



Ampli OTL  
pour casque



Programmateur  
de P87LPC76x

Interface  
IrDA

Projétomètre  
à PIC16F84

RC-Analyser :  
pour modèles  
réduits

# CLIMAT- LOGGER

pour  
température  
et humidité



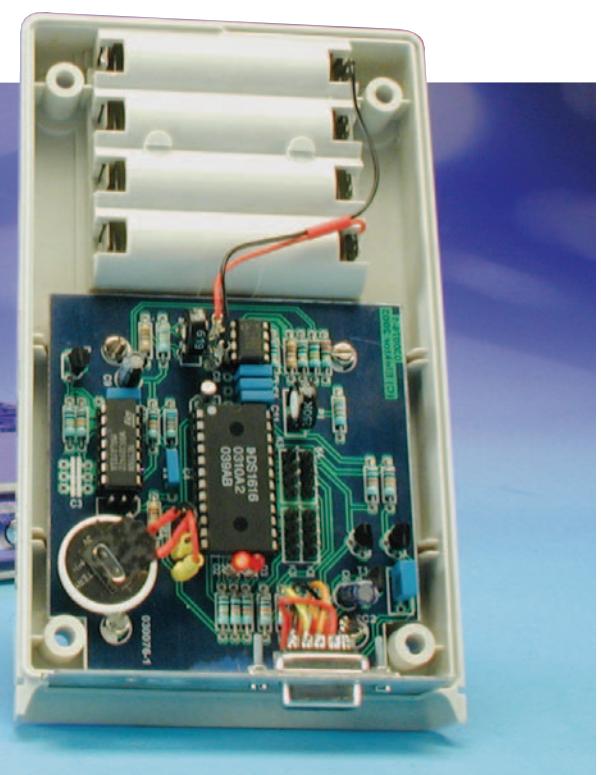
# Climat-Logger

Acquisition de températures et degrés d'humidité

Thomas Poms

<http://chripo.icb.at/Rooney>

Cette alternative aux thermo- et xérographes mécaniques peut se targuer d'être extrêmement compacte, autonome et de consommer très peu de courant. Le DS1616 de Dallas intègre déjà un capteur de température. Il est possible d'y connecter d'autres capteurs qu'un capteur d'humidité. L'évaluation des données recueillies se fait par le biais d'une interface RS-232 au travers d'un programme confortable tournant sous Windows sur un PC de bureau ou portable. Il est également possible d'envisager de connecter cette centrale d'acquisition à une carte à microcontrôleur.



Au cours de ses « missions » (série de mesures), le Climat-Logger travaille de façon autonome et tire son alimentation de piles (ou d'accus) et cela pendant une durée qui pourra être assez longue. On pourra ainsi, sans arrière-pensée, le mettre par exemple dans un compartiment réfrigéré ou dans un cellier à vin climatisé en vue de relever continûment les valeurs de température et d'humidité. Pour cette opération, le PC sur lequel tourne le programme Windows ne sert pas uniquement à la lecture des données recueillies au cours d'une telle mission et à leur évaluation, mais aussi à la définition des paramètres de la centrale d'acquisition (d'où le *logger* du titre) avant de lancer une « campagne » (pour rester dans la terminologie militaire) de mesures.

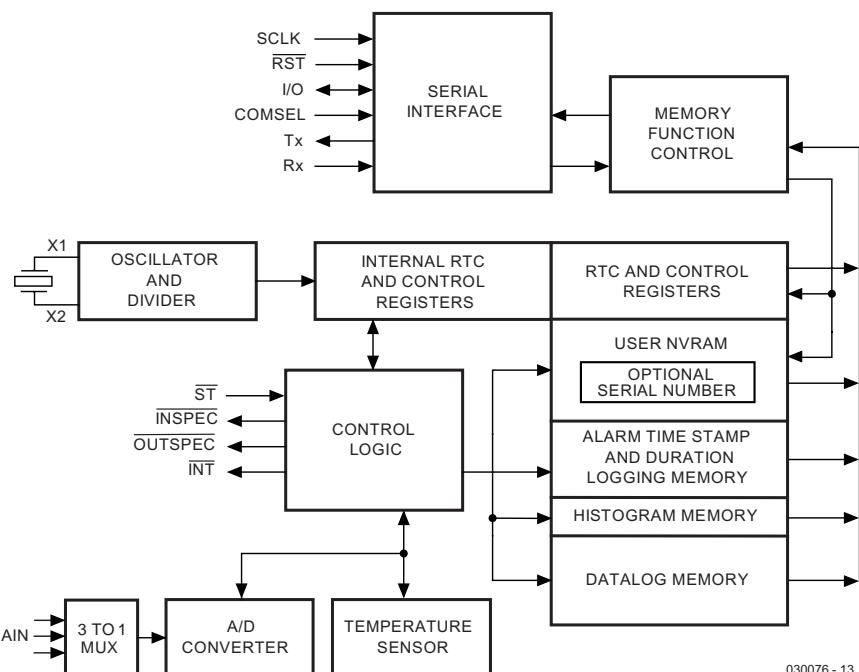
Par le biais du logiciel qui tourne sur le PC on paramètre ce (les canaux d'entrée) que l'on veut mesurer et comment (la fréquence d'échantillonnage).

Sur la présente version du Climat-Logger la liaison avec le PC se fait par le biais de l'interface série. Une version USB est en cours de développement (sans promesse de publication cependant, nous verrons en temps voulu).

## Le DS1616

Nous avons déjà eu l'occasion, dans plusieurs montages publiés dans Elektor, d'utiliser des capteurs de température « intelligents » de l'écuie Dallas Semiconductor. Il existe toute une famille de circuits intégrés disposant chacun de périphériques différents de sorte qu'il n'y a pas de problème de trouver un composant taillé sur mesure en fonction de la tâche à remplir et de l'application envisagée. La présence d'une mémoire intégrée en permet une mise en oeuvre confortable en tant qu'électronique d'acquisition de températures. Nous avons affaire, avec le DS1616 utilisé ici, à un membre « complexe » multi-fonctionnel de la famille des capteurs de température :

Si l'on fait abstraction du capteur de température et de la mémoire intégrée déjà évoqués plus haut, la fonction additionnelle la plus intéressante du DS1616 est indubitable-

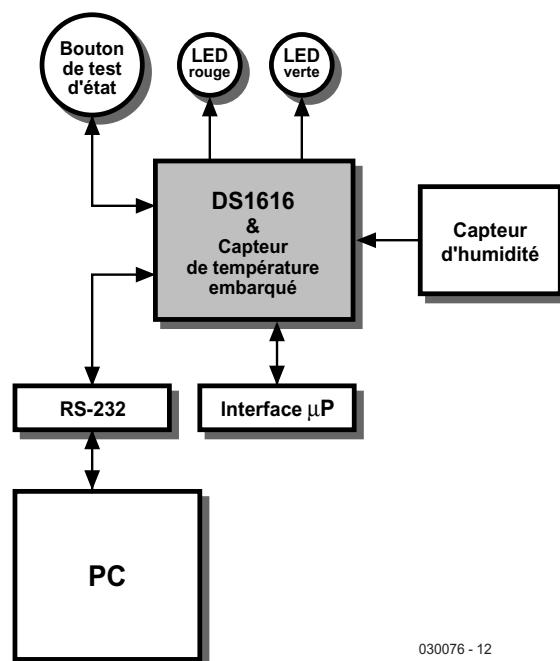


030076 - 13

Figure 1. Structure interne du DS1616.

ment, comme le montre le synoptique de la figure 1, la présence d'un convertisseur analogique/numérique (CAN) 8 bits en amont duquel on trouve un multiplexeur 3-vers-1. Il est possible ainsi de mesurer et de mémoriser 3 signaux analogiques externes se trouvant dans la plage

allant de 0 à 2 V de façon aussi précise que le signal de température interne. Outre cela, le DS1616 possède une horloge en temps réel (RTC = Real Time Clock) cadencée par un oscillateur interne (piloté par un quartz horloger de 32,768 kHz). L'horloge n'a pas pour seule fonction de cadencer une interface RS-232 sérieuse rustique, mais aussi



030076 - 12

Figure 2. Synoptique du Climat-Logger.

d'appeler les heures d'alarme, taux d'échantillonnage, repères chronologiques et temporisations programmés en mémoire; elle est indispensable pour l'échantillonnage des canaux de mesure et pour l'exécution de certaines instructions telles qu'effacement de la mémoire (*Clear Memory*).

Le capteur de température intégré du DS1616 définit la plage d'utilisation du Climat-Logger : elle va ainsi de -40 à +80 °C. De ce fait, les composants de la circuiterie connexe et de l'électronique de mesure de l'humidité se doivent d'être prévu au moins pour cette plage de températures si l'on ne veut pas qu'ils aient une influence négative sur les spécifications du Climat-Logger (**figure 2**).

## Cartographie mémoire

La mémoire du DS1616 a une taille de 2 048 octets subdivisés en pages de 32 octets (cf. **figure 3**). L'horloge en temps réel, le registre de commande (*Control Register*) et la RAM utilisateur (*User NV RAM*) ont à leur disposition une possibilité d'écriture. Nous aborderons un peu plus loin, dans le paragraphe « Logiciel », les fonctions de l'horloge TR et du registre de commande si tant est qu'elles soient pertinentes pour le Climat-Logger.

Pour l'utilisateur, tous les autres registres sont à lecture seule. Chaque instruction de lecture comporte l'adresse de début (*start*) à partir de laquelle doit se faire la lecture.

Le transfert des données se poursuit jusqu'à ce que soit atteinte la fin de page. Cela signifie par conséquent que l'on aura, par instruction de lecture, émission vers le maître (le PC), d'un maximum de 32 octets de données auxquels s'ajoute une paire d'octets CRC générés matériellement par le DS1616. L'octet de contrôle de redondance cyclique (CRC = *Cyclic Redundancy Check*) détecte une transmission comportant des erreurs, de sorte que l'on pourra, si besoin était, démarrer un nouveau processus d'émission.

La RAM est de type non volatile (NV), ce qui signifie que le contenu de la mémoire reste conservé même en cas de disparition de la tension d'alimentation. Cette RAM est utilisée pour le stockage dans la puce de paramètres importants lors de la clôture du programme tournant sur le PC et leur récupération lors d'un redémarrage du logiciel.

Avec la version de programme pour le PC actuelle, on a mémorisation des paramètres suivants :

- Les canaux de mesure sélectionnés lors de la dernière série de mesures démarrée
- Le taux d'échantillonnage
- Des paramètres d'optimisation des performances

## DS1616 MEMORY MAP

Address	Register definition	Page(s)
0000H 003FH	RTC and Control Registers	0 - 1
0040H 005FH	User NV RAM	2
0060H 0217H	(Reserved for Future Extensions)	3 - 16*
0218H 021FH	Serial Number	16**
0220H 027FH	Alarm Time Stamps and Durations	17 - 19
0280H 07FFH	(Reserved for Future Extensions)	20 - 63
0800H 087FH	Temperature Histogram (63 Bins of 2 Bytes Each)	64 - 67
0880H 08FFFH	ADC Channel 1 Data Histogram (64 Bins of 2 Bytes Each)	68 - 71
0900H 0FFFH	(Reserved for Future Extensions)	72 - 127
1000H 17FFFH	Datalog Memory (64 pages)	128 - 191
1800H and higher	(Reserved for Future Extensions)	192 +

\* First 8 bytes  
\*\* Last 8 bytes

Figure 3. Cartographie mémoire du DS1616.

Le schéma de la **figure 4** nous permet d'entrer au cœur du Climat-Logger. L'interface serielle du DS1616 requiert une tension d'alimentation de 5 V, qui n'est bien entendu uniquement nécessaire s'il y a effectivement transfert de données par le biais de cette interface. Cette tension est fournie par un régulateur de tension intégré du type 78L05, qui dérive sa tension d'entrée du port COM du PC.

Les 2 LED permettent de connaître l'état du DS1616.

Il faut, pour une interrogation d'état, appuyer pendant 2 s au minimum sur la touche connectée à l'embase JP1. Lors de l'initialisation d'une série de mesures (mission) les LED rouge (OUTSPEC, D2) et verte (INSPÉC, D3) clignotent simultanément 4 fois. Si l'on procède à une interrogation d'état après le démarrage d'une mission mais avant que la première mesure n'ait déjà eu lieu, les LED clignotent alternativement 4 fois encore, la LED rouge s'allumant en premier. Le démarrage d'une série de mesures pourra se faire tant depuis le PC que, en mode autonome, par une action sur la touche prévue à cet effet (cf. description du logiciel et mise en oeuvre à la fin de l'article).

Le logiciel permet de définir des

valeurs de seuil dont tout dépassement, d'un côté ou de l'autre, par l'une des valeurs (de mesure) analogiques (en externe ou par le capteur de température) entraîne le positionnement (mise à « 1 ») d'un bit du registre d'état 1 (*Status 1 Register*). Si cela a été le cas, la LED rouge (D2) clignote 4 fois. S'il n'y a pas eu de sortie du domaine défini par les seuils, la LED verte, D3, clignote 4 fois.

À l'intention de ceux qui voudraient utiliser le Climat-Logger sans PC, le montage a été doté d'une interface permettant de le connecter à une (n'importe laquelle) carte à microcontrôleur. La transmission se fait, contrairement à ce qui est le cas avec le PC, en synchrone par le biais des lignes SCLK (*Serial Clock*), IO (*Input/Output*) et RST (*Reset*). Il faut, pour passer d'un mode asynchrone au mode synchrone, forcer la broche 6 (COMSEL) au potentiel V<sub>CC</sub> (+ 5 V).

## Capteur d'humidité

Le capteur d'humidité capacitif HS1101 mesure l'humidité relative de l'air ambiant. La capacité du capteur varie de façon proportionnelle à l'humidité. Cette variation de capacité est convertie en un signal MLI (PWM

pour *Pulse Width Modulation* = Modulation en Largeur d'Impulsion). Ceci explique qu'il faille une circuitterie relativement complexe pour pouvoir appliquer le signal de mesure à la broche AIN2 (Analog IN) du DS1616.

À cet effet, une moitié du double temporisateur TS556 monté en multivibrateur libre génère un signal rectangulaire impulsional constant. Le réseau RC constitué par R1, R2 et C1, définit une pseudo-période de 1 seconde environ. IC1.A déclenche la seconde moitié du double temporisateur, montée elle en multivibrateur monostable et qui est chargée de la modulation en largeur d'impulsion. La sortie est dotée d'un filtre passe-bas qui dérive du signal MLI une tension analogique proportionnelle à l'humidité de l'air se trouvant

dans une plage allant de 990 à 1 190 mV (en fonction de la tolérance du capteur). Il ne reste plus qu'à adapter le signal à la plage de modulation (0 à 2 V) de l'entrée analogique du DS1616 de manière à obtenir une bonne précision. Il est à noter que, si l'on en croit la fiche de caractéristiques, le capteur possède une capacité pouvant aller de 161 à 200 pF, et que la tolérance ces composants (condensateurs électrochimiques) utilisé est encore plus importante, il faudra étailler individuellement chacun des montages. L'amplificateur de mesure IC3 offre cette possibilité. À son niveau la tension de mesure subit un gain proche de 10x. C'est à dessein que les valeurs de résistance choisies sont très élevées de manière à limiter les courants. Le signal de sortie lui aussi

arrive à l'entrée analogique du DS1616 par le biais d'une résistance de valeur élevée.

Un LM385-2.5 sert à générer la tension VCC\_HUM qui dérive de la tension de pile dont la constance est inévitablement sujette à caution, une tension de référence qui alimente tous les composants faisant partie du sous-ensemble de mesure d'humidité exception faite de IC3. Cette approche garantit une quasi-inexistance de variations de la tension d'alimentation (PSRR).

## Une centrale d'acquisition compacte

Le Climat-Logger prend place sur une platine simple face; en dépit de cela la réalisation de ce montage ne devrait pas poser de problème. Il n'y a pas de pont de câblage à mettre en place; les circuits intégrés seront montés sur support. Toutes les sorties accessibles de l'extérieur sont dérivées vers des

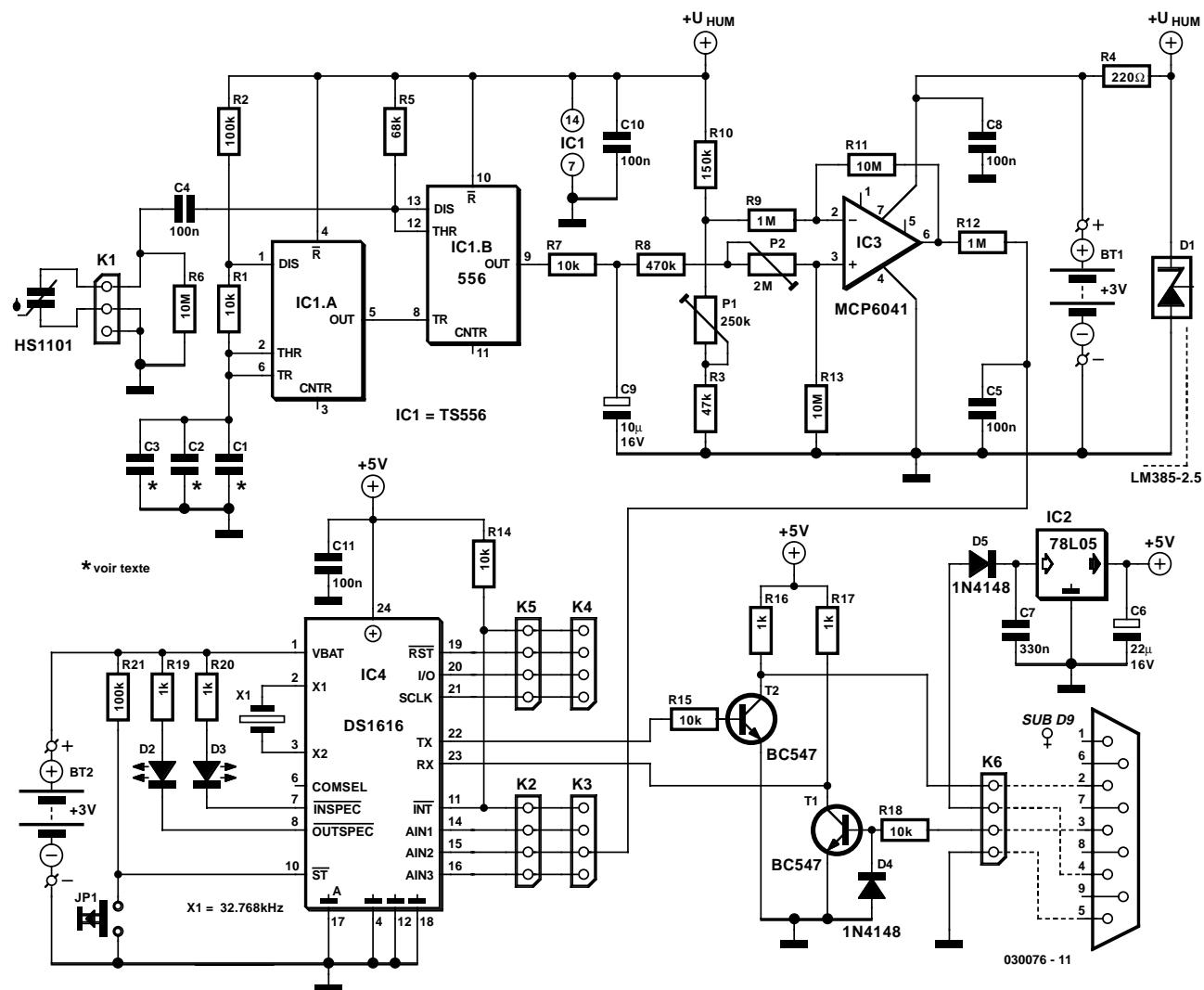


Figure 4. L'électronique du Climat-Logger.

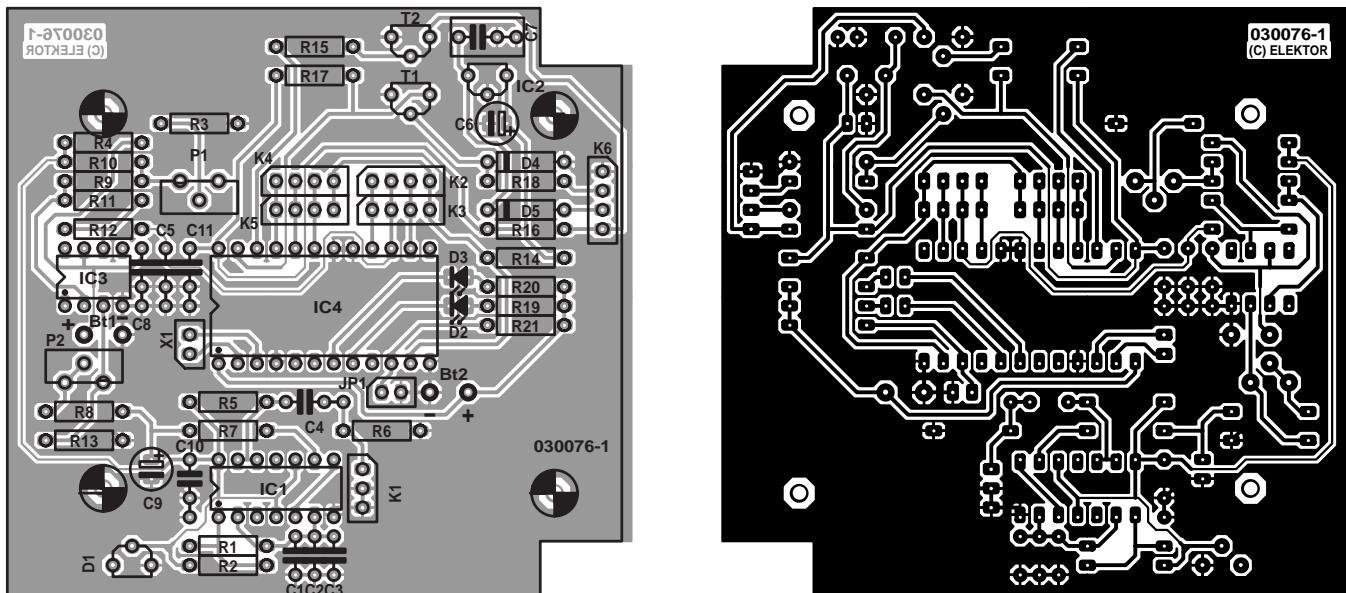


Figure 5. Dessin des pistes et sérigraphie de l'implantation des composants de la platine simple face dessinée à l'intention de cette réalisation.

embases de contacts. Cela est également le cas pour l'embase sub-D 9 points (RS-232) ainsi que pour les paires de contacts de branchements des sources de courant, BT1 et BT2. BT1 est sensiblement plus chargé par le circuit de mesure d'humidité que l'« accu de sauvegarde » BT2 qui fournit son courant au DS1616. Dans ces conditions, on comprend que l'on puisse connecter à BT1 une paire de piles Mignon (R6) ou Lady (R1) prises en série, BT2 étant une simple pile-bouton (nous y reviendrons dans l'encadré). La capacité de BT2 dépend de durée pendant laquelle le Climat-Logger doit mémoriser ses données et paramètres, la taille de BT1 dépendant pratiquement uniquement de la durée de fonctionnement que l'on veut obtenir du sous-ensemble de mesure d'humidité.

Le boîtier devra abriter non seulement la platine dotée de ses composants, mais aussi les piles dans leur support, l'embase RS-232 et le bouton-poussoir. Le capteur d'humidité devra être soudé le plus court possible sur K1. Il n'est pas recommandé de le fixer sur l'un des côtés du boîtier si ce positionnement requiert des longueurs de câbles de connexion (relativement) plus importantes. Il va sans dire que le boîtier ne doit pas être étanche vu que sinon le capteur d'humidité se trouve dans l'impossibilité de déterminer l'humidité de l'air.

## Liste des composants

### Résistances :

R1,R7,R14,R15,R18 = 10 kΩ  
 R2,R21 = 100 kΩ  
 R3 = 47 kΩ  
 R4 = 220 Ω  
 R5 = 68 kΩ  
 R6,R11,R13 = 10 MΩ  
 R8 = 470 kΩ  
 R9,R12 = 1 MΩ  
 R10 = 150 kΩ  
 R16,R17,R19,R20 = 1 kΩ  
 P1 = ajustable 250 kΩ vertical  
 P2 = ajustable 2 MΩ vertical

### Condensateurs :

C1 à C3 = cf. texte  
 C4,C5,C8,C10,C11 = 100 nF  
 C6 = 22 µF/16 V radial  
 C7 = 330 nF  
 C9 = 10 µF/16 V radial

### Semi-conducteurs :

D1 = LM385-2.5 (National Semiconductor)  
 D2 = LED 3 mm à faible courant rouge  
 D3 = LED 3 mm à faible courant verte  
 D4,D5 = IN4148

T1,T2 = BC547

IC1 = TS556

IC2 = 78L05

IC3 = MCP6041-I/P (Microchip, chez, par exemple, Farnell n°. 396-8790)

IC4 = DS1616 (Dallas Semiconductor)

### Divers :

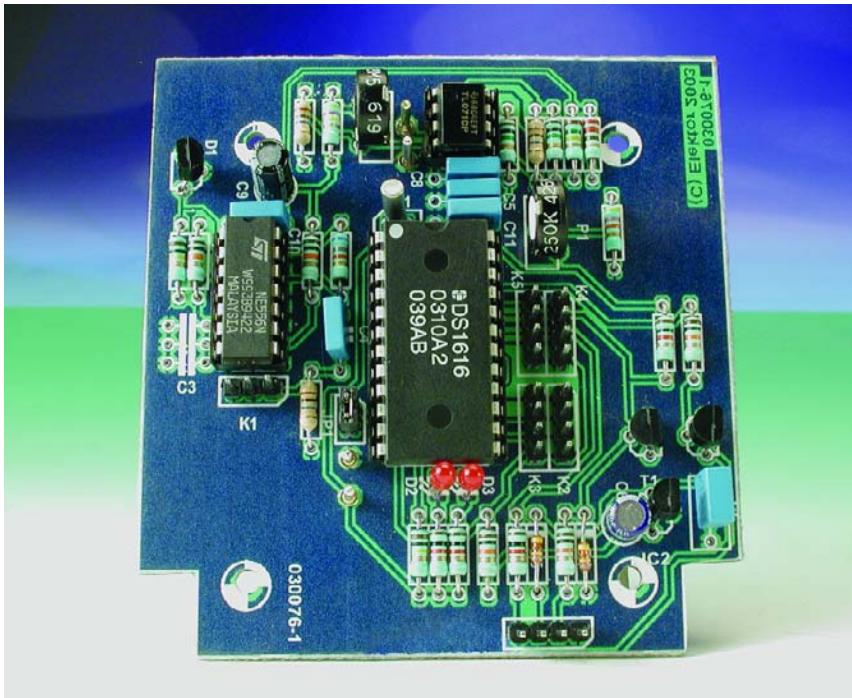
JP1 = bouton-poussoir unipolaire à contact travail  
 K1 = embase auto-sécable à 1 rangée de 3 contacts  
 capteur d'humidité HSI101 (Unitronic)  
 K2 à K6 = embase auto-sécable à 1 rangée de 4 contacts  
 BT1 = pile au lithium 3 V  
 X1 = quartz 32,768 kHz  
 BT2 = 2 piles Lady (R1) ou Mignon (R6) (en fonction du boîtier et du porte-pile)  
 Embase sub-D 9 points femelle en équerre encartable  
 boîtier avec compartiment à pile (par exemple, OKW taille 3 pour piles Mignon, chez, par exemple, RSComponents, n°. 583-218)

## Calibration

Cet étalonnage requiert que l'on dispose de 2 condensateurs de référence (ou de groupes de condensateurs) de 161 et 200 pF. Il est

impératif que ces valeurs soient respectées et vérifiées par mesure ! On commence par connecter le conden-

sateur de 161 pF à l'embase K1. On mesure, à l'aide d'un multimètre, la tension à la sortie du filtre passe-bas



(point nodal R8/P2). La tension d'offset (tension au point nodal R9/P1) doit avoir la même valeur. On devrait trouver, à la sortie de IC3, une tension comprise entre 2 et 7 mV. Il n'est pas possible de régler cette tension parfaitement à 0 V vu qu'il est impossible de modifier la tension d'offset de l'amplificateur opérationnel. On connecte ensuite le second condensateur au bornier K1 et on procède à une nouvelle mesure de la tension à la sortie du filtre passe-bas. La tension devrait présenter une valeur supérieure d'environ 200 mV à la valeur de tension mesurée lors de la précédente étape avec le condensateur de capacité plus faible. Comme la différence de tension en fonction de l'humidité de l'air approche de cette valeur de 200 mV et qu'elle devrait atteindre entre 0 et 2 V, il a été ajouté un circuit d'amplification monté en amplificateur non-inverseur à gain de 10x. L'ajustable P2 pris à l'entrée inverseuse permet un réglage fin. Si l'on mesure maintenant une tension de 2 V à la sortie de l'amplificateur opérationnel c'est que la calibration a réussi.

## Le logiciel

Le logiciel Windows tourne sous Windows2000 et XP; il n'existe pas, pour le moment, d'adaptation pour

Windows98/95. La fenêtre principale permet de procéder à tous les paramétrages; elle présente également les résultats de mesure actuels (**figure 6**). Il est possible, à ce niveau, de sélectionner, voire d'effacer, individuellement les canaux de mesure (*Temperature, ADC1, Humidity/ADC2 et ADC3*). Si le DS1616 ne se trouve pas « en mission » on a –saint est que les canaux de mesure aient été sélectionnés– mesure de la température et de l'humidité une fois par seconde, les résultats étant visualisés par les afficheurs correspondants. Cette mesure est implémentée par le biais d'un temporisateur (*timer*) dont la valeur par défaut a été fixée à 1 s. Sachant que ce temporisateur consomme du temps CPU et que cela pourrait, dans le cas de modèles d'ordinateurs plus anciens, se traduire par des pertes de performances, il est possible de modifier à loisir la valeur du temporisateur par le biais de l'option *Settings/Performance*. La nouvelle valeur est stockée en RAM de manière à ne pas avoir, à chaque redémarrage du programme, à modifier la valeur du temporisateur.

Lorsque le DS1616 démarre une série de mesure, seuls les canaux actifs sont échantillonnés et les informations obtenues mémorisées. La mémoire de stockage des valeurs du

DS1616 possède une capacité de 2 048 octets. Plus le nombre de canaux sélectionnés est important, plus la durée de saisie de mesures maximale diminue. Si l'on ne sélecte qu'un seul canal, il est possible de relever un maximum de 2 048 valeurs de mesure. Si l'on travaille sur 2 canaux, ce nombre passe à 1 024, tombant à 682 pour 3 canaux et 512 pour 4. La fenêtre principale ne visualise que les éléments les plus importants : *Humidity (ADC2)* et *Temperature*. Les autres facteurs peuvent être affichés par le biais de la fonction *ADC Channels>Show Channels*.

### Taux d'échantillonnage

Le taux d'échantillonnage (*sample rate*) indique l'intervalle de temps (en minutes) séparant 2 mesures. Comme certains des registres du DS1616 ont une largeur de 8 bits, le taux d'échantillonnage maximal est de 255 minutes. Il faut en outre, si l'on veut que l'horloge en temps réel fonctionne correctement, ce qui est une condition sine qua non pour pouvoir démarrer une série de mesures, le taux d'échantillonnage doit être égal ou supérieur à 1. Lors d'un changement de piles, le taux d'échantillonnage est remis à zéro (0). Si l'on a ensuite remis l'horloge à l'heure mais que l'on a oublié de paramétriser le taux d'échantillonnage à 1 l'horloge en temps réel ne fonctionne pas correctement.

### Démarrer Mission (Start Mission)

Ce bouton permet de démarrer une série de mesures. Il faut bien entendu qu'il y ait un (ou plusieurs) canal actif et un taux d'échantillonnage convenable –des erreurs à ces niveaux sont annoncées par un message (*popup*). L'actualisation des valeurs de température et d'humidité ne se fait plus une fois par seconde mais à l'intervalle de temps (*sample rate*) paramétré.

### Numéro de série (Serial Number)

Comme nombre de composants de Dallas Semiconductor, le DS1616 possède un numéro de série qui lui est unique, code qui, lors de graphiques de mesures, est sauvegardé lors de chaque instruction de sauvegarde (*save*) de manière à pouvoir réattribuer des données de mesure individuellement à chacun des modules.

### Seuils de température

#### (Temperature Threshold)

Ces valeurs de seuil sont mises à contribution pour positionner (mettre à « 1 ») l'indicateur (*flag*) du registre Status 1 en cas de dépassement dans un sens ou dans l'autre de la température paramétrée. Ce positionnement se traduit par la mise au niveau bas (« 0 ») de la sortie d'interruption INT. Ce signal est appli-

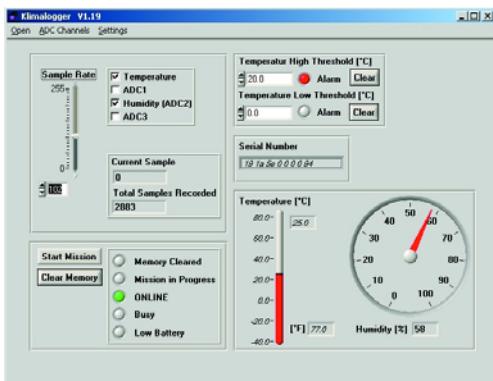


Figure 6. La fenêtre principale visualise tous les paramètres importants.

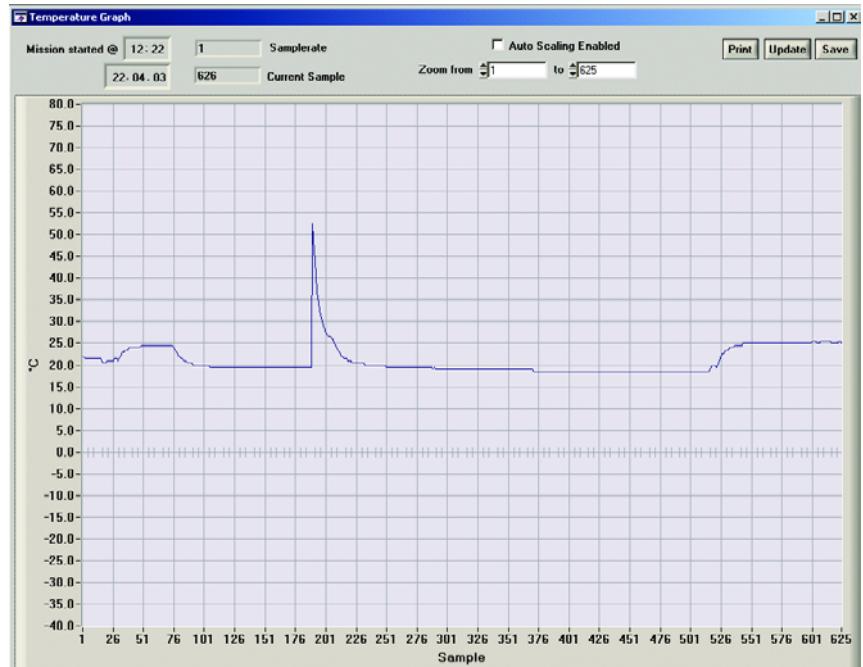


Figure 7. L'évolution des valeurs de mesure (température, humidité, voire les 2 entrées libres) pourra prendre la forme d'un graphique ou d'un tableau.

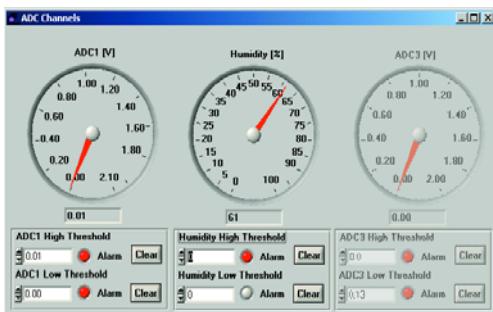


Figure 8. Les 3 entrées analogiques externes sont visualisées sous l'option ADC Channels.

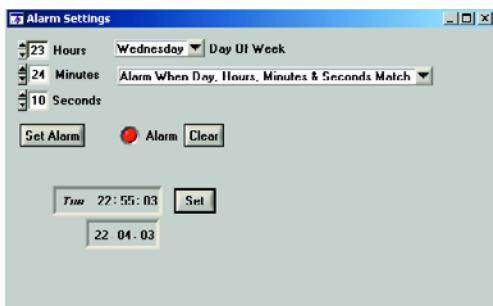


Figure 9. Le paramétrage de l'alarme (Alarm Settings) transforme le Climat-Logger en chronocommutateur.

qué aux embases K2/K3 et peut être utilisé pour, par exemple, produire un signal d'alarme. Il faut exécuter une instruction *Clear* pour effacer l'indicateur d'alarme.

## Température et autres graphiques

Cette fenêtre visualise les valeurs de température mesurées (**figure 7**). L'échantillon 1 (*Sample 1*) correspond au point identifié dans *Mission started @*. Il est possible ainsi d'at-

tribuer à chaque mesure mémorisée un tampon horodateur (date/heure) bien défini. Nous n'avons pas opté pour une remise à jour (*update*) automatique du processus pour éviter de consommer inutilement de temps CPU. On obtient une remise à jour soit par une réouverture de la fenêtre ou par action sur le bouton *<Update>*.

La fonction *Auto Scaling* permet un affichage clair de l'évolution de la température. Les limites ne sont plus de -40 et +80 °C, mais dépendent des valeurs minimale et maximale stockées dans la mémoire de données. L'instruction *Save* sauvegarde les valeurs de mesure en local sur le disque dur, l'utilisateur pouvant définir à loisir le cheminement (*path*) à utiliser.

Les 3 courbes Humidity, ADC1 et ADC3 présentent une identité tant au niveau du mode de fonctionnement que de leur aspect. Dans les versions ultérieures du programme, les 2 canaux encore libres pour le moment seront eux aussi utilisés.

## Autres fenêtres

On arrive dans la fenêtre *ADC Channels* par le biais de l'option *ADC Channels>Show ADC Channels*

du menu principal. Elle fournit des informations concernant les valeurs de mesure actuelles des entrées analogiques. Les entrées inactives sont, comme on le voit sur la droite de la **figure 8**, visualisées dans leur état d'inactivité (en grisé). Pour les entrées analogiques il est également possible de définir des valeurs de seuil. Leur fonctionnement est identique à celui décrit dans les seuils de température.

La fenêtre *Alarm Settings* (**figure 9**) comporte un certain nombre de paramètres ayant trait à l'horloge en temps réel (RTC).

L'horloge en temps réel est mise à l'heure-système du PC par le biais de la fonction *Set*. Les autres possibilités de paramétrage pourront servir, par exemple, à implémenter un chrono-programmateur. Lorsque l'on a concordance de l'heure de l'horloge avec l'heure d'alarme entrée, on a génération d'une interruption sur la ligne *INT*. On ne pourra modifier l'heure et les fonctions d'alarme qu'à condition que le DS1616 ne soit pas en train d'effectuer une « mission ». Il est possible, par le biais de l'option *Settings/Performance*, de paramétrier l'intervalle temporisateur, à la base de l'interrogation régulière de tous les capteurs et des entrées analogiques.

## Calcul de la durée de vie de la pile

La pile de sauvegarde du type CR1620 (3 V, 60 mAh) sert uniquement à alimenter le DS1616 et le capteur de température qu'il intègre. L'électronique de mesure d'humidité est elle, pour disposer d'une durée de fonctionnement acceptable, alimentée par 4 piles Mignon en raison de sa consommation de courant plus importante. Dans ces conditions la durée de vie de la pile-bouton dépend du taux d'échantillonnage choisi. Il suffit, pour avoir une idée de la durée de vie potentielle de la pile, de diviser sa capacité par la consommation de courant moyenne ( $I_{avg}$ ) du circuit.

