Marche et appuyez sur le bouton ILS pour simuler le passage de l'ILS près de l'aimant.

A.REBOUX

Nomenclature

- R₁, R₂ : 4,7 kΩ
- R₃, R₅, R₆ : 1 kΩ
- R₄ : 10 kΩ
- R₇ à R₁₁ : 100 Ω
- C₁ : 47 μF
- C₂ : 100 nF
- T₁, T₂ : 2N2907A
- Q : quartz 4 MHz
- CI₁ : PIC16F84 + support 18 broches
- CI₂ : 24C32 à 24C512, + support 8 broches
- CI₃ : ULN2003 + support 16 broches
- REG : 7805
- 8 LED bicolors
- 1 interrupteur ILS
- K₁ : 8 broches barrette mâle
- K₂ : 8 broches barrette femelle soudée
- 1 moteur (MECCANO)
- support (voir texte) MOTOR MODEL
- 95 rue Robespierre,
- 93000 Montreuil
- tél. 01 48 51 10 00

À propos de la programmation des PIC



Deux ou trois éléments à programmer

Lorsque l'on programme un microcontrôleur PIC, quelle que soit sa référence, on doit en fait programmer deux ou trois éléments différents, selon le contenu du circuit. Ces éléments sont les suivants :

- La mémoire de programme, qui peut être de type EPROM c'est à dire programmable électriquement une seule fois comme dans les 12C508 ou 12C671 par exemple, ou qui peut être de type EEPROM c'est à dire programmable et effaçable électriquement comme dans le cas des célèbres 16F84 mais aussi de tous les 16Fxxx (16F87x en particulier).

- La mémoire EEPROM de données qui peut être absente sur certains circuits.

Les «fusibles» ou bits de configuration dont le nombre et la fonction peuvent varier de manière importante d'un circuit à un autre mais qui sont toujours présents.

La mémoire de programme et les fusibles ou bits de configuration doivent toujours être programmés. La mémoire EEPROM de données, quant à elle, est beaucoup plus rarement concernée. Elle ne doit être programmée que lorsqu'elle est, par exemple, pré-chargée avec des don-

nées exploitées ensuite par le programme.

Un fichier au contenu élastique

Lorsque l'auteur d'un montage vous fournit le fichier à programmer dans votre PIC, ce dernier est toujours au format .hex, reconnu par tous les programmeurs de PIC mais, selon la façon dont le programme initial a été écrit par l'auteur, ou selon l'outil utilisé pour le compiler ou l'assembler, ce fichier ne contient pas nécessairement les informations relatives aux fusibles ou bits de configuration.

Si ce fichier contient les informations relatives à ces fusibles de configuration, aucun problème ne va se poser. En effet, lors du chargement du fichier dans votre programme de pilotage du programmeur, ces informations seront automatiquement prises en compte et le programmeur se configurera correctement.

Par contre, et c'est hélas la situation la plus fréquente, ces informations ne sont finalement qu'assez rarement présentes dans le fichier .hex. Il vous faut alors définir «à la main» l'état de ces fusibles au niveau de votre logiciel de pilotage du programmeur. En effet, même si ce dernier fait généralement un certain nombre de choix

«par défaut» qui correspondent à des situations standards, ceux-ci ne sont pas toujours adaptés à votre cas.

Nous allons donc voir rapidement quelles sont les fonctions de ces fusibles de configuration dans le cas des circuits les plus utilisés dans nos montages, à savoir les 16F84 et la série des 16F87x.

