

# elektor

l'électronique imaginative

N° 315  
SEPTEMBRE 2004  
mensuel

(B) 6,35 € • (CH) 10,90 FS (CAN) 8,10 \$Can.  
DDM surface 6,80 € • DDM avion 8,50 €  
ISSN 0181-7450  
www.elektor.fr

Plus :  
guide d'achat

Module µC  
Bluetooth

Contrôleur-  
hôte USB

# OÙ VONT

# les



**Créez-le,  
pas à pas:  
votre dé 3D**

# MICROCONTROLEURS

...Mesurer, réguler, piloter, commuter...



**SAK :**  
**le à-tout-faire  
numérique**

L 19624 - 315 - F: 5,75 €



# TOMI ENGDAHL

Le père de [epanorama.net](http://epanorama.net)

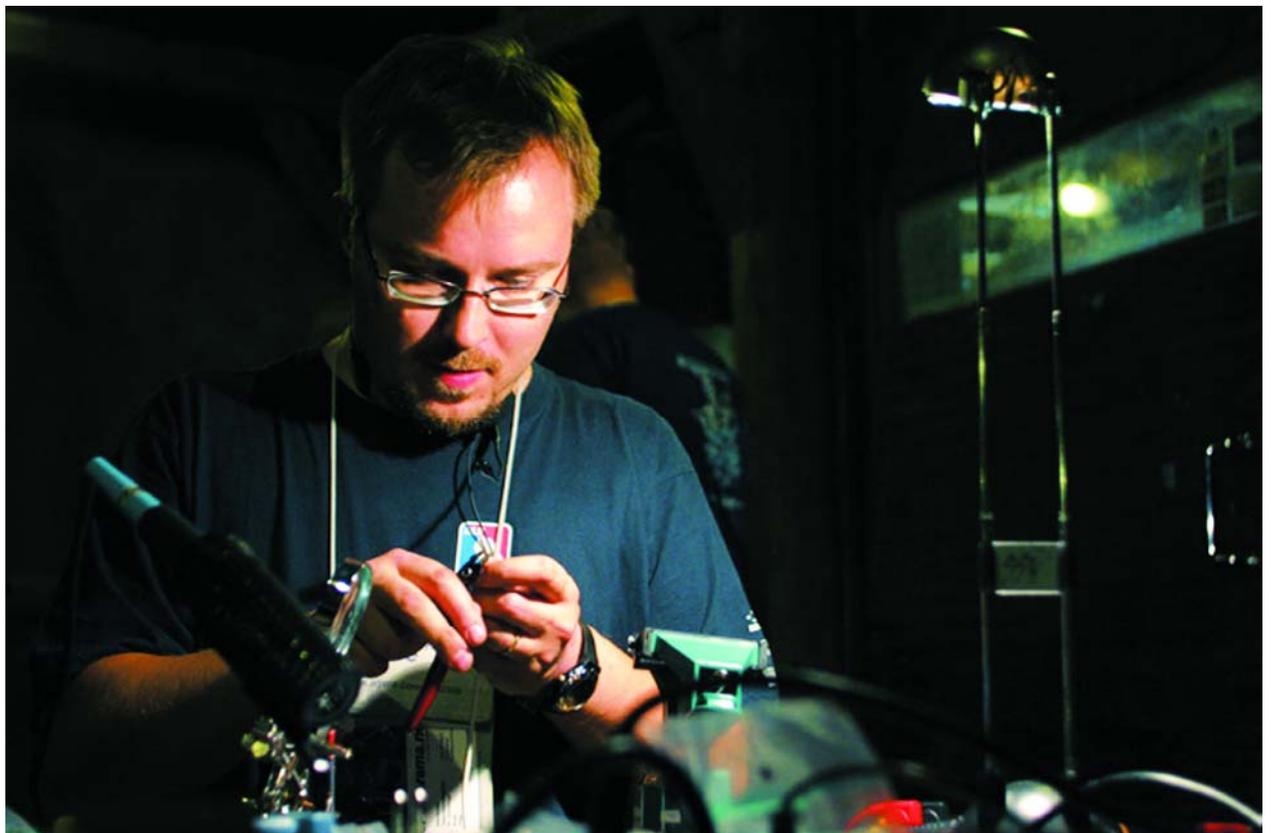


Photo: Oleg Hartsenko Copyright: Assembly Organizing

Fait-il également partie de vos favoris, ce [www.epanorama.net](http://www.epanorama.net) ? Sans doute bien. Cela fait en effet des années déjà que le site du Finlandais Tomi Engdahl sert non seulement de référence mais aussi de source d'inspiration pour nombre d'amateurs d'électronique. Nous n'allons pas, cette fois, nous commettre à une présentation du site, mais à une interview du maître des lieux.

## Sans vouloir enfoncer des portes ouvertes : l'électronique doit être votre violon d'Ingres. Comment cela a-t-il commencé ?

Je ne sais plus très précisément quand le virus m'a pris, mais la cause en est évidente. Mon père est ingénieur et je crois que c'est à cela qu'est dû mon intérêt de la technologie. Il est vrai que j'avais aussi démonté un certain nombre de postes de radio...

## La décision de faire de l'électrotechnique a donc été prise très tôt, mais avez-vous envisagé de poursuivre d'autres études ?

Non. Les choses étaient ou ne peut plus simples. Le seul point d'interrogation était Informatique ou Électrotechnique ? J'ai fini à la Faculté d'Informatique de l'Université Technique de Helsinki. J'y ai d'ailleurs obtenu mes diplômes. J'ai également suivi des matières ayant trait à l'électrotechnique telles que traitement du signal, architecture des ordinateurs et télécommunications à la faculté d'électrotechnique.

## Pourquoi avoir choisi Helsinki ?

En fait l'université d'Helsinki ne se trouve pas à Helsinki. ... Il y a quelques années elle a déménagé à Otaniemi, à Espoo, quelques kilomètres à l'Ouest de Helsinki. Un cousin y étudiait et la réputation aidant... Il me fallait une demi-heure de chez moi pour y aller. À noter que plus tard j'ai déménagé vers le campus.

## ...et vous vous êtes plongé dans la vie d'étudiant ?

Oui, j'ai été, entre autres choses, impliqué dans l'organisation d'événements sur le campus. Ainsi, un jour que j'ai eu un disc jockey pour l'éclairage lorsqu'il fit faux bond, je me suis trouvé forcé de le remplacer. Devant l'enthousiasme du public j'ai poursuivi dans cette voie. Nous avons ainsi, un ami et moi, joué aux disc jockey de nombreuses années lors de fêtes d'étudiants !

## Votre aventure sur Internet a-t-elle débuté à l'université ?

La grande aventure a en effet débuté à ce moment-là, encore que j'ai été, auparavant, en ligne quelques années durant, par le biais de BBS. Je crois que c'est en 1989 que j'ai écrit pour la première fois quelque chose au sujet de l'aspect matériel d'un PC. Je voulais en savoir plus mais dû constater qu'il n'existait que peu d'informations et rien du tout en finlandais. Je me suis alors mis à la recherche d'informations de livres et fiches de caractéristiques que j'ai condensées. Mes premiers articles, en finlandais, qui étaient consacrés aux ports parallèle et série du PC, sont encore d'actualité aujourd'hui. Je les ai cependant, au fil des ans, bien évidemment adaptés.

## Votre site est joliment actualisé. Le faites-vous toujours vous-même ?

Non, je ne fais plus à moi tout seul. J'ai créé, avec deux amis, la société ELH Communications Ltd. dont epanorama.net fait maintenant partie. Ils ont aussi été associés au changement de nom aux alentours de 1998-1999, vu qu'avant le site se trouvait sous ma page d'accueil à l'université.

## Où votre site tourne-t-il ?

Aux USA. Nous avons 2 serveurs, dont un back-up. Le serveur primaire tourne chez Rackhost. Un peu plus cher, mais cela ouvre de nouvelles possibilités. Nous ne pouvons pas, par exemple, faire exécuter de scripts au serveur de l'université. Les coûts occasionnés par la location et le trafic sont heureusement compensés par les revenus de bannières publicitaires.

The screenshot shows the ePanorama.net website interface. At the top, there's a search bar with the text "ePanorama.net" and a search button. Below the search bar, there are several navigation menus. On the left, there's a "Content" menu with links like "Links", "Documents", "Circuits", "Forums", "Chat Room", "Downloads", "Search", "Acronym", "RFCS", "New Links", and "Newsroom". Below that is an "Other" menu with "Submit a site", "Tell a friend", and "Feedback". At the bottom left is an "Info" menu with "About", "Advertise", "Privacy", "Ad FAQ", "Disclaimer", and "Legal notice". In the center, there's a "ePanorama.net - Links" section with a long list of links categorized by topic, such as "Amateur science", "Audio", "Automation", "Barcode technology", "Basics", "Basic components", "Books", "Business", "Cabling", "Career", "Circuit sites", "CPLA", "Companies", "Component dealers", "Component intergration", "Component manufacturers", "Computers", "Computer hardware", "Data communications", "Databases", "Digital Signal Processing", "EMC", "File viewing utilities", "FPGA", "Frequency Asked Questions", "FUP", "GPS", "HMI", "Home automation", "IC pinouts", "Industry links", "Intelligence", "IR remote control", "LAN", "Laser", and "Logic and built controllers". On the right, there's a "Site News" section with three entries: "April 12 2004: Major links update. Hundreds of links added and updated.", "March 28 2004: More acronyms and descriptions to acronym search. Now over 15000 acronyms on the list.", and "February 1 2004: Multi search system taken to use and search box added to link pages. Multi-search gives you one-stop access to all originally separate site search engines."

## Le trafic doit être impressionnant... avez-vous des statistiques ?

Début 2004 nous avions une moyenne de 14 615 visiteurs par mois qui ouvraient 51 562 pages. Cela se traduit par un trafic d'environ 67 gigaoctets par mois.

## Pas mal, en effet, mais vous visez sans doute plus ?

Nous poursuivons en permanence d'améliorer le site et rassemblons articles et liens et en intégrons le contenu. Nous modernisons le système de gestion de façon à accélérer les mises à jour. Une fois que cela sera réglé, nous passerons à l'amélioration et à l'extension des options les plus avancées, telles que fonction de recherche et évaluation des liens.

Autre souhait, vérifier les documents et tester les circuits avant qu'ils ne soient intégrés dans notre enveloppe, mais il s'agit là d'un projet à moyen-terme.

## Il vous faut sans doute, parallèlement, travailler encore...

En effet, je travaille chez Netcontrol, une société fournisseur de systèmes pour l'industrie de production et de distribution d'énergie. J'écris en outre pour le magazine d'électronique finlandais Processori. J'y ai une sorte de colonne dans laquelle je décris mois après mois quelques sites Internet intéressants. Un couteau à double tranchant en quelque sorte !

## Pourriez-vous nous dire quels sont vos marques-pages.

Bien. Un condensé des pages que j'utilise journallement : www.google.com (si vous cherchez quelque chose qui ne se trouve pas encore sur www.epanorama.net, www.processor.fi (n'est intéressant que si vous connaissez le finlandais), slashdot.org (actualités micro-informatiques) et pour la détente : www.dilbert.com et www.userfriendly.org.

## En guise de conclusion : que ne doit-on pas rater lors d'un passage sur epanorama ?

Que dire, un certain nombre de documents valent la peine d'être lus. Si nous devions en choisir un seul, lisez donc « Ground loop problems and how to get rid of them », cet article a été bien accueilli dans de nombreux forums.

(040130)

# Clé mémoire USB

## Flash, la mémoire idéale

Harry Baggen

En quelques mois, la mémoire USB (USB Flash-drive) est devenu un support de mémoire portable extrêmement populaire : petit robuste, capacité importante, sans pièce mobile et... sans pile. En ces jours où Elektor offre à ses nouveaux abonnés une telle mémoire USB en prime, il est temps de voir comment elle fonctionne.



Pratique une clé mémoire USB. Elle trouve toujours une place dans la poche du pantalon, ne requiert pas de pile et peut se targuer d'une capacité étonnante, (ne parlait-on pas, tout au début, de 22 disquettes pour une clé 32 Moctets ?). Bien qu'il n'existe que depuis quelques années, les clés sont devenues un « accessoire indispensable » étonnamment populaire.

Que contient-elle et comme fonctionne-t-elle ? En prenant, comme cobaye, la prime offerte aux nouveaux abonnés à Elektor (pour 1,5 an) nous allons le voir ici. La marque et le type n'y changent pas grand chose.

La photo montre qu'il n'y a pas grand chose à l'intérieur d'une clé mémoire USB. Le composant le plus important est le circuit de mémoire Flash proprement (le gros pavé noir) et le contrôleur (qui se trouve sur le dessous). On trouve en outre quelques CMS, un quartz et une LED signalant un accès aux données. Le petit interrupteur à glissière sur le côté sert à la protection en écriture.

### 2 circuits intégrés

La **figure 1** donne le synoptique de nos 2 circuits intégrés. Le contrôleur de mémoire Flash, un OTi-2168 (de Ours Technology inc.) ici, a été conçu spécialement pour

ce type d'applications. Il comporte un noyau RISC rapide se chargeant de la communication avec la mémoire Flash. L'ordinateur auquel est connecté cette clé pense qu'on vient de connecter un disque dur à l'un de ses ports USB. La puce OTi possède en outre une interface rapide, de type USB 2.0, pour un transfert de données à 480 Mbits/s au maximum.

Pourquoi faut-il une telle puissance de calcul ? Il ne s'agit ici pas uniquement de conversion série-parallèle mais aussi de piloter correctement la mémoire Flash. Celle-ci n'a pas la structure de RAM ou

d'(E)EPROM classique avec des lignes d'adresses et de données, mais elle doit pouvoir, au travers de quelques broches d'E/S et par le biais d'un protocole, communiquer avec la mémoire Flash.

Nous avons ici, une mémoire Flash NAND de 1 024 Mbits (128 Moctets) de Samsung, piloté au travers de 8 lignes d'E/S. Le contrôleur décide de l'opération, lecture, écriture ou effacement, des adresses et des données.

Le synoptique permet de se faire une bonne idée de la complexité d'une telle mémoire Flash qui intègre une bonne quantité d'intelligence.

## Modifier la lettre de station

Normalement, Windows XP attribue automatiquement à une mémoire USB, lorsque celle-ci est enfichée dans un port USB, la lettre de station suivante disponible sur le système. Dans le cas d'un réseau où l'on a accès aux lecteurs d'un autre ordinateur cela peut poser des problèmes. Il se peut que la mémoire USB se voit attribuer la même lettre qu'un lecteur du réseau ou pire encore qu'elle soit invisible. Il est possible de résoudre ce problème en attribuant une autre lettre de station à la mémoire USB. Dans le menu **Démarrer**, allez dans

**Paramètres/Panneau de configuration/Outils d'administration** puis **Gestion de l'ordinateur** et ouvrez **Stockage** à cet endroit. Cliquez, dans la fenêtre gauche, sur **Gestion des disques** Vous verrez apparaître dans la fenêtre à droite tous les lecteurs. Cliquez souris droit sur le lecteur amovible représentant la mémoire USB, choisissez **Modifier la lettre de lecteur et les chemins d'accès...** et attribuez-lui une autre lettre de station, Z par exemple.

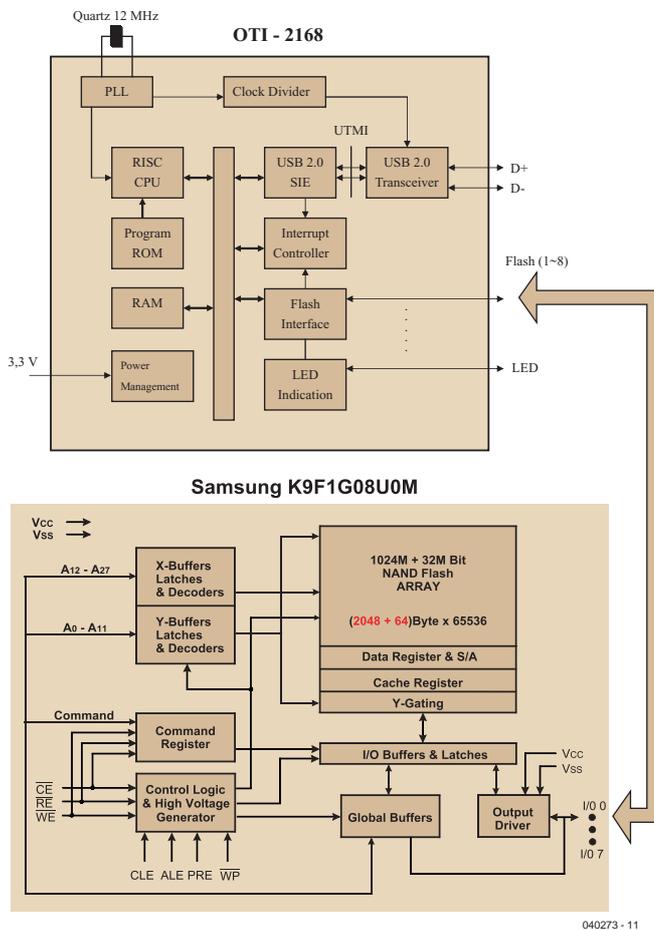


Figure 1. 2 circuits intégrés complexes pour une mémoire USB : un contrôleur spéciale à interface USB et une mémoire Flash de type NAND.