On aurait ainsi, par exemple, dans le cas d'un taux d'échantillonnage de 60 s, une durée de vie de pile = capacité de la pile /  $I_{avg}$  formule dans laquelle le courant moyen

$$I_{avg} = (t_{tc} \cdot i_{tc} + (T - t_{tc}) \cdot i_{osc})/T$$

avec :

$t_{tc}$ , la durée d'une mesure = 150 ms

$i_{tc}$ , la consommation de courant au cours d'une mesure de température = 0,5 mA

$i_{osc}$ , la consommation de courant au repos = 450 nA

$T$ , le taux d'échantillonnage = 60 s

Cela nous donne une durée de vie de près de 4 ans.

giques, à une valeur quelconque. Plus la valeur adoptée est élevée, plus l'intervalle séparant la prise de 2 échantillons devient important. Cette augmentation de l'intervalle permet à la CPU de passer moins de temps à l'exécution du programme. Il est possible, par le biais de la fonction *System Monitor* de Windows, se suivre exactement la charge CPU requise et trouver, par le biais du paramètre Performance, une valeur optimale pour le dit intervalle. Les PC de bureau et les ordinateurs portables disposant de suffisamment de RAM et d'un processeur récent ne devraient cependant pas avoir de problème avec un intervalle temporisateur d'une seconde. Le point de menu Help explique (en anglais) le mode de travail du *System Monitor*.

### Mise en oeuvre

Le Climat-Logger est connecté à un port série libre du PC ou du portable à l'aide d'un câble série ordinaire (**pas** de câble zéro modem (*null modem*) et **pas** de lignes croisées donc). Il faut, lors du premier lancement du programme du Logger, indiquer le port COM utilisé. Ce paramètre est mémorisé et n'aura plus à être modifié ultérieurement sauf bien

entendu si l'on envisage de connecter le Logger à un port COM différent de celui paramétré précédemment.

Dans le cas d'un DS1616 n'ayant encore jamais été utilisé, la RAM est remplie de 0xFF. Il faudra, la première fois que l'on connecte le Climat-Logger au PC, mettre l'horloge à l'heure et définir un taux d'échantillonnage cohérent. Ce double paramétrage devra être repris après chaque interruption de la tension fournie par la pile de sauvegarde (*backup*) du DS1616 (c'est-à-dire lors de tout remplacement de BT2). Tant que l'on n'a pas paramétré de taux d'échantillonnage, le PC lit une valeur de mesure par seconde, la remise à jour de l'affichage à l'écran se faisant elle une fois par minute.

Tant que la centrale d'acquisition de données reste reliée au PC, il est possible de jouer sur et de modifier le taux d'échantillonnage et les autres paramètres en fonction des impératifs de la mesure. Une fois déconnectée du PC, la centrale conserve les derniers paramètres qui lui ont été fournis et continue de travailler en mode autonome. Comme nous le disions plus haut, ces paramètres ne sont perdus qu'en cas d'interruption de la tension de la pile de sauvegarde alimentant le DS1616.

Avec la mouture la plus récente du programme disponible lors de l'écriture de ces lignes, la version V1.23, il est également possible d'envoyer le DS1616 « en mission » par le biais du cavalier JP1. Voilà comment procéder :

Sur le PC, on configure, de la manière indiquée dans le paragraphe « Le logiciel », le taux d'échantillonnage, les canaux de mesure et le reste des paramètres. On clique ensuite sur « *Enable External Start* » et on efface le contenu de la mémoire –la LED « *Memory Cleared* » s'allume à l'écran. Il est possible maintenant de déconnecter le Logger du PC et de le transporter sur le « lieu de l'action », là où doivent se faire les mesures. Une fois sur place, il suffit d'appuyer pendant au moins 0,5 s sur le bouton-poussoir pour démarrer la mission; dès lors les valeurs de mesure sont prises en compte et mémorisées.

Le fait que le Logger soit encore relié au PC n'interdit pas de démarrer une série de mesures par action sur le bouton-poussoir. Il n'est pas impératif dans ce cas-là de fermer le logiciel, par conséquent pourra rester ouvert. Le programme vérifiera s'il y a eu démarrage d'une mission.

Si l'on modifie l'un des paramètres (canal de mesure ou taux d'échantillonnage) après action sur le bouton « *Enable External Start* », le bouton « *Enable External Start* » sera automatiquement remis en attente. On l'aura compris, il ne faudra activer l'option « *Enable External Start* » qu'après avoir réellement terminé la totalité du paramétrage. Car c'est à ce moment-là qu'a lieu la mémorisation dans le DS1616 du taux d'échantillonnage et du canal de mesure. S'il devait se faire que l'on modifie a posteriori un paramètre sur le PC, ce changement ne sera pas transmis au DS1616 de sorte qu'il n'y aura pas concordance entre le paramétrage du Logger et les informations correspondantes visualisées sur l'écran du PC.

(030076)

## Téléchargements

pour le présent projet :

Dessin de la platine (EPS030076-1) au format .pdf et logiciel pour WIN2000/XP (EPS030076-11) à l'adresse :

[www.elektor.fr/dl/dl.htm](http://www.elektor.fr/dl/dl.htm)

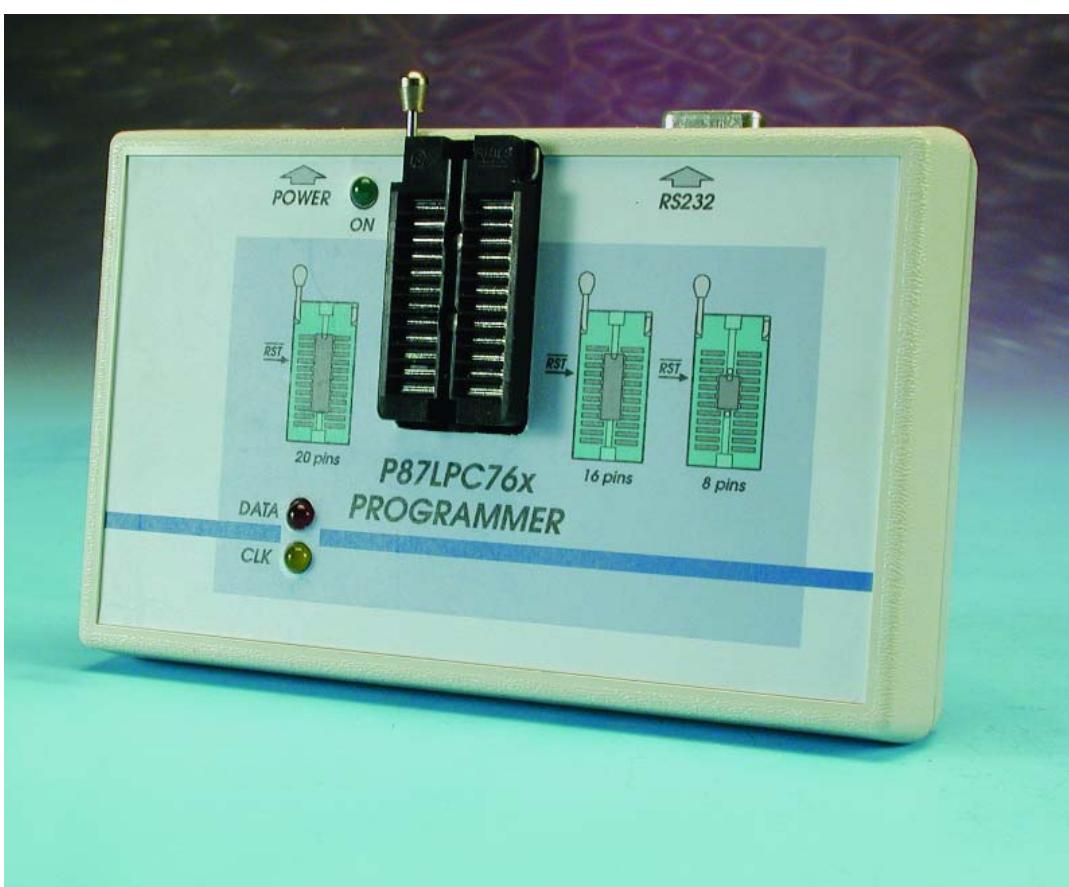
On pourra également télécharger le logiciel depuis le site Web de l'auteur à l'adresse <http://chripo.icb.at/Rooney>. On trouvera à cet endroit les informations les plus récentes (très peu en anglais majorité en allemand) concernant de projet et d'autres projets de l'auteur.

# Programmateur de P87LPC76x

Une électronique simple pour une famille de µC populaire

Peter Luyckx

Le programmateur décrit dans le présent article est conçu spécifiquement pour les P87LPC76x, une famille de microcontrôleurs 8 bits de chez Philips, descendant du 8051 et qu'il est possible de programmer sérielement. Le matériel nécessaire est remarquablement simple, le logiciel nécessaire étant lui disponible au téléchargement gratuitement depuis notre site Internet.



La grande popularité des microcontrôleurs 8 bits de la famille P87LPC76x tient au fait de leur universalité. La palette de leurs applications est très riche. Cela tient, entre autres, à leur taille compacte, au nombre important de sorties de commande, à leur courant de sortie on ne peut plus raisonnable et, last but not least, à la présence d'un bus I<sup>2</sup>C. Il est possible de programmer sérielement, de sorte qu'il est également possible de le programmer *in situ* (au cœur même du montage). Cette caractéristique présente un attrait indéniable pour les fabricants d'appareils vu qu'elle leur permet de mettre la dernière version de leur progiciel (*firmware*) dans le contrôleur dans leur

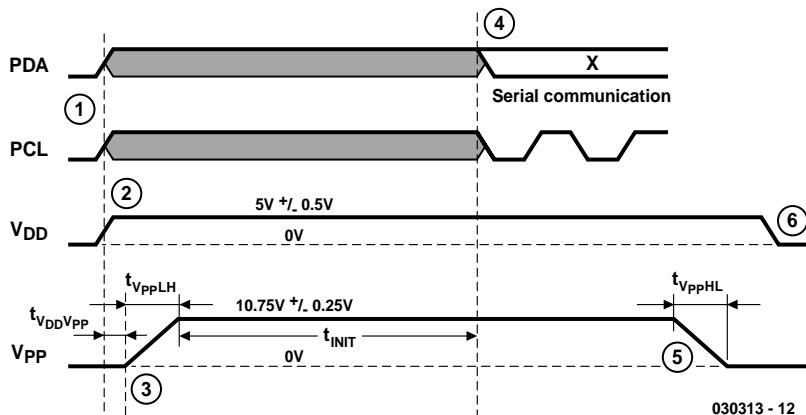


Figure 1. Chronodiagramme du mode de programmation.

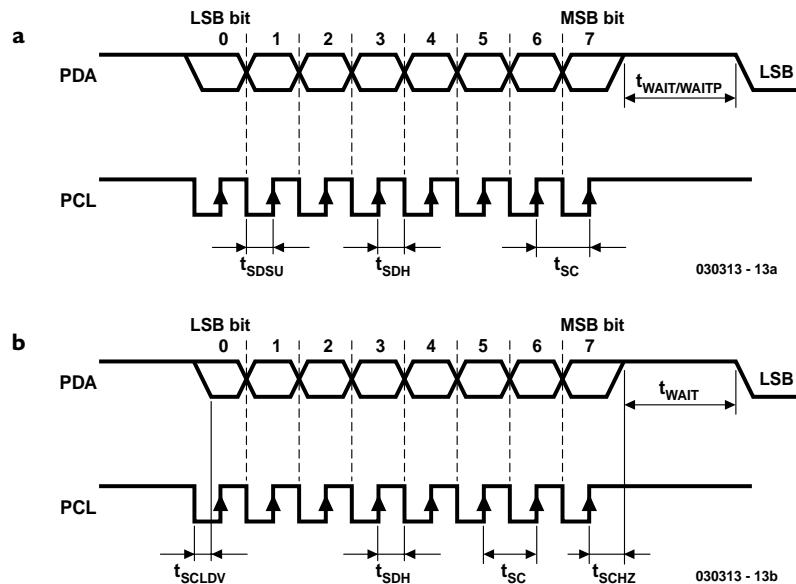


Figure 2. La chronologie lors de la programmation (a) et lors de la lecture du microcontrôleur (b).

appareil pour ainsi dire juste avant de fermer la boîte d'emballage. Ce type de contrôleur présente cependant un inconvénient : bien qu'ils soient dotés d'EPROM ils ne possèdent pas de fenêtre et sont du type OTP (One Time Programmable) et ne peuvent partant être programmés qu'une seule et unique fois.

Note : si donc vous avez un montage à mettre au point il est préférable de le développer sur une autre carte dotée de Flash telle que notre carte à 89S8252 Flash (en raison de sa compatibilité 8051) et une fois que tout fonctionne correctement, de programmer le 76x. L'avantage de cette

approche est que le cycle téléchargement/test est plus rapide sur la carte Flash vue qu'elle est programmable en circuit.

Le montage proposé ici est piloté par le biais du port série d'un PC. Il peut sembler, à première vue, que l'idée d'utiliser le port série soit quelque peu dépassée, mais l'une des raisons principales de ce choix est de limiter au strict minimum le coût du matériel nécessaire. Le choix d'une approche via l'interface USB se traduirait inévitablement par un surcoût de 10 à 15 € en raison de la présence d'un indispensable contrôleur USB.

Comme ne manquent pas de prouver le schéma et, plus loin, le dessin de la platine, le matériel nécessaire est extrêmement simple et la réalisation d'une très grande simplicité. Inutile de rechercher un composant exotique, le montage n'en comporte pas. Le logiciel pour PC requis par le programmeur est écrit en Visual Basic et pourra être téléchargé gratuitement depuis notre site Internet ([www.elektor.fr](http://www.elektor.fr)). Si vous ne disposez pas d'un accès à Internet, il vous reste la possibilité de commander la disquette (EPS030313-11) auprès des adresses habituelles.

## La programmation

La mise du circuit intégré en mode de programmation requiert le respect d'un certain protocole, environnement qu'illustre le chronodiagramme de la **figure 1** :

- Il faut commencer par mettre les broches PDA et PCL en flottement.
- On peut ensuite appliquer la tension d'alimentation de +5 V; le circuit intégré ne supporte pas d'être inséré tension d'alimentation présente (*hot insert*).
- après 20 µs, on peut appliquer la tension de programmation VPP à la broche prévue à cet effet. Cette tension doit avoir un temps d'établissement compris entre 1 et 100 µs.
- Le circuit intégré se trouve maintenant en mode de programmation. Il faut encore attendre de l'ordre de 60 µs avant que les lignes PCL et PDA puissent être sorties de leur état de flottement et que l'on puisse passer à la programmation.
- La sortie du mode de programmation est moins critique elle; tout ce que l'on exige est que la tension de programmation ait un temps de chute compris, à nouveau, entre 1 et 100 µs, la tension d'alimentation devant être coupée ensuite.

Lors de la programmation, les instructions en question sont envoyées vers l'interface série du contrôleur par le biais des lignes PCL et PDA. Chaque instruction ou chaque octet de programme comporte 8 bits (logique pour un octet allez-vous dire). Après 8 cycles d'horloge les 8 bits ont été transférés dans le processeur et l'octet complet se trouve dans le registre de réception. L'interface série du P87LPC76x est identique à l'UART compatible 8051 travaillant en mode 0.

Le changement de donnée se fait sur le flanc descendant du signal d'horloge, son transfert vers le microcontrôleur se faisant lui sur un flanc montant. Le premier bit est le bit de poids faible (LSB = Least Significant Bit), le bit de poids fort (MSB = Most Significant Bit) arrivant, logiquement, en dernier. Ce mode opératoire est utilisé tant lors de la program-

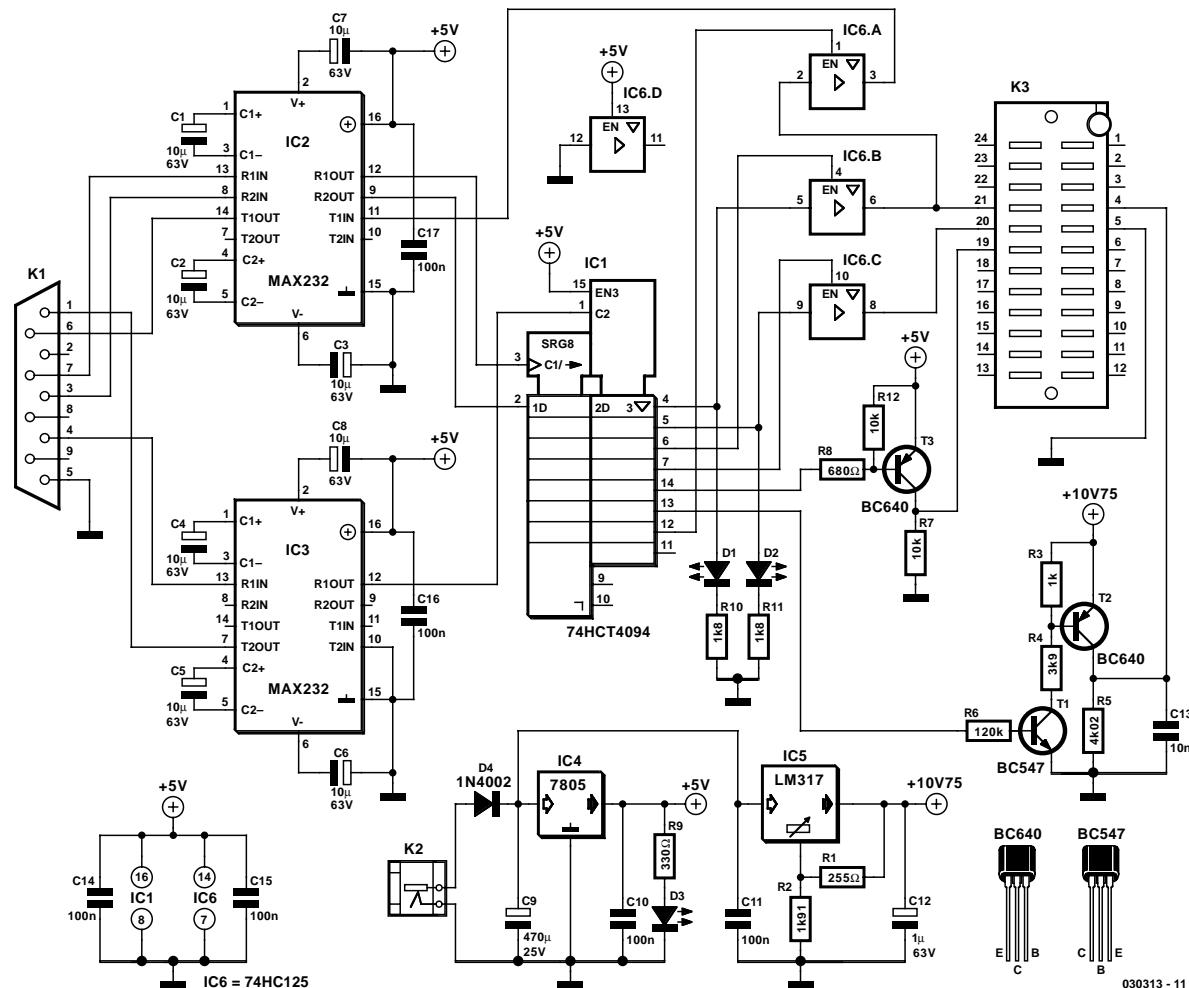


Figure 3. L'électronique du programmeur est étonnamment simple.

mation que lors de la lecture du contenu du microcontrôleur. Lors de la lecture de données, le premier bit est validé 40 ns après. Point important à noter : lors de la lecture du dernier bit, la broche PDA est mise, 40 ns après le dernier flanc montant de l'horloge, à l'état de haute impédance (*tri-state*) ; cela signifie partant qu'il faut échantillonner avant le dernier flanc d'horloge.

Il faut attendre un minimum de 2  $\mu$ s entre 2 instructions; dans le cas de **strt\_prgm** il faut attendre au minimum 250  $\mu$ s.

Pour vous permettre de mieux comprendre la chronologie, la **figure 2a** donne le chronogramme lors de la programmation, alors que la **figure 2b** représente celui de la lecture.

## Le matériel

Comme vous l'apprend un coup d'oeil, aussi rapide soit-il, au schéma représenté en **figure 3**, l'électronique concernée n'a rien de bien compliqué. Quelques circuits intégrés, une triplete de transistors, quelques résistances et condensateurs, voilà la recette de ce projet.

### Registre à décalage

IC1 constitue le cœur de ce programmeur. Il s'agit d'un double registre à décalage. Il y a, dans le premier registre, place pour 8 bits transférés séquentiellement au rythme de la fréquence d'horloge. Par l'envoi d'une impulsion d'échantillonnage (*strobe*) le contenu complet de ce registre à décalage est déplacé en parallèle (c'est-à-dire d'un seul coup) vers le second registre à décalage dont les sorties vont elles vers l'extérieur. Il est partant possible de transférer 8 bits vers le registre à décalage au rythme de l'horloge sans qu'il ne se passe rien au niveau des sorties.

Ce changement d'état ne prend place qu'après mise au niveau haut de l'entrée d'échantillonnage (broche 1). Cette caractéristique tombe à pic vu que nous avons besoin, sur notre montage, de pouvoir utiliser simultanément 7 sorties

de commande, alors que l'interface série ne dispose que de 3 sorties. Voici, en allant du bit de poids faible (LSB) au bit de poids fort (MSB), l'ordre d'attribution des fonctions aux différentes broches de sortie :

- Ligne de donnée pour le microcontrôleur à programmer
- Ligne de signal d'horloge pour le microcontrôleur
- Ligne de commande pour la mise à haute impédance de la ligne de données (active au niveau bas)
- Ligne de commande pour la mise à haute impédance de la ligne d'horloge (active au niveau bas)
- Ligne d'alimentation du microcontrôleur
- Ligne de la tension de programmation du microcontrôleur
- Ligne de commande pour la mise à haute impédance de la ligne de lecture direction PC (active au niveau bas)
- Le bit de poids fort du registre à

décalage n'est pas utilisé.

Il faut tenir compte du fait que lorsque l'on transfert un nombre du PC vers le registre à décalage, le bit de poids faible côté PC correspond au bit de poids fort côté registre à décalage. De plus, l'envoi d'un octet de donnée vers le microcontrôleur requiert 128 impulsions d'horloge. Cela tient au fait qu'il faut, par bit, transférer deux fois un octet dans le registre. Le seul bit de cet octet à changer est le bit d'horloge, tous les autres bits restent constants. Il faut partant que le registre à décalage reçoive  $2 \times 8$  soit 16 impulsions d'horloge pour chaque bit pris en compte par le microcontrôleur. Cette opération se répète 8 fois pour chaque octet de donnée, d'où nos 128 impulsions.

#### *Commande des tensions d'alimentation et des tampons*

Lors du passage au mode de programmation il faut, nous le disions plus haut, mettre les lignes de données et d'horloge soit à haute impédance soit en flottement, jusqu'à l'application des tensions d'alimentation et de programmation. C'est là la tâche de IC6, un quadruple tampon 3 états (*tri-state*) dont les sorties se trouvent à « 0 », « 1 » ou en flottement, cela en fonction de la tension appliquée à l'entrée de commande. Il faut, pour obtenir une sortie flottante, appliquer un signal de niveau haut à l'entrée.

Les tensions d'alimentation et de programmation nécessaires au microcontrôleur sont appliquées chacune au travers d'un étage à transistor vu qu'un circuit intégré CMOS est incapable de fournir un courant supérieur à 4 mA environ. La tension d'alimentation ne requiert qu'un étage simple à base de transistor PNP, T3. L'étage chargé de fournir la tension de programmation est un peu plus complexe, vu qu'il faut une adaptation de niveau de 5 vers 10,75 V. Pour cela, on commence par réaliser, à l'aide de T1, une sortie à collecteur ouvert, le transistor T2 servant à la commutation de la tension de programmation. Il serait impossible, en l'absence de T1, de mettre T2 complètement en blocage. C13 se charge de faire en sorte que la tension de programmation s'éta-

blisse avec un flanc compris entre 1 et 100 µs, tout en éliminant les parasites HF dont on se passerait bien. Les LED D1 et D2 visualisent, respectivement, le signal de données et le signal d'horloge. Il faudra, étant donné le peu de courant disponible, impérativement utiliser des LED de type faible courant.

#### *De RS-232 vers TTL*

La conversion de niveaux RS-232 vers des niveaux TTL fait appel à des composants d'interface bien connus, des MAX232, IC2 et IC3. Ce circuit intégré comporte une paire de pompes de charge qui se chargent, c'est bien le cas de le dire, de convertir le niveau de +5 en niveau de + et -10 V qu'exige le standard RS-232. Comme nous avions affaire à 3 entrées de commande nous avons opté, pour des raisons de disponibilité, pour une paire de MAX232.

#### *L'alimentation*

On pourra, pour alimenter le programmeur, tout simplement connecter, au bornier K2, un adaptateur secteur non réglé fournit une tension de 15 V. Notons qu'un adaptateur 12V quelconque fera sans doute également l'affaire vu que ce type de module fournit en règle générale, à charge faible, entre 15 et 17 V.

Un régulateur de tension intégré tri-pode 5 V, IC4, se charge de la régulation de la tension. La régulation de la tension de programmation de 10,75 V fait appel elle à un régulateur ajustable du type LM317. La diode D4 fait office de protection contre une inversion de polarité, la LED D3 servant d'indicateur marche/arrêt.

## **Le logiciel**

L'universalité du logiciel est son atout majeur. Il est constitué d'une série de routines diverses, ce qui offre la possibilité aux amateurs d'expérimentation, de modifier ou d'adapter l'un ou l'autre module à leurs besoins propres. Il est possible par exemple, sans trop de problème, d'implémenter une autre interface dans le logiciel; tout ce qu'il faudra modifier est la routine de sortie.

Nous ne pouvons entrer dans le détail de toutes les routines. Nous allons

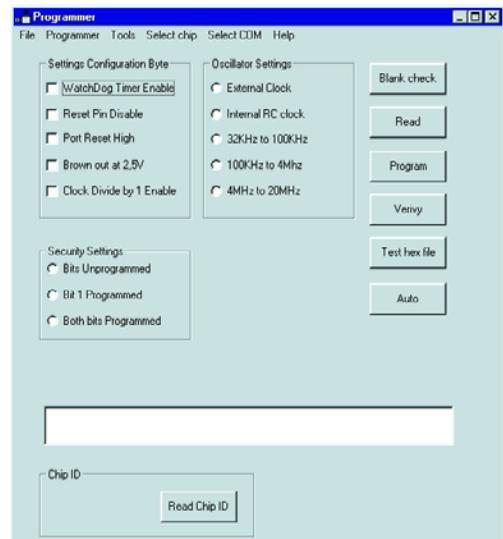


Figure 4. Recopie d'écran de l'interface utilisateur.

nous limiter à la description de la coquille du logiciel, son interface utilisateur et à sa mise en oeuvre.

La **figure 4** propose une recopie d'écran de la fenêtre utilisateur telle qu'elle apparaît sur l'écran du PC. Passons en revue les fonctions des différents boutons présents sur le côté droit de la fenêtre :

- « *Blank check* » : Sert à vérifier que le composant à programmer est vierge. N.B. : Vierge signifie que tous les emplacements de mémoire se trouvent à FF<sub>HEX</sub> (ils sont donc tous à « 1 »). La programmation peut faire un « 0 » de ce « 1 ».
- « *Read* » : Cette fonction sert à la lecture du contenu du circuit intégré.
- « *Program* » : Par le biais de cette fonction on programme le circuit en y transférant un fichier ouvert auparavant.
- « *Verify* » : Une action sur ce bouton démarre une routine qui vérifie le microcontrôleur après sa programmation; en cas d'erreur de programmation, incohérence entre le contenu du composant et le fichier à transférer, un message le signale.
- « *Test hex file* » : Sert à vérifier l'absence d'erreur au niveau de la somme de vérification (*checksum*) du fichier .hex et que la taille du fichier .hex en question ne dépasse pas la quantité de mémoire disponible dans le circuit intégré concerné.
- « *Auto* » : Par cette fonction, on exécute automatiquement toutes les fonctions énumérées plus haut (exception faite de « *Read* »).
- « *Read chip ID* » : Grâce à cette fonction il est possible de connaître le type de P87LPC76x en face duquel on se trouve, ce

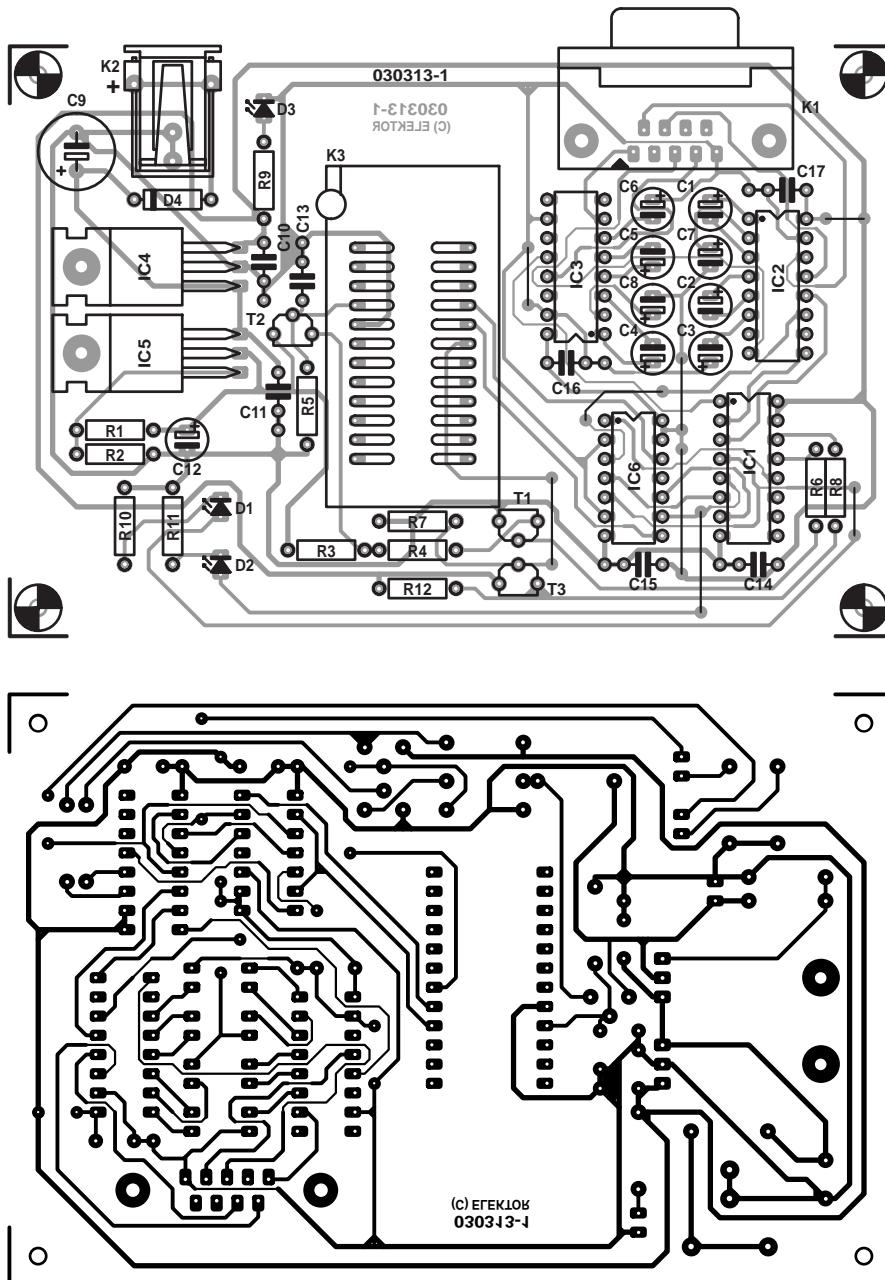


Figure 5. Dessin des pistes et sérigraphie de l'implantation des composants de la platine dessinée à l'intention du programmateur.

qui est d'une importance capitale lorsque l'inscription sur le circuit intégré lui-même est devenu illisible. Le résultat de cette interrogation apparaît dans la fenêtre de texte sur la gauche du bouton.

#### Paramètres

On peut, dans l'encadrement situé en haut à gauche, cocher des paramètres pour les octets de configuration. À côté à droite, on trouve les paramètres de l'oscillateur. Notons qu'il n'est possible, au niveau des paramètres « oscillator settings », de sélectionner qu'un seul paramétrage. Cette remarque vaut également

pour le bloc « *Security settings* » au milieu de la fenêtre utilisateur. Indiquons, pour être complet, que la routine de sécurité se charge de la valeur de l'octet de protection et qu'il y est fait appel à chaque fois que l'on a besoin de cet octet. Dans ces conditions c'est toujours la valeur la plus récente qui se trouve en mémoire.

Nous trouvons, entre le bloc « *Security settings* » et « *Chip ID* » une fenêtre de texte qui reproduit les

## Liste des composants

### Résistances :

R1 = 255  $\Omega$   
 R2 = 1k $\Omega$ 91  
 R3 = 1 k $\Omega$   
 R4 = 3k $\Omega$ 9  
 R5 = 1k $\Omega$ 5  
 R6 = 47 k $\Omega$   
 R7,R12 = 10 k $\Omega$   
 R8 = 680  $\Omega$   
 R9 = 330  $\Omega$   
 R10,R11 = 1k $\Omega$ 8

### Condensateurs :

C1,C2,C3,C4,C5,C6,C7,C8 = 10  $\mu$ F/63 V radial  
 C9 = 470  $\mu$ F/25 V radial  
 C10,C11,C14 à C17 = 100 nF  
 C12 = 1  $\mu$ F/63 V radial  
 C13 = 22 nF

### Semi-conducteurs :

D1 = LED à faible courant rouge  
 D2 = LED à faible courant jaune  
 D3 = LED verte  
 D4 = IN4002  
 T1 = BC547B  
 T2,T3 = BC640  
 IC1 = 74HCT4094  
 IC2,IC3 = MAX232 CP (Maxim)  
 IC4 = 7805 (TO220)  
 IC5 = LM317T (TO220)  
 IC6 = 74HC125

### Divers :

K1 = embase Sub-D 9 points femelle encartable  
 K2 = jack d'alimentation encartable  
 K3 = support FIN à 24 contacts large  
 8 ponts de câblage  
 adaptateur secteur 15 V (12 V)/300 mA  
 câble serial I:1  
 boîtier 145x90x30 mm tel que, par exemple, PacTek

chaînes hexadécimales programmées ou lues.

#### Barre de tâches

– La fonction « *File* » dotées des options « *Open file* », « *Close file* », « *Save as* » et « *Exit* » n'appelle guère de commentaire pour l'utilisateur chevronné de Windows que vous êtes.

– « *Programmer* » : cette fonction a été ajoutée pour pouvoir travailler avec le logiciel sans que le programmeur ne soit connecté au

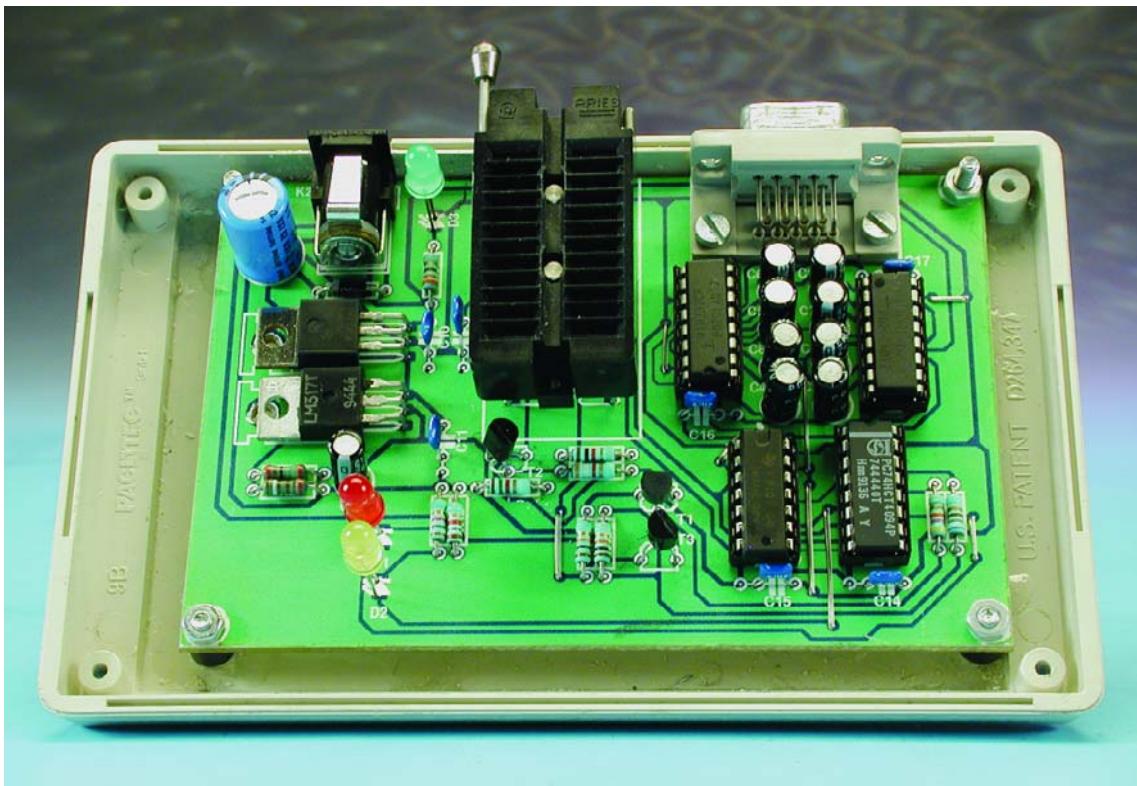


Figure 6. Exemplaire terminé du programmeur de P87LPC76x.

PC. L'option « *Online* » permet d'activer le programmeur, l'option « *Offline* » permettant, logiquement, de le désactiver.

- « *Tools* » : cette fenêtre à menu déroulant visualise les mêmes boutons que les boutons à accès direct de la fenêtre utilisateur, exception faite du bouton « *Read Chip ID* ».
- « *Select chip* » : Il est possible, de choisir, dans la liste des circuits disponibles, le type de composant à programmer. À noter que cette opération se fait automatiquement si l'on opte pour « *Read Chip ID* » ou « *Auto* ».
- « *Select com* » : On peut, par le biais de cette fenêtre à menu déroulant, choisir l'un des 4 ports sériels.
- « *Help* » : Avec l'option « *Index* » on ouvre un fichier d'aide (en anglais) qui donne une description succincte pour chacune des fonctions du programme; l'option « *About* » donne des informations concernant le programme, le copyright et l'auteur.

#### Fonction additionnelle

Il est une autre fonction qui n'apparaît pas dans la fenêtre utilisateur. Il

est en effet possible de stocker dans un fichier de configuration les paramètres concernant les octets de configuration, le composant et le port série. Voilà comment procéder : à chaque clôture du programme, on voit apparaître un message demandant si l'on veut mémoriser le paramétrage utilisé. Si l'on opte pour « *Yes* » les différents paramètres sont stockés dans un fichier ASCII baptisé *settings.pl* et placé sur le disque dur C:\>. Lors du lancement suivant du logiciel, le programme demande si l'on veut utiliser le paramétrage sauvegardé la fois précédente. On a le choix entre « *Yes* » et « *No* ». Si l'on choisit l'option négative, on démarre une session sans antécédent. En cas de « *Yes* » alors que le programme en est à sa première utilisation, on verra apparaître un message d'erreur.

## La réalisation

La **figure 5** vous propose le dessin des pistes et la sérigraphie de l'implantation des composants de la platine dessinée à l'intention de cette réalisation. Comme on peut le constater, ce circuit imprimé est à la fois compact et très lisible. Le sup-

port FIN (à Force d'Insertion Nulle, ZIF pour Zero Force Insertion en anglais) destiné à recevoir le composant à programmer est placé pratiquement au centre de la platine. Cela est encore plus vrai sur la version définitive que sur le prototype de la photo. Les embases K1 (RS-232) et bornier K2 (adaptateur secteur) ont été placées à dessein sur le bord de la platine de manière à faciliter l'accès. Il n'est pas nécessaire, en raison de la consommation de courant relativement faible, de l'ordre de 50 mA

du circuit, de doter les régulateurs de tension IC4 et IC5 d'un radiateur; ils pourront être couchés sur la platine.

L'étape de soudage ne devrait pas poser de problème. Si tant est que l'on respecte au pied de la lettre la sérigraphie et la liste des composants, il y a peu de risque que les choses se passent mal. Attention à n'oublier aucun des 8 ponts de câblage, car sinon le risque de dysfonctionnement est grand... La photo de la **figure 6** vous montre à quoi devrait ressembler votre platine, une fois terminée.