Le cas des 16F84

Les 16F84 ne contiennent que cinq fusibles de configuration, qui n'apparaissent généralement que sous forme de quatre fusibles distincts dans la majorité des logiciels de programmation. Ils ont les appellations et fonctions suivantes :

- FOSC1 et FOSC0, généralement regroupés sous le vocable unique de «Oscillator» ou «Osc», pouvant prendre les valeurs : RC, LP, XT ou HS. «Ce» fusible doit être positionné sur le type d'oscillateur d'horloge utilisé par le 16F84 dans le montage considéré. Il faut donc choisir RC pour un oscillateur à cellule résistance/condensateur, XT pour un oscillateur à quartz jusqu'à une fréquence de 4 MHz incluse, HS pour un oscillateur à quartz de fréquence supérieure à 4 MHz (10 et 20 MHz en particulier). La valeur LP est réservée à un usage avec un oscillateur à

Vous êtes de plus en plus nombreux à contacter l'auteur de ces lignes, que ce soit par courrier ou par Email, via son site Internet (www.tavernier-c.com), à propos de la programmation des PIC et, plus particulièrement, en ce qui concerne les «fusibles» de configuration dont l'état est rarement précisé dans les articles mettant en œuvre ces microcontrôleurs. Nous avons donc estimé utile de préciser ces notions une fois pour toutes ; notions qui sont fort simples lorsqu'elles sont correctement exposées.

quartz à très basse fréquence tel celui utilisant un quartz horloger à 32,768 kHz. Un mauvais choix au niveau de «ce» fusible conduit à un fonctionnement erratique de l'oscillateur qui peut refuser de démarrer ou dont la fréquence peut être instable.

- WDT ou WDTE est le fusible de validation du timer chien de garde (Watch Dog Timer Enable). Il ne doit être laissé non programmé que si cette fonction est explicitement prévue et gérée par le programme utilisé, ce que doit donc vous indiquer l'auteur du logiciel. Si vous laissez ce fusible non programmé à tort, votre programme ne fonctionnera pas car, dès la première interruption générée par le timer chien de garde, il se «plantera» puisque cette fonction ne sera pas supportée.

- PWRT ou PWRT(E) est le fusible de validation du timer de reset à la mise sous tension (PoWer up Reset Timer Enable). Sauf mention contraire explicitement exprimée par l'auteur du logiciel (dont nous ne voyons d'ailleurs pas bien quelle pourrait être la raison dans le cadre de nos montages), ce fusible doit toujours être programmé. Il permet de bénéficier d'un délai d'attente après la stabilisation de la tension d'alimentation avant que le programme ne démarre son exécution ce qui est une attitude prudente.

- CP ou CP0 est le fusible de protection de la mémoire de programme contre une lecture non autorisée. Si vous programmez un PIC pour votre propre usage, ce fusible n'a pas besoin d'être programmé. Votre mémoire de programme reste ainsi accessible en lecture et vous permet de vérifier, si nécessaire, son contenu. Par contre, si vous avez développé un programme que vous voulez protéger des copies illicites, vous programmerez ce fusible avant d'envoyer le circuit dans la nature. Sa mémoire de programme ne pourra plus être lue par quiconque et son contenu sera donc protégé. Le **tableau 1** résume de façon synthétique la signification de ces fusibles et devrait vous suffire une fois que vous avez lu les quelques explications qui précèdent. Rappelons en outre que ces fusibles de configuration, malgré leur appellation de fusible héritée du passé, peuvent être effacés au même titre que la mémoire de programme. Il vous est donc toujours possible de corriger une erreur faite à leur niveau, ce qui est bien utile dans le cas où vous auriez

Fusible	Etat	Etat par défaut	Fonction
FOSC1 et FOSC0	00		Oscillateur LP
	01		Oscillateur XT
	10		Oscillateur HS
	11	X	Oscillateur RC
WDT(E)	0		Timer chien de garde inactif
	1	X	Timer chien de garde actif
PWRT(E)	0		Timer de reset à la mise sous tension actif
	1	X	Timer de reset à la mise sous tension inactif
CP(O)	0		Mémoire de programme protégée
	1	X	Mémoire de programme non protégée

Résumé de la signification des fusibles de configuration des 16F84 (un fusible vierge correspond à un 1 logique)

un doute quant à l'état du fusible WDT par exemple.