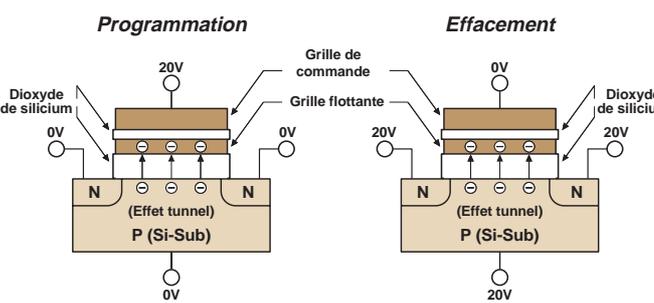


Figure 2. Principe de fonctionnement d'une cellule de mémoire Flash (source : Toshiba).

L'un des points d'intérêt importants dans le cas d'une mémoire Flash est indubitablement sa fiabilité. Les cellules de mémoire Flash ne sont pas éternelles et l'une d'entre elles peut rendre l'âme en cours d'utilisation. C'est là la raison d'un système de correction d'erreur (ECC). Lors de toutes les opérations (lecture, écriture mais aussi effacement) on a vérification des données. En cas de découverte d'erreur, celle-ci est corrigée et le contenu de la totalité du bloc où a eu lieu l'erreur est recopié vers un autre bloc encore libre de la mémoire.

Le fabricant donne à sa mémoire Flash, une durée de vie de 100 000 cycles d'effacement/écriture. Les données stockées restent conservées pendant 10 ans au moins dans les cellules de mémoire.

## Comment ça fonctionne ?

La structure de la mémoire Flash rappelle celle de l'EPROM. Chaque cellule de mémoire se compose d'un FETMOS à 2 grilles (figure 2). Il

existe, entre la grille de commande et le substrat du FET une porte flottante. Cette porte est isolée presque parfaitement par le biais d'une couche de dioxyde de silicium d'une très faible épaisseur. Les électrons injectés dans cette grille ne peuvent partant plus fuir après coupure de l'alimentation. Le nombre d'électrons présents sur la grille flottante déterminent le niveau (« 1 » ou « 0 ») stocké dans la cellule. La programmation et l'effacement des cellules se fait par l'application d'un champ puissant entre la grille et le substrat ce qui permet aux électrons de vaincre la barrière d'isolation (processus connu sous le nom d'effet tunnel de Fowler-Nordheim).

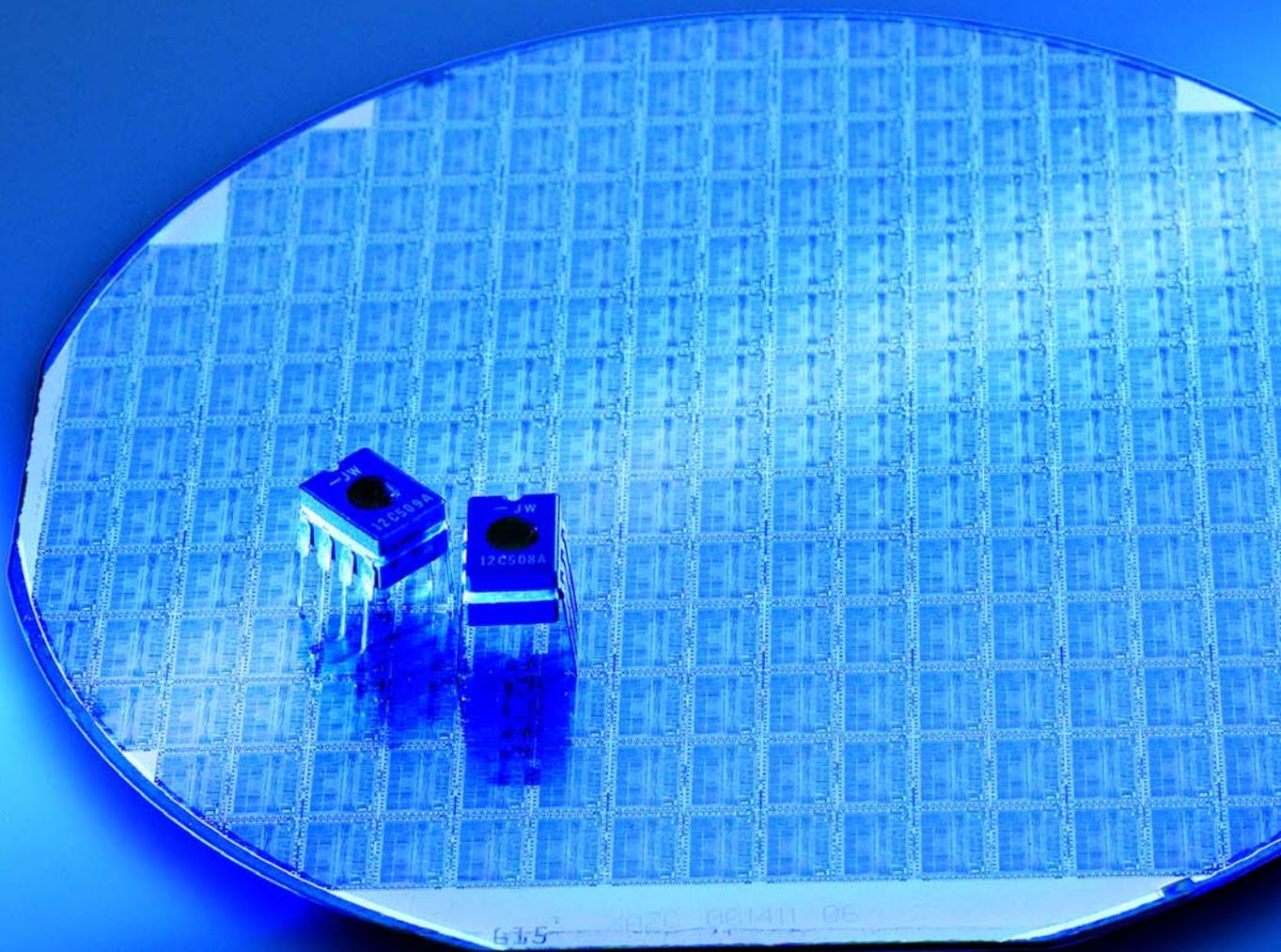
Aujourd'hui, des millions de mémoires USB et de cartes de mémoire ont prouvé que les mémoires Flash sont un medium fiable auquel on peut confier des données les yeux fermés et pour longtemps.

(040273-1)

# MICROCONT

*Christian Tavernier*

**S'il est un circuit intégré qui ne cesse de progresser depuis sa première introduction sur le marché, c'est bien le microcontrôleur puisqu'il ne se passe pas de semaine sans qu'un nouveau produit soit commercialisé.**



# RÔLEURS

## LES GRANDES TENDANCES

Faire son choix devient donc de plus en plus difficile et, afin de vous aider, nous vous proposons donc de découvrir, dans les lignes qui suivent, les grandes tendances du moment.

### Des fabricants toujours plus nombreux

Si les poids lourds des microcontrôleurs tels que Microchip, Motorola, Atmel ou dans une moindre mesure Philips sont toujours bien présents sur le marché, d'autres se sont lancés dans l'aventure parfois assez tardivement ou bien ne sont pas connus pour être des spécialistes en ce domaine, au point qu'on les oublie fréquemment. Sachez donc par exemple que Toshiba fabrique des microcontrôleurs avec sa gamme TMP86xxx mais aussi Zilog, longtemps célèbre pour son Z80 lors des débuts de la micro-informatique grand public, avec ses familles Z8encore ! et eZ80, sans oublier Dallas Semiconductor avec son DS89C420, Cypress avec ses PsoC™ (Programmable System-on-Chip™) spécialisés comme le CY8C27x ou bien encore une société totalement inconnue il y a peu de temps : Cygnal avec ses C8051xxx.

Ces produits, pour intéressants qu'ils soient, arrivent cependant un peu tardivement sur un marché déjà bien encombré et, sauf à présenter des particularités remarquables, il est peu probable qu'ils arrivent à faire une percée notoire comme ce fut le cas par exemple des PIC de Microchip il y a de cela quelques années.

Ce n'est pas nécessairement un problème pour leurs fabricants ; Toshiba par exemple fabrique suffisamment de produits finis pour pouvoir y intégrer ses propres microcontrôleurs, dont certains sont d'ailleurs très orientés électronique grand public.

En ce qui nous concerne, nous allons nous intéresser plus en détail aux produits des poids lourds évoqués ci-dessus car leur expérience en ce domaine leur permet de conserver une longueur d'avance et de proposer de ce fait des produits parfois très innovants.

### Une progression ... vers le bas

Tout le monde sait que les microcontrôleurs ne cessent de progresser vers le haut : plus de mémoire, plus d'entrées/sorties, plus de ports intégrés, etc. Ce que l'on sait moins c'est que la grande tendance du moment est de les faire aussi progresser vers le bas, plus particulièrement chez Microchip et, dans une moindre mesure, chez Atmel.

Microchip nous propose ainsi aujourd'hui une évolution de sa gamme 12Cxxx avec les 12Fxxx. Ce changement de lettre n'est pas anodin en effet, les 12C ne se programment qu'une fois car leur mémoire de programme

est de type OTP (*One Time PROM* ou PROM programmable une seule fois) tandis que les 12F sont équipés de mémoire de programme de type Flash, programmable et effaçable électriquement plusieurs milliers de fois.

Proposés en boîtiers à 14 ou à 8 pattes, ces circuits ont pour but de conquérir toutes les applications à faible coût dans lesquelles on répugnait jusqu'à ces dernières années à mettre un microcontrôleur en raison de son prix et/ou de sa taille. Si vous êtes un fidèle lecteur d'Elektor vous connaissez déjà le 12C508 (**figure 1**), qui fut le premier membre de cette famille, mais nous vous invitons aujourd'hui à découvrir par exemple le 12F629 qui, du « haut » de son boîtier 8 pattes, vous propose 1 K de mémoire de programme de type Flash, 128 octets d'EEPROM, 64 octets de RAM, 6 lignes d'entrées/sorties un compteur programmable (*timer*) 8 bits et un de 16 bits. Si cela ne vous suffit pas, sachez que le 12F675 (la logique de la numérotation nous échappe quelque peu !) est identique mais dispose en plus en interne d'un convertisseur analogique digital 10 bits à 4 canaux. Ces circuits peuvent fonctionner avec une horloge RC interne, qui les dispense alors de tout quartz, mais ils sont dans ce cas limités à 4 MHz, tandis qu'un quartz externe leur permet de grimper allègrement jusqu'à 20 MHz.

Atmel n'est pas en reste en ce domaine avec ses circuits de la famille ATtiny (*tiny* peut se traduire par petit en anglais ne l'oublions pas) qui offre 9 modèles différents. Même si ce n'est pas à proprement parler une nouveauté, l'ATtiny 2313 mérite votre attention puisqu'il dispose quant à lui de 2 K de mémoire de programme de type flash, 128 octets d'EEPROM, 128 octets de RAM, un UART (émetteur-récepteur de données séries asynchrones), 18 lignes d'entrée/sorties parallèles et qu'il est capable de fonctionner à 20 MHz de fréquence d'horloge ; le tout par contre dans un boîtier 20 pattes cette fois-ci.

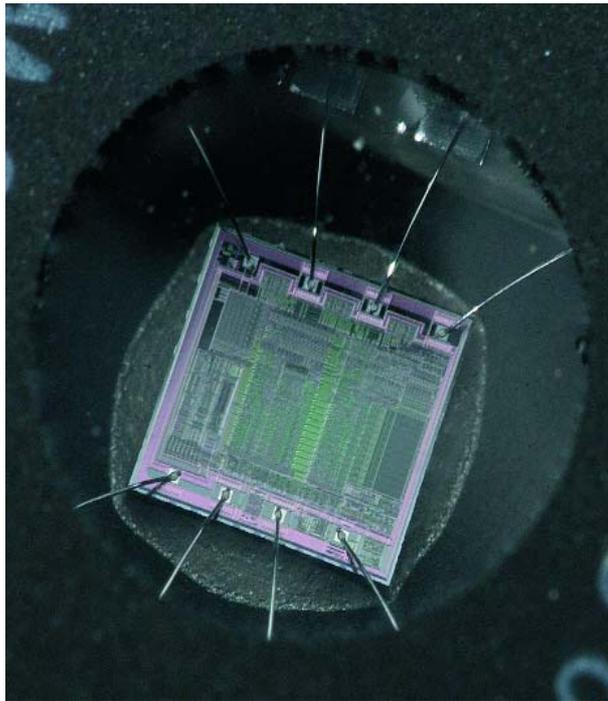
Si l'on reste dans ce domaine des « petits » microcontrôleurs, c'est cependant Microchip qui crée l'attraction avec ...

### Les rfPIC ou « radio » PIC

Ayant fait le constat que des microcontrôleurs sont de plus en plus fréquemment intégrés dans des télécommandes radio, Microchip a décidé de mettre sur le marché ses rfPIC soit, mot à mot, ses PIC radio-fréquence. Ils appartiennent à la famille des « petits » PIC que nous venons d'évoquer, mais intègrent dans un seul et même boîtier, le microcontrôleur lui-même bien sûr, ainsi que toute la circuiterie nécessaire pour réaliser un émetteur radio modulé en amplitude (ASK) ou en fréquence (FSK). Afin d'assurer la stabilité de la fréquence d'émission un oscillateur contrôlé par une tension (VCO) et une boucle à verrouillage de phase (PLL) intégrés asservissent celle-ci à la fréquence du quartz d'horloge du circuit.

Si le rf12C509 est déjà intéressant en soi car c'est un 12C509 muni de cet émetteur ; le tout nouveau rf12F675 devrait susciter encore plus de vocations parmi les plus créatifs d'entre-vous. C'est en effet un 12F675, brièvement présenté ci-dessus, muni de cette partie radio. Et ne croyez pas qu'elle soit délicate à mettre en œuvre : la **figure 2** est là pour vous prouver le contraire.

Figure 1. Une image qui va bientôt disparaître avec l'omniprésence des mémoires Flash : la fenêtre en quartz des microcontrôleurs à mémoire effaçables aux ultra-violets.



## Toujours plus

Hormis l'augmentation traditionnelle de la taille des mémoires intégrées dans les microcontrôleurs, qui reste la tendance de ces dernières années, deux autres axes d'évolution se dessinent.

Le premier concerne l'augmentation de la vitesse de fonctionnement. Si de nombreux circuits disposent encore d'une classique horloge à 4 MHz, les catalogues de Microchip, d'Atmel ou bien encore de Motorola regorgent de circuits fonctionnant à 16 voire 20 MHz. Attention toutefois à ne pas vous laisser abuser par ces chiffres. Un ATmega128 d'Atmel par exemple fonctionne "seulement" à 16 MHz de fréquence d'horloge mais exécute quasiment toutes ses instructions en un seul cycle d'horloge. A contrario, un MC9S12D de la famille HCS12 de Motorola peut fonctionner jusqu'à 25 MHz

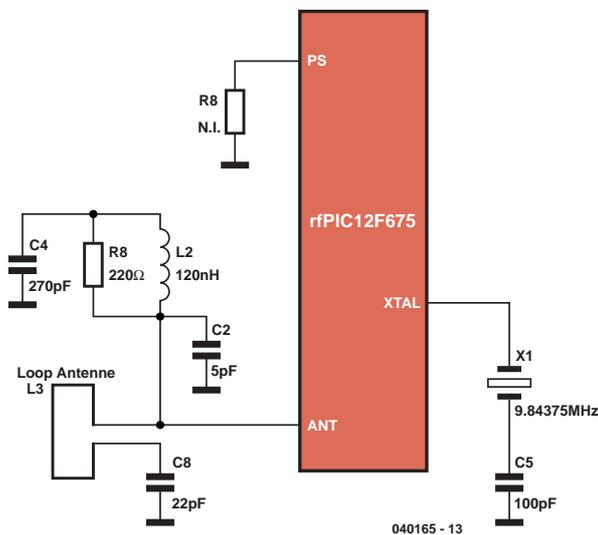


Figure 2. La mise en oeuvre de la partie haute fréquence d'un rfPic reste très simple.

mais demande plusieurs cycles d'horloge par instruction. Le second axe d'évolution concerne la spécialisation des périphériques internes. Si les UART (interfaces séries asynchrones) et autres SPI (interfaces séries synchrones) sont présents depuis déjà de nombreuses années dans les plus richement dotés des microcontrôleurs de tous les fabricants ; ce sont aujourd'hui des interfaces plus spécialisées qui font leur apparition. Le bus I<sup>2</sup>C existe ainsi chez quasiment tous les fabricants. Parfois il ne sait fonctionner qu'en mode esclave (le plus simple !) mais de plus en plus souvent il fonctionne aussi en mode maître. Stimulé par le marché du PC, c'est aujourd'hui le bus USB qui fait une entrée remarquée dans le monde des microcontrôleurs que ce soit chez Microchip avec les 16C745 ou 765 ou bien encore les futurs 18F2455, 2550, etc., chez Atmel avec l'AT91RM3400 ou bien encore chez Motorola ou même le « vieux » 6805 s'y met avec, par exemple, le 68HC705JBx.