Étant données les dimensions compactes de la platine, vous ne devriez pas avoir de difficulté à mettre la main sur un boîtier adéquat. Nous avons mis notre prototype dans un coffret PacTek de 145 x 90 x 30 mm. Comme le montre la photo en début d'article, nous avons dessiné une face avant donnant un aspect plus fini à notre programmeur. Si vous êtes intéressé par ce dessin, vous pouvez le télécharger sous la dénomination **EPS030313-F** depuis notre site Internet sis à l'adresse : [www.elektor.fr](http://www.elektor.fr).

En guise de conclusion, une dernière remarque qui a son importance : utilisez pour la connexion du programmeur au PC un câble de liaison 1:1 RS-232 et surtout pas un câble modem zéro (*null modem*).

À découvrir :

# Le moteur pas à pas

2<sup>ème</sup> partie : la commande unipolaire à 4 canaux

Timothy G Constandinou

Après avoir exposé les généralités sur les moteurs pas et pas et les manières de les mettre à l'ouvrage, il est temps, dans cette seconde partie, de se constituer un système de commande avec l'électronique d'interface pour le piloter par PC.



Nous allons donc voir en détail comment construire, vérifier et faire fonctionner une commande de moteur pas à pas à quatre canaux que vous pourrez adapter à la mesure de vos réalisations personnelles. Le projet comporte une interface RS-232 pour une communication directe avec le PC, un langage

évolué capable de faire exécuter les commandes adressées au contrôleur et enfin l'électronique de puissance pour fournir l'énergie au moteur. Et nous n'oublierons pas les tuyaux utiles pour adapter correctement le logiciel à vos besoins spécifiques. Ce

logiciel est compatible avec toute unité gérée sous Windows à 32 bits, il a été conçu en Borland Delphi.

## L'interface sérielle RS-232

La norme d'interface sérielle RS-232, vieille de plus de quatre décades, conserve les faveurs des utilisateurs d'ordinateur personnel, pour autant que la communication ne demande qu'une bande passante étroite. Comme la quasi-totalité des PC sont équipés de cette interface et que de nombreux microcontrôleurs en disposent également ou du moins sont capables de s'y raccorder, le port RS-232 reste bien pratique pour assurer une liaison directe avec les projets que l'on construit soi-même.

Facile à reconnaître, avec sa prise sub D généralement à 9 broches, référencée COM1 ou COM2, le port série totalise aussi neuf lignes de connexion, mais on peut se contenter de trois, (broches 2, 3 et 5) pour exploiter une voie en alternat (*half-duplex 2-way*). La **figure 1** reprend le brochage complet de ce port. Contrairement aux niveaux normés TTL, les signaux RS-232 sont bipolaires : entre +3 V et +25 V, ils représentent un « 0 » logique, entre -3 V et -25 V, un « 1 » logique. Cette disposition permet le transfert de données à

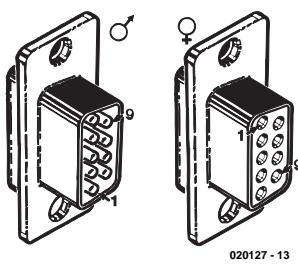


Figure 1. Brochage du port RS-232.

Broche	Signal
1	Data Carrier Detect (DCD)
2	Received Data (Rx)
3	Transmitted Data (Tx)
4	Data Terminal Ready (DTR)
5	Signal Ground (SG)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

relativement longue distance, mais réclame une interface électronique supplémentaire pour convertir les niveaux TTL en tensions RS-232 et

vice versa.

Côté ordinateur, communiquer avec un port RS-232 est aussi simple que de manipuler un fichier de données

sur disque. Chaque port a son adresse propre. On l'ouvre pour y avoir accès, on transfère les données voulues et on le referme. La seule condition, c'est d'instaurer au préalable les paramètres à respecter (la configuration), comme le débit binaire, le bit de parité et la gestion des temps morts.

## Matériel

La **figure 2** vous propose le schéma de la commande de moteur pas à pas et de l'interface, une structure toute droite. Partons de l'entrée RS-232 (K1) d'où les lignes d'émission (Tx) et de réception (Rx) se branchent à une puce de conversion de niveau (IC2). Il s'agit pour elle, nous l'avons dit, de transformer les signaux bipolaires RS-232 –par exemple d'un

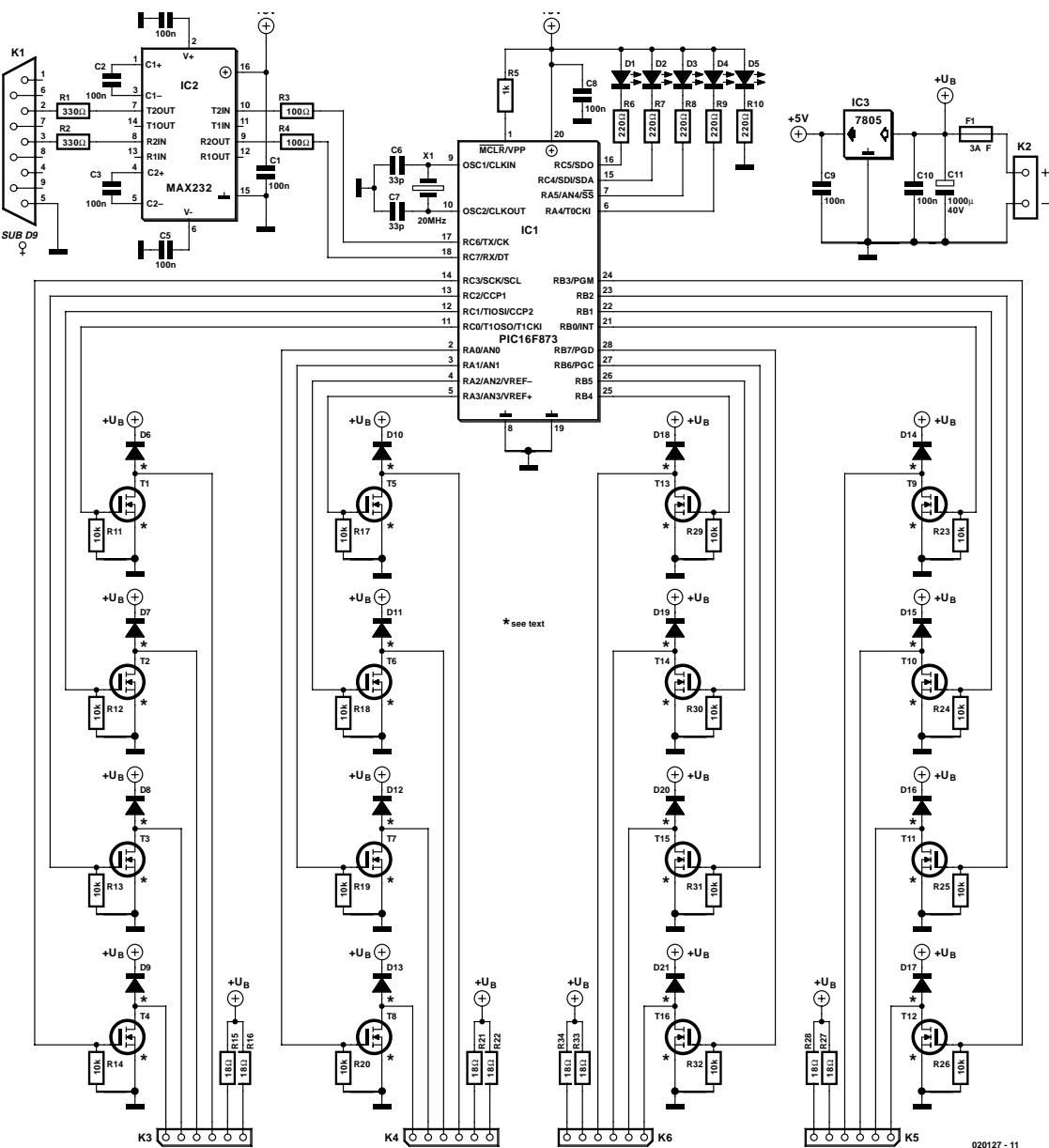


Figure 2. Schéma de la platine de commande.

débattement entre  $-9\text{ V}$  et  $+9\text{ V}$  – en une excursion TTL, définie, elle, entre  $+5\text{ V}$  et  $0\text{ V}$ . Notons que le processus est interne, par la technique de commutation de condensateur, de manière à former une tension supérieure à

double polarité ( $\pm 9\text{ V}$ ).

Les signaux convertis en niveaux TTL sont alors envoyés aux broches d'un UART (*Universal Asynchronous Receiver-Transmitter*) intégré au

microcontrôleur PIC IC1. Les bornes d'E/S RS-232 sont connectées à travers des résistances en série, telles que R1 et R2, de même côté convertisseur, R3 et R4, aux fins de protec-

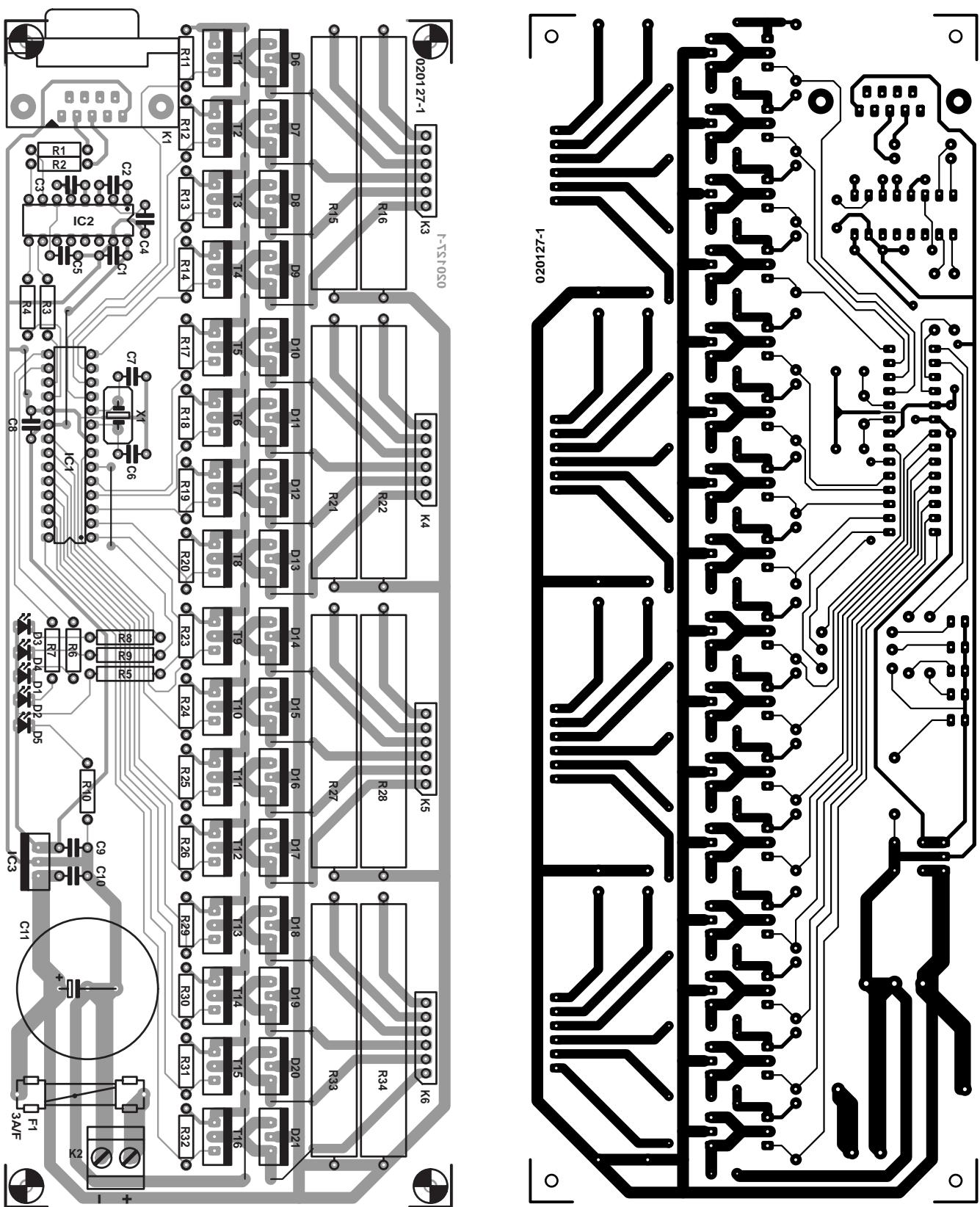


Figure 3. Tracés du circuit imprimé de commande de moteur pas à pas.

tion : on ne sait jamais !

Le régulateur linéaire IC3 sert à fournir une tension stable de +5 V au PIC MCU et aux puces d'interface RS-232. IC1 utilise les condensateurs C6 et C7, le quartz X1 et une bascule bistable interne pour former un oscillateur de précision sur 20 MHz dont le UART a besoin. La broche 1 du PIC est forcée à l'état haut par la résistance R5, puisqu'il n'est pas nécessaire de remettre à zéro le microcontrôleur. Toutes les autres broches des ports d'E/S (il y en a 20) sont en configuration de sortie et

connectées aux commandes des phases du moteur pas à pas et aux voyants à LED.

La technique de commande du moteur pas à pas retenue ici est unipolaire, avec résistances de limitation, utilisable sur les moteurs pas à pas de basse puissance qui comportent 5, 6 ou 8 fils. Elle fournit un moyen simple et bon marché d'excitation d'un bobinage unipolaire. Mais évidemment son rendement souffre de la dissipation dans les résistances de chute.

Pour transférer la puissance aux

**Tableau 1. Valeurs des résistances de ballast (exemples)**

V <sub>alim</sub> [V]	I <sub>moteur</sub> [A]	R <sub>moteur</sub> [Ω]	R <sub>ballast</sub> [Ω]	P <sub>ballast</sub> [W]
15	1,00	5	10	5,0
20			15	7,5
25			20	10,0
30			25	12,5
15	0,500	15	15	1,9
20			25	3,1
25			35	4,4
30			45	5,6

bobinages du moteur, le circuit de commande de phase fait appel à des composants MOSFET à niveau logique à l'entrée, attaqués directement par les sorties du microcontrôleur. Vous pouvez choisir parmi différents modèles de FET logiques, reportez-vous au tableau. Il faut utiliser des diodes de recouvrement rapides pour assurer un chemin de retour possible à l'énergie emmagasinée dans les bobines du moteur et protéger les composants MOSFET des décharges provoquées par la force contre-électromotrice. Vous en trouverez également une liste dans le tableau adéquat. Les résistances de limitation servent à doser le courant dans les enroulements de phases, elles dissipent inévitablement une certaine puissance. Leur valeur doit se calculer en fonction du moteur utilisé. Il est donc important de disposer des données du constructeur sur le moteur en question, y compris l'impédance des bobines et les valeurs nominales de tension et de courant. Si vous les ignorez, ne vous contentez pas d'en mesurer la résistance à l'aide d'un multimètre, vous avez besoin de connaître les caractéristiques exactes de puissance admissible du moteur. Le **tableau 1** donne un exemple de choix de résistance en série et les caractéristiques correspondantes de deux moteurs pas à pas sous différentes tensions d'alimentation.

Vous pouvez calculer ces valeurs comme suit :

$$R_{\text{série}} = V_{\text{alim}} / (I_{\text{moteur}} - R_{\text{moteur}})$$

$$P_{\text{série}} = 0,5 (I_{\text{moteur}}^2 - R_{\text{série}})$$

## Quelques remarques.

Comme le moteur est excité en mode à plein temps, chaque bobinage n'est alimenté que pendant la moitié du temps, donc la puissance de la résistance en série ne s'élève qu'à la moitié de celle à laquelle on s'attendrait normalement. Il convient de choisir une tension d'alimentation comprise entre 10 V et

## Liste des composants

### Résistances :

R<sub>1</sub>,R<sub>2</sub> = 330 Ω  
R<sub>3</sub>,R<sub>4</sub> = 100 Ω  
R<sub>5</sub> = 1 kΩ  
R<sub>6</sub> à R<sub>10</sub> = 220 Ω  
R<sub>11</sub> à R<sub>14</sub>,R<sub>17</sub> à R<sub>20</sub>,R<sub>23</sub> à R<sub>26</sub>,  
R<sub>29</sub> à R<sub>32</sub> = 10 kΩ  
R<sub>15</sub>,R<sub>16</sub>,R<sub>21</sub>,R<sub>22</sub>,R<sub>27</sub>,R<sub>28</sub>,R<sub>33</sub>,  
R<sub>34</sub> = 18 Ω/ 5 W (cf. texte)

### Condensateurs :

C<sub>1</sub> à C<sub>5</sub>,C<sub>8</sub>,C<sub>9</sub>,C<sub>10</sub> = 100 nF  
C<sub>6</sub>,C<sub>7</sub> = 33 pF  
C<sub>11</sub> = 1 000 μF/40 V radial

### Semi-conducteurs :

D<sub>1</sub> à D<sub>4</sub> = LED 3 mm verte  
D<sub>5</sub> = LED 3 mm rouge  
D<sub>6</sub> à D<sub>21</sub> = MBR2060CT (Farnell #  
247-157, cf. encadré)  
IC<sub>1</sub> = PIC16F873-20/SP  
**(EPS 02127-41)**

IC<sub>2</sub> = MAX232CPE (Maxim  
Integrated)

IC<sub>3</sub> = 7805CP  
T<sub>1</sub> à T<sub>16</sub> = FETMOS Logic-Level, tel  
que, par exemple, RFD14N05L  
(Farnell # 516-399, cf. encadré)

### Divers :

F<sub>1</sub> = porte-fusible encartable +  
fusible 3 AF (rapide)  
K<sub>1</sub> = embase sub-D 9 points femelle  
encartable  
K<sub>2</sub> = bornier encartable à 2 contacts  
au pas de 5 mm  
K<sub>3</sub> à K<sub>6</sub> = embase autosécable à  
1 rangée de 6 contacts  
X<sub>1</sub> = quartz 20 MHz  
La platine (**020127-1**) est disponible  
auprès des adresses habituelles  
La disquette **EPS020127-11**, contient  
tous les fichiers de code-source code  
(téléchargeable également depuis  
notre site : [www.elektor.fr](http://www.elektor.fr))

## FET Logic Level et diodes à rétablissement rapide

Sur ce circuit, le choix de FET Logic Level (positions T<sub>1</sub> à T<sub>16</sub>) et de diodes à rétablissement rapide(positions D<sub>6</sub>-D<sub>21</sub>) dépendra de leur disponibilité et de la puissance du (des) moteur(s) pas-à-pas utilisé(s).

### FETs

Type	I <sub>max</sub> [A]	U <sub>max</sub> [V]	R <sub>i</sub> [mΩ]	Note
RFD14N05L	14	50	100	Farnell # 515-399, Fairchild
BUK100-50GL	13.5	50	125	
BUK101-50GS	30	50	50	
IRL12203N	61	30	7	

### Diodes

Type	I <sub>max</sub> [A]	U <sub>max</sub> [V]		
MBR1045CT	10	45		Farnell # 878-364
MBR1545CT	15	45		Farnell # 878-194
etc.				

## Listage I. Code-source du progiciel.

```

// main.c - Main program code

#include <16f873.h>
#include <ports.h>
#include <protocol.h>
#use delay (clock=20000000)
#use rs232(baud=38400, xmit=tx, rcv=rc)

int astep=1, bstep=1, cstep=1, dstep=1;
long max=800, min=470;

// initialises the ports by defining whether the tri-state buffers should be input or output
void setup_ports(void) { set_tris_a(0x00);set_tris_b(0x00);set_tris_c(0xF0);set_uart_speed(38400); }

// resets one motor to initial state
void reset_motor(int motor) {
    if (motor==1) {output_low(a_1);output_low(a_2);output_low(a_3);output_low(a_4);output_high(led_a);}
    if (motor==2) {output_low(b_1);output_low(b_2);output_low(b_3);output_low(b_4);output_high(led_b);}
    if (motor==3) {output_low(c_1);output_low(c_2);output_low(c_3);output_low(c_4);output_high(led_c);}
    if (motor==4) {output_low(d_1);output_low(d_2);output_low(d_3);output_low(d_4);output_high(led_d);} }

// resets all ports to initial states
void reset_ports(void) { reset_motor(1);reset_motor(2);reset_motor(3);reset_motor(4);putc(ACKNOWLEDGE); }

// creates a delay which constitutes the step pulse duration
void delay_micro(long delay) { long n;for(n=1;n<=delay;n+=3)delay_us(6); }

// changes powered phases according to current step required
void power_motor(int axis, step) {
    if (axis==1) {
        if (step==1) {output_bit(a_1,1);output_bit(a_2,0);output_bit(a_3,0);output_bit(a_4,1);}
        if (step==2) {output_bit(a_1,0);output_bit(a_2,1);output_bit(a_3,0);output_bit(a_4,1);}
        if (step==3) {output_bit(a_1,0);output_bit(a_2,1);output_bit(a_3,1);output_bit(a_4,0);}
        if (step==4) {output_bit(a_1,1);output_bit(a_2,0);output_bit(a_3,1);output_bit(a_4,0);}
        output_low(led_a); }
    if (axis==2) {
        if (step==1) {output_bit(b_1,1);output_bit(b_2,0);output_bit(b_3,0);output_bit(b_4,1);}
        if (step==2) {output_bit(b_1,0);output_bit(b_2,1);output_bit(b_3,0);output_bit(b_4,1);}
        if (step==3) {output_bit(b_1,0);output_bit(b_2,1);output_bit(b_3,1);output_bit(b_4,0);}
        if (step==4) {output_bit(b_1,1);output_bit(b_2,0);output_bit(b_3,1);output_bit(b_4,0);}
        output_low(led_b); }
    if (axis==3) {
        if (step==1) {output_bit(c_1,1);output_bit(c_2,0);output_bit(c_3,0);output_bit(c_4,1);}
        if (step==2) {output_bit(c_1,0);output_bit(c_2,1);output_bit(c_3,0);output_bit(c_4,1);}
        if (step==3) {output_bit(c_1,0);output_bit(c_2,1);output_bit(c_3,1);output_bit(c_4,0);}
        if (step==4) {output_bit(c_1,1);output_bit(c_2,0);output_bit(c_3,1);output_bit(c_4,0);}
        output_low(led_c); }
    if (axis==4) {
        if (step==1) {output_bit(d_1,1);output_bit(d_2,0);output_bit(d_3,0);output_bit(d_4,1);}
        if (step==2) {output_bit(d_1,0);output_bit(d_2,1);output_bit(d_3,0);output_bit(d_4,1);}
        if (step==3) {output_bit(d_1,0);output_bit(d_2,1);output_bit(d_3,1);output_bit(d_4,0);}
        if (step==4) {output_bit(d_1,1);output_bit(d_2,0);output_bit(d_3,1);output_bit(d_4,0);}
        output_low(led_d); } }

// Moves a specified motor by a specified amount of steps in a specified direction.
int move(short direction, long steps, int axis, step) {
    long n, delay, accsteps;
    delay=max; accsteps=max-min;
    for(n=1;n<=steps;n++) {
        if(n<=accsteps)delay--;
        if(steps-n<=accsteps)delay++;
        if(direction==0)step--;else step++;
        if(step==0)step=4;
}

```

```

        if(step==5)step=1;
        power_motor(axis, step); delay_micro(delay); reset_motor(axis); } return(step); }

// Reads in 2 bytes from the UART and returns a 16-bit integer (range 0-65535)
long readlong(void) { return(256*getc() + getc()); }

// Main Program
void main(void) {
    char incomm;
    long steps;
    setup_ports(); reset_ports();
    while(0==0) {
        output_low(led_a); output_low(led_b); output_low(led_c); output_low(led_d);
        incomm=getc();
        output_high(led_a); output_high(led_b); output_high(led_c); output_high(led_d);
        switch(incomm) {
            case RESET:      reset_ports();                                break;
            case SETUP_ACC: min=readlong(); max=readlong();                break;
            case MOVE_A_FW:  steps=readlong(); astep=move(0, steps, 1, astep); break;
            case MOVE_A_RV:  steps=readlong(); astep=move(1, steps, 1, astep); break;
            case MOVE_B_FW:  steps=readlong(); bstep=move(0, steps, 2, bstep); break;
            case MOVE_B_RV:  steps=readlong(); bstep=move(1, steps, 2, bstep); break;
            case MOVE_C_FW:  steps=readlong(); cstep=move(0, steps, 3, cstep); break;
            case MOVE_C_RV:  steps=readlong(); cstep=move(1, steps, 3, cstep); break;
            case MOVE_D_FW:  steps=readlong(); dstep=move(0, steps, 4, dstep); break;
            case MOVE_D_RV:  steps=readlong(); dstep=move(1, steps, 4, dstep); break; } putc(ACKNOWLEDGE); } }

// ports.h - defines pin assignments

#define tx      PIN_C6
#define rc      PIN_C7
#define a_1     PIN_C3
#define a_2     PIN_C2
#define a_3     PIN_C1
#define a_4     PIN_C0
#define b_1     PIN_A0
#define b_2     PIN_A1
#define b_3     PIN_A2
#define b_4     PIN_A3
#define c_1     PIN_B3
#define c_2     PIN_B2
#define c_3     PIN_B1
#define c_4     PIN_B0
#define d_1     PIN_B7
#define d_2     PIN_B6
#define d_3     PIN_B5
#define d_4     PIN_B4
#define led_a   PIN_A5
#define led_b   PIN_A4
#define led_c   PIN_C5
#define led_d   PIN_C4

// protocol.h - defines communication protocol

#define RESET      1
#define ACKNOWLEDGE 2
#define SETUP_ACC  10
#define MOVE_A_FW  20
#define MOVE_A_RV  21
#define MOVE_B_FW  22
#define MOVE_B_RV  23
#define MOVE_C_FW  24
#define MOVE_C_RV  25
#define MOVE_D_FW  26
#define MOVE_D_RV  27

```

## Listage 2. Programme de test à faire tourner sur le PC.

```

unit main;

interface

uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, QCCom32, Buttons, ExtCtrls;

type
  TForm1 = class(TForm)
    QCPort: T_QCCom32;
    Commport: TComboBox;
    xclgroup: TRadioGroup;
    setup_acc, move_a_rv, move_a_fw, move_b_rv,      move_b_fw, move_c_rv, move_c_fw, move_d_rv, move_d_fw, reset: TRadioButton;
    parameter1, parameter2: TEdit;
    commportlabel, parameterlabel: TLabel;
    Executebutton: TBitBtn;
    autoreset: TCheckBox;
    procedure CommportChange(Sender: TObject);
    procedure ExecutebuttonClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure setup_accClick(Sender: TObject);
    procedure move_a_fwClick(Sender: TObject);
    procedure move_a_rvClick(Sender: TObject);
    procedure move_b_fwClick(Sender: TObject);
    procedure move_b_rvClick(Sender: TObject);
    procedure move_c_fwClick(Sender: TObject);
    procedure move_c_rvClick(Sender: TObject);
    procedure resetClick(Sender: TObject);
    procedure move_d_fwClick(Sender: TObject);
    procedure move_d_rvClick(Sender: TObject);
  private { Private declarations }
  public { Public declarations }
  end;

var Form1: TForm1;

Implementation {$R *.DFM}

procedure TForm1.resetClick(Sender: TObject); begin parameter1.Enabled := FALSE; parameter2.Enabled := FALSE; end;

procedure TForm1.setup_accClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := TRUE; end;

procedure TForm1.move_a_fwClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_a_rvClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_b_fwClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_b_rvClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_c_fwClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_c_rvClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_d_fwClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.move_d_rvClick(Sender: TObject); begin parameter1.Enabled := TRUE; parameter2.Enabled := FALSE; end;

procedure TForm1.CommportChange(Sender: TObject);
begin QCPort.Port := Commport.ItemIndex + 1; end;

procedure TForm1.FormShow(Sender: TObject); begin QCPort.Port := 1; CommPort.ItemIndex := 0; end;

```

```

procedure TForm1.ExecutebuttonClick(Sender: TObject);
var
  commandcode : char;
  command      : string;
begin
  Executebutton.Enabled := FALSE;

  if reset.Checked     then commandcode := char(1);
  if setup_acc.Checked then commandcode := char(10);

  if move_a_fw.Checked then commandcode := char(20);
  if move_a_rv.Checked then commandcode := char(21);
  if move_b_fw.Checked then commandcode := char(22);
  if move_b_rv.Checked then commandcode := char(23);
  if move_c_fw.Checked then commandcode := char(24);
  if move_c_rv.Checked then commandcode := char(25);
  if move_d_fw.Checked then commandcode := char(26);
  if move_d_rv.Checked then commandcode := char(27);

  QCPort.Open; setlength(command, 1);
  command[1] := commandcode; QCPort.Write(command);

  if (parameter1.enabled) then
    begin
      setlength(command, 2);
      command[1] := char(strtoint(parameter1.text) div 256);
      command[2] := char(strtoint(parameter1.text) mod 256);
      QCPort.Write(command); end;

  if (parameter2.enabled) then begin
    setlength(command, 2);
    command[1] := char(strtoint(parameter2.text) div 256);
    command[2] := char(strtoint(parameter2.text) mod 256);
    QCPort.Write(command); end;

  while(QCPort.Read = '') do;

  if autoreset.Checked then begin
    setlength(command, 1);
    command[1] := char(1);
    QCPort.Write(command);
    while(QCPort.Read = '') do; end;

  QCPort.Close; Executebutton.Enabled := TRUE;
end;
end.

```

30 V : plus elle sera grande, plus le moteur recevra de puissance. Celle-ci peut dépasser la puissance nominale du moteur lui-même, vu qu'une partie sera dissipée par les résistances en série. N'oubliez pas non plus que le courant maximum (par enroulement) que la platine peut conduire ne doit pas excéder 1 A.

## Construction

Tous les composants de ce circuit sont assemblés directement sur la platine, dont vous pouvez découvrir, à la **figure 3**, le côté pistes cuivrées et le côté composants. Utilisez des

supports pour les deux puces DIL (*dual-in-line*) mais IC3 doit se souder directement à la platine. Il est toujours préférable de monter pour commencer les composants au profil le plus bas, les ponts, résistances, supports DIL, condensateurs à la céramique, etc. Il y a lieu de vérifier la polarité des semi-conducteurs et des condensateurs électrolytiques avant soudage. Quant aux résistance en série, il vaut mieux les installer légèrement au-dessus de la surface de la platine, parce qu'elles chauffent fort en service. Ce serait une bonne idée d'intercaler, sur leurs fils de raccordement, des écarteurs en céramique

pour les maintenir à la distance voulue du circuit imprimé.

Si vous n'avez que faire de 4 canaux, vous pouvez hardiment ne garnir que ceux dont vous avez l'usage, deux sur quatre, par exemple.

Les soudures terminées, il reste à installer les puces du microcontrôleur PIC et le MAX232 dans leurs supports respectifs. Programmer votre propre PIC, vous pouvez le réaliser à partir de la disquette, dont la référence est **020127-11** ou acquérir gratuitement le code source sur notre site Internet [www.elektor.fr](http://www.elektor.fr) sous la rubrique Téléchargements. Le lecteur plus ambitieux qui souhaite assaisonner à sa mode le micrologiciel de son PIC ou lui adjoindre d'autres fonctions trouvera plus loin

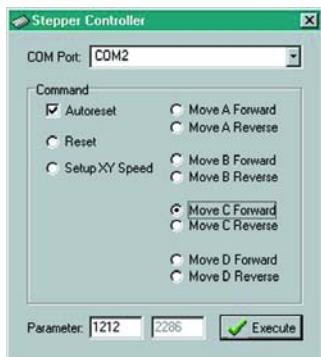


Figure 4. Le logiciel de commande de moteur pas à pas en action sur PC.

un panorama complet et quelques lignes de conduite. Mais il est malgré tout conseillé de tester le projet avec le micrologiciel d'origine avant d'entreprendre de le modifier.

## Le logiciel du contrôleur

La mission du microcontrôleur PIC consiste à recevoir du PC, par le port RS-232, les commandes et à les exécuter. Il porte la responsabilité de la génération des séquences de pas qui vont doser la puissance envoyée au moteur. Ces salves produiront aussi les cycles d'accélération et de décélération optimisés en fonction de la réaction spécifique du moteur. C'est l'entremise de cette interface à bas niveau qui va nous garantir la précision de la chronométrie.

Mais au fait, pourquoi s'embarrasser de ce microcontrôleur, alors qu'on dispose déjà d'un ordinateur pour commander le moteur pas à pas ? Le pilotage en temps réel, voilà qui était parfaitement réalisable dans le temps, avec des logiciels sous environnement DOS, mais c'est désormais impossible, hélas, à cause de l'exploitation en multitâche et du traitement simultané des menaces développés sur les systèmes récents à 32 bits de Windows. Le partage temporel du processeur l'empêche de poursuivre une gestion précise du temps. Le micrologiciel pour ce projet a été conçu à l'aide d'un compilateur C indépendant, peu onéreux, fourni par CCS, mais totalement compatible avec l'environnement MPLAB de Microchip. Pour obtenir plus de détails sur ce compilateur, une référence complète du langage est disponible en ligne sur le site Web de CCS.

Le code est divisé en trois fichiers, main.c,

protocol.h et ports.h. Le logiciel principal se trouve dans main.c, avec la définition des broches (sous des noms de variables différents) dans ports.h et le protocole de communication adapté dans protocol.h. Le code source du micrologiciel est repris dans le **listage 1**.

Le protocole de communication utilisé dans ce projet est très simple. Chaque commande est transmise sous forme d'un octet et si elle réclame d'autres paramètres, ils seront envoyés l'un après l'autre. Par exemple, pour demander au contrôleur de faire tourner le moteur de 1 000 pas dans un sens, il faut trois octets, le premier détermine la commande, les deux autres spécifient le nombre de pas (dans la plage de 0 à 65 535). D'après l'octet initial de commande, on connaît la longueur totale de la transmission. Après avoir exécuté la commande, le microcontrôleur répondra par un octet d'accusé de réception pour dire au PC qu'il est prêt à en recevoir d'autres, si nécessaire.

Le module de programme principal commence par initialiser et mettre à zéro tous les ports de sortie, y compris le UART, auquel il impose un débit binaire de 38 400 bits/s. Il entre alors en boucle sans fin, dans l'attente d'un octet en provenance du UART. Dès qu'il reçoit un octet de commande, il passe la main à la section de commande appropriée qui pourra alors recevoir du UART d'autres octets en complément. Voici la liste des commandes disponibles.

**RESET** (octet 1) : met à zéro tous les ports d'E/S.

**SETUP\_ACC** (octet 10) : suivi de quatre octets qui représentent deux entiers de 16 bits, il fixe l'intervalle minimum et maximum entre pas pour déterminer le mouvement du moteur. Lors de l'exécution de la commande MOVE, l'intervalle est maximal, puis raccourcit graduellement jusqu'à atteindre le minimum. Les suivants resteront alors au minimum. À l'approche de la fin du cycle de commande, les pas s'espaceront progressivement pour en revenir à l'intervalle maximal. Ce processus façonne l'accélération et la décélération de chaque commande MOVE.

**MOVE\_A\_FW** (octet 20) : suivi de deux octets (un entier de 16 bits), il indique combien de pas le moteur A doit franchir vers l'avant.

**MOVE\_A\_RV** (octet 21) : suivi de deux octets (un entier de 16 bits), il indique combien de pas le moteur A doit franchir en marche arrière.

**MOVE\_B\_FW** (octet 22)

**MOVE\_B\_RV** (octet 23)

**MOVE\_C\_FW** (octet 24)

**MOVE\_C\_RV** (octet 25)

**MOVE\_D\_FW** (octet 26)

**MOVE\_D\_RV** (octet 27)

Ce sont les mêmes commandes que les deux premières, mais pour les moteurs B, C et D.

Lorsque vous programmez votre PIC, n'oubliez pas d'éteindre la fonction DEBUG\_MODE. Assurez-vous que le POWER\_ON\_RESET est actif et débranchez les fonctions WATCHDOG\_TIMER et BROWN\_OUT\_DETECT. Vérifiez également que la fréquence d'horloge est bien positionnée sur 20 MHz.

Comme programmateur et outils de développement pour le microcontrôleur utilisé ici, nous vous recommandons le Programmateur de PIC (Eletor septembre 2003) et le PICSTART de Microchip ainsi que son module ICD (lequel réclame un support additionnel à 28 broches). Autre choix possible, Taylec Ltd propose à un prix abordable (moins de 50 f, 70 euros) un équivalent du module ICD, totalement compatible avec le logiciel Microchip qui est disponible gratuitement en téléchargement.

## Le logiciel du PC

Le logiciel du PC est rédigé en Borland Delphi 4. Nous avons utilisé le graticiel QCCOM32 de VCL (Visual Component Library) pour accéder au port série.

Le **listage 2** contient un programme de test qui illustre la manière de transférer les commandes par le port RS-232 vers le module de commande de moteur pas à pas. Il est également disponible sous la rubrique de téléchargements sur notre site [www.eletor.fr](http://www.eletor.fr).

Il importe de s'assurer que les

régagements de QCCOM32 correspondent exactement à ceux initialisés par le micrologiciel dans le UART, spécialement que le débit binaire (*b bitrate*) est à 38 400. Lors de l'envoi de chaque commande au contrôleur, il faut ouvrir le port, on transmet alors les octets nécessaires puis on attend le signal d'accusé de réception et finalement, on referme le port.

## Essais et mise en service

Avant d'appliquer l'alimentation, il est important de vérifier que tous les composants sont correctement placés et que les soudures sont propres. Nous débranchons tous les moteurs avant de mettre sous tension. L'occasion d'évaluer à l'ampèremètre le courant débité par l'alimentation. Puis de mesurer au voltmètre l'exactitude de la tension sur les bornes d'alimentation. Au moindre doute, débranchez immédiatement et vérifiez la platine et les connexions. Les cinq LED doivent s'allumer quand le circuit est convenablement alimenté. Si c'est le cas, le microcontrôleur est en marche. S'il n'y a qu'une LED éclairée, le circuit est sous tension mais le micrologiciel du contrôleur ne s'exécute pas comme il faut. En supposant que le microcontrôleur a été programmé avec succès, on peut vérifier que l'alimentation lui parvient sur les broches adéquates. Si tout cela est en ordre, vérifiez les composants de l'oscillateur (X1, C6 et C7). À part cela, il n'y a qu'à reprogrammer ou remplacer le microcontrôleur.

Lorsque le circuit démarre bien, on se sert d'un cordon D-9 mâle/femelle droit (non croisé) pour relier la carte de commande au port RS-232 du PC. On lance le logiciel de test sur le PC

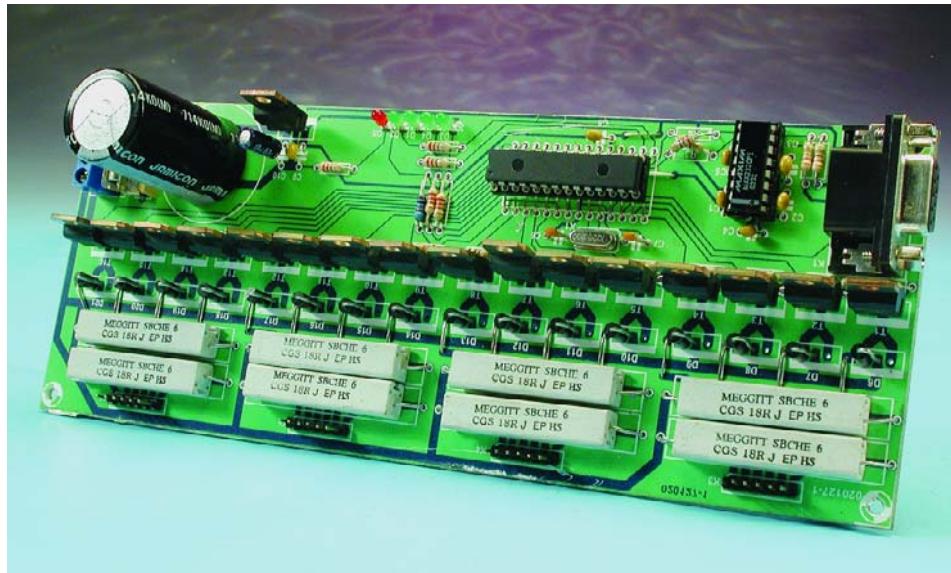


Figure 5. Et notre prototype bon pour le service.

et on choisit les réglages adéquats du port COM. Vous pouvez alors essayer les différentes commandes. Lors de l'envoi d'une commande, quatre LED doivent s'éteindre et une être allumée pour indiquer quel canal est en service, après quoi les quatre LED se rallument. Si tout marche comme vous voulez, éteignez le circuit de commande et branchez un moteur à l'un des canaux, non sans vérifier que les phases et le commun sont à la bonne place. Alors, on remet sous tension et on recommence les essais. Le moteur doit tourner doucement, accélérer et freiner quand on le fait démarrer et s'arrêter. S'il semble sautiler ou cahoter, vérifiez l'ordre des phases et la progressivité de l'accélération en fonction des possibilités du moteur. Des intervalles plus courts correspondent à une rotation plus rapide. Si vous choisissez une vitesse à intervalles constants, par exemple 800-800, il n'y aura ni accélération, ni décélération. La plupart des moteurs peuvent fonctionner avec des intervalles de 500 à 1 000.