Corrigeons aussi une stupidité que nous avons déjà lue plusieurs fois. Le fusible CP de protection de la mémoire ne rend pas cette dernière inaltérable. Il interdit juste sa relecture depuis l'extérieur afin de protéger son contenu de toute copie non autorisée. Par contre, ce fusible étant effaçable comme les autres, il donne bien évidemment à nouveau accès à la mémoire lorsqu'il est effacé. La sécurité n'est cependant pas mise en cause dans ce cas puisque l'effacement des fusibles n'est possible qu'en même temps qu'un effacement global de la mémoire de programme ! Lorsque le circuit vous est livré vierge ou, ce qui revient au même, après un effacement de celui-ci et donc des fusibles de configuration, ceux-ci sont non programmés ce qui, dans la terminologie officielle de MICRO-CHIP, donne la configuration suivante :

- Oscillateur en mode RC,
- WDT ou timer chien de garde validé,

- PWRT ou timer de reset à la mise sous tension non validé.

- CP ou protection du code non activée. Munis de ces informations, vous ne devriez donc plus avoir aucun mal à configurer correctement les fusibles de vos 16F84 avant programmation.

Le cas des 16F87x

Ces circuits, que l'on commence à rencontrer de plus en plus fréquemment car ils présentent les mêmes avantages que le 16F84 tout en offrant des ressources internes plus nombreuses et une mémoire de programme beaucoup plus vaste, disposent également de ce fait de fusibles de configuration nettement plus nombreux. Voyons donc ci-dessous ce qu'il en est, étant entendu que notre exposé est valable à ce jour pour les 16F873, 874, 876 et 877 :

- FOSC1 et FOSC0 sont généralement regroupés sous le vocable unique de



le PIC 16F84 en situation

«Oscillator» ou «Osc» comme pour le 16F84. Ils peuvent prendre les mêmes valeurs et ont les mêmes fonctions que pour ce dernier. Nous vous renvoyons donc à ce que nous avons écrit ci-dessus.

- WDT ou WDTE est le fusible de validation du timer chien de garde (Watch Dog Timer), ici aussi tout ce que nous avons écrit ci-dessus pour le 16F84 reste valable.

- PWRT ou PWRTÉ est le fusible de validation du timer de reset à la mise sous tension (PoWer up Reset Timer Enable). Comme pour le 16F84, et sauf mention contraire explicitement exprimée par l'auteur du logiciel, ce fusible doit toujours être programmé.

- CP1 et CP0 sont les fusibles de protection de la mémoire de programme contre une lecture non autorisée. Ils fonctionnent comme le seul fusible CP du 16F84 mais permettent de protéger diverses zones de la mémoire de programme si nécessaire.

Le **tableau 2** indique les différentes combinaisons possibles selon les versions de circuits 16F87x. Comme pour le 16F84, la mémoire est non protégée si ces fusibles ne sont pas programmés.

- DEBUG est le fusible de validation du mode de «debug» en circuit. Si ce mode n'est pas utilisé, ce qui est généralement le

CP1 et CP0	Zone protégée	Zone protégée
	16F873 et 16F874	16F876 et 16F877
00	0000 à 0FFF	0000 à 1FFF
01	0800 à 0FFF	1000 à 1FFF
10	0F00 à 0FFF	1F00 à 1FFF
11	Aucune	Aucune

T2 Signification des fusibles CP0 et CP1 dans le cas des 16F87x

cas de toutes les réalisations que nous vous proposons, ce fusible ne doit pas être programmé.

- WRT est le fusible d'interdiction d'écriture dans la mémoire EEPROM de programme à partir du registre EECON. Il doit être programmé uniquement si cette fonction est requise par le programme exécuté par le circuit. C'est donc à l'auteur de ce dernier de vous fournir cette indication.

- CPD est le fusible de protection du contenu de la mémoire EEPROM de données. Si ce fusible est programmé, le contenu de la mémoire EEPROM de données est protégé. En règle générale, ce fusible est laissé non programmé.

- LVP est le fusible de sélection du mode de programmation basse tension des PIC 16F87x. Il faut savoir, en effet, que ces circuits peuvent se programmer avec une «haute» tension de 12V comme les 16F84 ou avec une seule et unique tension de 5V.