## ARM

Nous ne pouvons pas, dans ce panorama du monde des microcontrôleurs, ne pas parler des produits ARM. Si ARM ne produit pas de microcontrôleurs dans le sens propre de ce terme, cette société propose, depuis plusieurs années déjà, des noyaux 32 bits que d'autres fabricants de microcontrôleurs renommés, pour le parler que d'AMI, Atmel, Cirrus, Philips, Samsung, STMicroelectronics, Texas Instruments, et même Intel !, utilisent allègrement dans leurs propres produits.

Il nous semble qu'il s'agit là aussi d'une tendance très forte. Pourquoi se casser la tête à réaliser son propre microcontrôleur 32 bits alors qu'il existe déjà un noyau parfaitement accepté dans le monde industriel. Le besoin de produits, de plus en plus sophistiqués, se devant quand même de rester confortable d'utilisation et maintenance aisée, a inévitablement un impact sur l'intelligence de tout système, le microcontrôleur. Le besoin de fonctionnalités de plus en plus nombreuses se traduit par le passage du 4 ou 8 bits traditionnel au 32 bits, le 16 bits semblant être une station dépassée, le 32 bits offrant des performances bien meilleures et une flexibilité sans égale pour le moment.

## Applications

Le développement de la robotique, ou bien encore l'utilisation de plus en plus importante des microcontrôleurs en automobile, conduit également les fabricants à intégrer des contrôleurs pour bus CAN dans leurs circuits, mais aussi des contrôleurs de moteurs qui peuvent être très évolués. C'est le cas chez Microchip avec le 18F2331 par exemple qui se paye en outre le luxe de fonctionner jusqu'à 40 MHz, mais c'est aussi vrai chez Motorola avec les MC68HC908MRx.

Dans cet océan de circuits numériques, le monde analogique n'est pas non plus oublié et tous les fabricants ont aujourd'hui à leurs catalogues des circuits intégrant des convertisseurs analogique digitaux parfois très évolués. Citons par exemple celui de l'ATmega128 d'Atmel qui est un modèle 10 bits à 8 canaux pouvant fonctionner en mode "normal" (single ended disent nos amis anglophones) sur 8 canaux, en mode différentiel sur 7 canaux ou bien encore en mode différentiel sur 2 canaux avec un amplificateur à gain programmable de 1 à 200 intégré. Pour la majorité des applications, il n'a donc plus

rien à envier à un convertisseur externe. Cet ajout de périphériques internes ne se fait pas au détriment de la mémoire, bien au contraire puisque c'est souvent dans les circuits les mieux équipés que l'on trouve les tailles de mémoires les plus impressionnantes. Nous vous proposons ainsi de découvrir, grâce au **tableau 1**, le contenu des trois plus « gros » microcontrôleurs actuels de Microchip (famille 18Cxxx), d'Atmel (famille ATmega) et de Motorola (famille HCS12). Si vous lisez ce tableau avec attention, vous verrez que l'on frôle presque la démesure avec les 99 entrées/sorties parallèles et les 256 K de mémoire Flash du MC9S12H256B de Motorola (**figure 3**) Il est bien loin le premier 68705P3 avec ses 20 entrées/sorties et ses 1,6 K de mémoire ...

### Tel le Phoenix ...

... cet oiseau mythique qui renaît de ses cendres dit-on ; il est un microcontrôleur dont on doit parler, même en ce second semestre 2004 : c'est le 8051. En effet, bien qu'il ait été mis sur le marché pour la première fois dans sa version de base vers la fin des années 80, il est toujours présent aujourd'hui et fait encore l'objet de nouveaux développements. Malgré son architecture vieillissante, surtout face aux PIC et aux AVR qui ont, rappelons-le, une architecture de type RISC ; le 8051 survit toujours par adjonction de nouveaux périphériques internes, par l'augmentation de la taille de sa mémoire de programme mais aussi par

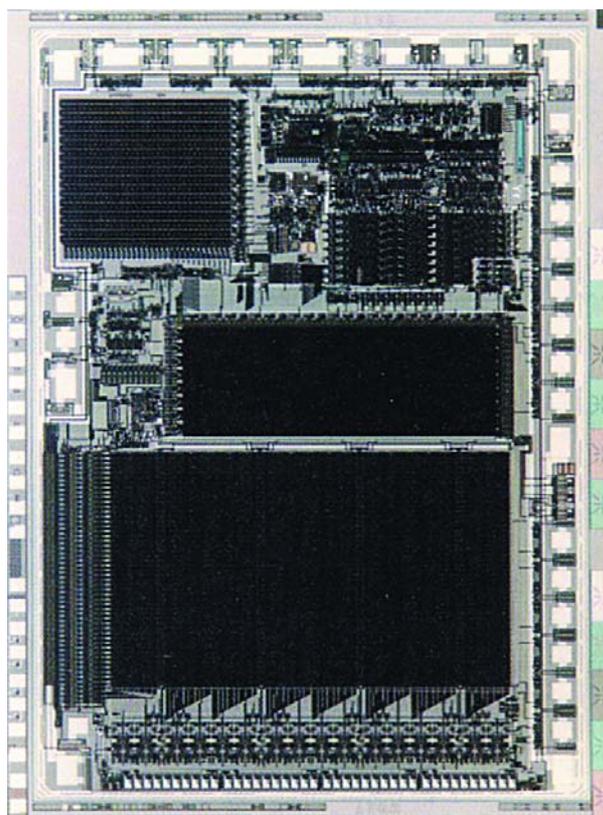
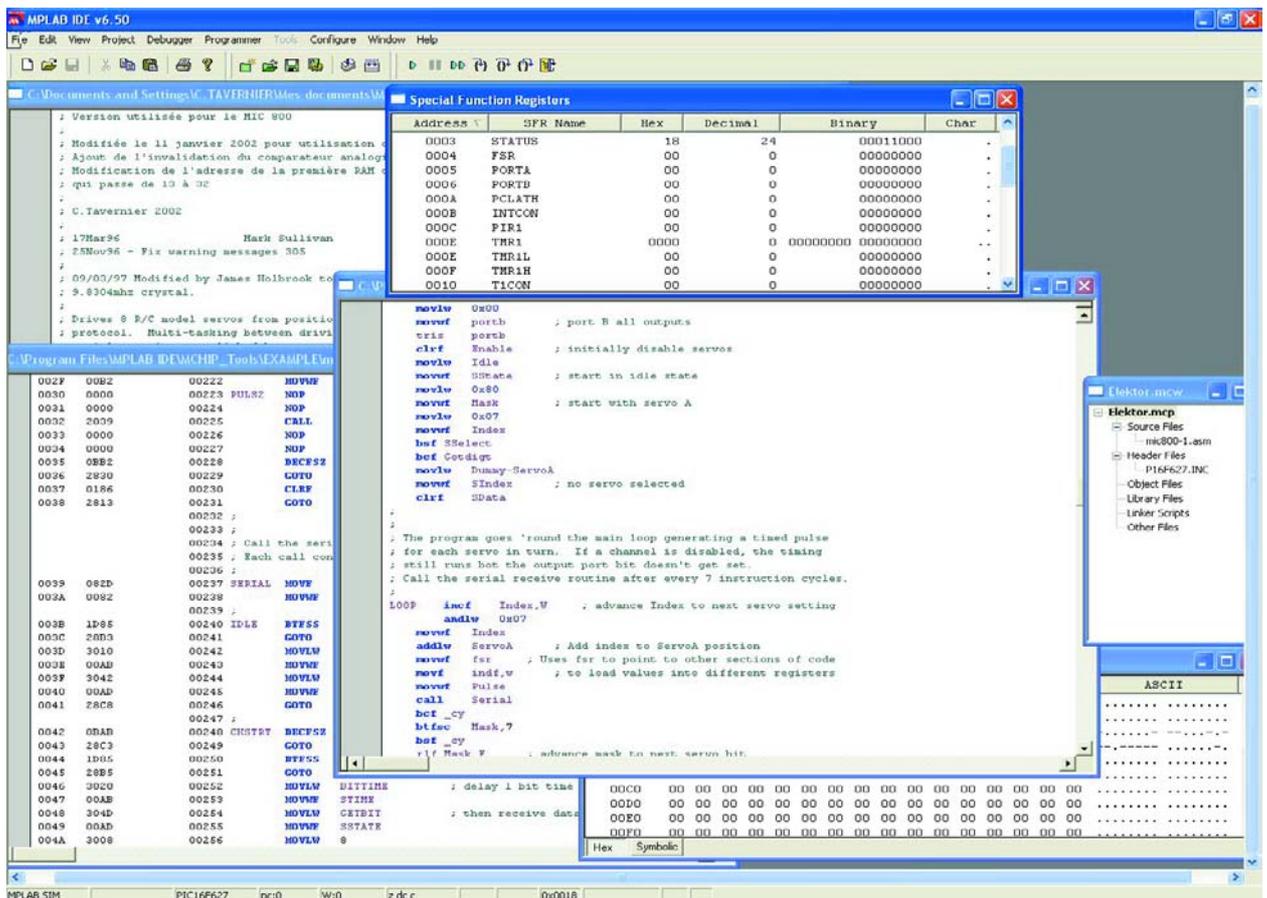


Figure 3. Sans légende (photo d'une puce de microcontrôleur).

**Tableau 1. Tableau comparatif du contenu de trois " monstres " récents de chez Atmel, Microchip et Motorola.**

Paramètre	ATmega128 (Atmel)	18F8720 (Microchip)	MC9S12H256 (Motorola)
Mémoire de programme (Flash)	128 K	128 K	256 K
Mémoire EEPROM de données	4 K	1 K	4 K
Mémoire vive (RAM)	4 K	3,8 K	12 K
Timers 8 bits	2	2	0
Timers 16 bits	2	3	1
Timers de capture et comparaison	4	5	8
Canaux PWM	8	5	6
Entrées/sorties parallèles	53	68	99
Interface série asynchrone (UART)	2	2	2
Interface série synchrone (SPI)	1	1	1
Interface I <sup>2</sup> C	-	1	1
Convertisseur analogique/digital	8 canaux	16 canaux	16 canaux
	10 bits	10 bits	10 bits
Comparateur analogique	1	2	-
Fréquence d'horloge maximum	16 MHz	40 MHz	32 MHz

Figure 4. Malgré sa gratuité, MPLAB intègre tout ce qu'il faut pour développer avec un maximum de confort une application à base de microcontrôleur PIC.



l'adjonction de possibilités dont on le croyait incapable telle que la programmation en circuit (ISP ou ICSP, voir ci-dessous).

Si l'assembleur 8051 n'a plus de secrets pour vous, un détour du côté de chez Atmel, mais cette fois-ci en vous intéressant à la famille AT89xxx, devrait vous satisfaire. Tous les circuits dont la référence commence par cet acronyme sont en effet des évolutions du 8051 mais disposent de la programmation en circuit, de nouvelles ressources internes ou bien encore de vastes mémoires de programme.

Si vous trouvez malgré tout que ces circuits ont un petit air de « réchauffé » pourquoi alors ne pas vous intéresser à la toute nouvelle famille proposée par Philips sous le nom générique P89LPC900. Si ce nom ne vous dit rien, plongez-vous dans votre collection d'Elektor et consultez le numéro de novembre 2003 qui leur a consacré un article.

## Logiciels et outils de développement

A l'origine, c'est-à-dire lorsque les microcontrôleurs contenaient péniblement 1 ou 2 K octets de mémoire de programme, il n'était possible de les programmer qu'en assembleur. Quelques lignes de langage évolué auraient en effet suffi à saturer leur mémoire car, rappelons-le, une ligne de langage évolué génère très souvent de nombreuses lignes d'assembleur, c'est ce que l'on appelle le taux d'expansion du compilateur.

Même si l'assembleur reste encore indispensable aujourd'hui, si l'on souhaite réaliser de la véritable programmation temps réel, ou bien encore si l'on veut exploiter au mieux la très grande vitesse d'exécution permise par les

microcontrôleurs les plus récents ; la tendance actuelle est au langage évolué lorsque ces contraintes n'existent pas. De nombreux programmeurs préfèrent en effet manipuler du Basic ou bien encore du C plutôt que de l'assembleur. Il existe donc aujourd'hui des compilateurs C pour toutes les familles de microcontrôleurs. Cela va des produits haut de gamme, hélas assez coûteux, tel que le compilateur CCS C PIC Compiler de CCS, pour toutes les familles de PIC par exemple, à la version *freeware* PICC-Lite C compiler de Htsoft qui convient pourtant fort bien pour de nombreuses applications.

Pour ceux d'entre vous qui sont réfractaires au C ou à l'assembleur, sachez qu'il existe aujourd'hui d'excellents compilateurs Basic, notamment pour les familles PIC de Microchip avec le PicBasic Pro Compiler de Micro Engineering Labs par exemple ou bien encore le Proton de Crownhill Associates, ainsi que pour la famille AVR d'Atmel avec le Bascom AVR de Melselec.

La famille 8051 n'est pas oubliée avec, pour ce qui est du Basic, le ... Bascom de Melselec mais pour 8051 cette fois-ci, ou bien encore le SDCC, totalement gratuit quant à lui, mais pour le langage C. Seuls les produits Motorola font un peu figure de parents pauvres en ce domaine car l'offre en compilateurs les concernant est minimaliste !

Partant du constat que tout développeur d'application à base de microcontrôleur fait souvent appel à des modules logiciels identiques, signalons l'initiative originale de Microchip qui propose, avec son nouveau logiciel baptisé Maestro, de construire la majeure partie d'une application quasiment sans programmer. Cet environnement, disponible gratuitement sur le site de Microchip, permet en effet d'accoler ces modules standards au fur et à mesure de la réalisation de l'application, écrivant ainsi de manière quasi automatique la majorité du programme qui la constitue.

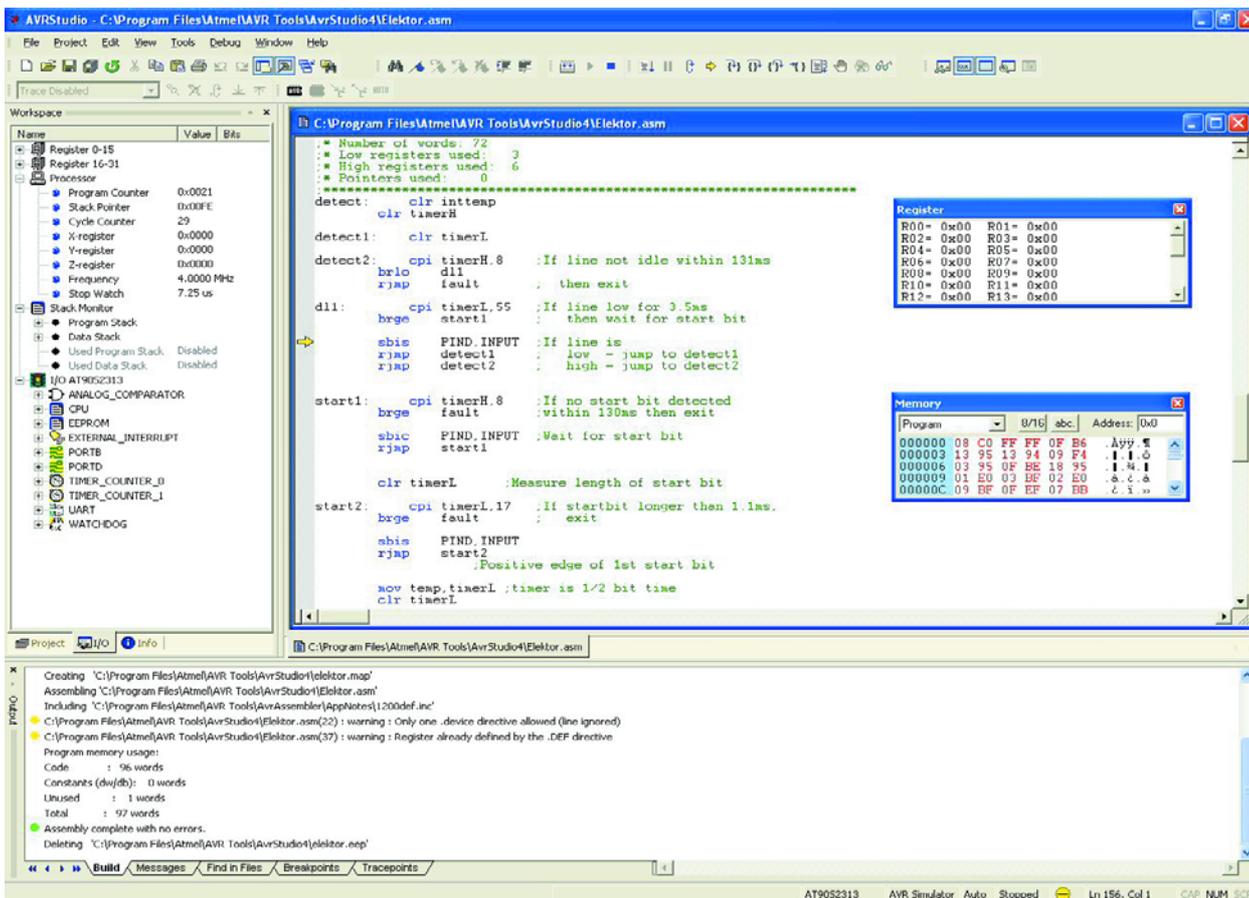


Figure 5. Avec des possibilités similaires, AVR Studio d'Atmel s'adresse à la famille AVR de ce fabricant.