Après l'essai de tous les canaux, si tout va comme espéré, vous pouvez adapter le logiciel de commande sur le PC ou celui du PIC pour y ajouter vos propres commandes et améliorations. Une variante porteuse serait le multiplex des moteurs, pour faire tourner plusieurs axes en même temps. Des applications du circuit décrit dans cet article, il y en a quan-

tité en robotique, pour le positionnement précis de pièces mécaniques, pour le pointage d'un télescope, d'une caméra, la commande des évolutions d'un robot ou des mouvements de précision d'une machine-outil pilotée par ordinateur, par exemple.

(020127-II)

## Liens utiles

[www.microchip.com/download/lit/pline/pic-micro/families/16f87x/30292c.pdf](http://www.microchip.com/download/lit/pline/pic-micro/families/16f87x/30292c.pdf)  
Fiche de caractéristiques du microcontrôleur PIC Microchip 16F87X.

[http://geocities.com/scottpinkham/delphi/qc\\_com32.zip](http://geocities.com/scottpinkham/delphi/qc_com32.zip) – lien direct de téléchargement vers le QCCOM32 de VCL pour E/S RS-232 en Borland Delphi.

[www.taylec.co.uk](http://www.taylec.co.uk) – outils de développement peu coûteux sur PIC compatibles avec l'environnement MPLAB de Microchip.

[www.ccsinfo.com](http://www.ccsinfo.com) – vers le compilateur C pour PIC compatible avec l'environnement MPLAB de Microchip.

[www.programmersheaven.com](http://www.programmersheaven.com),  
[www.torry.net](http://www.torry.net), [www.codeguru.com](http://www.codeguru.com) – des VCL pour l'accès matériel et la commande du port.

## Bibliographie

Serial Port Complete par Jan Axelson,  
ISBN: 0965081923.

Programmateur de PIC à large spectre,  
Elektor septembre 2003.

## Téléchargements gratuits

Programme PIC et PC (fichiers du code-source). Dénomination du fichier : **020127-II.zip**

Dessin de la platine au format .pdf.  
Dénomination du fichier :  
**020127-I.zip**

[www.elektron-fr/dl/dl.htm](http://www.elektron-fr/dl/dl.htm),  
choisissez le mois de publication correspondant.

# RC-Analyzer

## Mesure de vitesse sur modèles auto

Dipl.-Ing. Udo Kischel

udo.kischel@t-online.de

**Le RC-Analyzer a été conçu en vue de mesurer la vitesse de véhicules radio-commandés (d'où le RC). Le lectorat de cet article sont les réalisateurs de modèles réduits automobiles télécommandés qui aimeraient bien savoir quelle est la vitesse de pointe atteinte sur route par leur(s) petit(s) bolide(s).**



Le RC-Analyzer mesure la vitesse de modèles réduits automobiles. Le calcul de la vitesse ne se fait pas par comptage du nombre de tours, c'est-à-dire par calcul rétroactif à partir du temps requis pour accomplir chaque tour, mais de façon continue au niveau même du modèle réduit. Il est possible ainsi de mesurer les vitesses et en particulier la vitesse maximale atteinte dans les virages et les lignes droites. On dispose, comme produit résiduel, d'un profil de vitesse sur l'ensemble du parcours.

Un compte-tours mesure en permanence le régime (vitesse de rotation) de l'engrenage

principal. De par cette mesure en continu, il est possible, outre de mesurer la vitesse, également de mesurer l'effet de différentes tailles d'engrenage principal, d'analyser et de tester le fonctionnement du moteur et de la transmission de manière à trouver le réglage optimal de votre modèle radio-commandé et utiliser le moteur dans sa plage de fonctionnement optimale. Un avantage indéniable pour le compétiteur inconditionnel.

Les vitesses mesurées le sont une

fois par seconde avant d'être stockées. La version de programme proposée permet de mesurer, stocker puis lire une fois les tours de test effectués, pas moins de 120 valeurs. Il est possible ainsi de prendre en compte les éléments d'un test d'une durée de 2 minutes.

À l'avenir il est envisagé d'ajouter une EEPROM sur la platine de manière à obtenir une durée de mesure qui ne dépende plus que de la capacité de l'accumulateur du véhicule (de 8 à 10 minutes donc). Le taux d'échantillonnage d'une seconde peut être modifié par le biais du logiciel.

### Le système Analyzer

Le système Analyzer se compose de l'Analyzer proprement dit, un (des 3 types possibles) capteur de régime, un module d'affichage servant à... l'affichage des données sur la piste de compétition et pour terminer une interface RS-232 en permettant la connexion à un PC.

Le cœur, ou plus exactement le cerveau du système Analyzer dont on retrouve le schéma en **figure 1**, est un microcontrôleur PIC de chez Microchip du type PIC16F627 doté de 1 Kbit de mémoire Flash et d'une EEPROM embarquée de 128 octets mise à contribution pour le stockage

des valeurs de mesure. Le microcontrôleur à 18 broches sera à monter sur support de façon à permettre, le cas échéant, une remise à niveau (*update*) du logiciel qu'il intègre. L'horloge de 4 MHz pour le cadencement du microcontrôleur est fourni par un résonateur céramique à 3 broches. On découvre, à proximité du microcontrôleur, une paire d'embases à cavalier, JP1 et JP2, qui servent à la définition de l'un ou l'autre paramètre/fonction.

Le contrôleur est épaulé par un convertisseur de niveau RS-232 qui sert à établir la communication avec le PC. Seules les lignes TxD (*Transmit Data*) et RxD (*Receive Data*) sont utilisées, il n'y a pas, de ce fait (merci Mr Lheureux), de processus d'accordéon (handshake). Un régulateur de tension à faibles

pertes qui n'introduit qu'une chute de tension de 0,5 V, garantit un fonctionnement du circuit qu'il soit alimenté par piles ou par accumulateurs. La touche S1 qui attaque la ligne MCLR (*Memory CLeaR*) démarre la mesure. Comme d'habitude, l'ajustable P1 permet de jouer sur le contraste de l'afficheur LCD. La broche 5 (R/W) de l'affichage est forcée à la masse, vu que le logiciel n'a pas pour objet la lecture de données de l'affichage mais d'y écrire (W = Write actif au niveau bas). Tous les contacts servant à la connexion de l'alimentation, du capteur de vitesse de rotation, de l'affichage, d'une touche externe et du câble RS-232 sont accessibles de l'extérieur.

Un programme écrit en assembleur et servant à la présentation gra-

## Caractéristiques :

- Tension d'alimentation 5 à 8 volts
- Toutes connexions accessibles
- Sélection de menu par cavalier
- Réalisation aisée à l'aide de composants classiques
- Microcontrôleur PIC
- Affichage à contraste réglable
- Interface RS-232
- Résultats sous forme graphique
- Connexion pour touche externe
- Mesure, mémorisation, lecture
- 3 variantes de capteurs de régime IR et inductif
- Électronique de commande sur la platine de l'Analyzer

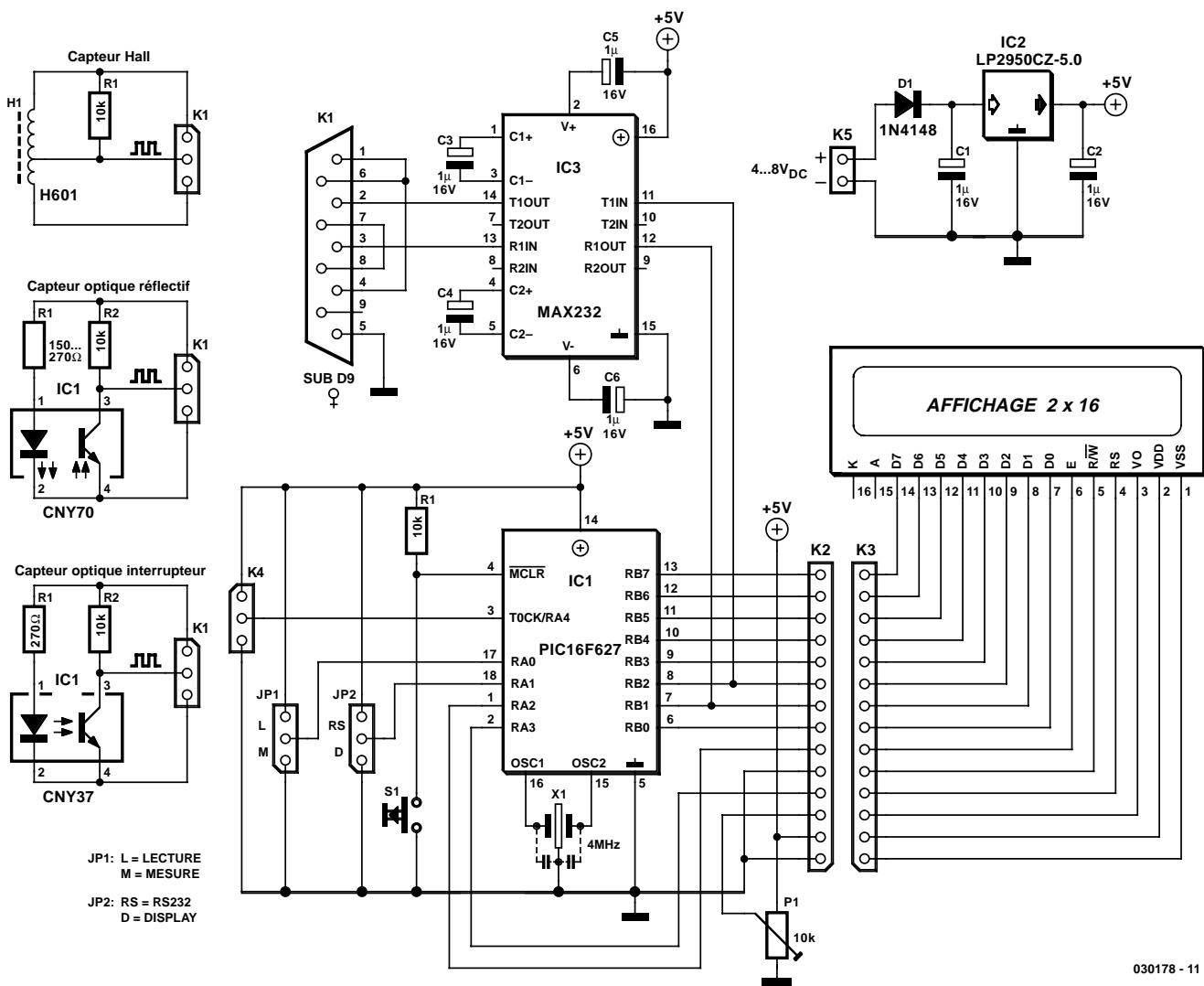


Figure 1. Le RC-Analyzer peut s'accommoder de 3 variétés de capteurs.

phique des données et à l'analyse couronne le système pour lui donner sa fonctionnalité. Le système RC-Analyzer est un système ouvert, de sorte qu'il est facile, ultérieurement, d'intégrer des fonctions nouvelles tant au matériel qu'au logiciel.

L'embase K4 pourra se voir connecter l'un des 3 capteurs représentés sur le schéma de la figure 1 : un capteur à effet Hall, qui réagit à un champ magnétique (rotatif), une barrière lumineuse optique à réflexion (opto-coupleur réfléctif) et un capteur à détection d'interruption optique (opto-coupleur à fenêtre). En cas d'utilisation de l'opto-coupleur à fenêtre il faudra doter l'engrenage principal du modèle RC d'un petit orifice par lequel la lumière infrarouge (IR) passera une fois par tour. Ce type de capteur requiert le plus de place. L'écartement de l'opto-coupleur utilisé dépend bien entendu de l'épaisseur de l'engrenage principal.

Il faudra, si l'on opte pour la barrière réfléctrice, disposer une minuscule surface réfléctrice sur l'engrenage principal (film d'aluminium collé ou tâche blanche). L'écartement entre l'opto-coupleur réfléctif et la surface réfléchissante peut atteindre en règle générale jusqu'à 15 mm sans que cela n'ait d'influence néfaste sur la fonction.

Dans le cas du capteur à effet Hall, un minuscule aimant plaqué sur l'engrenage principal génère, lors de chaque tour, l'impulsion nécessaire. L'écartement entre le capteur et l'aimant ne devrait pas dépasser 5 mm.

## Lecture et affichage des valeurs de mesure

3 principes – 1 même résultat : le rayon de lumière IR interrompu ou réfléchi selon le cas par l'engrenage principal voire l'induction dans le cas du capteur à effet Hall, génère, dans le capteur de régime concerné, un signal rectangulaire appliqué à l'entrée de comptage RA4 du microcontrôleur.

Cette entrée à trigger de Schmitt transfert les impulsions au compteur TMR0. Le temporisateur (timer) TMR0 comptabilise les transitions haut-bas et écrit leur nombre par seconde dans le registre correspondant et également dans l'EEPROM interne en vue d'une mémorisation à terme. Il est possible de paramétrier par logiciel la durée de la fenêtre à 1/20 (0,05), 1/10 (0,1), 1/2 (0,5) ou à 1 s et par conséquent le créneau du nombre de valeurs de mesure possibles.

Lors du lancement de la mesure par une action sur la touche S1, le programme introduit une temporisation de 10 s (qu'il est facile de modifier) de manière à ce que le pilote puisse aller sur le point de départ ou ait le temps de se concentrer sur le départ. Le logi-

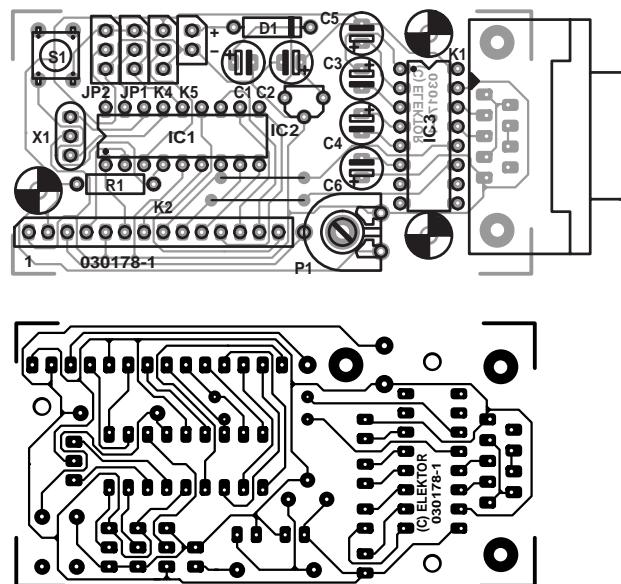


Figure 2. Une platine simple face, des composants classiques, compacte pourtant.

## Liste des composants Platine principale

### Résistances :

R1 = 10 kΩ  
P1 = ajustable 10 kΩ

### Condensateurs :

C1 à C6 = 1 µF/16 V radial

### Semi-conducteurs :

D1 = IN4148  
IC1 = PIC16F627-4/CP  
(EPS030178-41)  
IC2 = LP2950CZ-5.0  
IC3 = MAX232CP

### Divers :

JP1,JP2 = embase autosécable à une rangée de 3 contacts + cavalier  
K1 = embase sub-D à 9 contacts encartable en équerre  
K2 = embase autosécable à 1 rangée de 14 contacts  
K3 = affichage LCD à 2 lignes de 16 caractères et embase autosécable à 1 rangée de 14 contacts  
K4 = embase autosécable à une

rangée de 3 contacts  
S1 = bouton-poussoir unipolaire  
X1 = résonateur céramique 4 MHz à 3 contacts et capacité intégrée

### Capteur à effet Hall

RI = 10 kΩ  
HI = capteur à effet Hall H601  
KI = embase autosécable à une rangée de 3 contacts

### Capteur réfléctif

RI = 150 à 270 Ω  
R2 = 10 kΩ  
IC1 = CNY70  
KI = embase autosécable à une rangée de 3 contacts

### Capteur à interruption

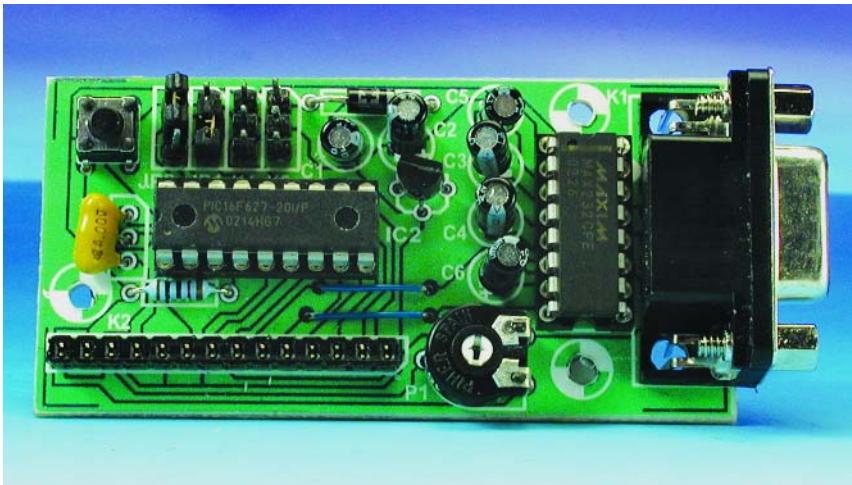
RI = 270 Ω  
R2 = 10 kΩ  
IC1 = CNY37  
KI = embase autosécable à une rangée de 3 contacts

ciel profite de cette temporisation pour effacer les anciennes valeurs dans l'EEPROM interne du PIC.

Le bouton-poussoir S1 ne sert pas uniquement à démarrer la mesure, mais aussi à la lecture des données et à leur transfert soit vers l'affichage LCD soit, en fonction des posi-

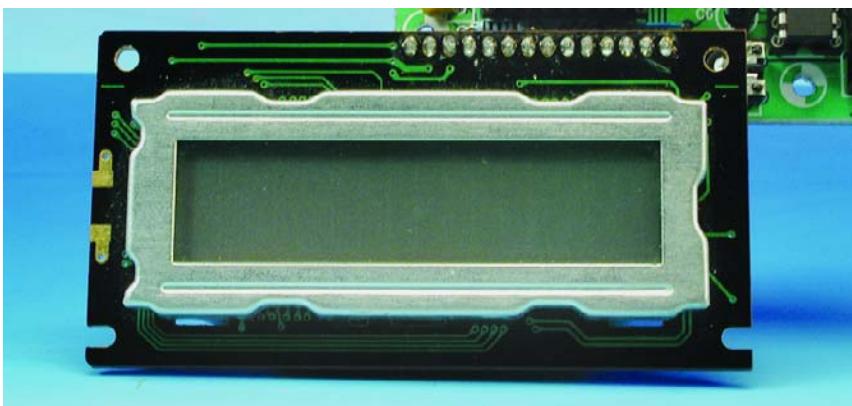
tions des cavaliers JP1 et JP2, vers l'interface RS-232 (cf. tableau 1).

La visualisation des données de mesure se fait par le biais d'un module d'affichage à 2 lignes de 16 caractères. À chaque fois on voit apparaître le numéro d'ordre de la mesure et la valeur de mesure pro-



rement dite, le nombre de tours par minute (tr/mn) de l'engrenage principal. La vitesse se calcule à l'aide d'une conversion simple. Dans une prochaine version du programme, il

est prévu une possibilité d'affichage direct de la vitesse en kilomètre/heure (km/h). Il est possible de paramétriser logiciellement le facteur de conversion (distance par-



courue par tour de l'engrenage principal). Le programme de lecture des données par le biais de l'interface sérielle et leur visualisation sur le PC a été écrit en Visual Basic pour Excel. Une interface de communication simple permet le choix du paramètre correspondant et visualise la vitesse et le régime du moteur dans sa propre fenêtre graphique. Cela offre une possibilité idéale de découvrir, par exemple, quel est l'effet d'un choix de roue dentée primaire (prise directement sur l'axe du moteur) et d'engrenage principal différents sur le régime (vitesse de rotation).

### Une petite platine

Bien qu'il faille que la platine de l'Analyzer prenne place dans le modèle réduit et que par conséquent, tout centimètre carré excédentaire est à éviter, nous avons trouvé une solution satisfaisante en dessinant une platine simple face dotée de composants « classiques » représenté en **figure 2**. Pas de misère avec des CMS, l'implantation des composants reste un jeu d'enfant. La platine ne comporte que 2 ponts de câblage. Le microcontrôleur sera placé, nous le disions plus haut, sur support.

Si la place disponible (ou plutôt le manque de place) le requiert, on pourra remplacer l'embase sub-D par une embase droite dans laquelle viendra s'enficher l'adaptateur correspondant.

### Le logiciel

L'intelligence du système RC-Analyzer se trouve dans le *programme en assembleur*. Le

**Tableau I**

Mesure	Lecture par affichage	Lecture par RS-232
Relier K5 à l'accu de marche ou à un canal de réception libre (ne dériver que + et -)		
Relier K4 au capteur	Relier K2 à l'affichage	Relier K1 à l'interface RS-232 du PC
JPI sur M	JPI sur R	JPI sur R
JP2 ouvert	JP2 sur D	JP2 sur RS
—	—	Ouvrir le fichier Excel rclogex3.xls et appuyer sur le bouton-poussoir. Le programme attend l'arrivée de données (boucle fermée, sortie par <Esc>)
Appuyer sur S1		
Après écoulement de la durée de temporisation, on a prise de la première des 60 (durée de porte de 0,5 s) ou de 120 (durée de porte de 1 s) mesures.	On a, sur l'affichage, visualisation, successive-ment, du numéro de mesure (0 à 59 ou 0 à 119) et de la valeur de mesure (tr/s) corres-pondante	Lecture des valeurs de mesure. Après prise en compte complète des données (20 s environ) on a affichage d'un message « Ready »; il reste à appuyer sur le bouton « OK ».
Démarrer une nouvelle mesure par une nouvelle action sur le bouton-poussoir (les données anciennes sont effacées).	Démarrer nouvelle lecture par une nouvelle action (1 ou 2 fois) sur le bouton-poussoir.	Le fichier RSAPI.DLL avec la librairie de fonctions pour la commande de l'interface sérielle du PC soit être recopiée dans le répertoire System de Windows avant le début de la visualisation.

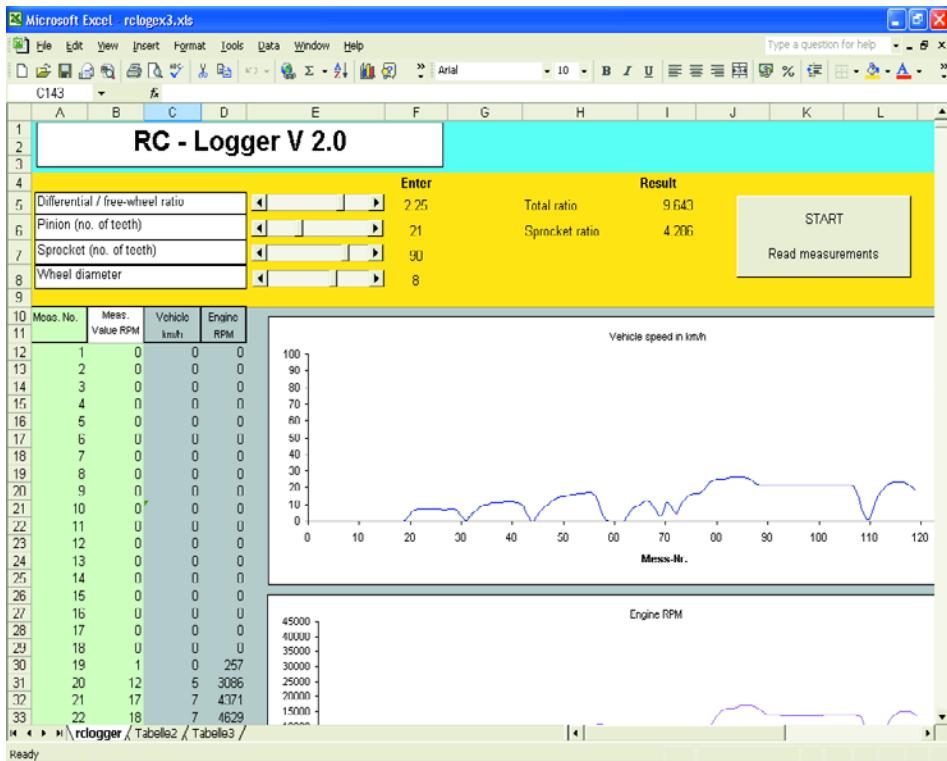


Figure 3. Visualisation des paramètres et des valeurs de mesure en Excel.

programme principal accède aux différentes fonctions programmées sous la forme de modules au fur et à mesure des besoins. Lorsque l'opération requise est exécutée, le programme retourne comme prévu dans l'ar-

borescence principale. Au nombre des fonctions du programme principal citons l'initialisation des entrées et sorties du PIC, la lecture des cavaliers et l'effacement des anciennes

valeurs de mesure de la mémoire EEPROM interne.

## Commande de l'interface sérielle

La librairie de fonctions **RSAPI.DLL** permet la commande de périphériques par le biais de l'interface sérielle du PC. Mesures et commandes sont des applications typiques pour cette .DLL. La commande se fait par l'intermédiaire de macros standard pour les programmes de la suite Office tels que Word et Excel, macros écrites en VB. L'un des avantages majeurs de cette approche est qu'il n'est pas nécessaire de procéder à une conversion préliminaire des valeurs de mesure transférées qui peuvent ainsi être placées directement dans des feuilles de tableau. Le fichier est ensuite sauvegardé sur le disque dur du PC, dans le répertoire système de Windows plus précisément.

Dans le programme Excel **rclogger3.xls**, on a déclaration, dans une macro, des fonctions et procédures (sub) de RSAPI.DLL correspondantes. C'est ainsi que l'on apprend au Basic qu'il faut utiliser de nouvelles fonctions externes. Ces déclarations se trouvent toujours au début du module VB, suivie par la routine de lecture des valeurs de mesure.

L'interface est initialisée au paramétrage suivant : 2 400 bauds, pas de bit de parité, 8 bits de donnée, 1 bit d'arrêt avant d'être ouverte.

La feuille de tableau Excel **rclogger** est activée et la colonne B des anciennes valeurs est effacée. La macro attend alors l'arrivée de nouvelles valeurs.

Tant que l'on n'a pas actionné la touche du RC-Analyzer en vue d'une lecture des valeurs de mesure, le programme reste dans une boucle fermée (dont on peut sortir par action sur la touche <Esc>). En cas d'action sur la touche, la fonction READBYTE procède à la lecture, ligne après ligne, des valeurs de mesure. Après transfert des données l'interface est clôturée par CLOSECOM et on a apparition d'un message de dialogue. La **figure 3** montre le résultat de l'exécution du programme.

(030178)

## Extensions planifiées, idées à moyen terme

Le programme (ainsi que le dessin des pistes) de ce projet est disponible au téléchargement gratuitement sous la dénomination **EPS030178-II** sur le serveur d'Elektor. Il est également possible, sous la dénomination de **EPS030178-41**, d'obtenir le microcontrôleur programmé auprès des adresses habituelles. La platine est disponible par le biais de PCBShop. Notons l'existence d'un site Internet spécifiquement réservé au RC-Analyzer à l'adresse :

<http://www.georgeii.de/analyzer/analyzer.htm>

Vous pourrez trouver à cet endroit la version la plus récente du logiciel, surtout lorsque les modifications, extensions et idées auront vu le jour.

- Mémoriser plus de 1 000 valeurs de mesure dans une EEPROM externe
- Démarrage du processus de mesure par le biais de la télécommande RC
- Utilisation de l'Analyzer comme module de commutation pour des fonctions additionnelles
- Visualisation des feux de stop en cas d'action correspondante sur le manche de commande
- Télétransfert des données par voie ISM
- Capteurs de température
- Lecture des données accu (pour, par exemple, tracer la courbe de décharge)

Le fichier .dll RSAPI.DLL est disponible sur le site Internet de B. Kainka :

<http://home.t-online.de/home/H.Kainka/rsapidll.zip>

# Interface IrDA pour le 8051

Transmission de données sans fil de la carte Flash vers un Palm

Burkhard Kainka

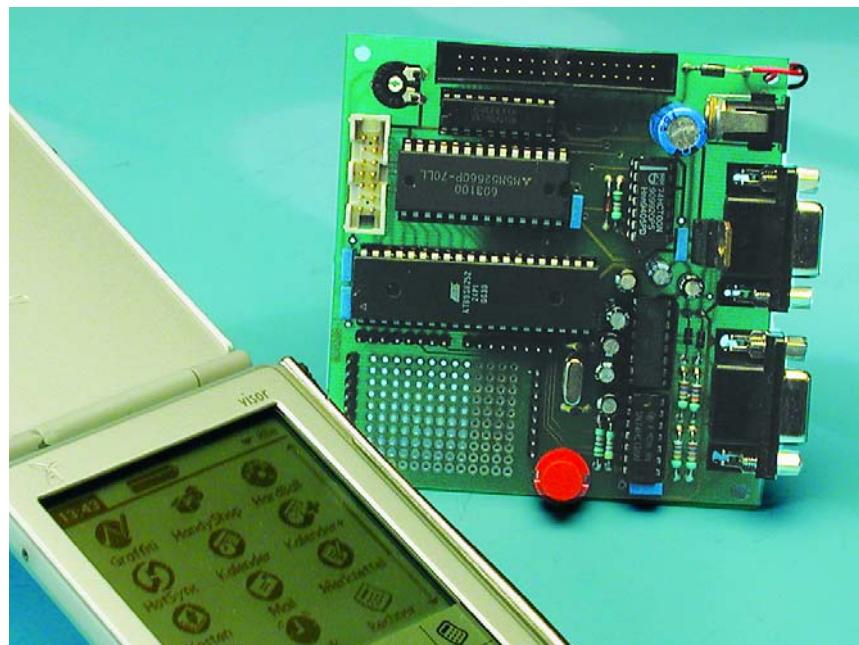
La transmission de données sans fil par le biais de l'infrarouge (IR) présente de très nombreux avantages dans le combat contre la pléthore de câbles. Ceci n'est pas uniquement vrai dans le cas des PC, qu'ils soient de bureau ou portables, mais également dans celui des cartes à microcontrôleur.

C'est dans le cas des organisateurs, ces mini-ordinateurs de poings comme le Palm, en particulier que la communication par IR se fait plus facilement que par le biais de l'interface RS-232. Certains des modèles les plus récents, le Palm Zire par exemple, ne comportent même plus d'interface RS-232, la liaison avec un PC se faisant par le biais d'une interface USB, l'interface IrDA servant à l'échange de données avec d'autres organisateurs, téléphones portables et autres appareils similaires. On en vient inévitablement à l'idée de doter des systèmes à microcontrôleur comme la carte à 89S8252 Flash décrite dans Elektor d'une interface IrDA.

## UART IrDA pour BASCOM-8051

Au premier abord, l'interface IrDA paraît relativement complexe, en particulier si l'on pense aux différents niveaux du logiciel. En fait, au niveau le plus élémentaire de la transmission physique d'octets distincts, les choses sont très simples. L'interface IrDA est une interface série relativement normale à impulsions raccourcies. De courts flashes lumineux représentent le bit de début (*start*) et chaque bit se trouvant à « 0 », c'est-à-dire aux positions auxquelles la ligne TxD de l'interface RS-232 produit un niveau haut d'une longueur d'un bit.

La longueur d'impulsion est définie avec précision. Elle est, pour des taux de transmission ne dépassant pas 115 200 bauds, de 1,6 µs au



minimum, sa longueur maximale ne dépassant pas 3/16èmes de la longueur de bit RS-232. Avec le Palm on utilise les signaux les plus courts possibles, c'est-à-dire des flashes lumineux de 1,6 µs de long. Cette façon de procéder permet d'économiser de l'énergie. En outre, des impulsions IR très courtes et de forte intensité possèdent une portée importante, vu qu'un récepteur IR

idoine peut aisément les distinguer de la lumière environnante vu que cette dernière ne comporte pas d'impulsions courtes de ce genre. La **figure 1** représente les signaux sériels d'une interface RS-232 comparés à ceux produits par une interface IrDA. Il suffit qu'un récepteur IrDA allonge quelque peu les impulsions lumineuses pour que les signaux résultants puissent être lus

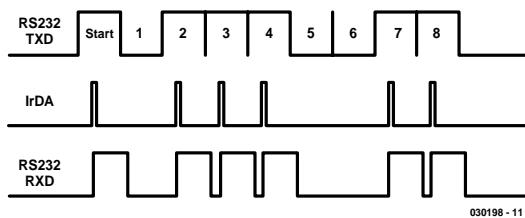


Figure 1. Signaux RS-232 et IrDA.

par une interface RS-232 standard. Il suffit, à un émetteur IR, de convertir les signaux sériels en brèves impulsions. Il existe à cet effet des circuits intégrés spécialisés. En ce qui concerne le Palm, cette fonction est déjà intégrée dans le processeur. À condition de paramétrier l'UART interne en conséquence, on a génération directe de signaux IrDA. L'UART du respectable ancêtre qu'est le 8051 ne connaît bien évidemment pas un tel mode. Il faudra partant écrire son propre UART (*Universal Asynchronous Receiver/Transmitter* = Émetteur/Récepteur Asynchrone Universel) logiciel pour IrDA.

Il est possible, à l'aide de routines en assembleur écrite sous BASCOM, de réaliser des routines IrDA simples. Les routines Putir et Getir constituent notre UART logiciel. Le transfert de données entre les routines en assembleur et le programme en BASIC se fait par le biais de l'accumulateur par attribution directe ou lecture du registre Acc. La routine d'émission attend un octet dans l'accumulateur. On a ensuite production d'une impulsion de bit de début sur la ligne de port P3.3 paramétrée en sortie. La longueur utilisée correspond à 3 cycles d'instruction. Cela se traduit, en cas d'utilisation d'un quartz

de 11,059 MHz, par une longueur d'impulsion de l'ordre de 3 µs.

C'est à dessein qu'il n'a pas été choisi de travailler aux temps de commutation les plus courts possibles, de manière à éviter de pousser à ses limites l'électronique de commande. Le bit de début est suivi d'une pause d'une longueur égale à une durée de bit. Ensuite, les 8 bits de l'accumulateur sont traités et pour chacun d'entre eux se trouvant à 0 on a génération d'une impulsion. Dans le **Listage 1** le commentaire de chaque instruction indique le nombre de cycles nécessaires. Les boucles d'attente ont été paramétrées de manière à ce que l'on très exactement la durée de bit correcte de 104 µs pour un taux de transmission (*baudrate*) de 9 600 bauds.

La routine de réception ne doit pas effectuer une interrogation directe de la ligne d'entrée parce qu'il y aurait risque dans ce cas-là de ne pas détecter des impulsions brèves. Ceci explique qu'il ait été fait appel à une interruption. La ligne d'entrée P3.2 est l'entrée d'interruption Int0. On crée pour cette raison une routine de traitement d'interruption (*interrupt-service*) et valide l'interruption au niveau de Int0. Lors de l'initialisation on a paramétrage d'une interruption déclenchée par flanc.

Chaque flanc d'impulsion descendant arri-

#### Listage 1. Accès au port par IrDA.

```

'_____
' IrDA1.BAS
'_____

$regfile = "89s8252.dat"
Dim N As Byte
Dim Flag As Bit
Declare Sub Putirda
Declare Sub Getirda

Enable Interrupts
Enable Int0
Set Tcon.0      'INT0 falling edge.
On Int0 Puls Nosave 'Nosave: timing!
While 1 = 1
    Getirda
    P1 = Acc
    Waitms 1
    Acc = P1
    Putirda
    Waitms 1
Wend
End

Sub Putirda
$asm
    clr  P3.3      ;1 start
    nop
    nop
    setb P3.3      ;1
    mov  r2,#43      ;2
S1:
    djnz r2,S1      ;43*2
    mov  r3,#8      ;2
S2:
    jb   ACC.0,S3      ;2,
                ;(10+86)*1.085µs=104µs
    clr  P3.3      ;1 data puls
    nop
    nop
    setb P3.3      ;1
    sjmp S4      ;2
S3:
    nop
    nop          ;delay
    nop
    nop
    S4:
    mov  r2,#41      ;2
S5:
    djnz r2,S5      ;43*2
    rr   a           ;1
    nop
    djnz r3,S2      ;2
                ;(14+82)*1.085µs=104µs
    clr  Flag
    ret
$end Asm
End Sub

Sub Getirda
Flag = 0
$asm
Startbit:
    djnz r2,S10     ;43*2
    djnz r3,S7      ;2,
                ;(9+7+80)*1.085µs=104µs
    ret
$end Asm
End Sub

Puls:
Flag = 1
Return

```

## Listage 2. Émission et réception d'états de ports avec le Palm.

```
#airport.bas
open "com1:",9600,ir as #5
draw -1
while 1
    input n
    put #5,n
    a= fn wait(0.1)
    i=fn serial(5)
    if i>1 then dummy=get$(#5,0)
    n =get$(#5,0)
    t$=str$(n)+" "
    draw t$,75,60,2
    dummy =get$(#5,0)
wend
```

vant à l'entrée de réception se traduit par un appel de la routine d'interruption Puls. La seule fonction de cette dernière est le positionnement d'un indicateur (*flag*) de bit. Ce bit pointe vers une impulsion en entrée jusqu'à ce qu'elle ait été lue et remise à zéro. On crée de cette façon, logiciellement, une sorte de prolongateur d'impulsion. Il devient possible ainsi de traiter des impulsions d'entrée de très courte durée, avec un rien de retard.

La routine de réception Getir est programmée d'une façon similaire à ce que l'on ferait dans le cas d'une entrée RS-232 directe. On commence, au départ, par remettre l'indicateur à zéro de manière à effacer toute impulsion parasite au cas où il y en aurait l'une ou l'autre. On attend ensuite le bit de début (*startbit*). Dès que l'on détecte la présence d'un état « 1 » de l'indicateur, on intercale une pause d'une longueur égale à 150% de la durée de bit. Dans ces conditions on atterrit, lors de la première interrogation de bit, au milieu du bit de donnée. Bien qu'une éventuelle impulsion d'entrée soit passé depuis longtemps déjà, elle n'est traitée que maintenant. Cette approche donne une tolérance relativement importante par rapport d'éventuels décalages chronologiques. Toute impulsion non entrée positionne (met à « 1 ») un bit dans l'accu, chaque impulsion fournit un bit à « 0 ». Le traitement des 7 bits suivants se fait à intervalle régulier, processus au cours duquel les bits de l'accumulateur subissent à chaque fois un décalage d'une position.

Le programme principal transfert directement les octets reçus au port 1. Après chaque transfert, on a lecture de l'état du port et envoi en retour par le biais de l'interface IrDA. Entre chaque opération d'émission et de réception il y a une pause de 1 ms de manière à donner au récepteur concerné un temps d'établissement au cours duquel il peut récu-

pérer du niveau d'impulsion plus important de son propre écho.

L'exemple proposé constitue une application fonctionnelle à 100% pour la commande et la lecture de ports. Ses utilisations potentielles vous de la commande de charges externes à l'interrogation de l'état d'interrupteurs, inverseurs et autres commutateurs. Si l'on ne doit utiliser que des entrées, il sera possible, par le biais de l'octet 255, de mettre simultanément le port en mode entrée et demander une réponse en retour.