Pour cela, ils doivent être mis en mode LVP en ne programmant pas ce fusible. Cela signifie donc que, par défaut, les circuits vierges sont toujours en mode LVP. Fort heureusement, même lorsqu'ils sont dans ce mode, ils peuvent toujours être programmés en mode «haute» tension par les programmeurs conventionnels.

- BODEN est le bit de validation du «brown out reset» ou reset automatique en cas de baisse anormale de la tension d'alimentation. Pour plus de précisions à ce sujet nous ne saurions trop vous recommander la lecture de notre ouvrage «Les microcontrôleurs PIC - Description et mise en œuvre» dans sa dernière édition (celle accompagnée des deux CD ROM de MICROCHIP) publié chez DUNOD. En général, il est vivement conseillé de ne pas programmer ce bit afin de valider cette fonction. Le **tableau 3** résume la signification des fusibles des 16F87x conformément-

Fusible	Etat	Etat par défaut	Fonction
FOSC1 et FOSCO	00		Oscillateur LP
	01		Oscillateur XT
	10		Oscillateur HS
	11	X	Oscillateur RC
WDT(E)	0		Timer chien de garde inactif
	1	X	Timer chien de garde actif
PWRT(E)	0		Timer de reset à la mise sous tension actif
	1	X	Timer de reset à la mise sous tension inactif
BODEN	0		«Brown out reset» inactif
	1	X	«Brown out reset» actif
LVP	0		Programmation en mode haute tension
	1	X	Programmation en mode haute et basse tension
CPD	0		Code de la mémoire EEPROM de données protégé
	1	X	Code de la mémoire EEPROM de données non protégé
WRT	0		Écriture dans la mémoire EEPROM de programme impossible via EECON
	1	X	Écriture dans la mémoire EEPROM de programme possible via EECON
DEBUG	0		Mode «debug» en circuit validé
	1	X	Mode «debug» en circuit non validé
CP0 et 1	—		Voir tableau 2
	11	X	Mémoire de programme non protégée

T3 Résumé de la signification des fusibles de configuration des 16F87x (un fusible vierge correspond à un 1 logique)

ment aux explications que nous venons de vous donner.

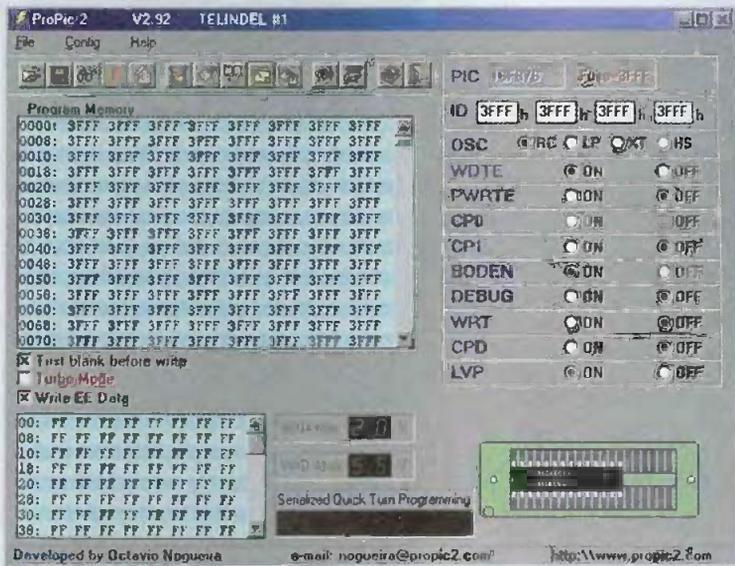
Lorsque le circuit vous est livré vierge ou après effacement de celui-ci, les fusibles configurent le circuit dans l'état suivant :

- Oscillateur en mode RC,
- WDT ou timer chien de garde validé,
- PWRT ou timer de reset à la mise sous tension non validé,
- CP0 et CP1 ou protections du code non activées,
- BODEN ou reset en cas de baisse de la tension d'alimentation activé,
- LVP ou mode programmation basse tension validé,
- CPD ou protection de code de la mémoire EEPROM de données non validée,
- WRT ou interdiction d'écriture dans la mémoire EEPROM de programme au moyen du registre EECON non validée,
- DEBUG ou mode «debug» en circuit non validé.