Indépendamment du compilateur, tout développement sur microcontrôleur fait appel à ce que l'on appelle, fort justement d'ailleurs, un outil de développement. Ici encore c'est Microchip et Atmel qui tiennent le haut du pavé avec MPLAB (figure 4) d'une part et AVR Studio (figure 5) de l'autre. Ces environnements de développement fonctionnant sous Windows contiennent en effet tout ce dont rêvaient encore de nombreux programmeurs il y a seulement quelques années, simulateur logiciel y compris. Ils évoluent sans cesse puisque MPLAB en est à sa version 6.5 et AVR Studio à sa version 4 de façon à supporter les nouveaux produits de leurs fabricants respectifs au fur et à mesure de leur commercialisation. Aussi surprenant que cela puisse paraître, la tendance actuelle est à la gratuité. En effet, malgré leurs remarquables possibilités, ces outils de développement sont totalement gratuits. Il vous suffit juste de prendre la peine de les télécharger sur les sites de leurs fabricants respectifs pour pouvoir commencer à travailler. Notons toutefois un très net retard de Motorola, ainsi que de quelques autres fabricants moins connus, chez lesquels rien d'aussi complet et performant n'existe, tout au moins si l'on considère les produits gratuits.

## La programmation en circuit

Même si ce n'est pas une nouveauté en tant que telle, la programmation en circuit, appelée aussi ISP pour *In System Programming* ou bien encore ICSP pour *In Circuit Serial Programming*, est sans conteste l'évolution actuelle la plus significative en terme de programmation des microcontrôleurs. Si elle est bien gérée par le concepteur d'une application, elle permet en effet de se passer de tout programmeur tout en autorisant la programmation des microcontrôleurs qui la supportent au

moyen de n'importe quel PC muni d'une interface dérisoire de simplicité.

La figure 6 rappelle le principe de ce concept pour ceux d'entre-vous qui ne le connaîtraient pas encore. Le microcontrôleur ne quitte plus la carte de l'application pour laquelle il est prévu ; carte sur laquelle il peut prélever son alimentation et son horloge. Grâce à deux ou trois de ses lignes de port qui changent temporairement de fonction, il est possible d'effacer et de programmer sa mémoire de programme. Si ces lignes de port ne sont pas utilisées par l'application, l'interfaçage est immédiat avec le PC chargé de réaliser la programmation. Dans le cas contraire, quelques straps ou interrupteurs DIL permet-

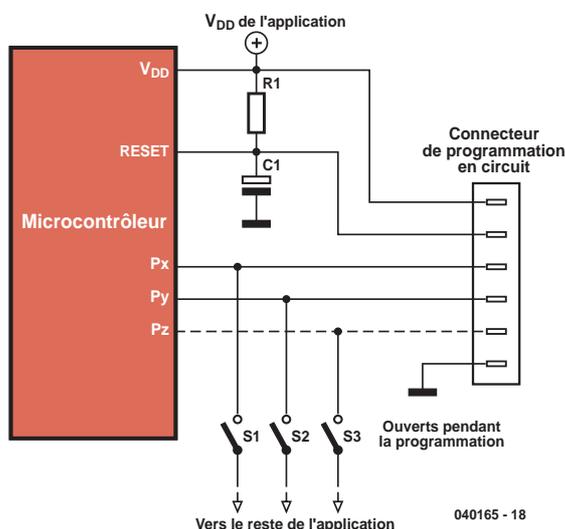
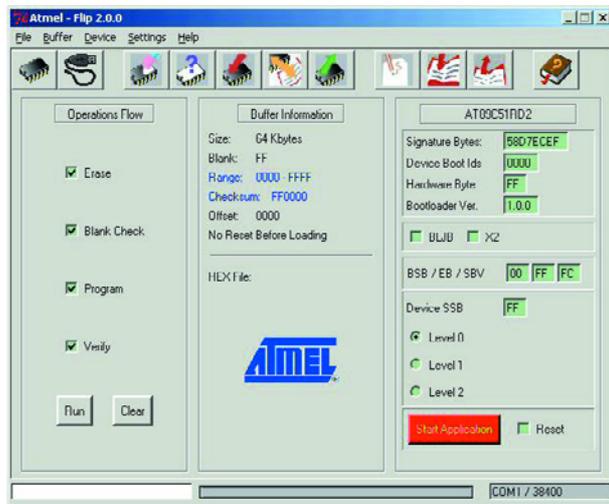


Figure 6. Principe général de la programmation en circuit (ISP ou ICSP).

Figure 7. FLIP d'Atmel ou comment programmer un microcontrôleur en circuit au travers de n'importe quelle interface série.



tent de les isoler du reste de l'application le temps de la programmation.

Tous les microcontrôleurs récents, équipés de mémoire flash, supportent ce mode de programmation et de très nombreux programmes gratuits sont disponibles sur Internet, tous fabricants confondus, pour gérer ce type de programmation.

Si l'on devait distinguer un tel logiciel, ce serait certainement FLIP (**figure 7**), proposé gratuitement par Atmel, et qui est capable de programmer en circuit de très nombreux microcontrôleurs de ce fabricant, pour peu qu'ils disposent d'au moins une interface série, peu importe qu'elle soit de type asynchrone (RS 232), SPI, USB ou même de type bus CAN. Aucun autre produit du marché n'est, pour l'instant, aussi polyvalent.

## Ces microcontrôleurs qui n'en sont pas vraiment

On ne saurait terminer cet article sans dire un mot de ces microcontrôleurs particuliers que sont les Basic Stamp, Pic Basic et autres Basic Tiger pour ne citer que les plus

connus. Ces « circuits », rappelons-le, intègrent en fait sur un minuscule circuit imprimé à la taille d'un boîtier 24, 28 ou 40 pattes un microcontrôleur rapide, pré-programmé avec un interpréteur de langage évolué. Le plus souvent il s'agit du Basic mais d'autres langages commencent à apparaître tel que Java avec le Javelin Stamp de Parallax ou bien encore la programmation en langage objet avec le OOPic.

Malgré leur succès sans cesse croissant, ces produits restent toutefois très coûteux si on les compare à de vrais microcontrôleurs, principalement en raison de leur mode de fabrication, et ils doivent donc être réservés à la réalisation de maquettes ou de très petites séries.

## En guise de conclusion

Il y aurait encore beaucoup à dire sur ce sujet mais, si l'on veut résumer en quelques mots l'évolution de ce marché on peut dire que la pénétration des microcontrôleurs est de plus en plus importante dans tous les secteurs de l'électronique, tant vers le haut avec des produits très performants, que vers le bas avec des produits de plus en plus petits et de moins en moins coûteux. La mise en oeuvre des microcontrôleurs devient aussi de plus en plus facile, tant en raison de la gratuité de nombre d'outils de développement que de la commercialisation de compilateurs performants.

Bien que cela contrarie certains de nos lecteurs, le microcontrôleur est devenu un composant indispensable à l'électronique et fait partant partie de notre quotidien, témoins les différents autres articles utilisant un microcontrôleur publiés dans Elektor ce mois-ci et les mois à venir. Lorsque vous lirez ces lignes, il se sera passé bien du neuf dans le monde des microcontrôleurs, témoin l'impact de Xtensa, le processeur se configurant par logiciel de Tensilica...

(040165-1)

## Adresses Internet des fabricants cités dans l'article :

Site de l'auteur <http://www.tavernier-c.com/>

### Microcontrôleurs classiques et outils de développement associés

ARM: <http://www.arm.com/>  
 Atmel: <http://www.atmel.com/>  
 Microchip: <http://www.microchip.com/>  
 Cygnal: <http://www.cygnal.com/>  
 Cypress: <http://www.cypress.com/>  
 Motorola: <http://mot-sps.com/>  
 Philips: <http://www.semiconductors.philips.com/>  
 Toshiba: <http://www.toshiba.com/taec/>  
 Zilog: <http://www.zilog.com/>

### Microcontrôleurs " spéciaux "

Parallax: <http://www.parallax.com/>  
 OOPic: <http://www.oopic.com/>  
 Basic Tiger: <http://www.wilke.de/>

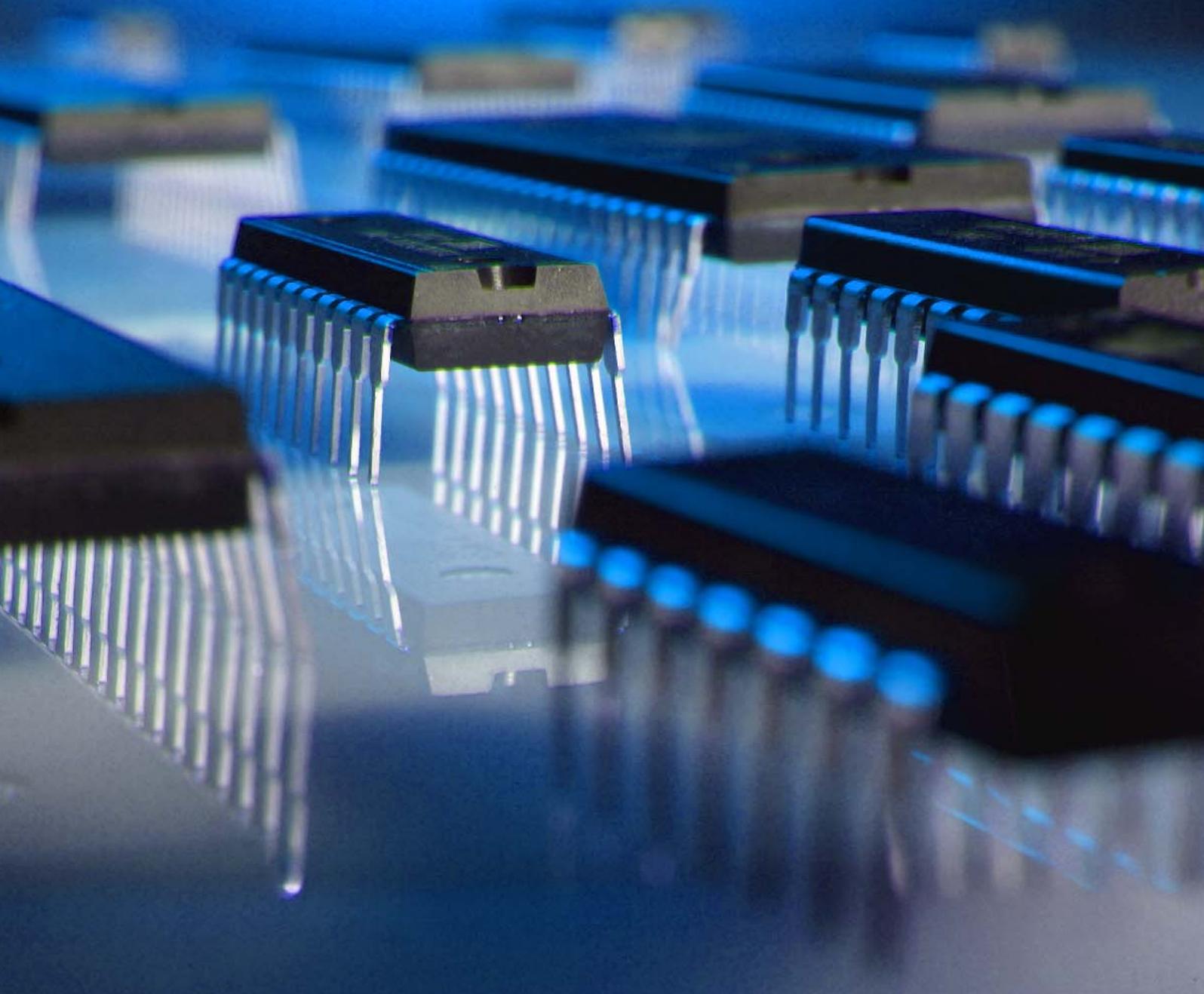
### Compilateurs

CCS C Pic Compiler: <http://www.ccsinfo.com/>  
 PICC-Lite C Compiler: <http://www.htsoft.com/>  
 Bascom AVR en Bascom 8051: <http://www.mcselec.com/>  
 Proton - Basic compiler: <http://www.picbasic.org/>  
 PicBasic (Pro) Compiler: <http://www.melabs.com/>  
 SDCC Compiler: <http://sourceforge.net/projects/sdcc>

# Le contrôl

## L'embarras du choix...

Il existe un nombre croissant de fabricants de semi-conducteurs proposant des microcontrôleurs. Il est extrêmement difficile, en raison de leur universalité, de choisir parmi eux le microcontrôleur idéal pour une application concrète.



# eur idéal

Il n'existe malheureusement pas de recette toute faite pour choisir le microcontrôleur adéquat, les caractéristiques de ces composants étant par trop différentes. Le point de départ est toujours l'application elle-même et ses exigences. Nous allons, ici, passer en revue les critères de choix et vous aider ainsi à définir les limites et faire votre choix.

## Vitesse

La « vitesse » d'un microcontrôleur ne dépend pas uniquement de la fréquence d'horloge maximale de la CPU et du générateur d'horloge (le quartz). Il est important aussi de savoir combien de cycles d'horloge il faut pour l'exécution d'une instruction et le langage utilisé (l'assembleur peut être bien plus rapide qu'un langage de haut niveau). Un contrôleur universel (cadencé à la bonne fréquence) pourra être utilisé jusque dans le domaine des mégahertz (applications vidéo).

## Mémoire de programme

Le programme est stocké dans de la mémoire non volatile. L'EPROM OTP interne ne se laisse programmer qu'une fois, il est préférable, en cours de développement, d'utiliser la version Flash du microcontrôleur, plus chère mais qui se laisse reprogrammer en quelques secondes. Ce processus peut se faire dans un programmeur ou en site, ISP (*In System Programmable*). Actuellement, la taille des mémoires de programme interne va de zéro à 1 024 Koctets (voire plus à la lecture de ces lignes). Les microcontrôleurs à « hublot » dotés d'EPROM effaçable aux UV ne sont plus de mode. On n'utilise plus de la mémoire EPROM externe à fenêtre d'effacement que dans le cas de programmes très gros, la Flash-RAM l'ayant remplacée.

## EEPROM

Lorsqu'il est nécessaire de conserver des variables même en cas de coupure de la tension d'alimentation, le microcontrôleur a besoin de mémoire non volatile EEPROM interne ou externe. Ces EEPROM ne sont PAS reprogrammables un nombre INFINI de fois.

La connexion des EEPROM externes au microcontrôleur se fait par liaison bifilaire. Elles entrent en jeu lorsqu'il faut, au démarrage du programme prendre en compte des valeurs individuelles telles que des données de calibration.

## RAM

La RAM sert au stockage des variables en cours d'exécution du programme. On trouve couramment des RAM d'une capacité allant jusqu'à 4 Koctets, mais les besoins sont souvent moindres. On peut envisager de la RAM externe.

## Entrées/Sorties numériques (I/O)

Il est relativement facile de déterminer le nombre des Entrées/Sorties numériques requises par l'application. Elles peuvent également être nécessaires pour la connexion de circuits intégrés périphériques, lorsque les ressources internes ne sont pas suffisantes. Pour les automates simples, le contrôleur peut s'en sortir avec très peu d'E/S (certains composants de la série Tiny d'Atmel ne possèdent que 8 broches), dans certains cas il faut utiliser des ports entiers en parallèle, de sorte que l'on se trouve en présence de « mille-pattes » difficiles à souder.