La réalisation d'un appareil IrDA simple est... simple. Nous allons utiliser comme base la carte à 89S8252 Flash (ref [1]).

L'approche la plus élémentaire ne requiert guère plus qu'une diode IR à la sortie d'émission (P3.3) et un phototransistor à l'entrée de réception (P3.2). Le schéma de la **figure 2** reproduit l'électronique de cette version minimalist, mais rien n'interdit de la doter d'un module IrDA amélioré pour obtenir une portée plus importante.

## Communiquer avec le Palm

Un mini-ordinateur portable requiert un langage de programmation permettant de développer rapidement des petites applications servant à solutionner des problèmes quoti-

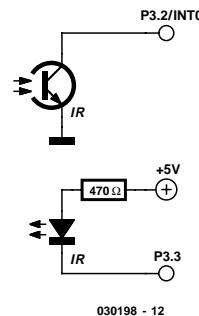


Figure 2. Interface IrDA élémentaire pour le 8051.

dens. Le BASIC, qui date du siècle dernier fait, à ce point de vue, parfaitement l'affaire. Il est partant d'autant plus réjouissant de constater qu'il existe, avec **HotPaw Basic**, un interpréteur Basic pour le Palm, que l'on peut essayer en toute tranquillité sans débourser un sou. On trouve en effet une version d'essai utilisable pendant un mois à l'adresse Internet :

[www.hotpaw.com/rhn/hotpaw](http://www.hotpaw.com/rhn/hotpaw).

Il suffit ensuite de payer de l'ordre de 20 \$US pour une version complète soit on continue d'utiliser la version d'essai avec une petite limitation : il est possible de mettre en mémoire et d'exécuter que 4 programmes seulement. La mise en oeuvre de (Virtual) Basic dans le monde de l'élec-

Broche	Dénomination	Fonction
1	Cathode IR	–
2	RxD	Sortie de données du récepteur, collecteur ouvert
3	V <sub>CC1</sub>	Tension d'alimentation 2,7 à 5,5 V
4	GND	Connexion de masse
5	SC	Sensitivity Control, Commande de sensibilité
6	NC	Non connecté
7	TxD	Entrée de commande de l'émetteur, actif au niveau haut
8	Anode IR	Connexion d'anode de la diode IR, 7 à 14 Ω vers V <sub>CC</sub>

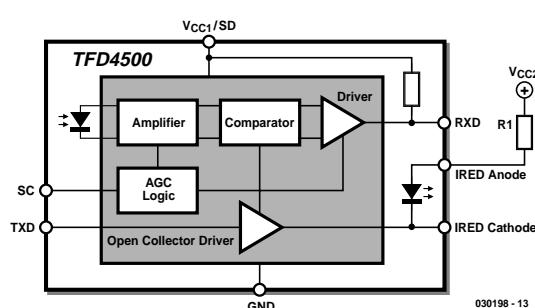


Figure 3. Synoptique du TFD4500.

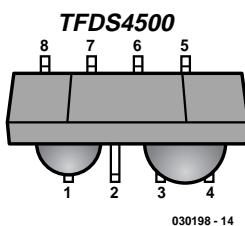


Figure 4. Forme du boîtier CMS du TFDS4500.

tronique est éloquemment décrite dans l'ouvrage cité en référence [2]. Les exemples de programmes en BASIC proposés ici sont disponibles au téléchargement depuis le site Elektor sous la forme de fichiers .txt. Pour pouvoir les utiliser il vous faudra recopier les listages dans le Bureau du Palm depuis un éditeur de texte en passant par une étape intermédiaire, celle d'un onglet. Lors du prochain HotSync, ces fichiers se retrouvent dans l'organiseur. Tous les modèles de PDA les plus récents tournant sous Palm OS sont dotés d'une interface IrDA. Il n'est pas difficile, avec **HotPaw Basic**, d'établir une communication par le biais de l'interface IrDA. Le **listage 2** fournit un exemple de commande et d'interrogation du port de la carte à 89S8252 Flash.

Le programme commence, après chaque envoi, par lire un éventuel écho. Pour ce faire, on traite le nombre d'octets se trouvant dans le tampon de l'interface. En cas de réception de plus d'un octet, le premier octet se trouvant dans le tampon doit être un écho de son propre envoi. Ce n'est que le second octet qui est alors la réponse du microcontrôleur. Après traitement, on a lecture d'un nouvel octet fictif (*dummy*) de manière à libérer le tampon d'éventuels octets erronés dus à des impulsions parasites.

Il existe, entre les différents modèles de Palm, des petites différences d'un point de vue matériel. Si le Palm M100 capte en principe l'écho de ses propres envois, le modèle plus récent, le Palm Zire, les élimine. Quoi qu'il en soit, le programme proposé fonctionne avec ces 2 modèles de Palm.

## Un circuit IrDA intégré

L'émetteur/récepteur IrDA intégré,

le TFDS4500, intègre un récepteur IR complet comportant une photodiode, un préamplificateur, un réglage de gain et un étage impulsional (**figure 3**). Il comporte en outre une diode d'émission dotée d'un étage de commande (*driver*). Le boîtier qui possède une lentille servant à la focalisation de la lumière est constitué d'un matériau filtrant qui n'est transparent que pour la lumière infrarouge. Ce composant relativement abordable facilite considérablement le travail avec IrDA. Le récepteur fournit des impulsions IrDA à logique négative. L'émetteur requiert d'être attaqué par des impulsions positives. Bien qu'il s'agisse d'un composant CMS (pour Montage en Surface), il reste relativement facile à monter sur une platine d'expérimentation à pastilles (**figure 4**).

Le **tableau 1** donne le brochage de ce composant. Il faudra prendre, entre les broches GND (Masse) et VCC (alimentation) un condensateur disposé le plus près possible, vu que sinon l'amplificateur de réception devient sensible aux parasites. Une résistance prise dans la ligne de la broche 8 sert à paramétriser le courant à travers la diode d'émission. Pour une valeur de  $10\ \Omega$  et à une tension de fonctionnement de 5 V le courant d'impulsion est de l'ordre de 250 mA. Il faut en outre, impérativement, un condensateur additionnel de  $4,7\ \mu F$  de manière à éviter la présence de parasites impulsionnels sur la tension d'alimentation.

La sortie du récepteur (broche 2) peut être reliée directement à l'entrée d'interruption du microcontrôleur. Dans le sens émission il nous faut cependant intercaler un inverseur vu que le TFDS4500 attend des impulsions positives. Il serait bien entendu possible de modifier la polarité des impulsions au niveau du programme, mais cela présenterait un inconvénient majeur. En principe, au repos, toutes les lignes de ports du processeur présentent, après une remise à zéro, un niveau haut. Dans ces conditions, l'émetteur IrDA serait activé pour la durée d'initialisation du processeur. Il est partant préférable d'intercaler un transistor de manière à inverser le signal.

En comparaison avec le circuit élémentaire de la figure 2, l'utilisation

d'un TFDS4500 présente d'indéniables avantages. Il est possible d'obtenir une transmission sûre à une portée pouvant atteindre de l'ordre de 1 mètre.

## Interface IrDA analogique

Il devient possible, à l'aide de l'UART logiciel présenté ici, de résoudre des tâches très diversifiées. Nous allons réaliser un module de mesure à base de TLC549, un CAN (Convertisseur Analogique/Numérique). Le microcontrôleur transmet à courir des données de mesure par le biais de l'interface infrarouge. Comme le processus se limite à une émission, il n'est pas nécessaire de disposer d'un canal de retour. L'interface ne requiert de ce fait qu'une simple diode d'émission IR mais pas de photo-récepteur. Un transistor PNP assure un courant impulsional élevé et une portée de l'ordre de près de 2 mètres. En dépit de cela, la consommation de courant moyenne ne dépasse pas 1 mA en raison de l'extrême brièveté des impulsions IrDA.

Le programme utilise une durée de pause de 250 mA et émet ainsi 4 valeurs de mesure par seconde, ce qui suffit largement dans le cas de processus à déroulement lent. Il est pos-

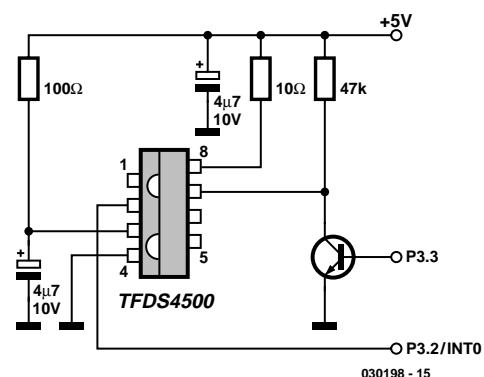


Figure 5. Adaptateur IrDA amélioré.

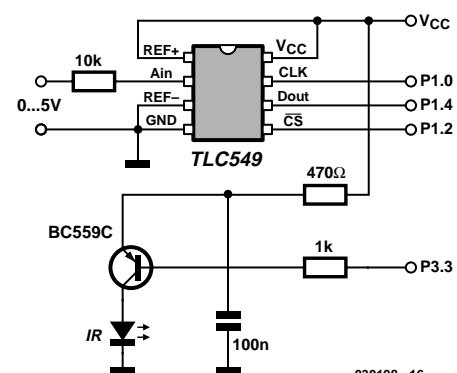


Figure 6. L'interface analogique.

### Listage 3. Transmission de valeurs de mesure analogiques.

```

' IrDAt1c549.BAS
'

$regfile = "89s8252.dat"
Dim N As Byte
Dim Addat As Byte
Declare Sub Adtlc549
Declare Sub Putirda

While 1 = 1
    Adtlc549
    Acc = Addat
    Putirda
    Waitms 250
Wend
End

Sub Adtlc549
    Clk Alias P1.0
    Cs Alias P1.2
    Adin Alias P1.4
    Dim Count As Byte
    Addat = 0
    Cs = 0
    Clk = 0
    Cs = 1
    Delay
    Cs = 0
    For Count = 1 To 8
        Shift Addat, Left, 1
        If Adin > 0 Then Addat = Addat + 1
        Clk = 1
        Delay
        Clk = 0
    Next Count
End Sub

Sub Putirda
$asm
    clr P3.3
    nop
    nop
    setb P3.3
    mov r2,#43
S1:
    djnz r2,s1
    mov r3,#8
S2:
    jb ACC.0,S3
    clr P3.3
    nop
    nop
    setb P3.3
    sjmp S4
S3:
    nop
    nop
    nop
    nop
    nop
S4:
    mov r2,#41
S5:
    djnz r2,S5
    rr a
    nop
    djnz r3,S2
    ret
$end Asm
End Sub

```

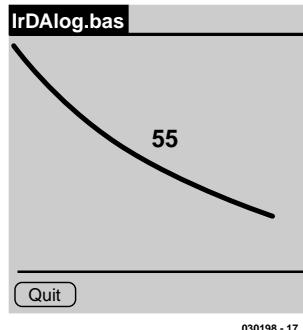


Figure 7. Les données plotées sur l'écran du Palm.

sible, si nécessaire, de d'accélérer le processus pour atteindre une vitesse de l'ordre de 500 mesures par seconde.

Le programme Palm suivant écrit en **HotPaw Basic** illustre une application type de l'appareil. On a visualisation graphique sur l'écran, en quasi-temps réel, des données de mesure où il est possible de les examiner comme sur un oscilloscope. La valeur de mesure du moment s'affiche en outre sous forme textuelle. La chronologie de cette application est totalement enfouie dans le programme du microcontrôleur, qui envoie, dans notre exemple, 4 valeurs de mesure par seconde. En fonction du cahier des charges de l'application, il est possible de procéder à des mesures plus rapides, qui seront transférées partiellement

dans le programme du Palm pour créer sur l'organiseur, par exemple, un axe de temps ajustable.

(030198)

### Bibliographie

- [1] Carte à 89S8252 Flash, Elektor
- [2] B.Kainka :  
Petites expériences d'électronique avec mon PC - Publitronic 2001

### Liens Internet

[www.hotpaw.com/rhn/hotpaw](http://www.hotpaw.com/rhn/hotpaw).

### Listage 4. Visualisation des données de mesure sur le Palm.

```

#IrDalog.bas
open "com1:",9600, ir as #5
while 1
gosub AD
Ualt =U
draw -1
for x=1 to 158
gosub AD
draw x-1,150-Ualt/2,x,150-U/2
Ualt=U
t$=str$(U)+"      "
draw t$,75,60,2
next x
wend
close #5
end

sub AD
U=-1
while U=-1
    U =get$(#5,0)
wend
return

```

## Récepteur IrDA pour le PC

Nombreux seront sans doute ceux d'entre nos lecteurs qui ne possèdent pas de Palm et dont le PC ne comporte pas d'interface IrDA et qui aimeraient quand même entrer dans le monde de la communication sans fil. Nous vous avons présenté, dans le numéro double de juillet/août 2003, un petit circuit à base de TFDU6102 de Vishay. Le circuit de la figure A à canal de réception seulement est encore plus simple. Il ne requiert que des composants standard (un phototransistor, une paire de transistors NPN et un condensateur servant à l'allongement des impulsions), circuit qui n'a guère plus à faire qu'à allonger les impulsions reçues. Le circuit est paramétré pour 9 600 bauds et pourra être connecté directement à l'interface série du PC, interface dont il dérive d'ailleurs sa tension d'alimentation. La ligne DTR doit être activée à cet effet. Il ne faut cependant pas s'attendre, dans le cas d'un circuit aussi rustique, à des miracles. Il est sensible à la lumière ambiante, il faut partant assombrir le couloir de transmission. De plus, la portée de 10 cm environ, n'a rien de très impressionnant.

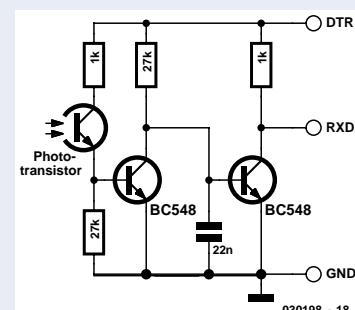


Figure A. Récepteur IrDA ultra-simple à l'intention d'un PC.

# Roulette à LED

## Les LED se substituent à la bille

Paul Goossens

Par essence, l'être humain est joueur. Vous arrive-t-il, occasionnellement, de tenter votre chance à une loterie voire même au casino du coin ? Qui d'entre nous n'a pas, un jour ou un autre, rêvé d'être riche sans avoir à trimer ? La roulette décrite ici ne vous permettra pas de gagner grand chose il est vrai, mais elle vous offre une chance, à peu de frais, de vous entraîner chez vous à la roulette et cela sans possibilité de tricher. Le microcontrôleur chargé de piloter les LED est en effet... incorruptible !

La roulette est un jeu pratiqué depuis de siècles et que l'on peut jouer partout dans le monde. Pour vous éviter d'avoir à parcourir des dizaines de kilomètres, nous vous proposons une roulette de salon électronique. Cette version électronique ne remplace que la roulette proprement dite, c'est-à-dire le réceptacle dans lequel la bille et les chiffres tournent en sens inverse l'un de l'autre. Il vous

faudra réaliser vous-même (voire les acheter) et le tapis et les fiches.

### Roulette

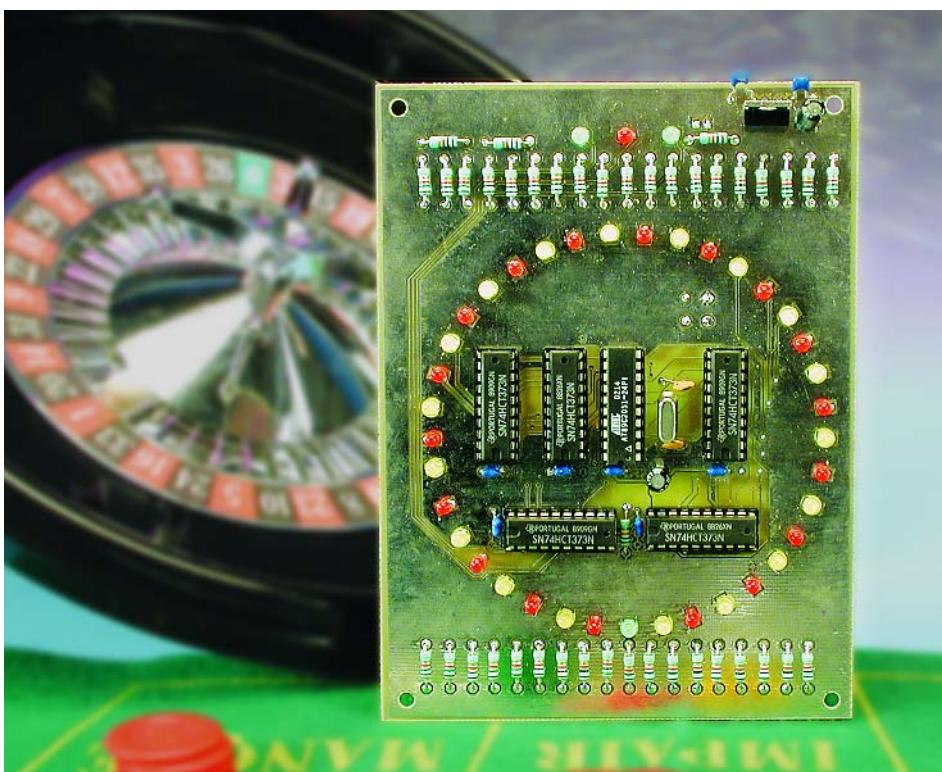
Nous n'allons pas vous faire l'injure de vous expliquer ce qu'est une roulette et comment on y joue. À chaque tour, une petite bille de métal qui

tourne dans le sens inverse du cylindre, car c'est là la dénomination officielle du bac central, détermine au plus grand des hasards, par la case dans laquelle elle s'arrête, quel est le chiffre gagnant. En fin de course la bille s'arrête dans l'une des 37 cases (1 à 36 et 0), chacune des cases ayant une couleur (noir, rouge ou vert). Les joueurs peuvent choisir un chiffre, une couleur, etc. sur lequel s'arrêtera, à leur avis, la bille.

Rares sont sans doute ceux d'entre nos lecteurs à jouer à la roulette chez eux avec un tel cylindre. Toute tolérance de construction se traduit par une anomalie de distribution des chances, de sorte que le jeu n'est plus totalement honnête – si tant est que l'on puisse parler d'honnêteté dans le jeu. Un cylindre de roulette a le double inconvénient de coûter très cher et d'être très encombrant. La variété électronique ne connaît que des avantages : son prix et la place qu'elle occupe sont faibles.

### L'approche

Sur ce montage, le déplacement de la bille est simulé par un certain nombre de LED disposées en cercle. Notre cylindre électronique ne se contente pas de simuler la rotation de la bille. Si vous avez déjà été dans un casino et que vous avez suivi de



près le fonctionnement de la roulette vous n'aurez pas manqué de constater qu'au début, en raison de sa vitesse élevée, la bille roule très souplement. Lorsqu'elle a perdu sa vitesse et qu'elle arrive au niveau des godets, elle ne se blottit pas immédiatement dans l'un d'entre eux mais saute de l'un à un autre de façon complètement aléatoire. Notre cylindre présente le même comportement, mais si vous préférez un comportement du type « Roue de la Fortune » vous pouvez le lui faire adopter par simple basculement d'un interrupteur.

Notre cylindre connaît une seconde option que vous ne trouverez pas au casino (où l'on indique cependant, sur une colonne d'affichage, les derniers chiffres sortis) : dès que la bille s'est arrêtée dans une case, notre roulette indique si le nouveau chiffre est supérieur ou inférieur au chiffre sorti juste auparavant. Il est possible ainsi, parallèlement aux méthodes de jeu que connaît la roulette, de parier sur le numéro à sortir, plus grand, plus petit que le précédent.

## Le schéma

On retrouve, en **figure 1**, le schéma de la roulette à cylindre. Son cœur est, comment pourrait-il en être autrement dès que l'on a affaire à une électronique numérique complexe, un microcontrôleur. Il s'agit dans le cas présent d'un AT89C2051, IC1, qui se contente d'un tout petit nombre de composants externes. C12 et R41 servent à la génération du signal de réinitialisation (reset) à appliquer dès la mise sous tension de la roulette. Le quartz X1 épaulé par C1 et C2 fait en sorte que l'oscillateur intégré dans IC1 fournit le signal d'horloge de 12 MHz. Les seuls organes de commande à la disposition de l'utilisateur sont un bouton-poussoir, S1, et un inverseur, S2, qui attaquent chacun une ligne d'E/S du microcontrôleur.

La visualisation de la bille se fait par le biais de 40 LED, D1 à D40, chacune de ces LED étant dotée de sa propre résistance de limitation de courant, R1 à R40. La commande de ces LED se fait par l'intermédiaire de verrous 8 bits, IC2 à IC6.

Les LED D1 à D37, prises en cercle, représentent les différentes cases chif-

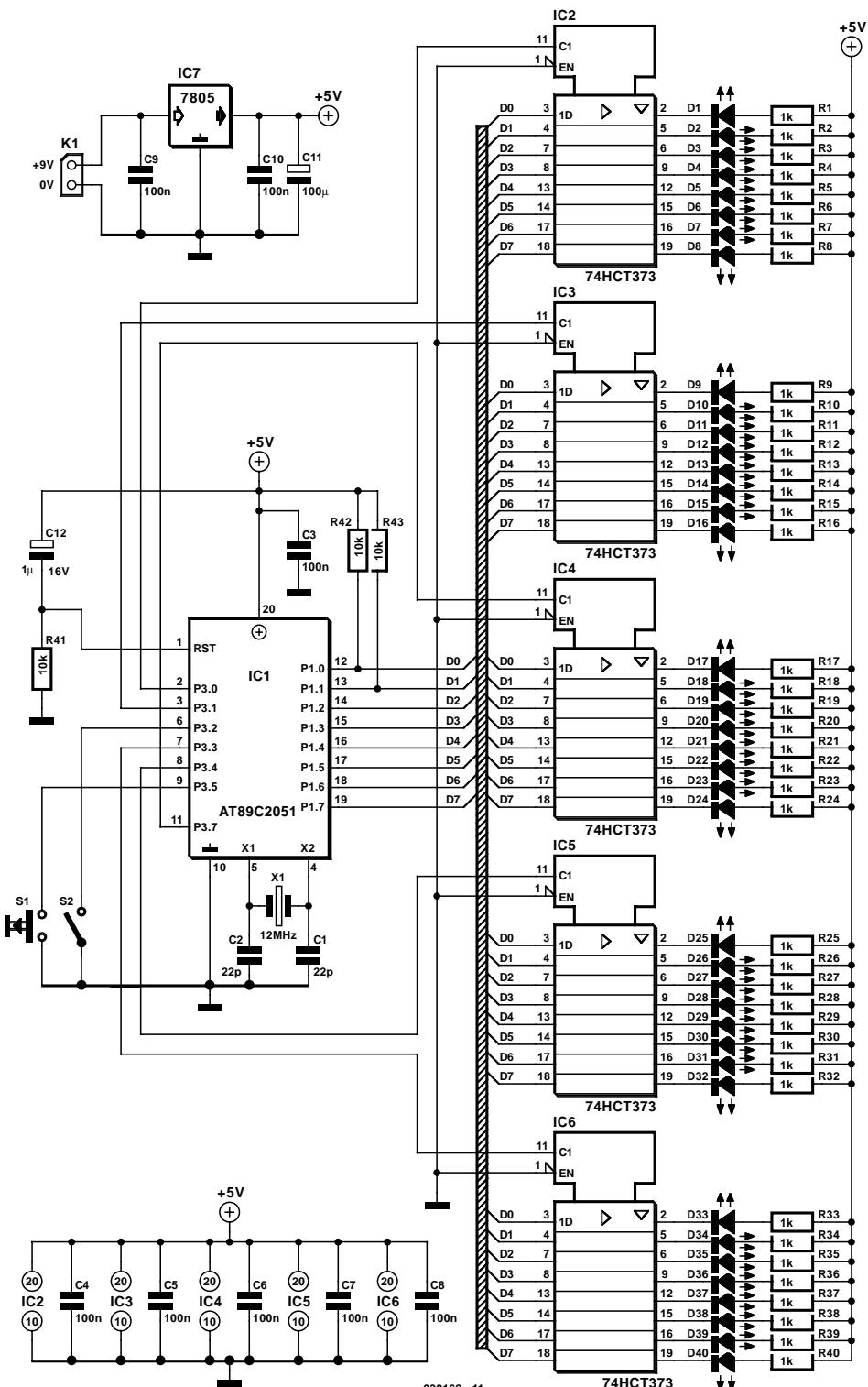


Figure 1. Le schéma de notre roulette : une quarantaine de LED pilotées par un microcontrôleur.

frées du cylindre. La LED D38 signale, par son allumage, que le chiffre qui vient juste de sortir est supérieur au chiffre sorti juste avant. D40 au contraire indique que le dernier chiffre à sortir est inférieur au chiffre qui le précédait. La LED D39 s'allume dès

que la bille a cessé son mouvement. L'alimentation de la roulette se fait par le biais d'une pile compacte de 9 V dont la tension de base est abaissée à la valeur de 5 V requise par les circuits intégrés numériques. On ne sera guère étonné de trouver, sur ce montage, nombre de condensateurs de tam-

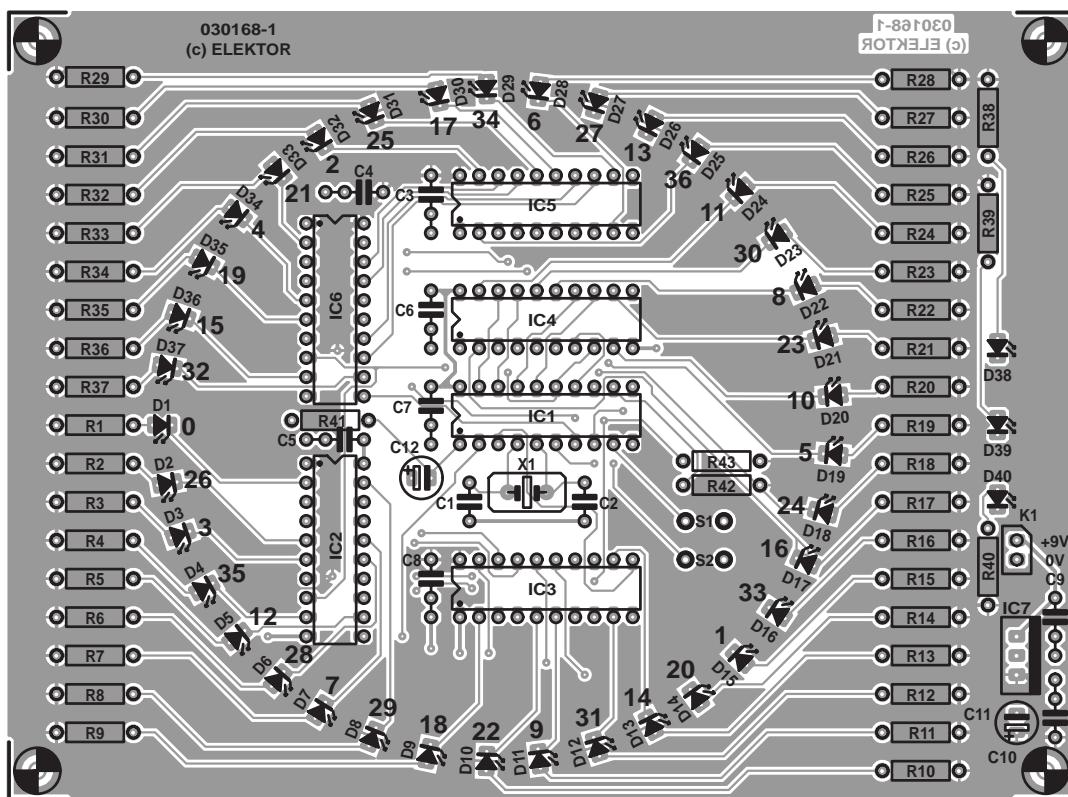


Figure 2. La platine double face reçoit tous les composants exception faite de la pile compacte 9 V.

ponnage et de découplage.

Une remarque concernant R42 et R43 : les lignes d'E/S du port P1 d'un AT89C2051 comportent toutes des résistances de forçage au niveau haut (*pull up*) internes, exception faite de P1.0 et P1.1. Ceci est dû au fait que les dites lignes peuvent également être utilisées en entrées analogiques. Comme sur le présent montage il est nécessaire de disposer de résistances de forçage au niveau haut sur toutes les lignes (les bits) du port 1, nous avons ajouté R42 et R43 de manière à ce que les bits 0 et 1 soient aussi attaqués par un signal de niveau haut défini lorsque la sortie ne se trouve pas au niveau bas.

## La réalisation

La **figure 2** vous propose la platine dessinée à l'intention de notre roulette électronique. Il s'agit d'une platine double face à trous métallisés sachant que sinon le nombre de ponts de câblage nécessaire pour obtenir une disposition circulaire des LED aurait été beaucoup trop important.

La construction n'appelle guère de remarque. Il suffit tout simplement de souder les composants à leurs emplacements respectifs et l'affaire sera réglée. Il va sans dire qu'il faudra bien veiller à respecter la polarité des LED, des circuits intégrés et des condensa-

## Liste des composants

### Résistances :

R1 à R40 = 1 kΩ  
R41 à R43 = 10 kΩ

### Condensateurs :

C1,C2 = 22 pF  
C3 à C10 = 100 nF  
C11 = 10 µF/16 V radial  
C12 = 1 µF/16 V radial

### Semi-conducteurs :

D1,D38,D40 = LED 3 mm verte  
faible courant  
D2,D4,D6,D8,D10,D12,D14,D16,  
D18,D20,D22,D24,D26,D28,D30,  
D32,D34,D36,D39 = LED 3 mm

jaune faible courant  
D3,D5,D7,D9,D11,D13,D15,D17,  
D19,D21,D23,D25,D27,D29,D31,  
D33,D35,D37 = LED 3 mm rouge  
faible courant  
IC1 = 89C2051-I2PC (programmé  
**EPS030168-41**)  
IC2 à IC6 = 74HCT373  
IC7 = 7805CP (boîtier TO-220)

### Divers :

K1 = embase à 1 rangée de 2 contacts  
pour connecteur à pression + pile  
9 V, éventuellement interrupteur  
marche/arrêt  
S1 = bouton-poussoir à contact travail  
S2 = inverseur unipolaire  
X1 = quartz 12 MHz

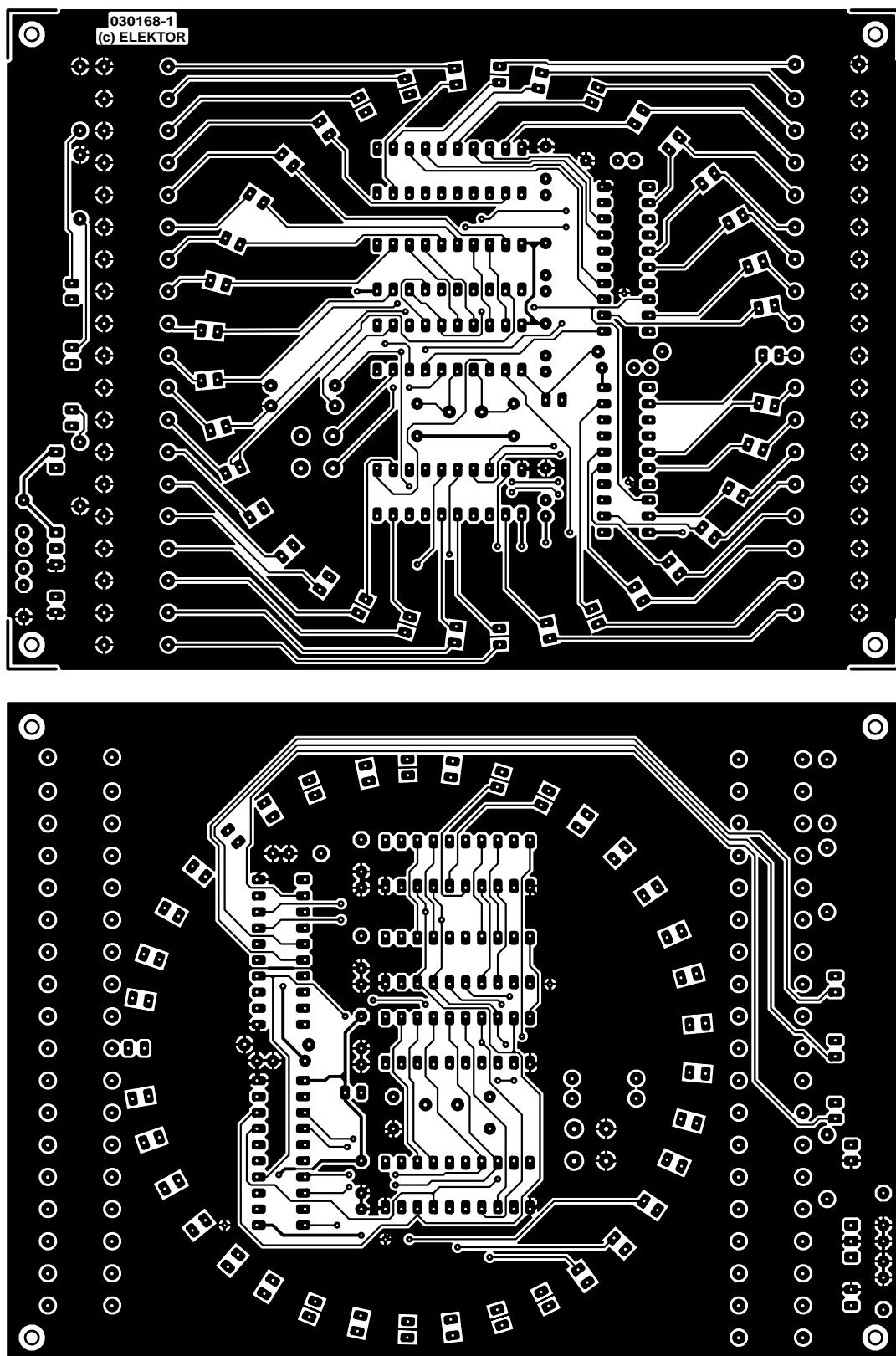
teurs électrochimiques. Rien n'interdit d'utiliser des supports pour les circuits intégrés mais cela n'a rien d'impératif si tant est que vous ayez une certaine expérience du soudage. Il est éventuellement envisageable de disposer les LED côté « pistes » de la platine, ceci dépendant bien entendu du boîtier dans lequel vous voulez l'implanter.

L'inverseur et le bouton-poussoir pourront être reliés au circuit par le biais

de câble souple, de manière à ce qu'ils puissent, le cas échéant, être montés sur le côté du boîtier. On pourra, au même endroit, planter l'interrupteur marche/arrêt que l'on prendra alors en série avec l'embase K1.

## En guise de conclusion

Le programme implanté dans le AT89C2051 a été écrit en C et compilé à l'aide du compilateur C51 de



Keil dont il existe aussi, nous l'avons signalé à plusieurs reprises déjà, une version d'évaluation gratuite disponible à l'adresse Internet : ([www.keil.com/demo/eval/c51.htm](http://www.keil.com/demo/eval/c51.htm)). Comme tant le code-source que le fichier .hex sont à votre disposition,

sous la dénomination de **EPS030168-11**, tant auprès des adresses habituelles que sur notre site Internet, rien ne vous empêche d'adapter le programme à vos propres exigences. Il vous faudra pour cela, outre le compilateur, avoir

un minimum de connaissances du langage de programmation C et disposer d'un programmateur pour le AT89C2051. Sachez qu'il existe également un contrôleur tout programmé (**EPS030168-41**) disponible auprès des adresses habituelles.

(030168)

# Projetmètre

## Multi-chronomètre à PIC16F84

Wilfried Wätzig

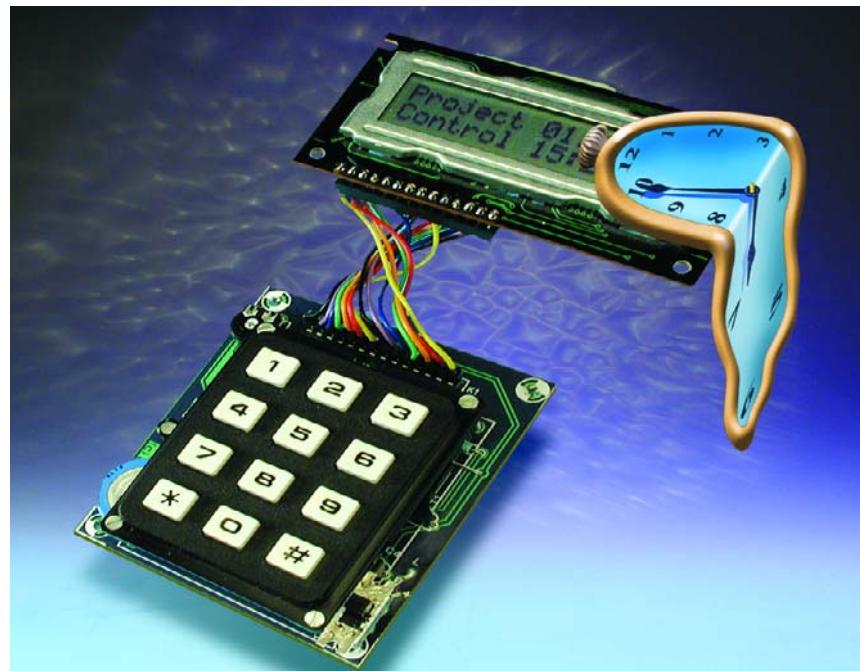
Ce chronomètre d'un type particulier mesure des intervalles de temps et les additionne pour donner une durée totale. Il utilise pour cela une horloge en temps réel et un tamponnage par GoldCap.

Si l'on veut connaître la durée totale investie dans un projet, il faut additionner les durées individuelles passées lors des différentes étapes. Avec un crayon et du papier on écrirait par exemple :

Étape 1 : le 7.10.03 de 8:13 à 8:55 (0:42)  
 Étape 2 : le 7.10.03 de 9:11 à 10:55 (1:44)  
 Étape 3 : le 8.10.03 de 19:25 à 21:05 (1:40)

et ainsi de suite. Il suffit ensuite d'ajouter les temps partiels (entre parenthèse) obtenus par les opérations de soustraction pour obtenir un temps total. C'est pour simplifier ce processus compliqué que nous avons développé ce projetmètre. Il est en mesure de comptabiliser les durées individuelles par projet et cela pour un maximum de 9 projets. Il est possible, grâce à lui, de mesurer n'importe quel processus ne requérant pas une mesure de temps très précise, la durée d'une compétition par exemple ou encore les durées d'activation d'appareils (domestiques) voire celles d'expériences.

Le cœur du système est un microcontrôleur 8 bits ô combien populaire, un PIC16F84 de Microchip, épaulé par une RTC (*Real Time Clock* = horloge en temps réel) du type DS1302 de Dallas Semiconductor. La visualisation des informations chronologiques se fait par le biais d'un affichage LCD à 2 lignes de 16 caractères, devenu classique lui aussi. La commande du projetmètre se fait au travers d'un clavier pour téléphone comportant les touches numériques 0 à 9 et les touches de fonction étoile (\*) et caret (#). Le clavier est structuré en une matrice de 3 colonnes sur 4 rangées connectée au port B, port auquel est également connecté le résonateur piézoélectrique BZ1, par le biais de T1, pour un



suivi acoustique de la saisie des chiffres par le clavier. La sélection du clavier et du résonateur se fait par le biais de RA4. RA2 met l'affichage en et hors-fonction. L'ajustable P1 sert, comme on en a l'habitude, à jouer sur le contraste de l'affichage.

### Fonctionnement

La saisie des durées d'un projet N est démarrée par une action successive sur les touches  $^{**}N$  (où N est un chiffre allant de 1 à 9). Par cette opération l'heure de base de l'horloge en temps réel est recopiée dans l'EE-

PROM du microcontrôleur. On voit alors apparaître sur l'affichage le message *N* suivi de la date et de l'heure de base. On pourra ensuite couper l'alimentation de l'appareil vu qu'un condensateur tampon continue d'alimenter l'horloge en temps réel.