Comme pour le 16F84, vous pouvez constater, à la lecture de ces informations, que la position de ces fusibles par défaut est la plus logique qui se puisse concevoir dans un cas général mais, munis des informations ci-dessus, vous êtes cependant à même de les programmer au mieux en fonction des besoins de votre application.

Des programmeurs parfois ambigus

Si les exposés ci-dessus sont facilement compréhensibles, un doute peut cependant subsister au vu de la syntaxe utilisée par certains logiciels de programmation. Pour ce qui est des deux plus utilisés que sont ProPic2 et ICProg, voici ce qu'il en est : Comme le montre la **figure 1**, ProPic2 indique en clair si la fonction correspondant à tel ou tel fusible est validée (On) ou non validée (Off). Il est donc inutile de savoir si le fusible correspondant doit être mis à 1 ou à 0. Il suffit juste de choisir les fonctions dont on désire ou non la validation. Pour ce qui est de ICProg, la présentation adoptée est un peu plus ambiguë comme le montre la **figure 2**. En fait, il suffit de se souvenir que la ou les fonctions que l'on désire valider doivent être cochées dans la partie droite de l'écran, sans avoir, là aussi, à se souvenir de l'état des fusibles correspondants.



1 Le logiciel ProPic2 indique clairement si telle ou telle fonction est validée

Enfin, en présence d'un logiciel dont vous ne connaissez pas le comportement, il est facile de lever tout doute éventuel. Il suffit pour cela de lui faire lire un circuit vierge et de regarder comment il positionne les indications relatives aux fusibles. Un coup d'œil au tableau 1 ou 3 vous permet alors facilement de déterminer la logique du programme.

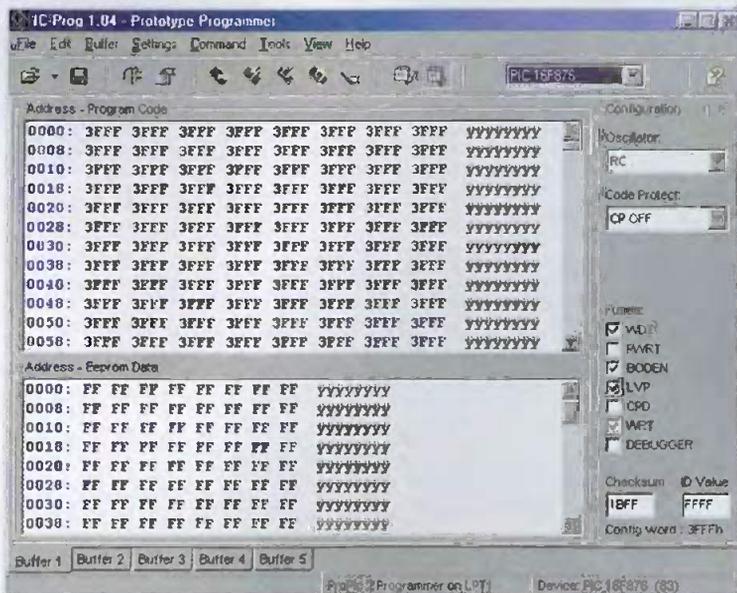
Conclusion

Les quelques informations données dans cet article devraient vous permettre de pro-

grammer vos 16F84 et 16F87x dans les meilleures conditions possibles. Si un doute subsiste à propos de l'état de certains fusibles, il ne vous reste plus, hélas, qu'à essayer les différentes configurations possibles.

Ces causes d'incertitudes ne concernant que le seul fusible WDT dans le cas du 16F84 et les fusibles WDT, CPD et WRT dans le cas des 16F87x, ces essais devraient être rapidement concluants.

C. TAVERNIER



2 Le logiciel ICProg est un peu plus ambigu. En fait, une fonction est validée lorsque la case correspondante est cochée