## Temporisateur/compteur

Si le programme a des durées à mesurer ou des événements à compter, le contrôleur a besoin d'un temporisateur/compteur. Les contrôleurs possèdent, jusqu'à 3 « timer/counter » 8 ou 16 bits pilotés par le biais de registres internes. On en a également besoin lorsqu'il faut fournir un signal indépendant (MLI, UART). Le temporisateur chien de garde en est un modèle spécial. Il est programmé à un intervalle que le programme exécuté doit réinitialiser à intervalle régulier.

## Interruptions externes

Les temporisateurs/compteurs ne sont pas les seuls à produire des interruptions. On dispose de jusqu'à 2 entrées d'interruption externe lorsqu'un événement externe doit interrompre le déroulement normal du programme et lancer une routine.

## Interfaces

Il est des interfaces intégrées standards très utiles, telles que I2C, I2S, SPI, CAN, USB, LIN voire celles qui servent à la connexion d'affichages LCD courants. S'il est également possible de les émuler, ceci demande d'excellentes connaissances de programmation (assembleur).

## Matériel analogique

En règle générale, les microcontrôleurs possèdent des interfaces intégrées vers le monde analogique, au nombre desquelles les CAN (à résolutions diverses et dotés en amont d'un multiplexeur analogique), comparateurs analogiques voire amplificateur opérationnel (à la sortie accessible de l'extérieur).

## Modes de fonctionnement

Si le microcontrôleur doit être alimenté par pile, les modes de fonctionnement prennent toute leur signification. Il est possible de mettre en sommeil différentes parties du contrôleur.

## « Programmabilité »

Dès l'instant où le contrôleur possède une interface ISP, il

peut être programmé dans l'application même. Il faut sinon disposer d'un programmeur plus ou moins complexe. Un microcontrôleur sans ISP soudé sur une platine devient pratiquement impossible à reprogrammer.

### Spécificités

Il existe des microcontrôleurs qui sont réservés à des types d'applications spécifiques, la commande de moteurs, DSP, contrôleur à module d'entrée HF... Il peut être intéressant, en fonction de l'application, d'envisager l'utilisation d'un microcontrôleur spécialisé.

### Prix, disponibilité, boîtier

À l'unité, ou pour les petites séries, le prix ne devrait guère jouer de rôle. Mais à quoi sert-il d'avoir trouvé le microcontrôleur « idéal » si l'on constate qu'il n'est disponible qu'en Corée et ce à un minimum de 10 000 unités. La forme du boîtier peut elle aussi jouer un rôle important, car qui d'entre nous est capable de souder, à

l'aide de moyens « normaux » une *flatpack* de 256 contacts espacés d'un 1/10<sup>ème</sup> de mm ?

### Environnement de développement

Il s'agit là d'un aspect important pour le concepteur semi-professionnel. Si l'on travaille en assembleur, on recommence à zéro pour chaque nouveau contrôleur. Si cela constitue un obstacle, il faudra utiliser toujours le même contrôleur ou utiliser un compilateur pour langage de haut niveau (C, Pascal, BASIC), utilisable pour toute une famille; cependant la taille du code généré est sensiblement supérieure à celui obtenu en assembleur. À cela s'ajoute le prix de l'environnement de développement. Pour certaines familles, le fabricant propose gratuitement un IDE complet, pour d'autres, il faut « tirer le porte-monnaie ». Les programmeurs du commerce et leurs pods complexes peuvent revenir très cher.

(040286-1)

# Fabricants de microcontrôleurs

## Familles 4, 8, 16, 32 et 64 bits

### Altera - [www.altera.com](http://www.altera.com)

32 bits : EPXxxx (ARM V4T), Nios (Nios)

### Analog Devices - [www.analog.com](http://www.analog.com)

8 bits : ADuC8xx (8051)

### AMD - [www.amd.com](http://www.amd.com)

32 bits : Au1x00 (MIPS)

### Atmel - [www.atmel.com](http://www.atmel.com)

4 bits : T48C510, ATAxxx (MARC-4)

8 bits : AVR (AVR), AT89xxxx (8051),  
Mega AVR (AVR)

16 bits : C251 (8051), AT91xxxx (ARM)

### ARC International - [www.arc.com](http://www.arc.com)

32 bits : ARC501 (ARCompact), ARC7xx, ARC6xx,  
ARCTangent (RISC)

### ARM - [www.arm.com](http://www.arm.com)

32 bits : ARM10xx, ARM11xx, ARM7xx, ARM9xx,  
SCxxx, MPCore (ARM)

### Cirrus Logic - [www.cirrus.com](http://www.cirrus.com)

32 bits : CS89712, EP73xx, EP93xx, PS7500xx (ARM)

### Cybernetic Micro Systems - [www.controlchips.com](http://www.controlchips.com)

8 bits : P-51 (8051)

### Cygnal (→ Silicon Laboratories) - [www.silabs.com](http://www.silabs.com)

8 bits : C8051Fxxx (8051)

### Cypress Microsystems - [www.cypressmicro.com](http://www.cypressmicro.com)

8 bits : CY8C2xxxx (M8C)

### Dallas Semiconductor (Maxim Integrated Products) - [www.maxim-ic.com](http://www.maxim-ic.com)

8 bits : DS2xxx, DS5xxx, DS80Cxxx, DS87Cxxx,  
DS89Cxxx, MAX765x (8051)

### Fujitsu Microelectronics - [www.fujitsu.com](http://www.fujitsu.com)

8 bits : MB89xxx (F2MC-8L)

16 bits : MB90xxx (F2MC-16)

32 bits : MB91xxx (FR)

### IDT - [www.idt.com](http://www.idt.com)

32 bits : RC32xxx (MIPS)

### Infineon Technologies - [www.infineon.com](http://www.infineon.com)

8 bits : C5xx, C868 (8051)

16 bits : C16xxx (C166 v1), XC16xxx (C166 v2)

32 bits : TC111B, TC19xx (TriCore V1.3), TC17xx (TriCore V1.2)

### Intel - [www.intel.com](http://www.intel.com)

8 bits : 8xC251x, 8xC51xx (MCS51)

16 bits : 80C18x, 8xCx96xx (=MCS-x96xxx)  
32 bits : 80960 (i960), IXC1100, IXP4 (StrongARM v5TE)

### **Microchip Technology - [www.microchip.com](http://www.microchip.com)**

8 bits : PIC12xxx, PIC14xxx, PIC16xxx, PIC17xxx, PIC18xxx, rPIC (PIC micro)

16 bits : dsPICxxxx (Modified Harvard RISC)

### **MIPS Technologies - [www.mips.com](http://www.mips.com)**

32 bits : 4Kxx, M4K, 24Kx (MIPS)

64 bits : 20k, 10Kx, 5Kx (MIPS)

### **Motorola Semiconductor - [www.freescale.com](http://www.freescale.com)**

8 bits : MC68Hxxx (HCOx)

16 bits : HCS12x, M68HCxx (HCS12)

32 bits : 68300 (68K), MCF5xxx (ColdFire), Mcore xxx (RISC), MAC7xxx (ARM), MPC5xxx (PowerPC)

### **National Semiconductor - [www.national.com](http://www.national.com)**

8 bits : COP8xxxx (Modified Harvard RISC)

16 bits : CR16xxxx (Compact RISC)

### **NEC Electronics - [www.necel.com](http://www.necel.com)**

8 bits : 78K0S/Kx1, 78K0/Kx1 (NEC K)

32 bits : v850ES/Kx1 (v800)

64 bits : VR41xx, VR5xxx (MIPS)

### **Oki Semiconductor - [www.okisemi.com/us](http://www.okisemi.com/us)**

4 bits : MSM6318xx, MSM6415xx (nX)

32 bits : ML67xxxx (ARM7TDMI)

### **Philips Semiconductors - [www.semiconductors.philips.com](http://www.semiconductors.philips.com)**

8 bits : P8xC5x, P89LPC9xx, P8xLPC76x (8051)

16 bits : PxAxxx (XA)

32 bits : LPC2x0x (ARM7)

### **Rabbit Semiconductor - [www.rabbitsemiconductor.com](http://www.rabbitsemiconductor.com)**

8 bits : Rabbit 2000, Rabbit 3000 (Z80/180)

### **Renesas Technology - [www.renesas.com](http://www.renesas.com)**

4 bits : M45xx (720), H4xxx (HMCS400)

8 bits : M38xx, M78xx, M3754x, H8/380xx (H8)

16 bits : M77xx, M79xx (740), H8/30xxx (H8), H8S2xxx (H8S), M16C/xx, M32C/xx (M16C), H8/36xx, H8SX/1xxx (H8)

32 bits : SH-xxxx (SuperH), M321xx (RISC)

### **Silicon Storage Technology - [www.sst.com](http://www.sst.com)**

8 bits : SST89xxxx (FlashFlex 51)

### **Sharp Microelectronics - [www.sharpsma.com](http://www.sharpsma.com)**

16 bits : LH754xx (ARM)

32 bits : LH7952x, LH7A4xx (ARM)

### **STMicroelectronics - [www.stm.com](http://www.stm.com)**

8 bits : ST62xx (ST6), ST72xxx, ST7FLite (ST7), uPSD3xxxx (8032)

16 bits : ST10xxxx (80C166), ST92Fxxx (ST9), STR7xxx (ARM)

32 bits : ST40RA (SH4)

### **SuperH - [www.superh.com](http://www.superh.com)**

32 bits : SH-4xxx (SuperH)

64 bits : SH-5xxx (SuperH)

### **Tensilica - [www.tensilica.com](http://www.tensilica.com)**

32 bits : Xtensa V, Xtensa LX (Xtensa)

### **Texas Instruments - [www.ti.com](http://www.ti.com)**

16 bits : MSP430xxxx (MSP), TMS470 (ARM)

### **Toshiba America Electronic Components - <http://chips.toshiba.com>**

8 bits : TMPx8xxxx (TLCS)

16 bits : TMP96xxxx, TMP91xxxx, TMP95xxxx, TMP93xxxx (TLCS)

32 bits : TMP92xxxx, TMP94xxxx (TLCS), TMPR19xxx, TMPR39xxx (MIPS)

64 bits : TMPR49xx, TMPR99xx (MIPS)

### **Triscend - [www.triscend.com](http://www.triscend.com)**

8 bits : E5 (8051)

### **Ubicom (ex-Scenix) - [www.ubicom.com](http://www.ubicom.com)**

8 bits : SXxxxx, IP2012/2022 (MASI)

32 bits : IP3023 (MASI V2)

### **Xemics - [www.xemics.com](http://www.xemics.com)**

8 bits : E88LC0x (RISC)

### **Xilinx - [www.xilinx.com](http://www.xilinx.com)**

32 bits : PowerPC 405 (PowerPC)

### **Zilog - [www.zilog.com](http://www.zilog.com)**

8 bits : eZ80xxx, Z8xxxx, Z8Fxxx, Z8 Encore! (Z80/180)

# Swiss Army

Jim Spence

Pressé ? Vous faut-il concevoir un projet à microcontrôleur ILLICO PRESTO ? Nous voulons prouver ici qu'il est possible de réaliser et de programmer une carte à microcontrôleur quasi-universelle. Son coeur est un Atmel 89C8252, processeur à architecture 8052. Il est épaulé par un logiciel très spécial (et gratuit) qui ouvre des perspectives intéressantes. La communication avec le circuit peut se faire de 2 manières : soit par USB 2.0 soit par RS-232.



# Knife

## Tiny BASIC, assembleur 8051, RS232 et USB dans un même projet

L'intention à l'origine de ce projet était de réaliser un contrôleur embarquant un BASIC qui retiendrait le programme même alimentation coupée, qui démarerait automatiquement à la mise sous tension et ne requerrait pas trop de circuits intégrés. Il devait en outre être très facile à utiliser et ne pas exiger de logiciel spécial sur le PC pour fonctionner. Ce défit prend, plus ou moins, la forme d'un unique composant. Ceci est possible en raison des 2 Koctets d'EEPROM de données qui peuvent être programmées à l'aide d'instructions de haut niveau que possède le 89C8252.

Le microcontrôleur AT89C8252 a fait sa première apparition dans Elektor avec la **carte 89C8252 Flash** en décembre 2001, carte devenue un classique et vendue à plus de k exemplaires.

Vous vous demandez peut-être pourquoi ne pas utiliser de compilateur C ou BASIC sur l'ordinateur pour ensuite transférer le code dans le micro. Il s'agit indubitablement d'une bonne approche, mais elle vous fait perdre l'instantanéité de vos actions. La carte SAK exécute *immédiatement* tout code lui étant envoyé. Si vous entrez :

```
Pz1=0
```

au clavier, vous mettez immédiatement toutes les lignes du port 1 au niveau bas. De plus, il faut un minimum de connaissances en ce qui concerne les compilateurs C et BASIC avant d'arriver à quoi que ce soit. Si tout ce que vous voulez est de continuer à expérimenter avec votre projet de robotique favori, voici la voie à suivre. Rien ne vous interdit, ultérieurement, de vous essayer aux compilateurs ou à l'assembleur...

### RS-232 et/ou USB

Au cours de la mise au point l'auteur en eut assez d'avoir à incorporer un système de communication sur chaque carte même si ce dernier n'était pas

nécessaire sur l'application finale. Cette frustration s'est traduite par une alimentation et une connectivité RS-232 distinctes pouvant être utilisées avec d'autres circuits. Cependant, la solution RS-232 est dépassée au point que nombre de PC modernes ne possèdent plus de port sériel, ce qui nous a amené à penser à une interface USB d'autant plus que celle-ci peut assurer l'alimentation du circuit. Vous avez ainsi le choix entre RS-232 et USB lorsqu'il vous faut parler au micro présent sur votre carte SAK.

La carte SAK (*Swiss Army Knife*) est constituée de 3 sections : microcontrôleur (MCU), interface RS-232 et interface USB. Nous les aborderons successivement dans les paragraphes à suivre.

### Le microcontrôleur et sa cour

L'électronique de l'unité centrale (MCU) représenté en **figure 1** incorpore un microcontrôleur fameux, un AT89C8252, cadencé à 22,118 MHz; il est doté d'une version de Tiny Basic connue sous le nom de Tiny Control Basic (TCB). Il est difficile de faire plus compact. Il est possible de programmer la mémoire de programme en utilisant un mélange d'assembleur et de langage de haut-niveau. IC2 a été ajouté en vue de faciliter ces opérations bien qu'il soit possible de le supprimer dans l'application finale ou si l'on n'a pas besoin de programmation en circuit (ICP).

Au niveau des embases, K1 reproduit toutes les lignes d'Entrée/Sortie sous la forme d'une embase à 40 contacts, K2 étant une embase à 14 contacts pouvant se connecter à chacune des 2 autres platines. Les broches 1 et 2 assurent la communication sérielle. Les broches d'acquiescement (*hands-*

*hake*) ne sont pas concernées en cas de communication sérielle.

Le AT89C8252 se laisse programmer sériellement par le biais de 3 lignes seulement : SCK (*Serial Clock*), MISO (*Output*) et MOSI (*Input*). Il est possible, en utilisant ces 3 lignes et en émettant des octets spéciaux, d'accéder directement à la mémoire de code sans extraire la puce du circuit d'où le terme de *programmation en circuit*.

Le broche 5 de K1, la ligne DTR, initialise ce processus; le tableau 1 décrit les autres broches. Le signal DTR valide la moitié supérieure de IC2, un tampon inverseur 3 états. IC2 inversant toutes les lignes, le logiciel doit en tenir compte. Le terminal ByVac auquel nous reviendrons intègre le programme requis à cet effet.