La clôture du projet en cours se fait par l'entrée d'un double # (##). Cette action se traduit par la lecture de l'heure de base fournie par l'horloge en temps réel, la soustraction, de cette information chronologique, de l'heure de début de projet stockée en mémoire et l'addition du résultat

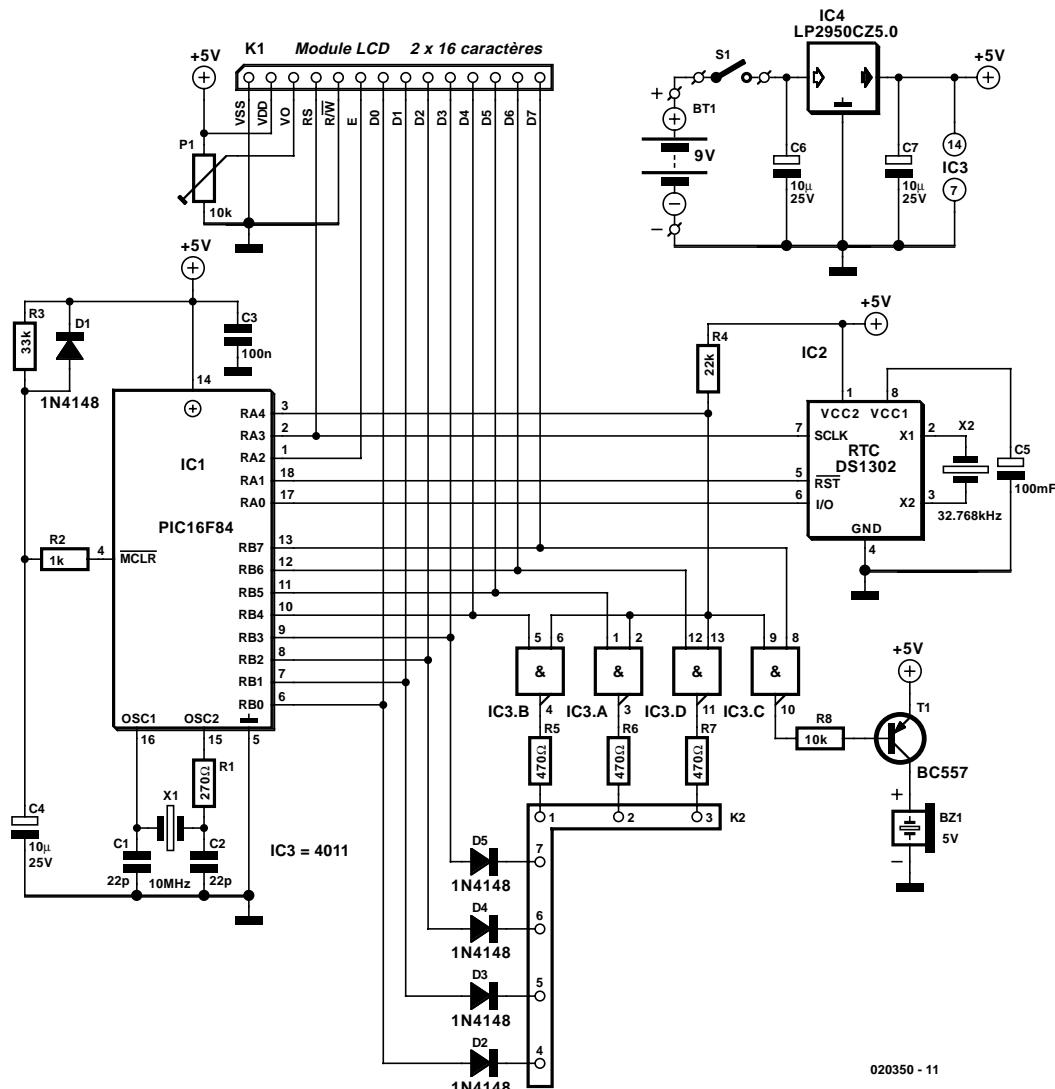


Figure 1. Le coeur de ce montage est un microcontrôleur 8 bits très populaire, un PIC16F84 de Microchip et un circuit RTC (Real-Time-Clock) DS1302 de Dallas Semiconductor.

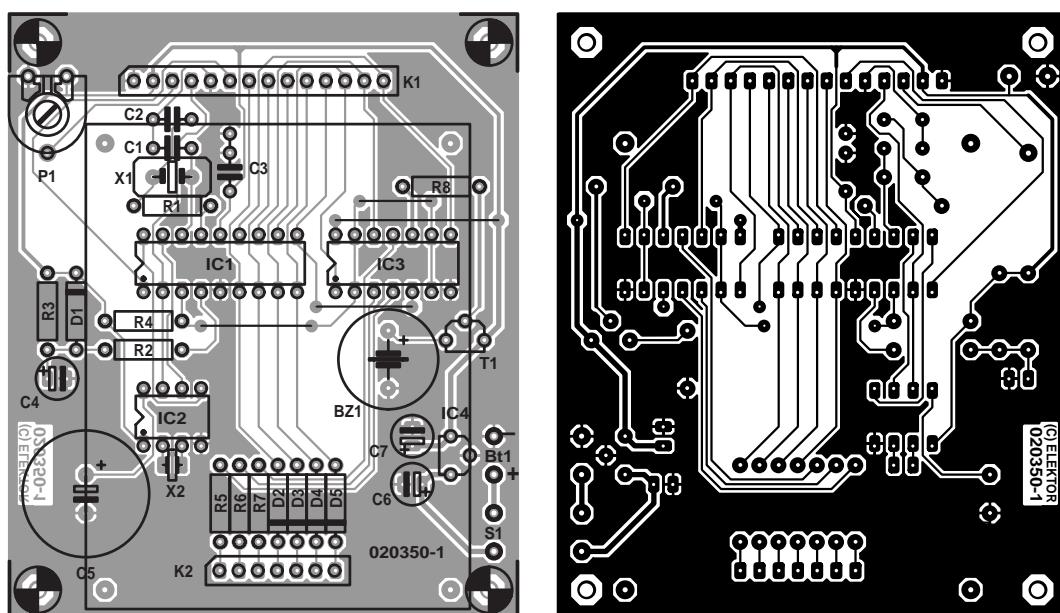


Figure 2. Dessin de platine et sérigraphie de l'implantation des composants.

## Liste des composants

### Résistances :

R1 = 270  $\Omega$   
 R2 = 1 k $\Omega$   
 R3 = 33 k $\Omega$   
 R4 = 22 k $\Omega$   
 R5 à R7 = 470  $\Omega$   
 R8 = 10 k $\Omega$   
 P1 = ajustable 10 k $\Omega$

### Condensateurs :

C1,C2 = 22 pF  
 C3 = 100 nF  
 C4,C6,C7 = 10  $\mu$ F/25 V radial  
 C5 = GoldCap 100 mF/5,5 V

### Semi-conducteurs :

D1 à D5 = IN4148  
 T1 = BC557  
 IC1 = PIC16F84-10P  
 (programme EPS 020350-41)\*  
 IC2 = DS1302 (Dallas Semiconductor)  
 IC3 = 4011  
 IC4 = LP2950CZ5.0 (régulateur 5 V low-drop, TO92)

### Divers :

K1 = embase autosécable femelle à 1 rangée de 14 contacts  
 K2 = embase autosécable mâle à 1 rangée de 7 contacts  
 BZ1 = résonateur 5  $V_{CC}$   
 BT1 = pile compacte 9 V + connecteur à pression  
 X1 = quartz 10 MHz  
 X2 = quartz 32,768 kHz  
 S1 = interrupteur unipolaire  
 module LCD alphanumérique à 2 lignes de 16 caractères  
 clavier téléphonique (4x3) (tel que, par exemple, Conrad RFA 709840)  
 dessin de platine EPS-020350-I\*  
 (cf. [www.elektor.fr/pcbs/pcbs.htm](http://www.elektor.fr/pcbs/pcbs.htm))  
 Diskette 020350-II\*

\*Le dessin de platine et le programme du microcontrôleur (fichiers source et .hex) sont également disponibles au téléchargement sous <http://www.elektor.fr/dl/dl.htm>

obtenu à la durée totale du projet partiel présente en EEPROM. Le calcul se fait avec des informations chronologiques au format jour/heures:minutes.

Par une action sur l'une des touches numériques de 1 à 9 on obtient l'affichage, pendant 3 secondes, de la durée totale du projet numéroté de 1 à 9 correspondant. Le format est alors dd hh:mm (pour days (jours) et hours (heures)).

Notre projetmètre reconnaît les instructions de commande énumérées dans le tableau 1.

## Tableau I

### Instructions de commande

**N	Démarrage du projet #N
##	Arrêt du projet en cours
#0*N	Effacement des informations chronologiques concernant le projet #N
N	Affichage du tampon horodateur du projet N
#*n	n=0 secondes xx = 00 à 59
	n=1 minutes xx = 00 à 59
	n=2 heures xx = 00 à 23
	n=3 jour du mois xx = 01 à 31
	n=4 mois xx = 01 à 12
	n=5 jour de la semaine xx = 01 à 07
	n=6 année xx = 00 à 99
*#n	Lecture du tampon RTC #n



Il va sans dire qu'il faudra, lors de la première utilisation du projetmètre, informer l'horloge en temps réel de la date et de l'heure exactes. Il est recommandé d'opter pour la succession des données logique suivante : année, mois, jour, heure, minute, seconde. Il faudra veiller à mettre l'information des secondes à 00 pour obtenir le démarrage de l'horloge en temps réel. En effet, après une réinitialisation (première mise sous tension), l'horloge en temps réel met le registre des secondes à 80 et entre dans une boucle d'attente.

La consommation de courant de l'électronique est de quelque 4 mA, de sorte qu'une pile compacte de 9 V permet de l'utiliser pendant une période relativement longue. Le régulateur de tension IC4 devrait être de type à faible perte de tension (low-drop) de manière à utiliser les

derniers milliwatts que puisse fournir la pile.

## La construction

Nous avons dessiné, pour le projetmètre, un circuit imprimé largement dimensionné dont la **figure 2** donne la sérigraphie de l'implantation des composants et le dessin des pistes. Nous ne nous attendons pas à des problèmes lors de la mise en place des composants pour peu que l'on fasse bien attention à la polarité de ceux qui en ont une. Les 3 circuits intégrés pourront prendre place sur un support. Le clavier matriciel vient s'enficher dans l'embase à 7 contacts K2, le module LCD venant lui se cliquer dans l'embase K1. On peut réaliser ainsi un appareil extrêmement compact qui sera doté d'un petit boîtier en plastique.

(020350)

# MAX4684/MAX4685

## Double commutateur analogique

### MAX4684/MAX4685

#### Double commutateur analogique

##### Fabricant :

Maxim [www.maxim-ic.com](http://www.maxim-ic.com)

<http://pdfserv.maxim-ic.com/en/ds/MAX4684-MAX4685.pdf>

##### Description :

Le MAX4684/MAX4685 est un double commutateur analogique unipolaire (SPDT = *Single-Pole/Double-Throw*) pour tensions faibles capable de travailler à une tension d'alimentation asymétrique comprise entre 1,8 et 5,5 V. Le MAX4684 possède une résistance de transfert maximale du commutateur normalement fermé de  $R_{ON} = 0,5 \Omega$ , alors que cette caractéristique passe à  $R_{ON} = 0,8 \Omega$  commutateur normalement ouvert. Dans le cas du MAX4685, ces valeurs sont identiques; quelle que soit la situation,  $R_{ON} = 0,8 \Omega$ . Cette résistance de transfert a été déterminée à une tension d'alimentation de +2,7 V.

Les commutateurs des 2 variantes travaillent (avec un délai de 2 ns par rapport au signal logique) en coupure avec un temps de montée de  $t_{On} = 50$  ns

et un temps de chute de  $t_{Off} = 40$  ns (à +3 V). Les entrées de logique numérique sont compatibles logique 1,8 V à une tension d'alimentation comprise entre +2,7 et 3,3 V.

##### Applications :

Commutateur de casque

Lecteur MP3

Appareils alimentés par pile

Remplacement de relais

Distribution de signaux AV (Audio/Vidéo)

Circuiterie de commutation

Carte PCMCIA

Téléphones mobiles

Modems

##### Application :

Inverseur de polarité automatique Elektor I/2004

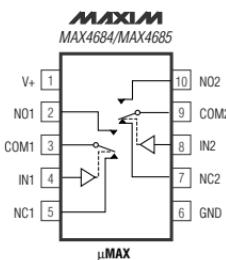
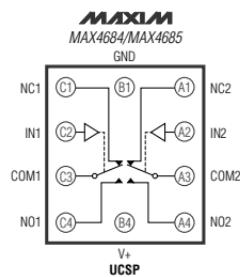
##### Boîtiers et brochage :

Suffixe	Boîtier	Dénomination
EBC	12 UCSP (Chipscale package)	AAF
ETB	10 Thin QFN (3x3)	AAG
EUB	10 mMAX*	—

\*Utilisable pour soudage au fer à souder

##### Caractéristiques techniques :

Résistance de transfert $R_{ON}$ contact repos	MAX4684, $V_+ = +2,7$ V	max. $0,5 \Omega$
	MAX4685, $V_+ = +2,7$ V	max. $0,8 \Omega$
Résistance de transfert $R_{ON}$ contact travail	$V_+ = +2,7$ V	max. $0,5 \Omega$
Dérive de $R_{ON}$ sur l'ensemble de la plage de signal		max. $0,15 \Omega$
Alimentation asymétrique		+1,8...+5,5 V
Traitement du signal rail à rail		
Compatible logique 1,8 V		
Diaphonie faible	100 kHz	-68 dB
Isolation si désactivé élevée	100 kHz	-64 dB
Taux de distorsion harmonique totale		0,03 %
Consommation de courant		max. 50 nA
Faible courant de fuite	$T_A = +25$ °C	max. 1 nA



IN_	NO_	NC_
0	ouvert	fermé
1	fermé	ouvert

### Description des broches :

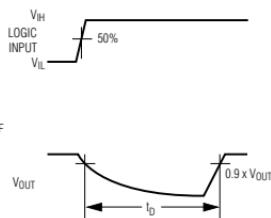
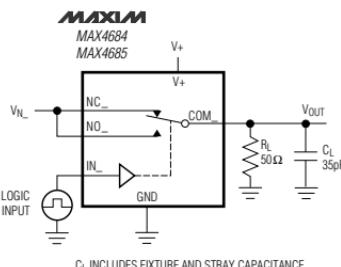
Nom	Broche		Fonction
	UCSP	µMAX	
NC_	A1, C1	5, 7	Commutateur analogique normalement fermé
IN_	A2, C2	4, 8	Entrée de commande numérique
COM_	A3, C3	3, 9	Commutateur analogique broche commune
NO_	A4, C4	2, 10	Commutateur analogique normalement ouvert
V+	B4	1	Tension d'alimentation positive
GND	B1	6	Masse

### Entrée de commande numérique

Les entrées de commande logique du MAX4684/MAX4685 acceptent des tensions allant jusqu'à +5,5 V quelle que soit la tension d'alimentation. On peut ainsi à une tension d'alimentation de +3,3 V par exemple, moduler IN de la masse à 5,5 V. La modulation jusqu'aux limites de la tension d'alimentation minimise la consommation de courant. Dans le cas d'une tension d'alimentation de +1,8 V, les niveaux logiques sont de 0,5 V (Bas) et de 1,4 V (Haut).

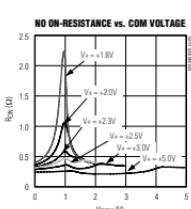
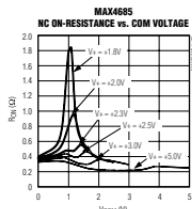
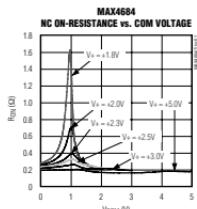
### Niveau de signal analogique

Les niveaux de signal analogiques situés à l'intérieur de la plage de la tension d'alimentation traversent les commutateurs en ne subissant qu'une très faible variation de la résistance de transfert. Les commutateurs sont bidirectionnels, de sorte que NO\_, NC\_ et COM\_ peuvent être utilisés tant en entrée qu'en sortie.



**Valeurs-limites :** (Toutes les tensions sont référencées à la masse)

V+, IN_	−0,3 à +6 V	
COM_, NO_, NC_	−0,3 à (V+ + 0,3 V)	
Courant permanent NO_, NC_, COM_	±300 mA	
Courant de crête NO_, NC_, COM_	impulsionnel 1 ms, Rapport cyclique 50%	±400 mA
	impulsionnel 1 ms, Rapport cyclique 10%	±500 mA
Dissipation permanente ( $T_A = +70^\circ\text{C}$ )	12-Bump UCSP µMAX 10 boches	909 mW 444 mW
Température de service	−40 à +86 °C	



**Caractéristiques électriques :** ( $V+ = +2,7 \text{ à } 3,3 \text{ V}$ ;  $V_{IH} = +1,4 \text{ V}$ ;  $V_{IL} = +0,5 \text{ V}$ ;  $T_A = T_{MIN} \text{ à } T_{MAX}$  sauf mention contraire)

Paramètre	Symbole	Conditions	$T_A$	Min.	Typ.	Max.	Unité
<b>COMMUTATEUR ANALOGIQUE</b>							
Plage du signal analogique	$V_{NO\_}$ , $V_{NC\_}$ , $V_{COM\_}$		*	0		$V+$	V
Résistance d'activation NC_	$R_{ON(NC)}$	$V+ = 2,7 \text{ V}$ ; $I_{COM\_} = 100 \text{ mA}$ ; $V_{NC\_} = 0 \text{ à } V+$	MAX 4684	+25 °C		0,3	0,5
			MAX 4685	+25 °C		0,45	0,8
			*			0,8	Ω
Résistance d'activation NO_	$R_{ON(NO)}$	$V+ = 2,7 \text{ V}$ ; $I_{COM\_} = 100 \text{ mA}$ ; $V_{NO\_} = 0 \text{ à } V+$	+25 °C			0,8	Ω
Dérive des 2 résistances d'activation	$\Delta R$	$V+ = 2,7 \text{ V}$ ; $I_{COM\_} = 100 \text{ mA}$ ; $V_{NC\_}$ ou $V_{NO\_} = 1,5 \text{ V}$	+25 °C			0,06	Ω
			*			0,06	Ω
Dérive de la résistance d'activation NC_	$R_{FLAT(NC)}$	$V+ = 2,7 \text{ V}$ ; $I_{COM\_} = 100 \text{ mA}$ ; $V_{NC\_} = 0 \text{ à } V+$	MAX 4684	*		0,15	Ω
			MAX 4685	*		0,35	Ω
Dérive de la résistance d'activation NO_	$R_{FLAT(NO)}$	$V+ = 2,7 \text{ V}$ ; $I_{COM\_} = 100 \text{ mA}$ ; $V_{NO\_} = 0 \text{ à } V+$	*			0,35	Ω
Courant de fuite commutateur NO_ ou NC_ ouvert	$I_{NO\_OFF}$ $I_{NC\_OFF}$	$V+ = 3,3 \text{ V}$ ; $V_{NO\_}$ ou $V_{NC\_} = 3 / 0,3 \text{ V}$ ; $V_{COM\_} = 0,3 / 3 \text{ V}$	+25 °C	−1		1	nA
			*	−10		10	nA
Courant de fuite commutateur COM_ fermé	$I_{COM\_ON}$	$V+ = 3,3 \text{ V}$ ; $V_{NO\_}$ ou $V_{NC\_} = 3 / 0,3 \text{ V}$ ou ouvert; $V_{COM\_} = 0,3 / 3 \text{ V}$ ou ouvert	+25 °C	−2		2	nA
			*	−20		20	nA

**Caractéristiques électriques :** ( $V_+ = +2,7$  à  $3,3$  V;  $V_{IH} = +1,4$  V;  $V_{IL} = +0,5$  V;  $T_A = T_{MIN}$  à  $T_{MAX}$  sauf mention contraire)

Paramètre	Symbol	Conditions		$T_A$	Min.	Typ.	Max.	Unité
<b>CARACTÉRISTIQUES DYNAMIQUES</b>								
Temps d'activation	$t_{ON}$	$V_+ = 2,7$ V; $V_{NO\_o.} = 1,5$ V; $R_L = 50 \Omega$ ; $C_L = 35$ pF		+25 °C		30	50	ns
				*			60	
Temps de désactivation	$t_{OFF}$	$V_+ = 2,7$ V; $V_{NO\_o.} = 1,5$ V; $R_L = 50 \Omega$ ; $C_L = 35$ pF		+25 °C		25	30	ns
				*			40	
Délai de coupure de contact	$t_{BBM}$	$V_+ = 2,7$ V; $V_{NO\_o.} = 1,5$ V; $R_L = 50 \Omega$ ; $C_L = 35$ pF	*		2	15		ns
Isolation ouvert	$V_{ISO}$	$C_L = 5$ pF $R_L = 50 \Omega$ ; $f = 100$ kHz; $V_{COM\_} = 1$ $V_{RMS}$	+25 °C		-64			dB
Diaphonie	$V_{CT}$	$C_L = 5$ pF $R_L = 50 \Omega$ ; $f = 100$ kHz; $V_{COM\_} = 1$ $V_{RMS}$	+25 °C		-68			dB
Distorsion THD	THD	$R_L = 600 \Omega$ ; $IN\_ = 2$ $V_{SS}$ ; $f = 20$ Hz à 20 kHz	+25 °C		0,03			%
Capacité de désactivation NC <sub>_</sub>	$C_{NC\_}(OFF)$	$f = 1$ MHz	+25 °C		84			pF
Capacité de désactivation NO <sub>_</sub>	$C_{NO\_}(OFF)$	$f = 1$ MHz	+25 °C		37			pF
Capacité d'activation NC <sub>_</sub>	$C_{NC\_}(ON)$	$f = 1$ MHz	+25 °C		190			pF
Capacité d'activation NO <sub>_</sub>	$C_{NO\_}(ON)$	$f = 1$ MHz	+25 °C		150			pF
<b>ENTRÉES NUMÉRIQUES</b>								
Haut (High) logique	$V_{IH}$		*		1,4			V
Bas (Low) logique	$V_{IL}$		*			0,5		V
Courant de fuite Entrée IN <sub>_</sub>	$I_{IN\_}$	$V_{IN\_} = 0$ ou $V_+$	*	-I		I		$\mu$ A
<b>TENSION D'ALIMENTATION</b>								
Plage de la tension d'alimentation	$V_+$		*		1,8		5,5	V
Consommation de courant	I <sub>+</sub>	$V_+ = 5,5$ V; $V_{IN\_} = 0$ ou $V_+$	+25 °C	-50	0,04	50	nA	
			*	-200		200		

\*: Sur l'ensemble de la plage de température

# Alarme d'effraction

À indication individualisée par capteur

Erik Martens

Le montage décrit ici est celui d'une alarme d'effraction tout à la fois simple et efficace, sur laquelle une LED visualise l'état de chacun des capteurs. Un simple coup d'œil suffit pour apprendre quelles fenêtres et portes ne sont pas encore fermées avant d'activer l'alarme.

## Caractéristiques techniques :

- Nombre maximal de capteurs : 8
- Surveillance de chacun des capteurs et signalisation en cas d'activation ou de rupture de câble
- Contact anti-effraction (sabotage)
- Bouton « Panique »
- Temporisation de sortie : 60 secondes
- Temporisation d'entrée : 60 secondes
- Tension d'alimentation : 230 V secteur ou accu de secours
- Indication des LED :
  - Alarme activée
  - Sabotage
  - Accu de secours en fonction
- Sorties : 2 (12 V<sub>CC</sub>, 500 mA)
- Durée d'activation de l'alarme : 60 secondes

Avec le présent concept d'alarme d'effraction il est possible de surveiller un maximum de 8 capteurs ou faudrait-il plutôt parler de détecteurs. Une LED signale individuellement l'état de chacun des capteurs. Cette LED signale que le capteur est actif et que son câblage est en bon état. Il va sans dire que notre alarme d'effraction comporte un interrupteur permettant de l'activer et qu'elle est dotée d'un contact de tentative de sabotage et d'une paire de sorties permettant la commande d'une sirène et/ou d'un tube à éclats. Notre alarme comporte en outre un bouton « panique » classique sur ce genre d'installations.

## Le système

Le cœur de notre alarme d'effraction est un AT89C51, un microcontrôleur de l'écurie Atmel. C'est à lui que notre alarme doit ses

fonctionnalités; il procède en outre à un filtrage (logiciel) de toutes les entrées. Il faut qu'une entrée présente le même niveau pendant 30 ms pour que le programme du microcontrôleur le prenne en compte.

Un coup d'œil au schéma de la **figure 1** montre que l'alarme d'effraction est simple et ne nécessite qu'un minimum de composants.

## Les détecteurs

Il est possible de connecter un maximum de 8 détecteurs à l'installation d'alarme. On les retrouve sur la gauche du schéma. Au repos, les contacts de ces capteurs doivent être fermés (NC = *Normally Closed*). Il faut en outre connecter un contact anti-effraction pour chacun des capteurs. On dispose, sur le bornier de connexion de chacun des capteurs, K3 à K10, de la tension d'alimentation continue de +12 V.

## Mise en oeuvre

L'ouverture de l'interrupteur à clé S1 met l'alarme en fonction. Dès cet instant, le sablier de la temporisation dite de sortie commence à couler. Au cours de cet intervalle (d'une durée de 60 s) il est possible de quitter la zone protégée sans que l'alarme ne soit déclenchée. La désactivation de l'alarme se fait par fermeture de l'interrupteur à clé S1.

Si, une fois la temporisation de sortie écoulée et que partant l'alarme est en mode actif, on a ouverture du contact

de l'un des capteurs, on aura lancement de la temporisation d'entrée (de 60 s elle aussi). Normalement, le locataire légitime aura désactivé l'alarme pendant cette temporisation d'accès par le biais de l'interrupteur à clé S1. En cas d'effraction, l'alarme se déclenchera effectivement lorsque la temporisation d'accès sera arrivée à son terme. La seule manière d'interrompre l'écoulement inéluctable de la temporisation d'accès est de désactiver l'alarme par le biais de l'interrupteur à clé S1.

Si l'on se trouve dans une situation requérant, pour quelque raison que ce soit, le déclenchement de l'alarme on pourra utiliser le bouton-poussoir prévu à cet effet, S2. Que l'alarme soit active ou non, une action sur le bouton « Panique » se traduira par un déclenchement instantané de l'alarme. Au bout de 60 s ou après une nouvelle action sur le bouton « Panique » l'alarme se taira à nouveau.

## Sabotage

Le contact anti-effraction K11 à pour tâche, alarme activée, de détecter, dès qu'elle prend place, toute tentative de sabotage. Au nombre de celles-ci, mentionnons l'ouverture du coffret dans laquelle est logée l'alarme d'effraction, la coupure de câbles, etc. Le contact de sabotage est du type NC, étant partant fermé en situation normalement, de sorte que l'on pourra prendre en série toutes sortes de contacts et câbles

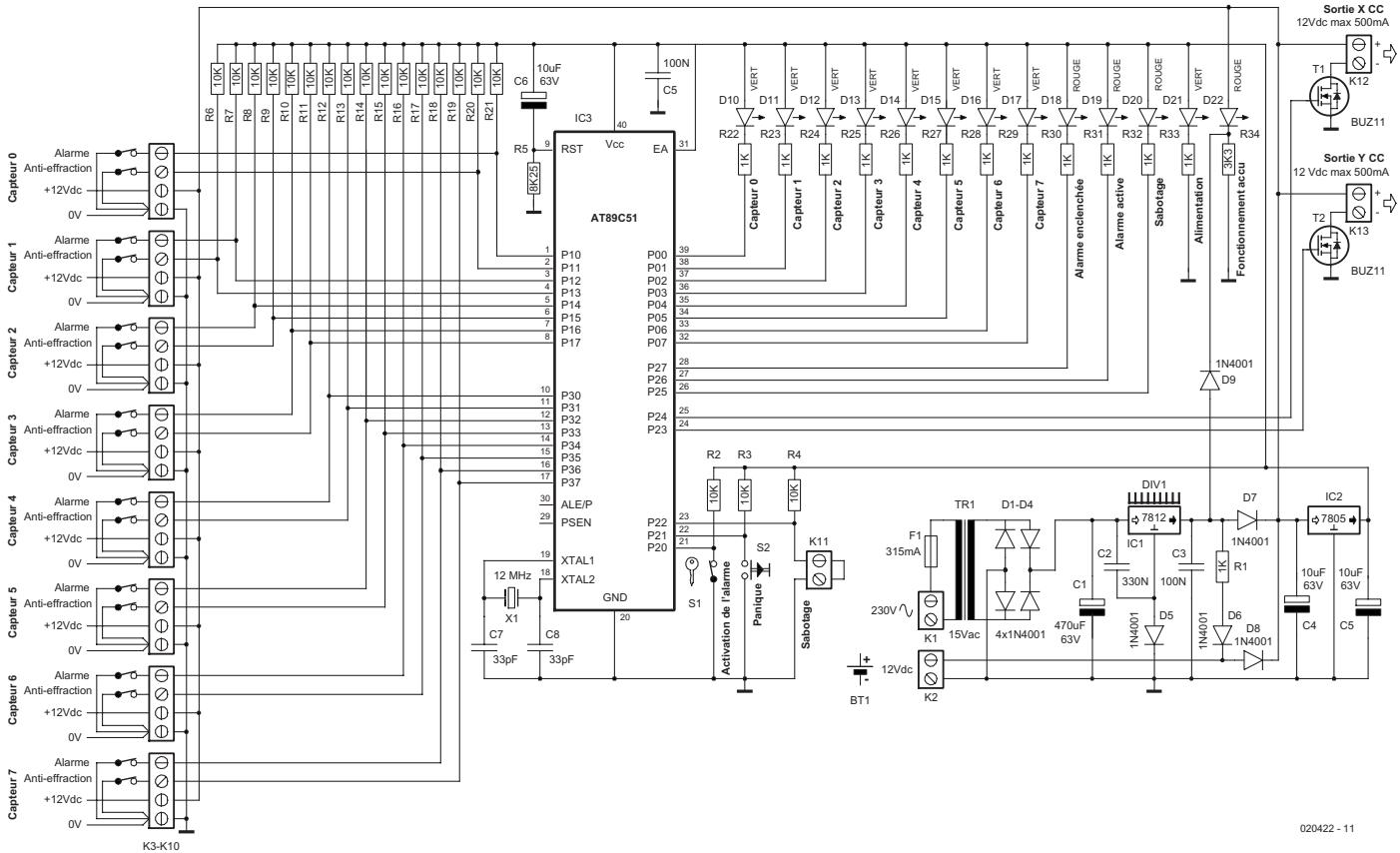


Figure 1. Les éléments principaux de notre alarme sont un microcontrôleur, un certain nombre de contacts de détection et une ribambelle de LED d'indication.

NC (les câbles allant aux sources de signaux d'alarme, sirène et/ou tube à éclats par exemple).

## Visualisation

8 LED, D10 à D18, visualisent l'état de chacun des capteurs qu'elles représentent. Tant que l'alarme est active, la LED correspondant au capteur ayant déclenché l'alarme restera allumée ou se mettra à clignoter en cas de rupture de câble.

Lors de l'activation de l'alarme, la LED « alarme enclenchée », D18, clignotera pendant la durée de la temporisation de sortie. Lorsque cet intervalle est écoulé, la dite LED reste allumée continûment. La LED D18 s'éteindra bien évidemment lorsque l'alarme sera désactivée.

La LED « alarme active », D19, clignote pendant la phase d'entrée et s'allumera fixement lorsque l'on a effectivement déclenchement de l'alarme. La seule manière d'éteindre D19 est de désactiver l'alarme par le biais de l'interrupteur à clé S1. Il est possible ainsi de toujours savoir s'il

y a eu déclenchement d'alarme et de déterminer quel détecteur (ou contact anti-effraction) a déclenché l'alarme.

La LED « sabotage », D20, s'allume en cas d'ouverture du contact anti-effraction K11. Cette LED aussi restera allumée jusqu'à ce que l'alarme soit désactivée

La LED « Fonctionnement accu », D22, signale, pour finir, que l'accumulateur au plomb BT1 a pris à son compte l'alimentation de l'alarme d'effraction.

## Sorties

L'alarme d'effraction est dotée, pour des raisons de fiabilité, de 2 sorties distinctes, K12 et K13. Chacune de ces sorties est pilotée par un BUZ11, T1 et T2 respectivement, et est en mesure de commuter un courant maximum de 500 mA sous 12 V, plus qu'il n'en faut pour alimenter des systèmes de génération d'alarme classique tels que lampes, tubes à éclats et autres sirènes. On pourra, si l'on a besoin de plus de puissance

ou que l'on veut mettre en circuit un périphérique travaillant à une tension d'alimentation différente, brancher directement aux sorties un relais 12 V par le biais duquel l'appareil en question sera mis en et hors-circuit.

### L'alimentation

Le montage est doté de sa propre alimentation secteur. Une combinaison classique transformateur + pont de redressement (constitué par les diodes D1 à D4) et un condensateur de lissage, C1, met à la disposition du régulateur intégré IC1 une tension d'entrée de quelque 18 V. La prise de la diode zener D5 dans la ligne de masse la tension de sortie fournie par IC1 est de 12,65 V environ. La diode D7 abaisse cette tension à 12 V environ. Un second régulateur de tension intégré, IC2, distille à partir de ces 12 V, une seconde tension régulée, de 5 V cette fois.

En cas de disparition de la tension secteur appliquée au bornier K1, l'accumulateur au plomb de 12 V BT1 connecté lui au bornier K2, prend instantanément à son compte l'alimentation de l'alarme. Cet accumulateur est rechargeé en permanence au travers de la résistance R1 et de la diode D6 dès que la tension du secteur est présente. Les diodes D7 et D8 empêchent la circulation de courants de charge dans la direction inverse.

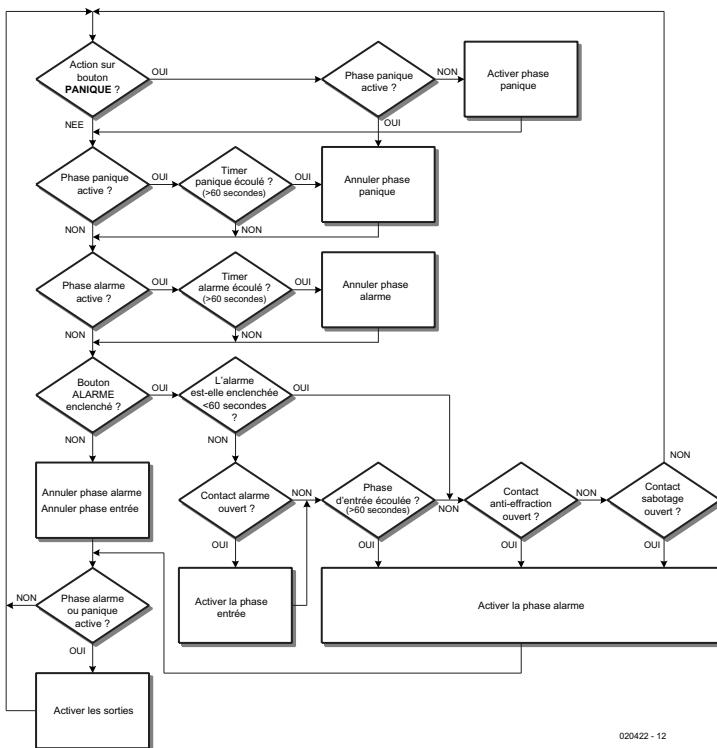


Figure 2. Cet ordinogramme illustre le principe de fonctionnement de l'alarme.

## Le programme

La structure du programme requis par l'alarme est relativement simple. Elle est rendue par l'ordinogramme de la **figure 2**. Tous comptes faits, il ne s'agit de rien de plus que d'un résumé logique de tous les points abordés dans les paragraphes précédents. Après réinitialisation du microcontrôleur ce dernier pilote l'ensemble des LED pendant une durée de 2 secondes. Ce processus permet de s'assurer que les LED sont en état de marche et qu'elles sont connectées correctement.

Il est possible, dans le code-source du programme, d'adapter à son goût les valeurs des différentes temporisations. Il s'agit des temporisateurs (timer) suivants :

Temporisation	variable	défaut (s)
Timer « Panique »	PANVAL	60
Timer « Sortie »	UITVAL	60
Timer « Entrée »	INLVAL	60
Activation Sortie X	ALXVAL	60
Activation Sortie Y	ALYVAL	60

Le programme de l'alarme d'effrac-

tion est disponible au téléchargement depuis le site Internet d'Elektor à l'adresse : [www.elektor.fr](http://www.elektor.fr) sous la dénomination **EPS020422-11**.

## Mise en coffret

Il est judicieux de répartir l'électronique de l'alarme d'effraction dans 2 coffrets de la manière illustrée par le croquis de la **figure 3**.

Le coffret de grande taille reçoit les sous-ensembles les plus importants du montage, y compris l'alimentation et l'accumulateur au plomb, mais pas le panneau de commande ni les dispositifs de visualisation. Ce coffret devra être placé à un endroit difficilement accessible. En fait, il s'agit d'un coffret dans lequel entrent et sortent des câbles et rien de plus. Veillez, lors de réalisation de l'alimentation secteur, au respect des règles de sécurité et à prévoir une isolation suffisante; n'oubliez pas de doter le câble allant à la prise secteur d'un bon dispositif anti-arrachement. Le coffret des interrupteurs et des LED sera quant à lui placé de façon à être visible. De par le concept du montage, les dommages que pourrait lui faire subir un cambrioleur resteront sans effet sur le fonctionnement de l'alarme (l'interrupteur à clé servant à l'activation de l'alarme est NC, normalement fermé). Il est recommandé bien entendu de prendre ce coffret dans la boucle anti-sabotage.

Il faudra, pour éviter que l'alarme d'effraction ne puisse être mise hors-fonction sans que cela ne soit signalé, tirer les câbles vers les détecteurs de manière à ce qu'ils soient d'accès difficile. On pourra également intégrer la boucle anti-sabotage dans les dits câbles.

La figure 3 montre une application bien pensée de la boucle anti-sabotage. On y voit la prise en circuit d'un interrupteur NC dans la boucle anti-sabotage. Cet interrupteur devra s'ouvrir en cas d'ouverture du coffret ou si ce dernier était maltraité.

Si vous n'utilisez pas tous les capteurs disponibles, le contact d'alarme et le contact anti-effraction de chacun des capteurs non utilisé devra être doté d'un pont de court-circuit (pour fermer le contact concerné).

(020422)

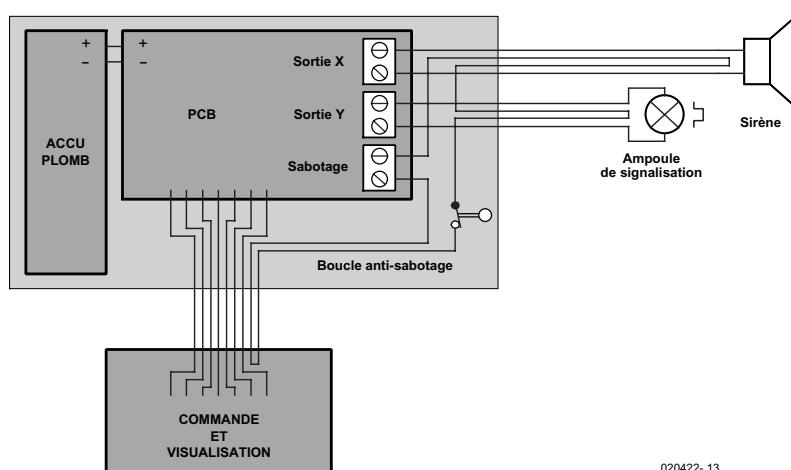


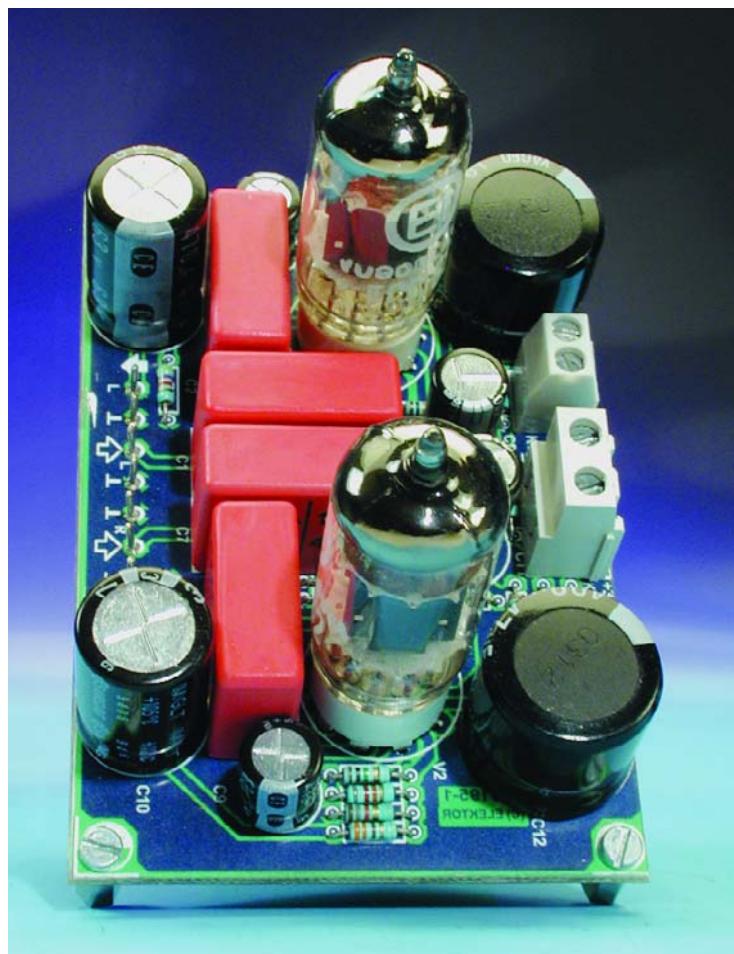
Figure 3. Lors de la réalisation pratique de l'alarme, les organes de commande et de visualisation sont séparés du corps de l'alarme proprement dit.