Outre la programmation en circuit, il existe d'autres signaux concernant l'interface USB. La broche 6 est baptisée Ring Indicator. Si tant est que le PC soit paramétré correctement et que la carte USB est configurée correctement elle aussi, cette ligne réveille un PC en sommeil. Par la mise de cette ligne au niveau bas, la puce USB enverra les signaux corrects permettant de faire sortir un PC de sa léthargie. Si l'on veut que ce processus fonctionne correctement le microcontrôleur ne peut pas se trouver en alimentation autonome (par l'interface USB) vu que le courant disponible en veille n'est pas en mesure d'activer le microcontrôleur si lui non plus ne se trouve pas en veille; et dans ce mode-là, le contrôleur sera dans l'incapacité d'initier tout signal. Les spécifications USB définissent strictement le courant pouvant être drainé de l'interface et à quel moment. Un port USB normal peut fournir un courant limité (100 mA) jusqu'à ce qu'il soit configuré correctement. Une fois activé, le bus peut fournir un maximum de 500 mA. La broche 7 passe au niveau bas signalant une configuration

# Caractéristiques :

- BASIC (TCB) nombre entier embarqué
- Le BASIC peut appeler l'exécution de code Instant de programmes en assembleur, sans phase de compilation
- Ne requiert, pour sa programmation, qu'un simple émulateur de terminal
- Auto-détection du taux de transmission (baud)
- Programme BASIC utilisateur, exécution de la ligne 10 en cas de réinitialisation
- Possibilité de programmation de la mémoire de code en circuit via ByVac-Terminal
- Connectivité d'E/S
- Option USB 2.0
- Option RS-232
- Tout le logiciel disponible gratuitement

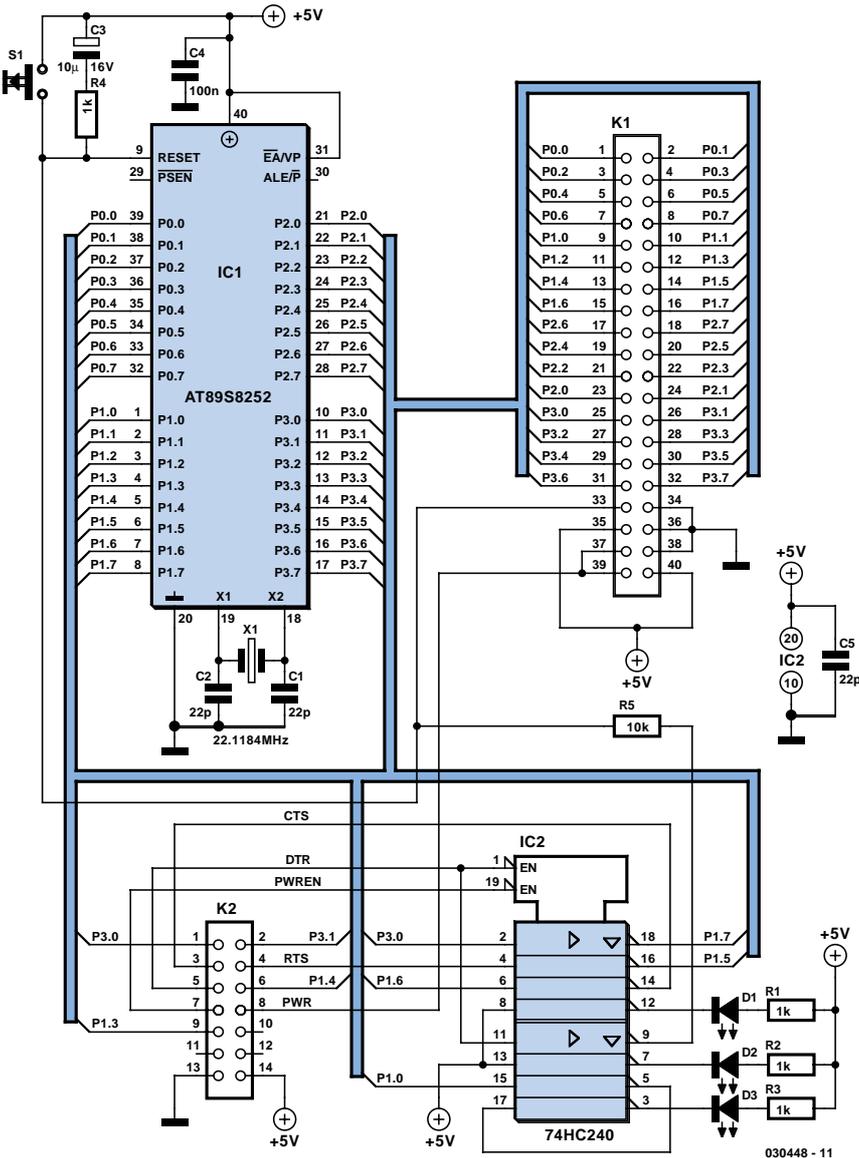


Figure 1. Le schéma de la carte à microcontrôleur (MCU). Son cœur est un micro Flash d'Atmel, un 89S8252.

correcte du port USB. Cette activation valide la moitié inférieure de IC2 ce qui se traduit par l'allumage de la LED PWR qui signale l'activation de la ligne PWR et sa capacité de fournir du cou-

rant. Le courant destiné à des périphériques externes est dérivé de la broche 8. Le circuit USB, nous y reviendrons, ne fournira aucun courant à cette broche tant que la configuration

n'est pas faite correctement. La broche 9 est un signal en provenance de l'interface USB passant au niveau bas en mode d'attente (*suspend*). L'utilisateur peut surveiller cette ligne. La broche 14 est une ligne d'alimentation fournie par le bus USB, qu'il soit ou non configuré. Nous le disions, elle ne peut fournir que 100 mA tant que le bus n'est pas configuré. Pour cette raison, des périphériques externes doivent utiliser les lignes PWR plutôt que Vcc.

Par un suivi des lignes mentionnées plus haut, il est possible de mettre IC2 en mode de veille (*power down*) lorsque le PC est mis en sommeil. La ligne *RT* peut servir pour réveiller le PC si tant est que les interfaces USB et celle du PC soient configurées correctement. Il faut, pour cela que IC1 soit lui-même auto-alimenté. La LED P1.0 est reliée, au travers d'une paire d'inverseurs, à la ligne d'E/S P1.0 de IC1; elle peut servir à n'importe quoi. Il est toujours pratique d'avoir de quoi effectuer des tests.

## Interface USB

L'interface USB représenté en **figure 2** repose sur une puce de FTDI, (Future Technology Devices Intl. Ltd.) et respecte le schéma de la fiche de caractéristiques. Le FT232BM est conçu pour remplacer quasiment l'interface RS-232 et une fois que l'on a installé les pilotes de périphériques il devient un port COM sur le PC. Il existe également des pilotes pour d'autres systèmes d'exploitation. Le circuit et les pilotes logiciels associés éliminent nombre de problèmes que sous-entend la réalisation d'une interface USB.

Quelques aspects de cette électronique méritent que l'on s'y intéresse. Le premier est le FETMOS à canal-N fournis-

# Jim Spence

Jim est né en 1953 et possède un titre en Construction (1979). Il est actuellement un manager de projet IT travaillant pour une compagnie IT internationale. Le dernier article de Jim ayant été publié l'a été, il y a 10 ans, dans ETI (Electronics Today International). Il s'agissait d'un autre projet de mono-carte à base de Z80 cette fois et tournant sous Forth. Avant de se consacrer au suivi de projets, Jim a donné des cours de CAD (Computer Aided Design). Ses domaines d'intérêt majeur sont l'électronique et les langages de programmation. Jim nous dit qu'il remonte au temps des tubes et qu'il a eu la chance de voir apparaître le transistor, les circuits intégrés et de voir l'industrie des ordinateurs personnels sortir du néant.

jim@byvac.com



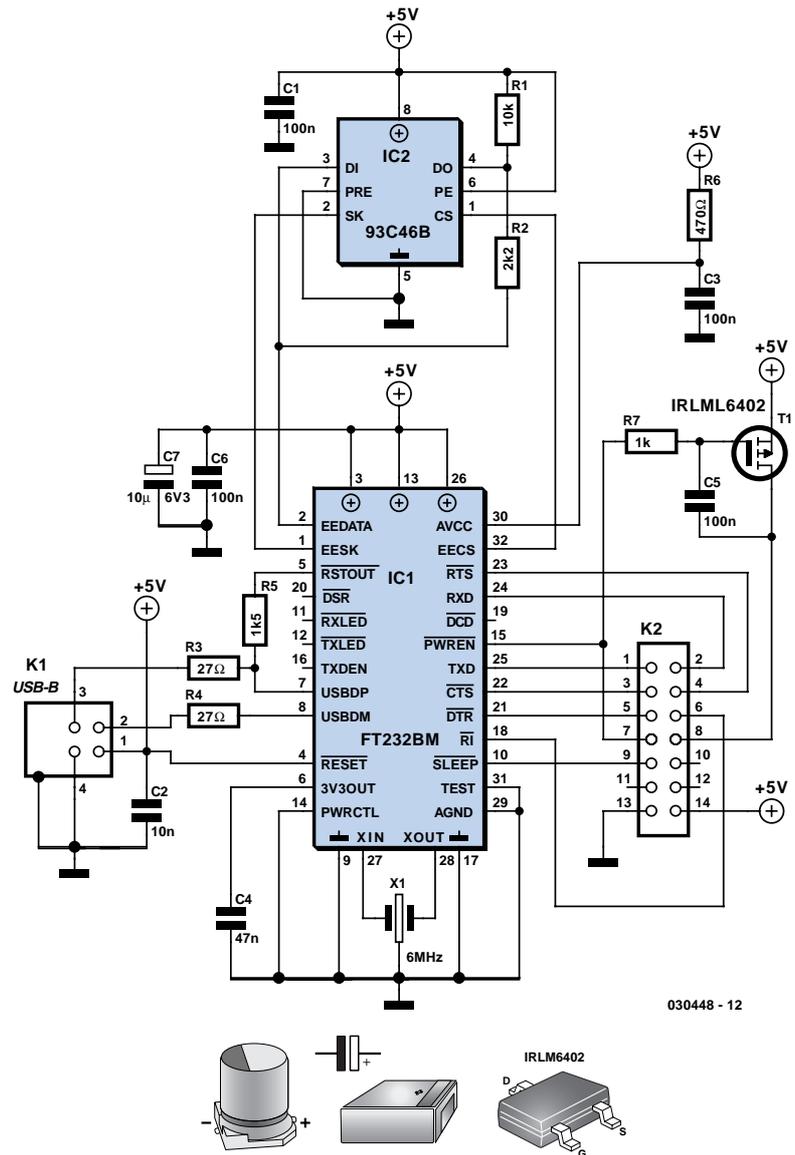
sant une tension à la broche 8 de K2 lorsque le signal PWREN# est au niveau bas. Il a été implanté ainsi de manière à pouvoir alimenter des périphériques externes depuis le bus USB une fois que le FT232BM est initialisé. Ce processus d'initialisation est connu sous la dénomination d'énumération. Le seul autre composant du circuit est IC2. Ce composant est optionnel et n'est nécessaire que si l'on veut utiliser les caractéristiques spéciales de la puce, USB 2.0 par exemple, ou si l'on met en oeuvre 2 puces sur le même système. L'électronique est parfaitement fonctionnelle en son absence.

Il est bon, avant de poursuivre, de s'intéresser quelque peu à l'interface USB. Cette interface est extrêmement pratique vu qu'elle constitue d'une part une interface rapide et que de l'autre elle fournit du courant.

Lors de la première connexion d'un périphérique celui-ci ne doit pas drainer plus de 100 mA jusqu'à ce qu'il soit énuméré. Une fois le bus énuméré, ce dernier peut fournir jusqu'à 500 mA. En mode de veille (*suspend*), ce courant tombe à 500  $\mu$ A.

Tous les périphériques USB possèdent un identificateur (ID) unique enregistré, le FT232BM disposant d'un ID par défaut enfoui. Une utilisation commerciale requiert l'enregistrement de l'ID. De même, si vous envisagez d'utiliser plus d'un périphérique sur le même système, chaque périphérique doit posséder son propre ID produit. Il est possible de paramétrer tout cela en implantant l'EEPROM, IC2.

Une fois l'énumération effectuée, ce qui est fait par le FT232BM, la ligne PWREN est mise au niveau bas. Cet état est transmis à la broche 7 de K2 de manière à ce qu'il soit possible, le cas échéant, de le suivre par le biais du microcontrôleur. Il active en outre T1 qui permet à son tour de drainer 500 mA du bus USB par le biais de la broche 8 de K2.



030448 - 12

Figure 2. L'électronique de l'interface USB.

Il existe 2 types de connecteurs : « A » et « B ». Le connecteur « A » est un fournisseur de courant; c'est le type que l'on trouve sur le PC. Le type « B » est un consommateur de courant; il

existe, pour autant que je le sache, sous 3 formes différentes. Ceci utilisé sur la platine est le plus courant, encore qu'il soit possible, avec difficultés, de mettre la main sur des versions

# Liste des composants

## Carte MCU

### Résistances :

R1-R4 = 1 kΩ  
R5 = 10 kΩ

### Condensateurs :

C1, C2 = 22 pF  
C3 = 10 μF/16 V radial  
C4, C5 = 100 nF

### Semi-conducteurs :

D1 à D3 = LED faible courant de la couleur que vous préférez  
IC1 = AT89S8252-24PC en boîtier DIP40 programmé (code **030448-41**)  
IC2 = 74HC240  
K1 = embase à 2 rangées de 20 contacts  
K2 = embase à 2 rangées de 7 contacts en équerre  
S1 = bouton-poussoir miniature unipolaire à contact travail  
X1 = quartz 22,1184 MHz

### Interface USB

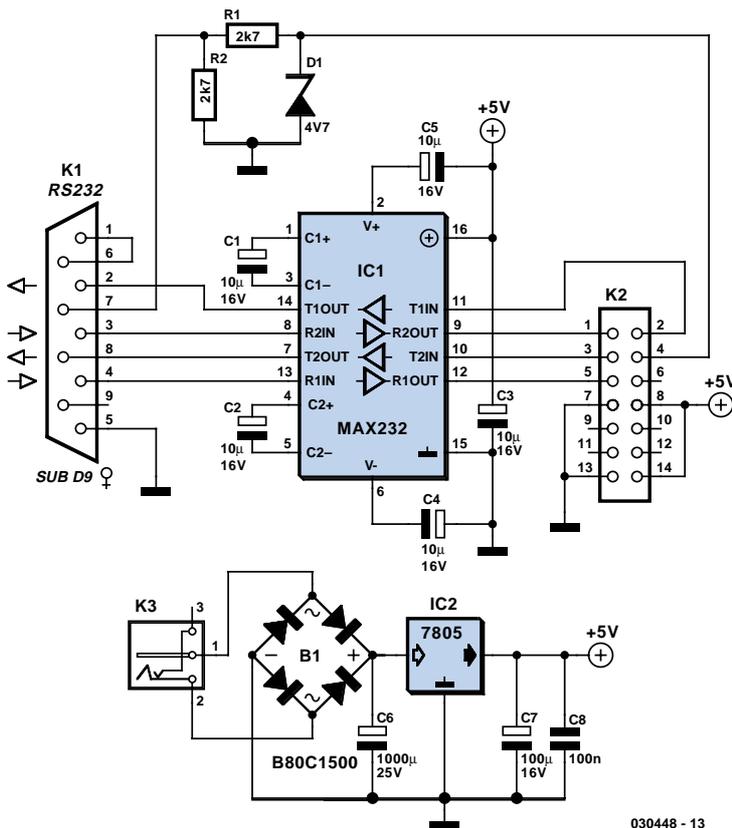
(Tous composants CMS forme 1206)

### Résistances :

R1 = 10 kΩ  
R2 = 2kΩ2  
R3, R4 = 27 Ω  
R5 = 1kΩ5  
R6 = 470 Ω  
R7 = 1 kΩ

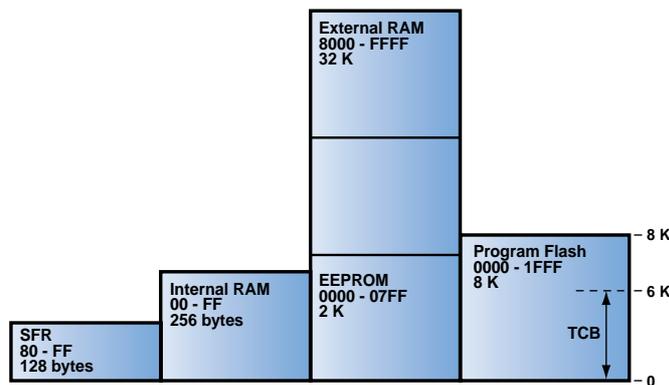
### Condensateurs :

C1, C3, C5, C6 = 100 nF  
C2 = 10 nF  
C4 = 47 nF



030448 - 13

Figure 3. Le schéma de l'interface RS-232.



030448 - 14

Figure 5. Architecture de mémoire Harvard applicable au contrôleur Flash 89S8252 de ce projet. TCB occupe 6 des 8 K disponibles dans la zone de Flash de programme.

miniatures telles qu'on les trouve sur les appareils photo numériques et les lecteurs MP3.

Le FT232BM requiert un pilote même sur un système tournant sous XP. On trouvera, sur le site de FTDI, tous les pilotes, logiciels et documentations nécessaires. Le pilote dont nous avons besoin ici est le Virtual COM Port (VCP). Il vous permettra d'utiliser le périphérique comme s'il s'agissait d'un port COM.

L'EEPROM, si installée, pourra être programmée en se servant des pilotes D2XX et l'un des nombreux programmes utilitaires proposés sur le site. À noter que **les pilotes ne coexistent pas**; il vous faudra désinstaller l'un pour utiliser l'autre.

### Bonne vieille RS-232

Ce paragraphe est destiné à ceux d'entre vous qui n'ont aucun atome crochu avec USB. La figure 3 donne le schéma de l'interface RS-232. Un port RS-232 n'étant pas en mesure de fournir suffisamment de courant, il faut prévoir, dans le schéma, une alimentation 5 V toute simple. Il se peut que la présence d'un redresseur en pont, B1, vous surprenne, mais il vaut son argent, vu qu'il vous permet d'utiliser une alimentation fournissant du courant alternatif mais vous permet également de ne pas vous faire de soucis quant à la polarité d'une alimentation donnant du courant continu – enfichez-là et ça marche.

La tension en provenance de l'alimentation est transmise à la broche 8 de K2 pour signaler à la carte principale la présence de courant. La broche 7 se trouve en permanence à la masse, donnant ainsi la même information à la carte principale. Le circuit constitué par R1, R2 et D1, limite la tension sur la ligne RTS et la rend utilisable pour l'alimentation de IC1 présent sur la carte principale.

C7 = 10  $\mu$ F/6,3 V CMS

Semi-conducteurs :

T1 = IRLML6402

IC1 = FT232BM (FTDI, [www.ftdichip.com](http://www.ftdichip.com))

IC2 = ne pas implanter (93C46B SO8)

Divers :

K1 = embase USB encartable type « B »

K2 = embase à 2 rangées de 7 contacts en équerre

X1 = résonateur céramique 6 MHz à 3 contacts

## Carte RS 232

Résistances :

R1, R2 = 2k $\Omega$ 7

Condensateurs :

C1-C5 = 10 mF/16 V radial

C6 = 1 000 mF/25 V radial

C7 = 100 mF/16 V radial

C8 = 100 nF

Semi-conducteurs :

B1=B80C1500, boîtier rond (80 V/1,5A)

D1 = diode zener 4V7/500 mW

IC1 = MAX232 (boîtier DIP16)

IC2 = 7805

Divers :

K3 = jack d'alimentation pour adaptateur secteur encartable en équerre

K1 = embase sub-D 9 points femelle en équerre encartable

K2 = embase à 2 rangées de 7 contacts en équerre

Platine, code **030448-1**

Logiciel du projet sur 2 disquettes, code **030448-11** ou à télécharger gratuitement

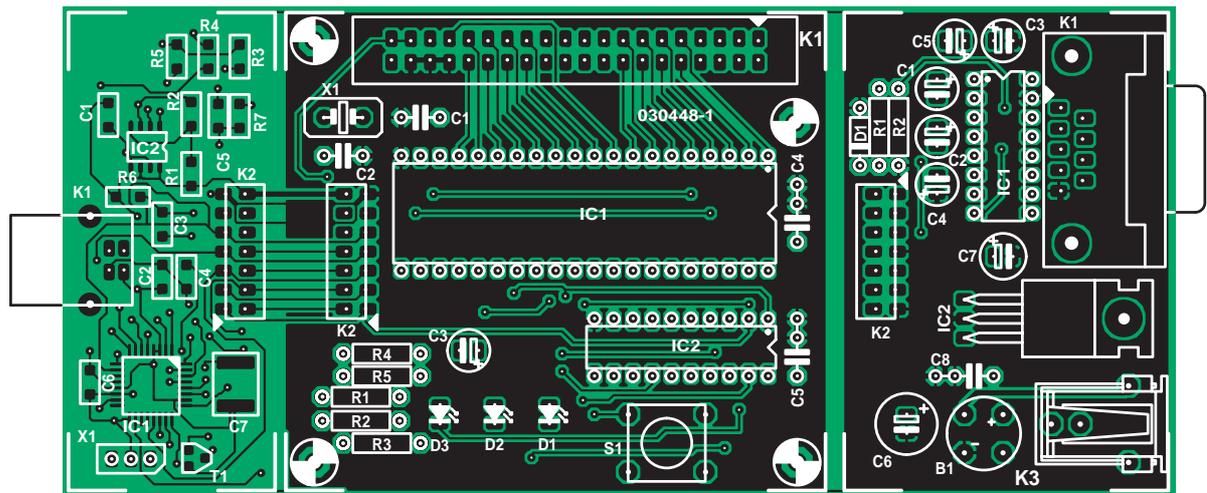


Figure 4. Sérigraphie des composants de la platine combinant MCU, RS-232 et USB (référence 030448-1). Si vous optez pour l'interface USB, vous pouvez soit laisser la section USB reliée à la partie MCU, soit les séparer et les relier à l'aide d'un morceau de câble en nappe.

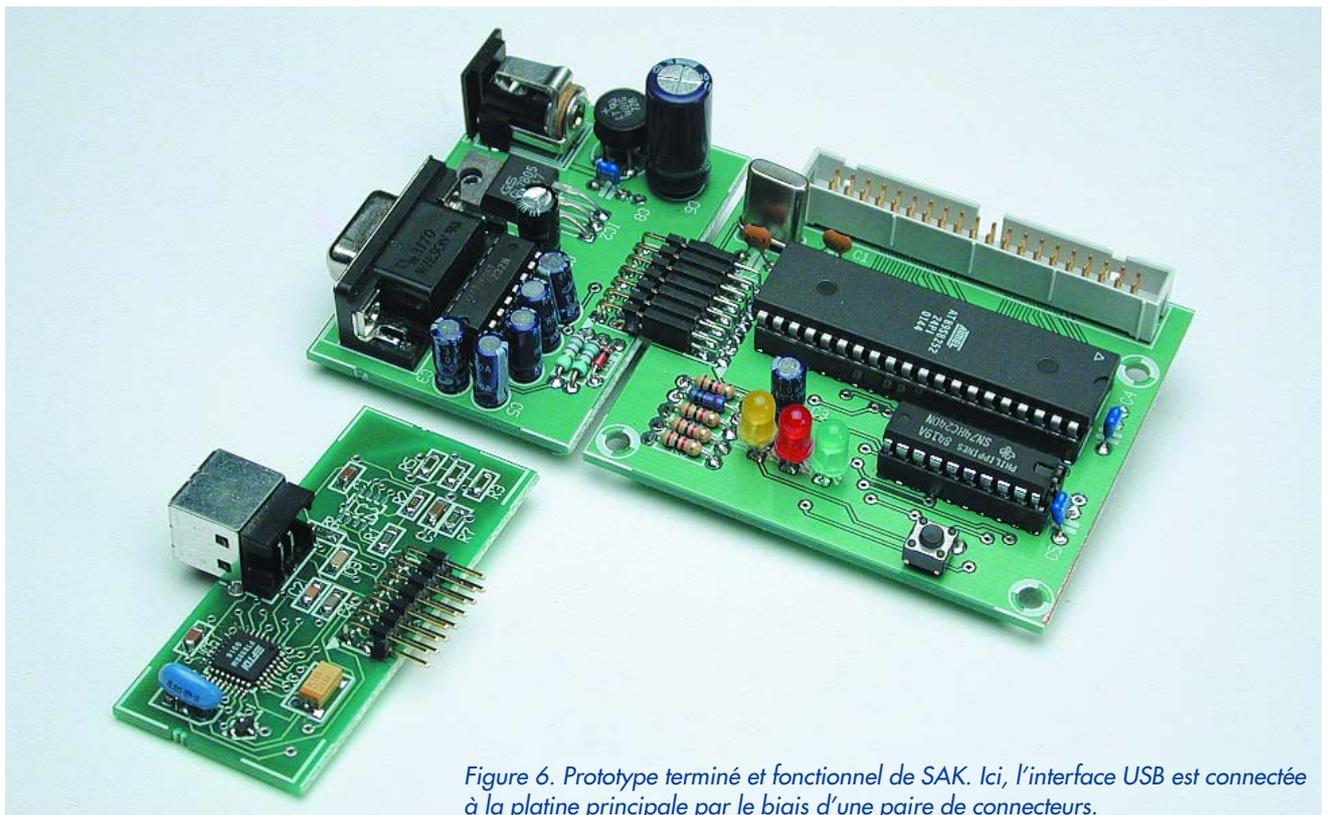


Figure 6. Prototype terminé et fonctionnel de SAK. Ici, l'interface USB est connectée à la platine principale par le biais d'une paire de connecteurs.

# Programme ByVac-Terminal

ByVac-Terminal possède quelques caractéristiques permettant de tirer le maximum de performances et de confort d'utilisation de SAK. En version 1.0, est disponible au téléchargement depuis notre site : [www.elektor.fr](http://www.elektor.fr). Par exemple, pour transférer un programme depuis votre éditeur de texte il vous suffit d'utiliser « Send File ». Cependant, si vous utilisez l'espace de mémoire en EEPROM, vous constaterez un certain délai lors de l'écriture de la ligne de programme. Il est possible, sur certains émulateurs de terminal, d'envoyer ligne après ligne et d'insérer un délai entre les lignes. Ceci fonctionne bien, mais peut se traduire par un transfert (download) sensiblement plus lent.

TCB connaît l'instruction « LOADB » au protocole très simple et fonctionnant de la manière suivante. L'instruction entrée, TCB attend une ligne de BASIC à envoyer; la ligne reçue, il la traite et envoie au terminal le code ASCII 6 (ACK) pour indiquer qu'il est prêt pour la ligne suivante. Ce protocole simple fonctionne parfaitement et le terminal ByVac l'intègre.

Autre avantage : il est capable d'utiliser la caractéristique de programmation en circuit de sorte que l'on peut écrire et transférer du code assembleur dans l'espace de code.

Nous avons vu comment allumer et éteindre la LED sur la carte en nous aidant d'un programme simple entré au clavier. Vous pouvez vérifier que tout fonctionne bien en entrant :

```
Pz1=254
```

```
Pz1=255
```

Cette double instruction a dû allumer et éteindre la LED. L'édition de programme en TCB consiste à réentrer la ligne; pour effacer une ligne, il suffit d'en entrer le numéro tout seul. Ceci peut devenir très ennuyeux. Une autre solution consiste à écrire le programme sous Notepad ou un autre éditeur de texte et de le transférer ensuite vers la carte.

Ouvrez votre éditeur de texte et entrez le programme suivant. Il est proche du programme d'introduction de TCB. À noter qu'au lieu d'assigner directement une valeur au port 1, les lignes 20 et 40 utilisent un opérateur logique (« and » et « or »). Le résultat est le même mais n'a aucun effet sur les autres lignes du même port. À noter le « end » sans numéro

Toutes les lignes de E/S au niveau de IC1 sont utilisées en vue de faciliter la programmation en circuit. Si l'on n'a pas besoin de cette fonction, seuls les signaux TxD et RxD sont nécessaires. Certaines des fonctionnalités de la carte USB ne sont pas disponibles pour l'interface RS-232, ce qui explique que les broches 6 et 9 de K2 ne soient pas connectées.

## Construction & test: MCU, RS-232...

Un coup d'oeil sur le dessin de circuit imprimé de la **figure 4** vous apprendra que les sections du microcontrôleur (MCU), USB et RS-232 sont proposées,

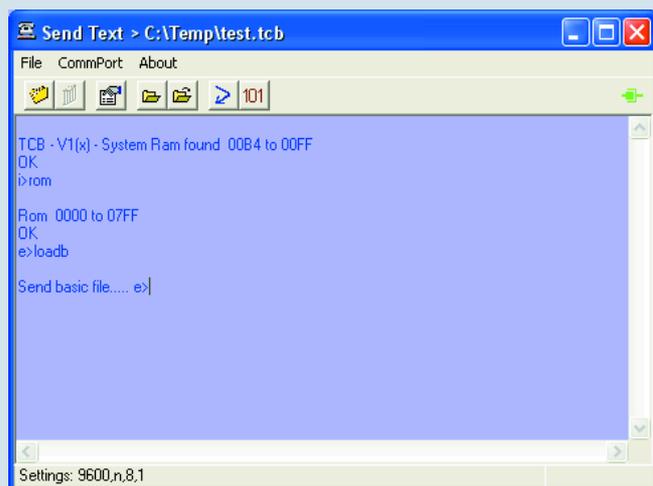
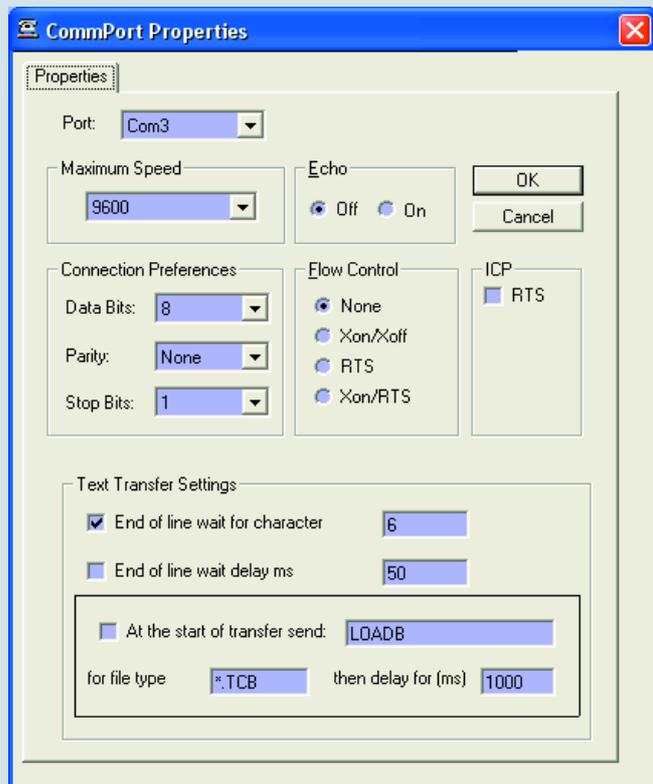
sous la dénomination **030448-1**, en une seule platine disponible auprès des adresses habituelles.

Il vous faudra, en fonction de l'interface que vous envisagez d'utiliser pour connecter SAK au PC, implanter la section USB ou RS-232. Rien n'interdit d'implanter tous les composants mais il n'est pas possible d'utiliser simultanément les 2 interfaces.

Les sections MCU et RS-232 n'appellent pas de remarque de réalisation spéciale; il suffit de respecter la liste des composants et la sérigraphie de l'implantation des composants. Il est recommandé de monter les circuits intégrés sur support. L'embase à 14 contacts pourra être découpée dans

un modèle plus grand si nécessaire. Le régulateur de tension IC2 ne requiert pas de radiateur.

Avant d'implanter les circuits intégrés dans leur support, connectez la carte à une alimentation et mettez-la ensuite sous tension. Vérifiez au multimètre que vous avez bien 5 V à la polarité correcte entre les broches 20 et 40 de IC1 et 10 et 20 de IC2. Si c'est bien le cas, déconnectez l'alimentation, insérez les circuits intégrés et rebranchez l'alimentation. SI vous disposez d'une sonde logique ou d'un oscilloscope, vérifiez le niveau de la broche 2 de K2 (broche 11 de IC1). Appuyez sur Reset et après 1 à 2 secondes vous devriez voir le signal « sign-on ». Il prendra la



de ligne en fin de programme. Ceci indique à TCB que le téléchargement est terminé. En pratique, si vous l'oubliez, cela n'empêchera pas le programme de tourner.

```
10 for j = 1 to 10
20 pz1 = pz1 and 254
30 gosub 500
40 pz1 = pz1 or 1
50 gosub 500
60 next j
70 end
```

```
500 for k = 1 to 20
510 next k
530 return
end
```

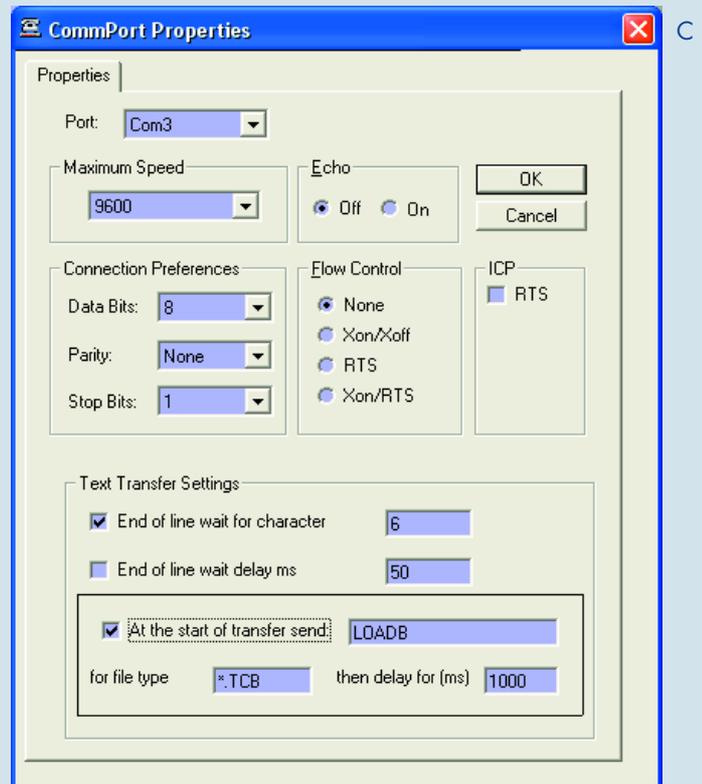
Sauvegardez le programme avec l'extension .tcb, LedFlash.tcb par exemple. Si vous utilisez Notepad, assurez-vous d'avoir bien choisi l'option « Tous types fichiers » avant de sauvegarder sinon vous vous trouverez en présence d'un fichier baptisé LedFlash.tcb.txt. Ceci n'arrive que lors de la première sauvegarde, mais il faut le savoir.