# Ampli OTL pour casque

## Ampli à tubes « sans fer »

Dr. phil. nat. Hansjörg Friedli

Il est possible de réaliser, à l'aide de composants modernes faciles à trouver, un amplificateur pour casque d'écoute au rendu superbe. Pas de transformateur de sortie, l'absence de « contre-réaction supra-nationale » et de bons condensateurs de couplage garantissent une réponse en fréquence bien linéaire ainsi que, de par le concept judicieux adopté, une diaphonie intercanaux faible.



### Valeurs de mesure

(avec filtre)

Tension d'alimentation	187 V
THD+N (1 mW/600 Ω)	2,3% droite 1,86% gauche
S/N (1 mW/600 Ω)	>93 dB >100 dBA
Ondulation de la tension de chauffage	30 mV <sub>CC</sub>
Puissance consommée totale	17 W

Les experts des tubes savent que même un tube de puissance, capable de délivrer quelques centaines de milliampères de courant d'anode, n'est pas en mesure d'attaquer un haut-parleur ayant une impédance de 8 Ω, en raison de la résistance interne d'un tube qui est de plusieurs kilohms. L'adaptation des 2 impédances est pratiquement toujours réalisée à l'aide d'un transformateur. En soi, il n'y aurait pas grand chose à critiquer à l'utilisation d'un tel convertisseur de tension selon la « loi des leviers » si chaque transformateur pris dans le trajet du signal ne signifiait pas des pertes au niveau du son. Si l'on veut minimiser ces pertes le plus possible, il faudra impérativement opter pour une technologie de bobinage complexe et très soignée, utiliser une tôle de bonnes caractéristiques pour l'armature du transformateur et un nouveau de bonne section.

Cependant, lorsque l'impédance de la charge n'est pas de quelques ohms seulement mais qu'elle est, comme c'est le cas avec de nombreux casques d'écoute, de plusieurs centaines d'ohms, et qu'en outre, la puissance requise en

sortie n'est pas très importante, l'amplificateur doit pouvoir se passer de transformateur de sortie ce qui lui permet de porter le qualificatif d'OTL (*Output Transformer Less*, sortie sans transfo). La charge à commander est attaquée directement par les tubes.

La version OTL présentée ici est une cathode-suiveuse, qui convient à la commande d'un (voire deux) casque(s) d'écoute d'une impédance nominale d'au moins  $300\ \Omega$ .

## Le circuit

Comme le montre le schéma de la **figure 1**, le tube utilisé ici comme amplificateur est une double triode du type ECC82 facile à trouver. On connaît des versions de meilleure qualité ou de durée de vie plus longue de ce type de tube sous la dénomination E802CC ou E82CC. Pour pouvoir moduler un casque d'écoute à un niveau suffisant on a besoin d'un étage préamplificateur. On utilise à cet effet le système triodes avec les contacts de support 1, 2 et 3. Le signal d'entrée arrivant sur la platine commence par rencontrer P1, un potentiomètre de  $50\ k\Omega$  (non représenté sur le schéma et qui sert d'organe de commande de volume) et est couplé directement par le biais du condensateur C1. La résistance R1 définit la tension de polarisation de grille négative nécessaire. Le facteur d'amplification (gain) est en fait défini par la résistance R8, le niveau maximal de la tension d'entrée l'étant par R2. La valeur attribuée à R9 est telle que le courant de repos d'anode se trouve dans une partie aussi linéaire que possible de la courbe de réponse.

La tension d'entrée inversée et amplifiée disponible est appliquée, par le biais de C2, à la grille d'un second système dont la résistance de cathode prend la forme d'une paire de résistances, R5 et R6. La résistance-série définit la résistance de fonctionnement alors que le rapport entre les 2 résistances permet un réglage correct de la tension de polarisation de grille. La paire R4/C3 servent à la découpler de la charge, à la réguler et appliquée, en provenance de R3, à la grille. Il circule, en fonction de la tension de grille, un courant d'anode répondant à la

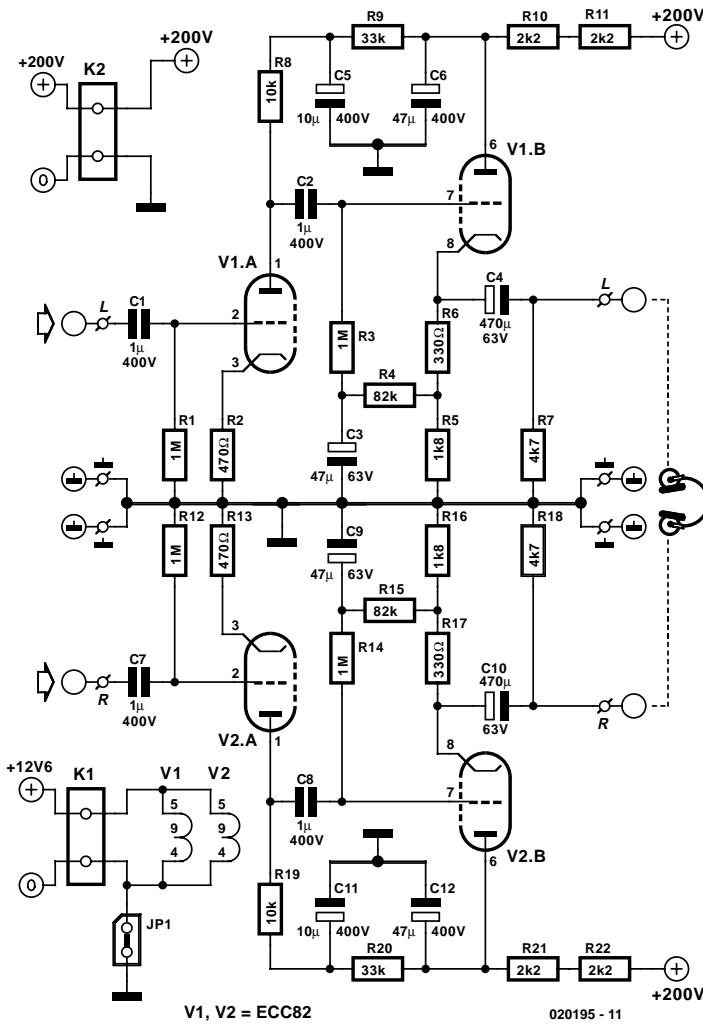


Figure 1. L'électronique de l'ampli est centrée sur 2 ECC82.

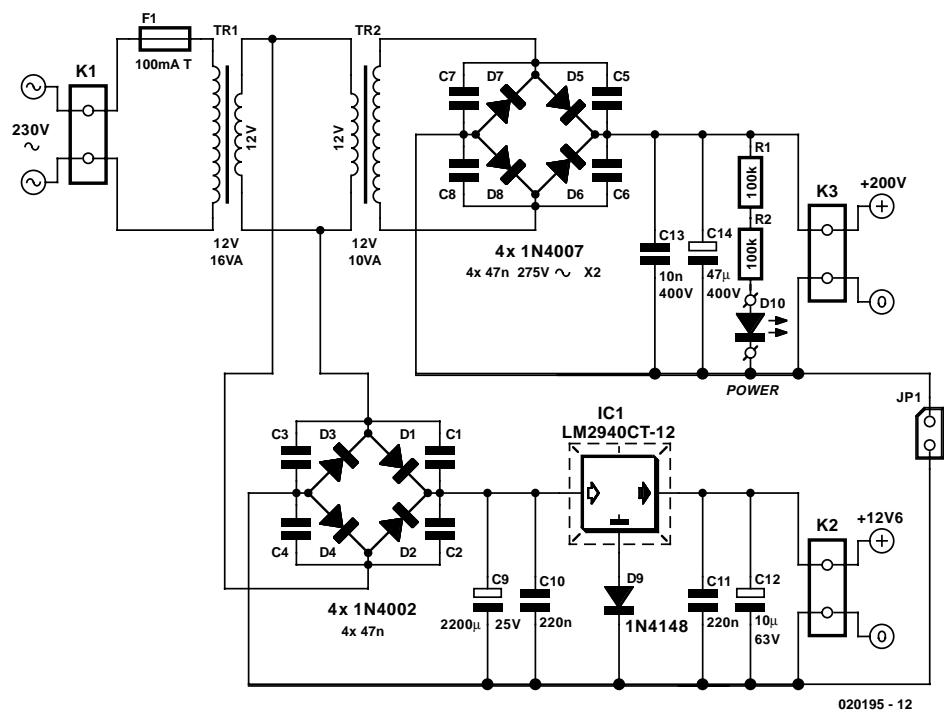


Figure 2. La Haute-Tension est fournie par une paire de transformateurs secteurs.

## Liste des composants de l'amplificateur

### Résistances :

R1,R3,R12,R14 = 1 MΩ  
 R2,R13 = 470 Ω  
 R4,R15 = 82 kΩ  
 R5,R16 = 1kΩ8  
 R6,R17 = 330 Ω  
 R7,R18 = 4kΩ7  
 R8,R19 = 10 kΩ  
 R9,R20 = 33 kΩ  
 R10,R11,R21,R22 = 2kΩ2

### Condensateurs :

C1,C2,C7,C8 = 1 µF/400 V (MKP4 ou MKS4 250 V)  
 C3,C9 = 47 µF/63 V radial  
 C4,C10 = 470 µF/63 V radial  
 C5,C11 = 10 µF/400 V radial (tel que, par exemple, B. ECA2GHG100 de Panasonic, Farnell 219-9320)  
 C6,C12 = 47 µF/400 V radial (chez Selectronic par exemple)

### Tubes :

V1,V2 = ECC82 (Conrad 120855) avec support Noval (Conrad 120529)\*

### Divers :

JP1 = embase autosécable à 2 contacts + cavalier  
 K1 = bornier à 2 contacts au pas de 5 mm (RM5)  
 K2 = bornier à 2 contacts au pas de 7,5 mm (RM7,5)

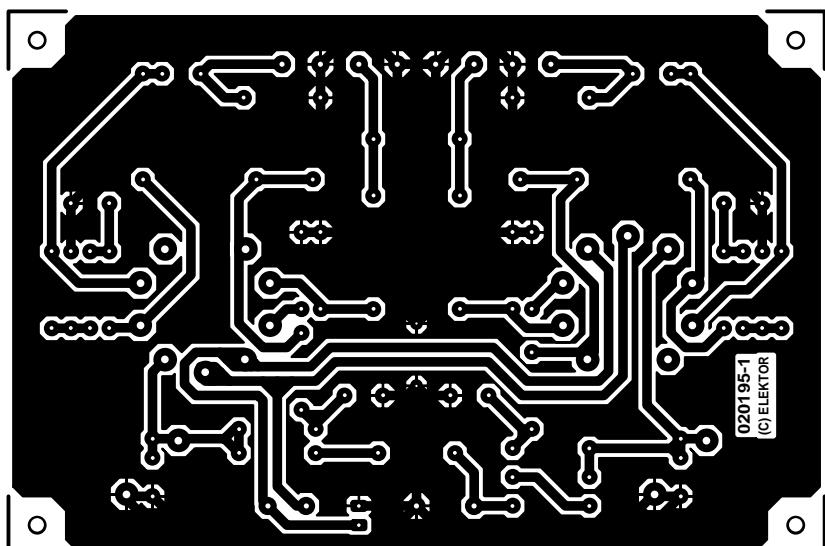
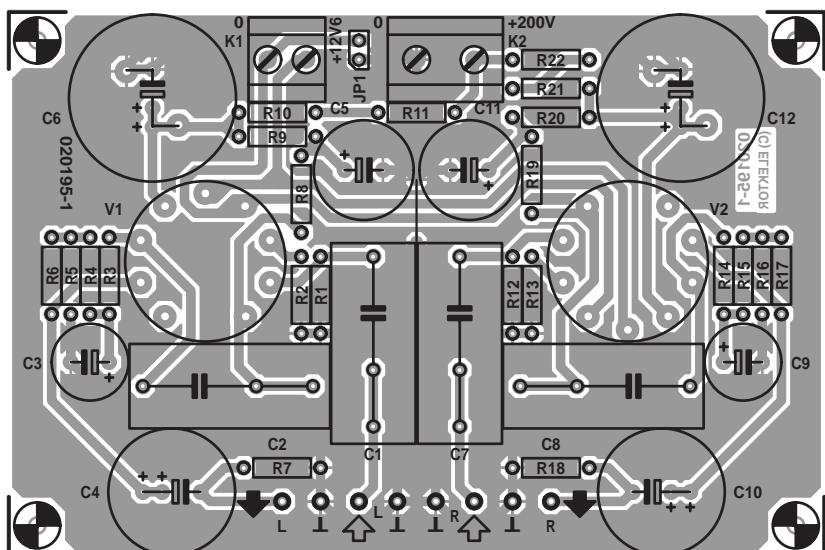


Figure 3. Dessin des pistes de la platine de l'amplificateur.

courbe caractéristique, courant produisant, au travers du couple de résistances R5+R6, une tension fortement proportionnelle. Cette tension attaque ensuite le casque d'écoute, par le biais du condensateur de couplage C4. R7 force la sortie, en ce qui concerne la tension continue, au potentiel de la masse et évite ainsi la production de craquements de commutation lors de la connexion du casque d'écoute.

## L'alimentation

La figure 2 représenté l'alimentation de l'amplificateur pour casque. La génération de la Haute-Tension (HT) fait appel à un transformateur secteur classique. L'enroulement faisant normalement office d'enroulement secondaire est connecté aux points 12 V du transformateur secteur proprement dit. Au niveau de l'enroulement primaire du trans-

formateur « renversé » (tout comme la crème du même nom), le transformateur d'anode, on dispose, à vide, d'une tension de quelque 2 000 V CA. Cette tension alternative (d'où le CA pour Courant Alternatif) subit un redressement avant d'être lissée par le condensateur C14. La tension servant ultérieurement de tension continue de chauffage est redressée par les diodes D5 à D8 et lissée par C15. Les condensateurs pris en parallèle sur les diodes servent à éliminer les parasites HF. À l'aide du régulateur de tension intégré du type LM2940CT12 (à faibles pertes) et d'une diode au silicium (Si) prise dans la ligne de masse on obtient sans trop de chichi, une tension de 12,6 V.

La LED D10 ne sert pas uniquement

à visualiser la mise sous tension, elle constitue, de concert avec R1 et R2, une charge à vide servant parallèlement à assurer la décharge de C14 au cas où il n'y aurait, occasionnellement, rien de connecté au bornier K3.

## Dessin des pistes

Le dessin des pistes de la platine d'amplification représenté en figure 3 a été conçu de manière à pouvoir utiliser les condensateurs « normaux » de type MKP4, mais aussi des condensateurs mieux adaptés aux applications audio mais plus encombrants (C1, C2, C7 et C8) au pas de 15 ou de 22,5, des MKP4 de WIMA à tension de service plus élevée par exemple. On pourra utili-

ser, pour les condensateurs de fort volume, C6 et C12, des versions au pas de 5 ou 7,5, voire de 10 mm pour des types enclicables.

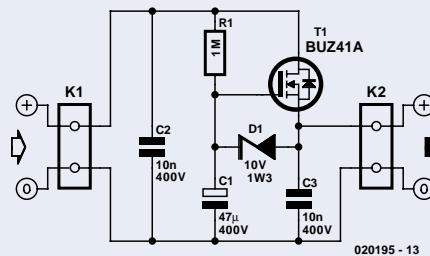
Les derniers types de condensateurs évoqués possèdent un diamètre plus important et captent ainsi moins de chaleur des tubes placés à proximité. On utilisera de préférence des condensateurs répondant bien aux impulsions et présentant une inductance faible; ils devront en tout état de cause être spécifiés pour une température de 105 °C.

La tension est appliquée à un canal au travers, respectivement, des paires R10+R11 et R21+R22. Les résistances « chauffent », en fonction de la tension d'alimentation, entre 0,18 et 0,25 W. Il faudra partant bien tenir compte de la puissance dissipable admissible. Il existe des résistances, qui à taille identique, les résistances 1/4 watt à film métal par exemple, sont capables de dissiper entre 0,3 et 0,4 W, les composants de la série PR01 de BCcomponents étant même en mesure de dissiper 1 W. Il n'est pas mauvais de penser à monter ces 4 résistances légèrement au-dessus du circuit imprimé pour leur permettre de mieux évacuer la chaleur.

Il a également été fait en sorte, lors du dessin de cette platine, à ce que les 2 canaux stéréo soient aussi écartés l'un de l'autre que possible. Les contacts de masse des 4 condensateurs de filtrage au niveau des anodes sont reliés, canal par canal, à une masse commune. À noter la présence, sur la platine de l'amplificateur, d'un unique pont de câblage (pris entre C5/C11 et C1/C7).

L'électronique de l'alimentation secteur prend place sur une seconde platine sur laquelle on pourra monter la platine de l'amplificateur en gigogne. Il est cependant à noter que l'amplificateur présente une certaine sensibilité aux champs parasites dont l'intensité dépend beaucoup du type de transformateur utilisé. Les composantes 50 Hz présentes dans le spectre (cf. les courbes de mesure A et B) sont une indication évidente de la présence des 2 transformateurs qui se trouvaient, lors de la mesure, à une vingtaine de centimètres. Les diodes D1 à D4 sont des 1N4002, solides déjà, mais il y a suffisamment de place

## Filtre pour la tension d'anode



Ce circuit est celui d'un filtre RC de puissance servant à diminuer l'ondulation (résiduelle) d'une tension d'alimentation non régulée. Ce filtrage se traduit inévitablement par la perte de quelques volts. Un FETMOS du type BUZ41A (500 V/4,5 A) prévu pour la haute-tension est monté en source-suiveuse simple. Par le biais de la paire R1/C1 la grille est amenée très progressivement à la tension de consigne; ce n'est qu'au

bout de 4 minutes (ce qui correspond à la durée de 5 durées RC) que l'on atteint 99% de la tension de sortie nominale. Cette temporisation n'a pas d'effet néfaste vu qu'il faut également aux tubes un certain temps pour chauffer et atteindre leur température de croisière. Un simple coup d'œil à la courbe B permet de constater l'effet positif de ce filtre. La chute de tension aux bornes de T1 dépend principalement de la tension de coupure (cut-off) entre la grille et la source; elle se situe aux alentours de 3,5 à 4 V. La grille est protégée contre les surtensions par la diode zener D1. La présence de C2 et C3 est indispensable pour éviter l'entrée en oscillation du circuit. Nous n'avons pas dessiné de platine pour ce petit circuit qu'il sera facile de réaliser sur un morceau de platine d'expérimentation à pastilles. La réduction de l'ondulation induite par ce filtre est impressionnante.

pour monter des modèles de diodes encore plus costauds. Les condensateurs C5 à C8 au pas de 15 doivent (!) être des exemplaires X2.

Si l'on veut éliminer toute ondulation

de la tension de chauffage, on pourra utiliser pour TR1, un transformateur de 15 V/20 VA, dont les dimensions sont légèrement supérieures ce qui ne permet pas de le monter sur la platine. Bien que dans ce cas-là la dissipati-



tion augmente, le radiateur prévu pour IC1 peut faire face à la situation. Nous avons également essayé un transformateur 15 V/16 VA qui dérivait cependant 23 VA du secteur (étant partant en surcharge évidente). Rien ne vous interdit, au contraire, d'essayer différents types de transformateurs; on peut obtenir des rapports très différents qui sont influencés par le type, la tension à vide et sous charge, la taille du noyau, le matériau utilisé. Le rapport des enroulements est toujours calculé en prenant la tension du secondaire hors charge; dans le cas d'un transformateur de 9 V (nominaux) cela donne de l'ordre de 12 V.

On pourra, pour compenser la puissance réactive, prendre un petit condensateur (de quelque 150 nF/250 VAC X2) en parallèle sur le côté HT de TR2. Ceci accroît même quelque peu la tension d'anode, la puissance consommée et le courant ne diminuant eux de 6 à 7% seulement.

## Liste des composants de l'alimentation

### Résistances :

R1,R2 = 100 kΩ

### Condensateurs :

C1 à C4 = 47 nF céramique  
 C5 à C8 = 47 nF/275 VAC/X2 RM15  
 C9 = 2 200 µF/25 V radial  
 C10,C11 = 220 nF  
 C12 = 10 µF/63 V radial  
 C13 = 10 nF/400 V RM7,5 ou 10  
 C14 = 47 µF/400 V radial (22215266109 BC Components, Farnell 322-7984)  
 (22215266479 BC Components, Farnell 322-8009)  
 ECA2GHG470 Panasonic (par ex. chez Farnell 319-9356)

### Semi-conducteurs :

D1 à D4 = IN4002  
 D5 à D8 = IN4007  
 D9 = LED rouge faible courant  
 IC1 = LM2940CT-12 avec radiateur tel que Fischer SK104 (50,8 mm)

### Divers :

JP1 = embase autosécable à 2 contacts + cavalier  
 K1,K3 = bornier à 2 contacts au pas de 7,5 mm (RM7,5)  
 K2 = bornier à 2 contacts au pas de 5 mm (RM5)  
 F1 = porte-fusible encartable + fusible 100 mA/retardé  
 TR1 = transfo secteur 12 V/16 VA tel que, par exemple, BV054-5383.0K (ERA)  
 TR2 = transfo secteur 12 V/10 VA tel que, par exemple, BV048-5383.0H (ERA)

L'approche à 2 transformateurs présente l'avantage d'une séparation de la tension de chauffage et de la

Haute-Tension. Il n'en cependant pas moins interconnecter les 2 potentiels de masse correspondants. Cette

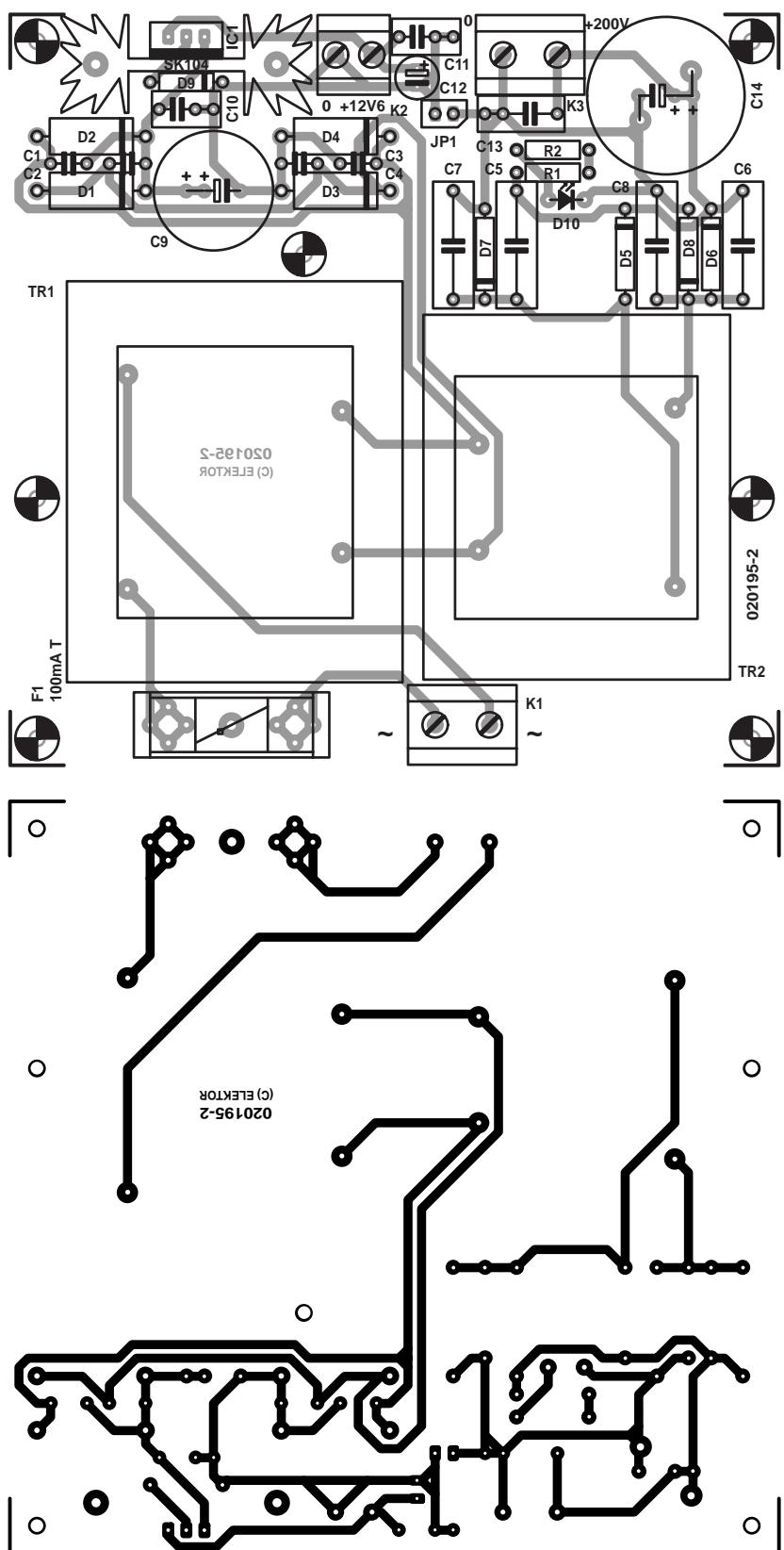


Figure 4. La platine de l'alimentation pourra être montée en sandwich sur celle de l'amplificateur.

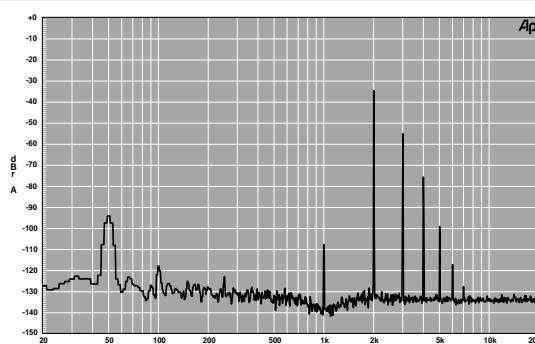
## Courbe et sonorité

La **courbe A** rend le spectre de fréquences d'un signal de 1 kHz (1 mW dans 600 W). Les distorsions (canal droit) atteignent 2,3%; elles sont principalement dues à l'harmonique du second ordre.

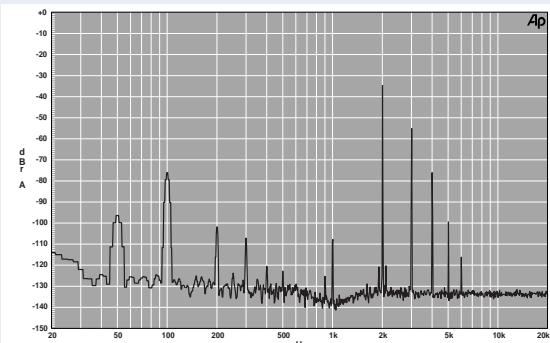
Il faut en outre remarquer la très grande influence de l'ondulation de la tension d'alimentation d'anode non régulée. Nous avons développé à cet effet un filtre actif (cf. l'encadré) qui élimine quasi-totalement la composante 100 Hz et les harmoniques. L'effet indéniable de ce filtre est visualisé par la **courbe B**. Pour finir, la **courbe C** rend la distorsion en fonction de la puissance de sortie. On constate que les distorsions augmentent assez linéairement avec la puissance de sortie. Aux alentours de 5 mW l'amplificateur arrive « en butée ».

Il est difficile, à l'aide de chiffres seuls, de décrire ce dont est capable un appareil audio. L'aspect marquant de la sonorité de l'amplificateur est sa dynamique : les parties à fort niveau sont présentes, les parties douces sont vraiment douces. Le son est chaud, spatial, aux niveaux élevés peut-être un peu moins précis dans la définition dans la partie supérieure du spectre de fréquences, ce qui pourrait se traduire par petite diminution du plaisir que peuvent offrir les voix d'opéra. Le son est très agréable et cela même si

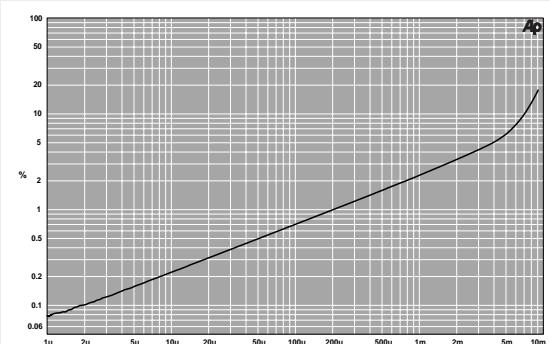
l'écoute dure un certain temps. Les graves ne posent pas de problème, en tout cas, une musique bien bardée de graves comme de la musique pop ou du jazz est extrêmement agréable à écouter !



**Courbe de mesure A.** Le spectre obtenu en l'absence de filtre de tension d'anode.



**Courbe de mesure B.** La prise dans le circuit du filtre s'est avérée payante : les composantes 100 Hz ont sensiblement diminué.



**Courbe de mesure C.** Les distorsions croissent de façon linéaire avec l'augmentation de la modulation.

connexion peut se faire soit au niveau de la platine de l'amplificateur soit au niveau de celle de l'alimentation (JP1), voire sur les deux.

### La réalisation

La mise en place des composants sur les platines n'appelle pas de remarque particulière, tous les composants trouvant place normalement, c'est-à-dire côté « composants ». Comme nous disions plus haut, le régulateur devra être refroidi. Sa surface de radiateur est reliée à sa broche centrale; dans le présent schéma, elle se trouve, en raison de la présence de la diode, 0,6 V au-delà du potentiel de la masse. Nous ne vous ferons pas l'injure de signaler qu'il faudra, dans le cas des condensateurs, veiller à respecter la polarité. Il faudra, dans le cas des 4 condensateurs de 1 F aussi, faire attention à la polarité cor-

recte. Comme la connexion de grille constitue le point le plus sensible d'un circuit à tubes, ce point devra être relié au feuillet intérieur du condensateur. Le feuillet extérieur est souvent identifié par un trait; il fait office de blindage. Les seules pièces externes restant à connecter sont les embases Cinch, le potentiomètre, le jack et l'interrupteur secteur, ce qui vous laisse une grande liberté quant au choix du coffret dans lequel prendra place l'ensemble du montage et le positionnement des différents organes de commande. Le dessin des pistes représenté en figure 3 présente une belle symétrie de sorte que rien n'empêche de l'exposer aux regards avec ses tubes et ses composants. Si vous ne voulez pas que la platine soit visible mais que vous aimeriez bien exposer les tubes, il vous faudra effectuer une gravure en miroir de la platine. Les supports pour les tubes

viendront dans ce cas-là prendre place côté cuivre de la platine où ils seront également soudés, le reste des composants étant monté comme d'habitude de sorte qu'ils seront, une fois montés, orientés vers le bas. Les polarités ne changent pas elles.

La totalité des lignes de 230 V doivent être réalisées en conducteur isolé de section suffisante. Il est également indispensable de veiller à l'isolation des points de soudure et de placer une plaquette d'isolation entre la tôle de montage et la face inférieure du transformateur. Il est important en outre de veiller à une bonne circulation de l'air. Le double interrupteur marche/arrêt est intercalé dans la ligne 230 V. Il faudra veiller, lors du positionnement des transformateurs, à les placer le plus loin possible des entrées. Si les transformateurs se trouvent trop près des entrées on ne manquera pas d'entendre, si les embases Cinch se trouvent en l'air et que l'on ouvre le potentiomètre, le crépitement clair typique du 50 Hz.

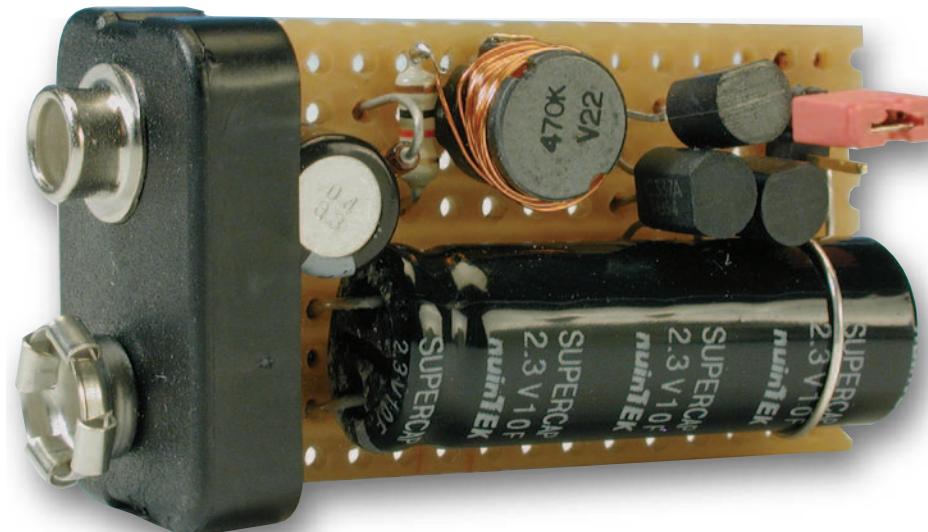
(020195)

# Condensateur en accumulateur

## Goldcap pour stockage d'énergie

Burkhard Kainka

Les Goldcaps constituent, avec leur énorme capacité, une alternative intéressante aux piles jetables et aux accumulateurs. Ils se laissent recharger très rapidement et peuvent être déchargés à un courant élevé. Leur seul inconvénient est de n'être conçu que pour des niveaux de tension faibles. Cependant il suffit d'un peu d'électronique pour résoudre de tels problèmes.



On trouve les piles compactes de 9 V dans des appareils à la consommation de courant faible et n'étant mis en fonction qu'occasionnellement. On ne sera partant guère étonné que la pile soit épuisée au moment précis où on en a le plus besoin. Il ne sert à rien dans ce cas-là non plus d'envisager l'utilisation d'accumulateurs CdNi vu que ce type d'accu supporte encore moins bien des périodes de décharge longues qu'une pile ordinaire. Une Supercap peut être la solution à ce genre

de problème vu qu'elle se laisse recharger très rapidement lorsque le besoin s'en fait sentir et qu'elle reste prête à rendre ses services même des années plus tard. Un petit convertisseur de tension rehausse la tension fournie par le condensateur à 9 V. De par la compacité d'une Supercap de 10 F/2,3 V, l'ensemble, électronique comprise, pourra prendre place sur une platine dont la

taille n'excède pas celle d'une pile compacte de 9 V.

Contrairement à ce qui est le cas avec les accumulateurs, les condensateurs acceptent d'être chargés et déchargés très rapidement (à ces courants de crête élevés), leur auto-décharge étant faible. Un condensateur est un dispositif de stockage d'énergie presque idéal, témoins son utilisation dans les flashes. Aucune pile ni accumulateur de taille équivalente n'est capable de fournir une quantité d'énergie aussi importante. Autre avantage, la charge et la décharge se font quasiment sans perte d'énergie. Il n'en faut cependant pas cacher un inconvénient : une prise d'énergie se traduit par une diminution de la tension aux bornes du condensateur. De ce point de vue, un accumulateur présente un comportement plus intéressant vu qu'il garde, pendant un certain temps, quasiment constante la tension qu'il fournit avant que celle-ci ne chute rapidement en fin de décharge. Il n'est partant envisageable de substituer un condensateur à un accumulateur qu'à condition d'utiliser un

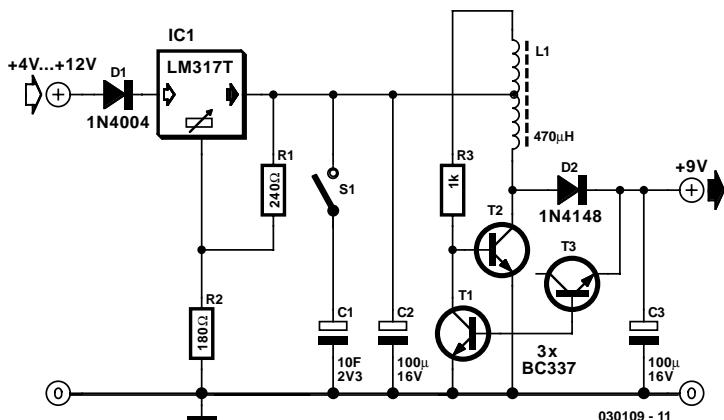


Figure 1. L'électronique de notre accumulateur à Goldcap.

convertisseur de tension régulateur capable de fournir une tension de sortie stable.

### Avec convertisseur bloqueur

Le montage décrit ici et dont la **figure 1** donne le schéma utilise une Supercap de 10 farads seulement (ce qui n'en correspond pas moins à 10 000 000 µF !), sa tension de service maximale étant de 2,3 V. Le convertisseur de tension rehausse la tension de sortie à quelque 9 V. Un petit interrupteur ou un cavalier monté sur la platine permettent de faire en sorte, le cas échéant, que le condensateur de stockage conserve son énergie pour une durée plus longue. Le montage comporte en outre un circuit de charge de sorte que l'on pourra brancher l'accumulateur Supercap à pratiquement n'importe quelle source de courant. Un régulateur de tension ajustable, un LM317T, évite toute surcharge de la Supercap. Ce régulateur intègre en outre une circuiterie de limitation

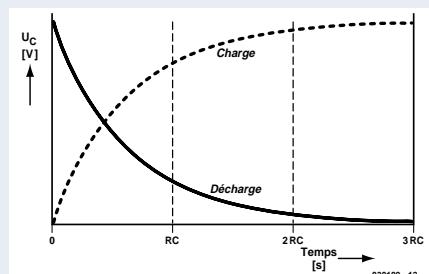
de courant et un dispositif de thermo-protection. Si l'on utilise, pour la recharge, un adaptateur secteur, c'est la résistance interne de ce dernier qui détermine le courant de charge. Dans le cas d'un courant de charge de 1 A, il faut de l'ordre de 20 s pour recharger l'« accu » !.

Le circuit comporte un convertisseur bloqueur réglé utilisant une petite bobine. Cette dernière prend la forme d'une self de 470 µH à corps de ferrite sur laquelle auront été bobinées une vingtaine de spires. Cette self se doit d'avoir une résistance ohmique faible (inférieure à 1 Ω). Avec ce type de bobine, les 2 extrémités de l'enroulement interne bobiné sur le noyau sont soudues aux broches de soudage de la self. On ajoute 20 spires de fil de cuivre émaillé plus fin bobinées dans le même sens de bobinage en vue de constituer la contre-réaction du convertisseur bloqueur (**figure 2**). On peut également, supprimer le bobinage d'origine et bobiner 40 spires de fil de cuivre émaillé en veillant à prévoir une prise intermédiaire cen-

## Mesure de la capacité

En présence de capacités de 1 farad ou plus, la quasi-totalité des capaciomètre jette le gant. Comment faut-il partant s'y prendre pour déterminer la capacité d'une Goldcap ou d'une Supercap ? Pas de problème pour peu que l'on revienne aux bases et que l'on opte pour une mesure « physique » de la capacité. Tout ce dont on a besoin pour cela est un voltmètre, une montre et une résistance.

Branché sur une résistance de valeur fixe, un condensateur ne se décharge pas de façon linéaire mais exponentiellement. La courbe de décharge ne cesse de s'aplatis vu qu'une tension plus faible se traduit par la circulation d'un courant moindre. De ce fait, théoriquement, le condensateur ne se décharge jamais complètement. La constante de temps  $\tau = R \cdot C$  est un facteur intéressant dans ce cadre. Après écoulement de cette durée, le condensateur s'est déchargé jusqu'à une valeur correspondant à  $1/e$  (= 0,37) de la tension de sortie. Ainsi, une Supercap d'une capacité C de 10 F possède, en combinaison avec une résistance de 100 Ω, une constante de temps  $\tau = 100 \Omega \cdot 10 F = 1\,000 s$  soit 16 mn 40 s. Un bon point de départ... Pour la mesure, on chargera le condensateur jusqu'à une tension de 2 V par exemple. Il suffit alors, pour mesurer la capacité, de mesurer le temps nécessaire pour que la tension tombe à  $2 V \cdot 0,37 = 0,74 V$ . La formule  $C = \tau/R$  permet de calculer la capacité. Lors d'une mesure nous avons relevé une constante de temps de 1 200 s ce qui permet de fixer à 12 F la capacité.



Si l'on manque de patience (ou de temps) on pourrait être tenté d'utiliser une résistance de valeur plus faible pour accélérer la mesure. Ceci pourrait amener à se trouver confronté à 2 problèmes. Le premier est que l'on pourrait fort bien se brûler les doigts (dans le sens propre de cette expression) vu qu'une Supercap est capable, avec son courant de crête de 6 A, de fournir une puissance non négligeable. Secundo, le condensateur possède une résistance-série qui se traduit par une erreur de mesure d'autant plus importante que la résistance de décharge est faible.



Figure 2. Voici comment effectuer le bobinage de la petite self.

trale. Un fil de cuivre émaillé de 0,5 mm de diamètre devrait faire l'affaire. Quelle que soit la solution adoptée, cette bobine devient une sorte de transformateur. Le résonateur bloqueur entre bien en oscillation et travaille sagement jusqu'à une tension minimale de 0,5 V.

La tension de sortie est stabilisée à quelque 9 V par le biais d'un réseau de régulation rudimentaire. Un transistor NPN est utilisé ici en diode zener à tension zener de 8 V environ. Rien n'interdit bien entendu d'opter pour une véritable diode zener. Le fonctionnement de cette régulation de base est extrêmement fiable. Nous avons ainsi, à une tension d'entrée de 2,3 V, mesurée 8,5 V en sortie, alors que nous avions toujours encore 8,4 V alors que la tension d'entrée était tombée à 0,7 V. Les caractéristiques techniques intéressantes d'un convertisseur bloqueur sont sa puissance de sortie et son rendement. Lors d'une mesure, nous avons chargé la sortie du convertisseur à l'aide d'une résistance de 1 kΩ. Dans le cas d'une tension d'entrée de 2,3 V nous avons mesuré un courant d'entrée de 50 mA. La dissipation primaire était partant de 115 mW. La tension de sortie était

exactement la même que celle mesurée à vide à savoir 8,5 V, le courant de sortie étant partant de 8,5 mA. Dans ces conditions, la puissance de sortie était de 72,3 mW, ce qui se traduit par un rendement de 63%.

Avec une charge de 8,2 kΩ cela donne un courant de sortie de 1 mA environ. La tension de sortie est maintenue à niveau constant jusqu'à ce que la Supercap se doit déchargée jusqu'à de l'ordre de 1 V, ce qui signifie qu'elle aura fourni à ce moment 80% de son énergie déjà. Au fur et à mesure de la diminution de la tension d'entrée en-deçà de cette valeur la tension de sortie chute elle aussi progressivement, sachant que nous avons mesuré, à une tension d'entrée de 0,6 V la tension de sortie était encore de 4,8 V. Comme cela est normalement le cas avec un convertisseur de tension, le courant d'entrée chute en cas d'augmentation de la tension d'entrée. De plus, plus la charge prise à la sortie est faible, plus la consommation de

courant en sortie est faible.

À vide, le convertisseur ne consomme que 2 mA à une tension d'entrée de 2,3 V. Dans ces conditions on peut espérer, en cas de consommation faible, une durée d'utilisation pouvant aller jusqu'à 2 heures, avant qu'il ne soit nécessaire de procéder à une recharge.

La Supercap d'une taille de 10 x 30 mm utilisée ici est fabriquée par la société coréenne Nuin TEK. On trouve sur Internet des sites proposant ce type de composant en VPC, comme, par exemple, la société AK-Modul-Bus ([www.ak-modul-bus.de](http://www.ak-modul-bus.de)). Sur le serveur au service des amateurs d'expérimentations (en allemand malheureusement), [www.elexs.de](http://www.elexs.de), de la même société, on découvrira d'autres expériences utilisant ces composants intéressants. On pourra utiliser également une Goldcap de 10 F/5 V plus facilement disponibles mais aux dimensions différentes.

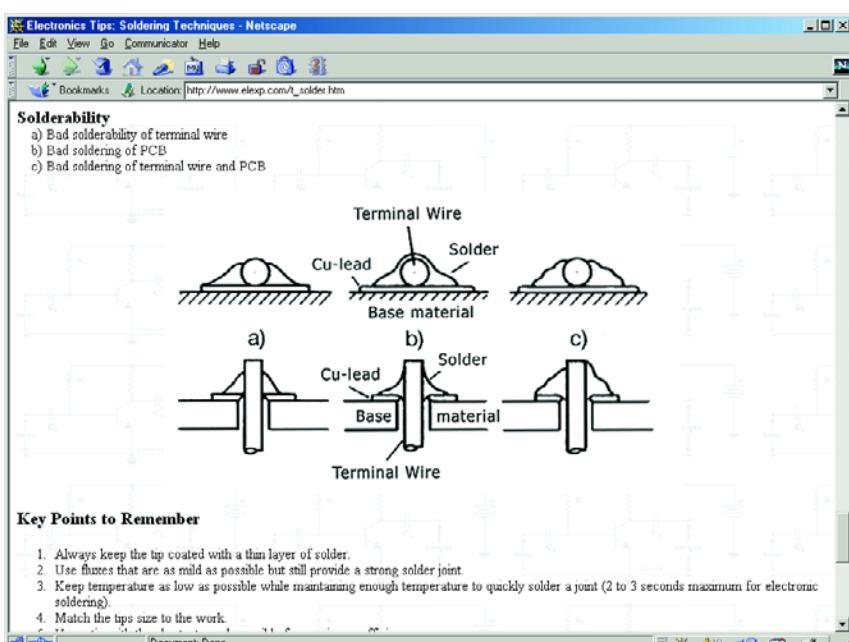
(030109)

# L'art du soudage

## Faire de belles soudures

Harry Baggen

Lors de la réalisation de montages électroniques la liaison entre les différents composants se fait par le biais de fils ou de pistes de cuivre. Les soudures ont alors pour fonction d'assurer une bonne fixation mécanique et une liaison électrique optimale.



Le soudage à la main n'a rien de bien compliqué surtout si l'on a déjà une certaine expérience en la matière. Notez que l'on trouve sur Internet de nombreux articles décrivant étape après étape l'*« art » du soudage*. Il est indispensable, qu'il s'agisse de la réparation d'appareils électriques ou de la réalisation d'un nouveau montage, de bien savoir manier un fer à souder. En fait il nous semble qu'il ne soit pas nécessaire de parler du sujet à un lecteur d'Elektor pour qui le fer à souder est sans doute devenu une sorte de troi-

sième main. Il n'en reste pas moins qu'il est toujours recommandé de jeter un coup d'œil critique à ses soudures car il est d'une importance capitale qu'elles soient de bonne qualité. Il nous arrive de constater, sur les montages que nous recevons à l'occasion, que leur auteur a peut-être pensé « un peu plus ne peut pas faire de mal » ou encore que les îlots de soudage n'ont pas été amenés à température suffisante. Tout ceci peut avoir plus de conséquences (négatives) qu'il n'y paraît à première vue, peut-être pas dans l'immédiat,

mais à moyen ou long terme. On constatera alors, lors d'une panne, qu'une liaison électrique présente une résistance de transfert plus importante que prévue ou qu'à la suite d'une tension mécanique ou thermique une des pattes d'un composant s'est désolidarisée.

Il est partant important, que l'on soit débutant ou électronicien « pratiquant » de toujours examiner ses soudures d'un œil critique. Pour vous aider dans cette opération nous avons examiné un certain nombre de « manuels » de soudage disponibles sur Internet.

Nous vous recommandons, si vous n'avez pas (ou que peu) de temps à perdre, de faire un saut sur la page **B2ModKits** [1]. Une dizaine de photos accompagnées d'un texte explicatif illustrent succinctement la technique de soudage.

Pour une approche plus exhaustive nous ne pouvons que recommander l'article « Online Soldering School » chez **Benchmarks, The Electronics Technician's Web Page** [2]. L'ensemble du processus de soudage est passé en revue en quelques étapes, du choix de la taille du fer à souder au dessoudage en passant par le nettoyage de sa pointe.

On trouvera, chez **Electronics Tips : Better Soldering** [3], un article plus général et informatif (en anglais) consacré au soudage. Il s'agit d'une reproduction d'une information fournie à l'origine par Coopertools, le fabricant des fameux fers et stations à souder Weller. On y aborde l'aspect de la composition de la soudure (et son point de fusion en fonction de l'alliage utilisé) sans oublier le fonctionnement des fers et stations à souder Weller.

Le site allemand Elexs propose, avec « **Der kleine Lötkurs**, Teil 1...4 [4] un excellent cours de soudage exhaustif (en allemand). Même si vous ne maîtrisez pas la langue de Goethe, il vaut sans doute la peine d'examiner les photos qu'elles soient un peu sombres. La pratique, rien que la pratique semble avoir été le moteur de l'auteur de ce site.

Il existe également, l'un ou l'autre site en français, encore qu'ils soient très rares, traitant du sujet. Celui proposé en référence **Interface-z** [5] est l'un des meilleurs que nous ayons trouvé.

Nous vous proposons en outre en référence [6] et [7] quelques autres sites présentant des cours de soudage (pour l'électronique).

Si le sujet vous intéresse et que vous aimeriez trouver encore plus d'informations sur le soudage, pourquoi ne pas faire un tour Outre-Quiévrain sur un site belge (en anglais cependant) **The Educational Encyclopedia** [8]. On y trouve une série de liens renvoyant vers des sites où l'on peut trouver des informations sur le soudage et ses techniques. Signalons au passage que ce site comporte une belle collection de liens renvoyant à d'autres sujets ayant trait à l'électronique.

Nous pouvons proposer aux plus expérimentés de nos « lecteurs-soudeurs » un cours de soudage des CMS. Ces composants miniatures sans pattes requièrent une technique de soudage spéciale qui nécessite souvent un certain nombre d'outils quels qu'ils soient. Le site **Soldering SMD** [9] explique, d'une façon très facile à comprendre, les bases du soudage de composants CMS.

Une fois que vous aurez maîtrisé le soudage des CMS vous pourrez vous essayer au défi suivant : le dessoudage et le remplacement de CMS sur des plaques existantes. Rato Elektronica propose un manuel bien illustré sur le sujet en page **How-To: SMD Rework** [10].

- Conseils**
  - Nos prestations
- Comment choisir**
- Tableaux synthétiques**
  - Par type de capteur
  - Par type de détection
- Comparaisons**
  - Comparaison tapis / caméra
  - Temps réel
  - Mesure de distance
  - Télémétrie ultrasonore
  - Mesure de vitesse
  - Détection de présence, position, mouvement
  - Tout ou rien
  - Intensité lumineuse
- Patches**
  - Pure-Data
  - Max
- Techniques**
  - Soudure de composants
  - Montage d'un Tapis sensible
- Protocoles**

**Conseils : soudure à l'étain - 3**

1 - Matériel nécessaire  
2 - Consignes de sécurité  
3 - Comment souder  
4 - Conseils pratiques

**3 - Comment souder**

La soudure des composants électroniques sur le circuit imprimé a deux buts :

- immobiliser le composant à sa place,
- permettre le contact électrique entre le composant et les pistes de cuivre du circuit imprimé.

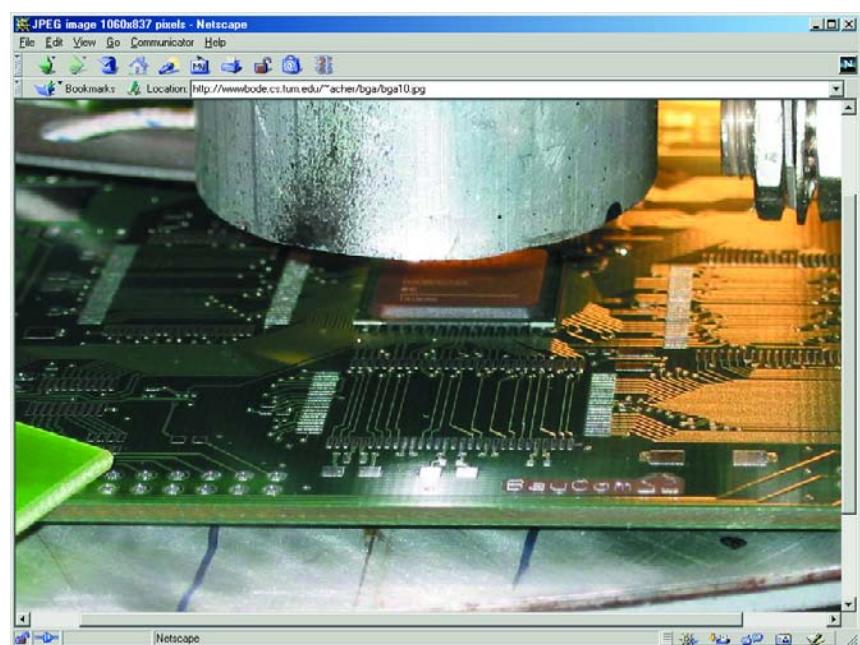
Si les soudures sont mal faites, le montage peut ne pas fonctionner ou les composants peuvent être arrachés.

Pas à pas :

**1 - Placer le composant sur le circuit imprimé.**

Un circuit imprimé courant comporte une face sans circuit et une face avec circuit de cuivre apparent. La soudure se fera du côté du Cuivre.

Le composant lui-même se retrouve du côté non cuivré et ses pattes dépassent du côté du circuit de Cuivre. Il y a des **pastilles** de Cuivre prévues du côté circuit pour faciliter la soudure.



## Adresses Internet

- [1] B2ModKits.com: <http://www.b2modkits.com/Resources/Soldering/>
- [2] Benchmarks: <http://www.mts.net/~rmcgill/>
- [3] Electronics Tips: Soldering Techniques: [http://www.elexp.com/t\\_solder.htm](http://www.elexp.com/t_solder.htm)
- [4] Elexs: Der kleine Lötkurs: <http://www.elexs.de/loet1.htm>
- [5] <http://www.interface-z.com/conseils/soudure.htm>
- [6] A primer on hand soldering electrical connections: <http://et.nmsu.edu/~etti/fall97/electronics/solder.html>
- [7] Fechner-Schulte Leiterplattentechnik GmbH / Tips zum Löten: <http://www.fs-leiterplatten.de/loet.htm>
- [8] The educational encyclopedia, soldering techniques, assembling process: <http://users.pandora.be/educypedia/electronics/soldering.htm>
- [9] Soldering SMD: <http://www.smdin.com/solderingsmds.html>
- [10] Rato Elektronica NV - SMD Rework: [http://www.rato.be/nl/smd\\_rework.htm](http://www.rato.be/nl/smd_rework.htm)
- [11] BGAs selber einlöten: <http://www.bode.cs.tum.edu/~acher/bga/index.html>

Mentionnons, en guise de conclusion, un site destiné aux spécialistes, celui de **Georg Acher** [11], qui explique, dans un long article, comment souder soi-même des BGA (*Ball Grid Arrays*). Les seuls contacts qu'aient ces circuits intégrés prennent la forme de demi-billes disposées sur le dessous du composant de sorte qu'il est parfaitement impossible d'y accéder à l'aide d'un fer à souder. Georg montre qu'il est cependant possible, avec quelques « outils » et une bonne dose d'ingéniosité, de souder ces BGA « manuellement ».

(035074)

APPLIKATOR est une rubrique servant à la description de composants intéressants récents et de leurs applications; par conséquent, leur disponibilité n'est pas garantie. Le contenu de cette rubrique est basé sur les informations fournies par les fabricants et les importateurs, ne reposant pas nécessairement sur les expériences pratiques de la Rédaction.

# Inverseur de polarité automatique

## Sans diodes ni relais

Klaus-Jürgen Thiesler

Il existe toujours le danger, avec des appareils alimentés par pile que l'on fasse une erreur de polarité lors de la mise en place des piles. L'implantation d'une diode ou d'un pont de redressement à diodes apporte indubitablement une solution à ce problème mais se traduit par des pertes d'énergie dont on se serait fort bien passé. L'alternative moderne à cette approche prend la forme d'un circuit intégré dernier-cri doté de commutateurs analogiques doubles.

Il existe depuis belle lurette des éléments de commutation CMOS, mais il a fallu attendre l'avènement des téléphones mobiles et des appareils de mesure portables pour que les fabricants de circuits intégrés se soient

vus forcés d'améliorer la famille logique 4000 qui avait bien vieilli. Ce montage à un circuit intégré basé sur un commutateur analogique CMOS du type MAX4684 de Maxim Integrated; ce composant corrige

automatiquement la polarité des piles montées dans le système. Ce composant se contente d'une tension de 1,8 V pour fonctionner ce qui signifie qu'il est actif dès lors que l'alimentation se compose d'au moins 2 piles ou accumulateurs CdNi ou NiMH même s'ils sont déchargés.

Un coup d'oeil à l'intérieur du circuit intégré représenté en **figure 1** nous permet de découvrir un double commutateur unipolaire réalisé à base de FETMOS canal-P et canal-N. Comme vous n'êtes pas sans le savoir, ce type de composant se caractérise par une résistance de commutation  $R_{DSon}$  étonnamment faible, caractéristique combinée à une vitesse de commutation très rapide et un courant de commutation élevé. Cette paire de commutateur remplace pratiquement un redresseur en pont pleine-onde.

Le circuit pris en aval est ainsi non seulement protégé, mais de plus le MAX4684 se charge d'une correction extrêmement rapide de la polarité d'une pile montée à l'envers. La **figure 2** illustre le comportement du circuit intégré dans le cas d'une pile montée correctement et d'une pile montée à l'envers. Les trajets du courant sont donnés en grisé. En

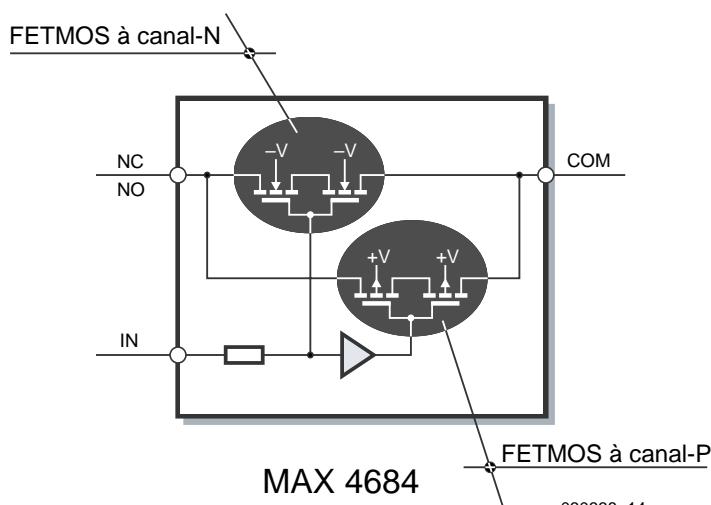


Figure 1. Chacun des commutateurs est constitué d'une paire de FETMOS (à canal-P et canal-N).

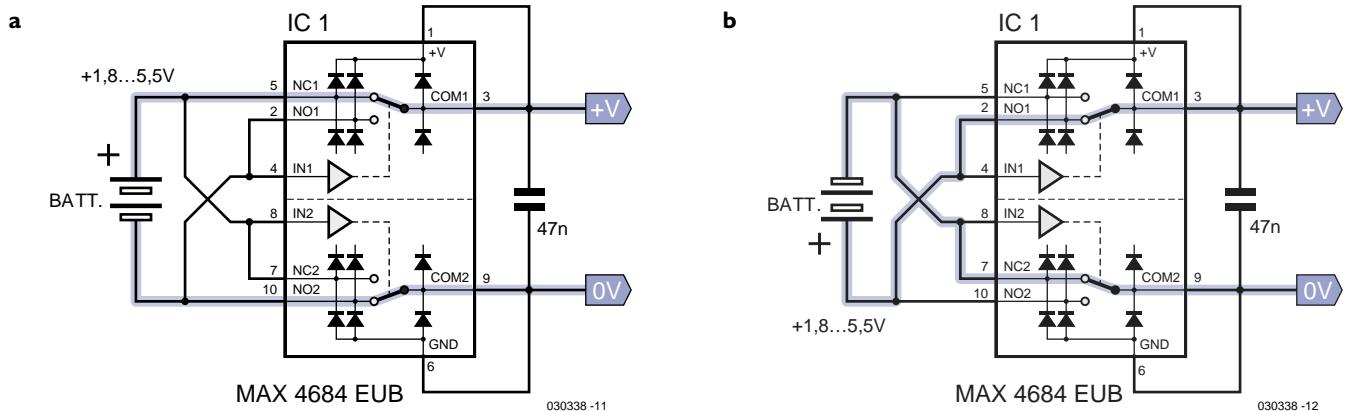


Figure 2. Trajets suivis par le courant en cas de polarité de pile correcte (a) et de polarité inversée (b).

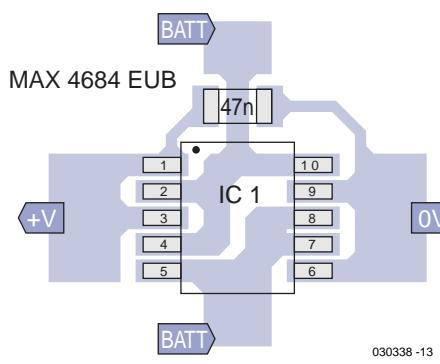


Figure 3. Un dessin de platine doté de surfaces de cuivre importantes (relativement).

l'absence de pile, les deux commutateurs se trouvent en position repos : les bornes COM sont chacune reliées aux points NC (*Normally Closed*). Dans ces conditions le pôle négatif de la pile se trouve relié à la sortie 0 V. Dans le second croquis, nous avons inversé la polarité de la pile. Le commutateur du haut change de position de travail, le commutateur

du bas reste dans la position qu'il avait au repos.

Les diodes de protection internes prises aux entrées et aux sorties garantissent un démarrage correct du MAX4684 lors de la mise en place de la pile. Avant que le commutateur concerné ne passe dans sa position de travail, les 2 sorties COM1 et COM2 (et partant aussi +V et 0V, la masse) se trouvent en court-circuit. En tout état de cause le MAX4684 possède un temps de basculement de 60 ns au maximum. Pendant cet intervalle de commutation les diodes de protection internes établissent la tension d'alimentation requise par le MAX4684 quelle que soit la polarité d'implantation de la pile. Ce n'est qu'une fois que l'entrée de commande concernée se trouve reliée au pôle plus d'une pile disposant d'une tension à ses bornes supérieure à 1,4 V que l'on a commandé des FETMOS internes et basculement du commutateur. Ce n'est qu'à partir de ce moment que la sortie se trouve reliée avec la polarité correcte à la tension de la pile.

L'attaque des FETMOS de commutation interne se fait par le biais d'un temps mort, intervalle au cours duquel tous les contacts de commutation sont ouverts. Ce mode de coupe avant contact connu sous l'abréviation de BBM (*Break Before Make*) d'une durée de 2 ns au moins évite la mise en court-circuit de la pile au cas où le condensateur de sortie serait encore chargé et qu'il alimente le circuit intégré lors d'un changement de pile.

À l'inverse des diodes dont la ten-

sion directe est pratiquement constante, la dissipation au niveau de la résistance d'activation d'un FETMOS varie en fonction du courant. La résistance  $R_{DSon}$  est, avec ses 0,3 Ω à une V+ de 1,8 V, remarquablement faible ; à 5 V, la résistance d'activation chute à 0,2 Ω. Le courant maximal admissible au niveau des 2 commutateurs et des diodes de protection internes est de ±300 mA.

De ce fait, le MAX4684 convient à la quasi-totalité des applications utilisant des piles Mignon (R6) ou Lady (R1), c'est-à-dire les appareils audio portatifs, les instruments de mesure mobiles ainsi que les téléphones sans fil.

Le **Tableau 1** indique les chutes de tension maximales aux bornes des deux commutateurs comparées à celles relevées sur des diodes au silicium et Schottky montées en pont redresseur.

Vous trouverez dans ce même numéro une double page, l'info-carte, consacrée à ce circuit intégré et en donnant les principales caractéristiques. Le MAX4684 est proposé en boîtier μMAX (3 x 5 mm) à 10 broches. En cas de problèmes de place, on pourra utiliser une autre version du MAX4684 proposée elle en boîtier UCSP de 1,5 x 2 mm. La **figure 3** vous propose un dessin de circuit imprimé pour la version μMAX. Signalons que le MAX4684 ne convient PAS au redressement de tensions alternatives pour en faire une tension continue lissée.

(030338)

## Tableau I

Comparaison entre diodes et commutateurs analogiques (à 100 mA - 2,4 V <sub>CC</sub> )		
Composants	Tension directe	Dissipation P <sub>tot</sub>
4 diodes Si	2 x 0,65 V = 1,30 V	130 mW
4 diodes Schottky	2 x 0,40 V = 0,80 V	80 mW
<b>MAX4684</b>	<b>2x0,02 V=0,04 V</b>	<b>4 mW</b>
2 MAX4624	2 x 0,08 V = 0,16 V	16 mW
2 ADG819	2 x 0,05 V = 0,10 V	10 mW

# CoolRunner II

Kit de développement pour CPLD de Xilinx

Paul Goossens

Nous nous trouvons ici en présence d'un kit de développement très intéressant de Xilinx tout à la fois complet et d'un prix très abordable.



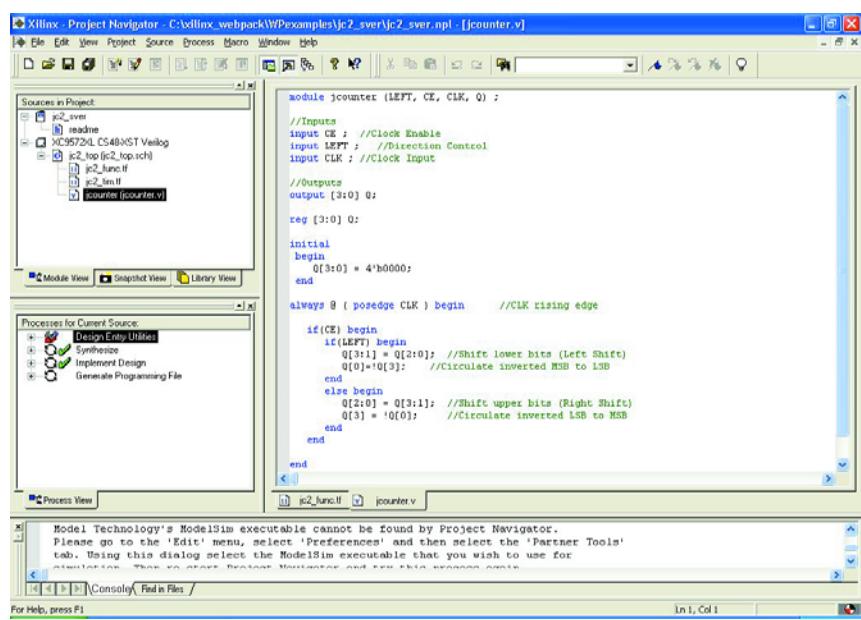
FPGA d'être programmées in situ (d'où le IC de *In Circuit* que l'on rencontre à quasiment toutes les pages de leur fiche de caractéristiques) et à la mise à disposition, par les fabricants de ces composants, de logiciels de développement gratuits. Le seul inconvénient, si tant est que l'on puisse parler d'inconvénient, est que la plupart de ces types de composants sont uniquement proposés en version CMS (SMD).

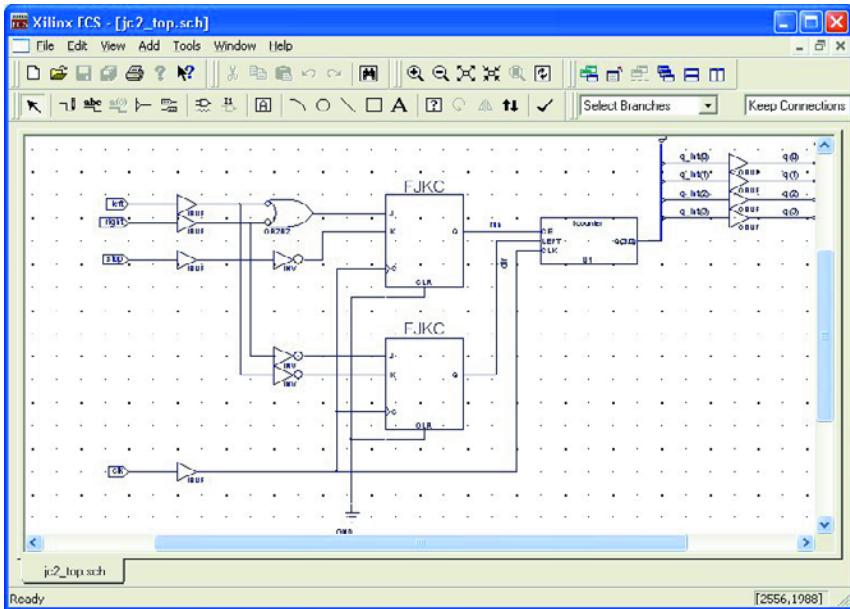
## CoolRunner II, kit de développement pour CPLD de Xilinx

Ce dernier obstacle est facilement franchi dans le cas du kit de déve-

Les PAL et les GAL n'ont pas réellement percé dans le monde de l'électronique amateur, et ceci bien que ce type de composant permette de réduire sensiblement le nombre de composants requis pour une réalisation et qu'il soit possible, dans le cas des GAL, d'en modifier le contenu sans qu'il ne soit nécessaire de faire un nouveau montage. Les avantages de la mise en oeuvre de ce type de composant ne contrebalancent pas, pour la plupart des amateurs, le coût important d'un programmeur et des logiciels de développement qu'il requiert.

Les CPLD et FPGA se voient eux de plus en plus utilisées dans des appareils tant professionnels que développés par des amateurs. Ceci ne tient pas uniquement au fait que ces composants offrent bien plus de possibilités que les PAL et les GAL, mais sans doute bien plus à la capacité qu'ont les CPLD et les





lancement (*development board*) de Xilinx. Cet ensemble comporte les éléments suivants :

- Une carte dotée d'une CPLD,
- Un câble parallèle servant lors de la programmation,
- Un logiciel de développement (ISE WebPack) sur CD,
- Un CD baptisé Design Kit Resource CD,
- Un ouvrage (en anglais), le Programmable Logic Guide et

- Un porte-pile (sans pile) prévu pour recevoir 2 piles AA (LR6).

La caractéristique la plus étonnante de cet ensemble est indubitablement son prix : 50 \$US hors frais de port. Si l'on les ajoute, cela nous amène à de l'ordre de 75 E (au prix actuel du dollar). Il est fort probable qu'il vous coûterait plus cher de rassembler les pièces nécessaires que d'acheter le kit tout fait !

La disponibilité est exemplaire : on

peut en effet, même si l'on est un particulier, effectuer une commande directe par le biais de l'E-shop de Xilinx à l'adresse :

[www.xilinx.com/xlnx/xbiz/onlinestore.jsp?sGlobalNavPick=PURCHASE](http://www.xilinx.com/xlnx/xbiz/onlinestore.jsp?sGlobalNavPick=PURCHASE).

Sur cette page, cliquer au centre sur « Development Boards » puis sur « CoolRunner-II Boards ». Et vous voici arrivé à la fin de votre quête. Il ne reste plus qu'à mettre le kit dans le chariot et à passer à la caisse (carte de crédit souhaitées).

## Le matériel

La carte est dotée d'une CPLD X2C256TQ144-7. Il également possible de mettre une seconde CPLD sur le circuit, une XC2C64 ou XC2C32. Un oscillateur de 1,843 2 MHz fournit le signal d'horloge à la (aux) CPLD. Le positionnement de cet oscillateur sur un support en permet un remplacement aisément par un oscillateur travaillant à une fréquence différente. Il faudra, dans ce cas-là, faire attention au brochage et à la tension d'alimentation (3,3 V) de ce nouveau bloc.

La tension d'alimentation pourra prendre 2 formes : soit celle d'un adaptateur secteur soit celle d'une paire de piles placées dans le porte-pile fourni avec la carte de développement. Selon le type d'alimentation choisi, il faudra veiller au positionnement correct du cavalier JP1. Contrairement à ce qui pourrait paraître logique au premier abord, si l'on opte pour une tension d'alimentation externe supérieure à 3,3 V, JP1 devra être mis en position « Int » (*Internal*). Si l'on utilise les 2 piles, JP1 devra être mis sur « Ext » (*External*). Dans ce cas-là la tension appliquée n'est pas régulée. Si donc on fait l'erreur d'appliquer une tension trop élevée (adaptateur et JP1 sur « Ext ») certains composants, la CPLD en particulier, peuvent subir des dommages irréparables. Vous êtes, par conséquent, prié de bien faire attention. Xilinx nous a promis d'insister sur ce point lors de la prochaine mouture de cette carte.

La carte comporte une embase sub-D 25 points par le biais de laquelle se fait la programmation de la (des) CPLD. Cette embase est reliée au port parallèle du PC par le biais du câble fourni avec le kit. Si vous possédez déjà une autre interface JTAG de Xilinx, vous pouvez également mettre à contribution l'embase JTAG présente sur la carte.

Le nombre et la qualité des Entrées/Sorties disponibles sont relativement restreints. En effet, seuls une LED et un bouton-poussoir attaquent la CPLD. Si cela peut être suffisant pour une toute première expérience, les perspectives offertes sont, c'est le moins que l'on puisse dire, limitées.

De manière à permettre à un utilisateur



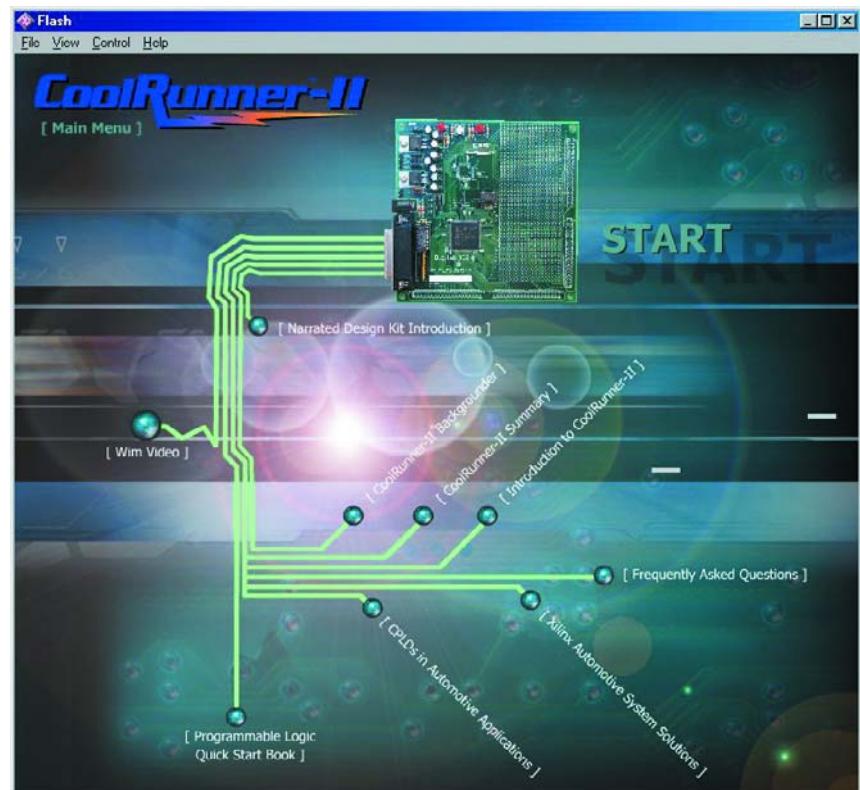
potentiel d'ajouter des E/S additionnelles, Xilinx a doté sa carte d'un champ d'expérimentation à pastilles de bonne taille sur lequel on aura vite fait d'implanter l'électronique périphérique nécessaire. De plus, tous les signaux importants et les lignes d'alimentation sont dérivées vers 4 embases à 2 rangées de 20 contacts (il vous faudra mettre en place l'embase requise) au cas où le (grand) champ d'expérimentation à pastilles s'avérerait trop petit pour les extensions envisagées.

## Le logiciel

L'environnement logiciel accompagnant cette carte d'expérimentation est de très bonne qualité, occupant pas moins de 2 CD. Le programme de développement ainsi que certains exemples se trouvent sur le CD baptisé ISE 5.1 (à noter que Xilinx prévoit une version 6.1 sur son site fin novembre (elle devrait sûrement être disponible lorsque vous lirez ces lignes). Le second CD est en fait une sorte de présentation multimédia de description de la carte en particulier et des CPLD en général, sans oublier une importante documentation technique.

Il est toujours recommandé, lorsque l'on utilise un outil de développement de voir s'il n'existe pas une version plus récente étant donné le temps nécessaire à la fabrication du kit (et des CD-ROM). Il vous faudra, au cours de l'installation du programme, vous enregistrer, ce qui pourra se faire soit par le site Web, soit par E-mail ou encore par télécopie. Il existe, lorsque l'on tourne sous ISE, différentes manières de saisir un projet : le programme reconnaît les langages suivants : Verilog et VHDL. Le logiciel comporte en outre son propre programme de saisie de schéma permettant de « dessiner » un circuit de manière plus traditionnelle. Notons, en extra, l'existence d'un éditeur de machine d'état (*state machine*). Cet éditeur simplifie énormément le développement d'une machine d'état. Il suffit de dessiner et de remplir les conditions requises, le logiciel créant à partir de là un fichier VHDL valide. Il est plus facile de modifier ou d'analyser une telle machine d'état que d'avoir à se plonger dans le code VHDL pour se faire une idée du fonctionnement de la machine d'état.

ISE comporte, heureusement, un didacticiel (*tutorial*) qui, pas à pas, explique à l'utilisateur comment utiliser le logiciel. Comme toujours, lorsque l'on se lance dans la découverte d'un nouveau produit, il est souhaitable de passer le didacticiel complètement en revue. Cela permet, finalement, de gagner du temps car cela évite d'avoir à chercher la fonction nécessaire. Et, pour finir, ISE intègre également un pro-



gramme permettant une programmation aisée de la CPLD depuis le PC. Comme on le constate, un seul et unique logiciel permet de clore le cycle de programmation complet.

## En guise de conclusion

Si l'on est intéressé par la mise en oeuvre de CPLD ou que l'on tient à se donner un minimum d'expérience dans la description de circuits en Verilog ou en VHDL, c'est exactement le kit de développement qu'il vous faut. Xilinx met à disposition,

sur son site, gratuitement, un certain nombre d'exemples que l'on pourra utiliser dans ISE. Le prix et la disponibilité sont des facteurs attrayants; il est à espérer que d'autres fabricants de processeurs proposent ce type de produits de développement. La rédaction est curieuse de savoir quels montages nos lecteurs auront développés à l'aide de ce kit. N'hésitez pas à nous faire part de vos succès, notre boîte postale électronique ([redaction@elektor.fr](mailto:redaction@elektor.fr)) vous est ouverte.

(030427-1)