Comme nous le disions, TCB connaît un protocole de transfert de fichier très simple mais efficace que le ByVac-Terminal met à profit. Démarrez le terminal et utilisez le paramétrage de départ de la **figure A**. Notez que l'option « End of line wait for character » est cochée. Dans le terminal, entrez LOADB comme l'illustre la **figure B**.

TCB est prêt à recevoir un programme en BASIC en utilisant le protocole simple. Utilisez « File » et « Transmit text file » ou encore la 4<sup>ème</sup> icône à partir de la gauche. Choisissez votre fichier « LedFlash.tcb » et il sera chargé dans TCB.

Le programme se charge, entrez RUN pour voir les fruits de vos efforts. Vous pouvez, pour accélérer les choses, modifier le paramétrage de transfert de texte selon les indications de la **figure C**. Le terminal entrera ainsi de lui-même LOADB dès que vous choisissez un fichier doté de l'extension « .tcb ». Un passage au paramétrage RS-232 exige de déconnecter le SAK du PC et à l'y reconnecter ensuite.

Utilisez la 5<sup>ème</sup> à partir de la gauche pour recharger le même fichier. Le cycle de développement sera alors le suivant :



1. Éditer le programme
2. Sauvegarder
3. Utiliser l'icône de rechargement
4. RUN
5. Retour au point 1

Pour arrêter le programme en cours d'exécution, faites un Ctrl+C ou appuyez sur le bouton Reset. L'attrait d'un système interactif de ce genre est que tout se passe devant vos yeux, vous pouvez expérimenter sur la ligne d'instruction pour une **exécution immédiate**. En fin de développement vous avez votre produit fini et final.

Vous trouverez sur notre site, 2 programmes de dé, l'un simple, l'autre moins, pour tester la séquence ci-dessus.

forme d'une brève salve d'impulsions à 9 600 bauds.

Il va sans dire qu'il faut, pour que l'on ait l'activité décrite tout juste, que IC1 ait été doté de TCB (Tiny Control BASIC). Tout seul, la platine du contrôleur est limitée. Le seul risque d'erreur se situe au niveau des soudures et de l'implantation des composants (tous les composants se trouvent-ils à leur place et sont-ils orientés correctement ?).

### ...et USB

Comme le montre un examen du dessin des pistes, des pistes relient l'interface USB à la section MCU. Si vous voulez placer l'interface USB à une certaine distance de la carte MCU, il faut

separer les parties de la platine et installer un câble de liaison entre les 2 embases concernées.

Venons-en aux points délicats. La puce FT232BM n'existe qu'en version CMS qui de plus est en version « miniature ». Il vous faut avoir le coeur bien accroché pour cette réalisation, mais il suffit en fait d'un fer à souder, d'un peu de soudeuse et de pâte à souder. Il ne s'agit pas d'un travail de débutant, mais « à coeur vaillant rien d'impossible »... Malheureusement, si la pâte à souder coûte cher elle simplifie notablement le processus. On peut aussi essayer en remplaçant la pâte à souder par un excès de soudeuse que l'on enlèvera à l'aide de tresse à dessouder.

La tresse à dessouder permet fort bien d'enlever tout excès de soudeuse de sorte; partant ce n'est pas une catastrophe s'il y a de la soudeuse un peu partout. Le plus important est de bien placer et orienter IC1. Il faut qu'il soit parfaitement aligné avec toutes les pistes et qu'il le reste lors de la première soudeuse. Attention s'il venait à glisser ! Prenez donc le temps d'aligner IC parfaitement et soudeuse ensuite quelques-unes de ses broches (une à la fois si vous y arrivez), encore que la pointe fine du fer à souder de l'auteur recouvrait de 2 à 3 contacts simultanément. Si tout va bien, soudeuse le reste des contacts, le circuit étant quasiment impossible à dessouder une fois en place. Enlevez toute la soudeuse excédentaire à la

**Tableau 1. Interface de programmation et USB**

Broche J1	Nom	Fonction
1	RXD (SCK)	Partie de l'interface de programmation en circuit sérielle de IC1. Est la ligne d'horloge
2	TXD	Sortie sérielle de IC1
3	CTS (MISO)	Sortie de l'interface de programmation sérielle de IC1
4	RTS (MOSI)	Entrée vers l'interface de programmation sérielle de IC1
5	DTR	Au niveau bas, active le mode de programmation, niveau bas suivi d'un niveau haut réinitialise IC1
6	RI	À mettre au niveau bas pour réveiller le PC (requiert un paramétrage spécial)
7	PWREN	Sortie de l'interface USB; si au niveau bas indique que l'on dispose de 500 mA sur la ligne PWR
8	PWR	5V @ 500mA
9	SLEEP	Passe au niveau bas pour indiquer que le PC connecté s'est mis en sommeil
10-12		non utilisées
13	GND	Masse (Ground)
14	VCC	100 mA, 5V

tresse à dessouder et vérifiez l'absence de court-circuit entre 2 broches.

L'embase K2 pourra prendre la forme d'un morceau de barrette autosécable à 2 rangées de 7 contacts.

Avant de connecter le montage au PC, téléchargez les pilotes de la puce FTDI et décompactez-les dans un répertoire adéquat. Si tout se passe bien, lors de la première connexion du montage, le PC vous demandera d'installer le pilote correspondant.

Une fois la réalisation terminée, prenez le temps de vérifier l'absence de court-circuit au multimètre le cas échéant. Connectez le montage à un port USB du PC ou mieux encore à un hub pour éviter tout risque d'endommagement du PC. Bien que les spécifications USB impliquent une protection contre les courts-circuits, on ne sait jamais, un nouveau hub coûte moins cher qu'un nouveau PC. Sachez, si cela peut vous rassurer, que l'un de nos premiers prototypes, insuffisamment vérifié avait entraîné, lors de sa connexion au PC, la fin du fonctionnement de tous les périphériques USB, la souris y comprise. Il a suffi d'un redémarrage du PC pour que tout rentre dans l'ordre – une expérience dont nous nous serions aisément passé.

Si tout va bien, le PC détecte le nouveau périphérique et demande où se trouvent ses pilotes. Installez-les comme vous le feriez pour n'importe quel autre pilote. Si vous avez des doutes, jetez un coup d'oeil sur le site Internet de FTDI pour de plus amples informations, des instructions d'installation étant fournies avec les pilotes

pour Windows et autres systèmes d'exploitation.

Vous devriez avoir maintenant un nouveau port COM. Pour connaître le numéro de port lui ayant été attribué, il vous faudra entrer au coeur de votre système d'exploitation. Sous Windows XP, il vous faut entrer dans le Panneau de Configuration, Système, Matériel, Gestionnaire de périphériques et cliquer sur le + de Ports. Vous devriez y trouver un port qui n'existait pas auparavant. Si ce n'est pas le cas, réinstallez le pilote et assurez-vous de l'absence de message d'erreur.

### Test de l'ensemble

Interconnectez les 2 platines, en optant soit pour la section RS-232 soit pour la section USB. Démarrez l'utilitaire de terminal gratuit ByVac (cf. l'encadré ByVac-Terminal).

Il vous faudra, en cas d'utilisation de la carte RS-232, un câble non croisé (1:1) broche 1 à broche 1, 2 à 2 etc. Vérifiez-le au multimètre. Il vous faudra également activer RTS dans le boîtier ICP. Appuyez sur le bouton Reset (S1), attendez quelques secondes et vous devriez voir apparaître un message de bienvenue. Si ce n'est pas le cas, vérifiez que IC1 envoie bien ce message. Si vous trouvez un signal et que le message n'apparaît pas, vérifiez les paramètres, le taux de transmission, etc. les embases, câbles et câblage.

### Et voici TCB

Au coeur de ce projet règne le logiciel TCB (Tiny Control BASIC) qui, notons-le au passage, tourne sur tout système

89C8252 avec ou sans RAM externe. En cas de présence de RAM externe il la détectera automatiquement; comme on le sait, le présent projet ne comporte pas ce type de RAM. Il existe 3 façons de transférer le logiciel TCB dans IC1 :

1. Utiliser l'interface de programmation;
2. Utiliser un composant préprogrammé (**030448-41**) disponible auprès des adresses habituelles;
3. Utiliser un programmeur pour 89C8252.

Bien que possible, nous ne recommandons pas la solution 1, l'interface et le logiciel de programmation étant conçus pour de courts programmes (< 100 lignes), de plus cela signifierait qu'il ne vous serait pas possible de tester le fonctionnement de l'électronique et tant que vous n'avez pas pu faire cela, comment savez-vous que le montage a été réalisé correctement ? Cela n'en est pas moins possible et vous prendra de l'ordre de 25 minutes (cf. l'encadré « Code assembleur »). Nous recommandons les solutions 2 et 3 si c'est la première fois que vous réalisez ce montage.

Il a été fait mention, dans différents articles, de ce processeur et de son architecture, l'aspect mémoire présentant une importance capitale, nous allons l'évoquer succinctement ici.

### La mémoire

L'espace de mémoire du 8052 repose sur une architecture qui partage des zones de mémoire parallèles. Par une

**Tableau 2. Spécifications principales du Tiny Control BASIC**

<b>Nombres:</b>	entiers signés à 16 bits allant de -32767 à 32767
<b>Variables:</b>	à une lettre de A à L (12)
<b>Arithmétique:</b>	+, -, *, /, et MOD
<b>Logique:</b>	NOT, AND, OR, XOR
<b>Comparaisons:</b>	>, <, =, <>, >=, <=
<b>Instructions:</b>	RUN, LIST, NEW, DUMP, RND, ABS, IF, THEN, GOTO, FOR, TO, NEXT, REM, CALL, RETURN, GOSUB, ROM, RAM, IRAM, LOADH, LOADB, DECIMAL, HEX, LET, PRINT, INPUT, BAUD
<b>Registres à fonction spéciale:</b>	PZO, PZ1, PZ2, PZ3, TCON, TMOD, TL0, TL1, TH0, TH1, SCON, SBU, IE, IP, T2CON, WMCON, SPCR, SPSR, SPDR, PCON
<b>Interruptions:</b>	ONEX0, ONEX1, ONT0, ONT1, ONT2, ONSP, EI, DI

combinaison logique des signaux OE et PSEN pour accéder à la RAM, cela signifie qu'il est possible d'utiliser de la RAM externe en mémoire de programme. L'architecture du 8051 date et à l'époque on pensait que séparer la mémoire de données de la mémoire de programme était une bonne idée. C'était l'architecture Harvard et, en théorie du moins, cela signifie qu'il est possible, au cours d'un même cycle-machine, de saisir une instruction de code en même temps que des données. La **figure 5** nous donne la cartographie de mémoire; elle peut prêter à confusion si l'on ne connaît que les agencements de mémoire conventionnels. L'espace EEPROM est réservé aux seules données, les programmes en assembleur ne pouvant pas y tourner, des programmes en langage de haut niveau le pouvant eux (cf. plus loin). On dispose de 256 octets de RAM interne qui partage la moitié supérieure de cet espace avec des registres à fonction spéciale (SFR). La technique d'accès aux différents espaces est fonction du set d'instructions. Entrer dans le détail nous amènerait trop loin; à titre d'exemple, l'instruction MOVC est utilisée pour accéder à la mémoire de code, MOVX l'étant pour l'accès à la mémoire externe. On utilise d'autres techniques d'adressage direct ou indirect pour accéder à l'espace de RAM interne.

La mémoire Flash embarquée contient TCB en version 1, programme qui n'occupe que les 6 premiers Koctets. Si cela vous suffit pour vos applications, oubliez toutes ces histoires de modes d'adressage de la mémoire, TCB s'en chargera.

## Premiers pas de programmation en TCB

Le BASIC entier est une version modifiée de Tiny BASIC appelée Tiny Control BASIC (TCB) conçue spécialement pour permettre aux utilisateurs de tirer le maximum d'un microcontrôleur le plus rapidement possible sans avoir de courbe d'apprentissage raide ou avoir à installer de logiciel spécial. L'architecture 8052 connaît 3 espaces de mémoire supportés par TCB. La RAM interne, l'EEPROM et la RAM externe. On accède à ces domaines par les mots-clé IRAM, ROM et RAM respectivement. Entrez IRAM et vous verrez apparaître le message :

```
i>iram
```

```
Internal Ram 00B4 to 00FF
```

```
OK
```

```
i>
```

L'invite « i> » indique que nous nous trouvons dans le domaine de la RAM interne. Comme on le voit, il ne s'agit que de quelques octets, mais ils suffisent pour 2 à 3 lignes de programme. TCB vous indique lorsqu'il ne reste plus d'espace mémoire. Pour accéder à l'EEPROM, entrez ROM :

```
i>rom
```

```
Rom 0000 to 07FF
```

```
OK
```

```
e>
```

Nous nous trouvons maintenant dans l'espace de ROM et pouvons constater

que nous avons 2 K de mémoire, de 0000 à 07FF. Il suffit, à n'importe quel moment, d'entrer « DUMP » pour afficher le contenu de la mémoire. L'intérêt d'utiliser cet espace est qu'il conservera le programme même après coupure de l'alimentation. Son inconvénient est que l'écriture s'y fait plus lentement qu'en RAM et que le nombre d'écritures est limité, on parle de quelque 100 000 fois.

Ce projet ne possède pas de RAM externe, de sorte l'entrée du mot-clé RAM se traduira par une erreur. Tous les programmes TCB seront écrits dans l'EEPROM.

TCB est capable d'exécuter un programme au démarrage mais ce programme doit démarrer avec la ligne 10. En l'absence de ligne 10, tout programme présent sera en fait effacé. Le processus de démarrage est le suivant :

- Tester entrée utilisateur (barre d'espace) – attendre de l'ordre de 1 à 2 secondes

- En l'absence d'entrée utilisateur, tester l'espace en (EEP)ROM pour y trouver un programme commençant à la ligne 10 et l'exécuter le cas échéant. Sinon, tester l'espace en RAM pour y trouver un programme commençant à la ligne 10 et l'exécuter.

- En l'absence de ligne 10 et s'il existe de la RAM, on a effacement (instruction NEW) du programme en RAM et le pointeur se trouve en espace RAM. En l'absence de RAM, il passe dans l'espace de RAM interne (IRAM). Il n'y a pas effacement de la mémoire si l'on appuie sur la barre

# Code assembleur

À un moment donné on peut avoir besoin de plus de contrôle ou de plus de vitesse que ce que permet un langage de haut niveau. La seule alternative est l'assembleur, ce qui revient en fait à programmer directement le processeur.

Il vous faudra un assembleur; il s'agit d'un programme qui traduit les mnémoniques de 2 à 4 lettres en nombres que le processeur peut comprendre. Il en existe plusieurs, gratuits, pour ce processeur. Nous avons utilisé, pour les exemples, « ASM51 ». Il existe plusieurs sources où le télécharger.

Il vous faudrait, pour l'exécution de code assembleur utilisant la fonction LOADH, disposer de RAM, mais ce projet n'en comporte pas. Nous disposons cependant d'espace de code libre en EEPROM débutant à 1700<sub>HEX</sub>, espace accessible par le biais de la fonction de programmation en circuit.

## Prêt, partez !

Reste toujours un point délicat. Il est difficile de programmer en assembleur si l'on ne sait pas un minimum de choses quant au processeur et quelques notions du set d'instructions du 8051/8052. L'exemple suivant vous mettra sur la voie.

## Exemple de programme

Nous restons en terrain connu en faisant clignoter la LED présente sur la carte. Nous allons pour cela, vous le savez, mettre P1.0 à « 0 » puis à « 1 ».

```
; Exemple de clignotement de LED
; Utilise Asm51
; La LED est connectée à P1.0
;
$MOD8252
    seg
    org          1700h
flash: ; mise au niveau haut et bas de P1.0
    setb p1.0
    call delay
    clr      p1.0
    call delay
    jmp     flash
;
delay: ; delay
    mov     r0, 0f0h
del2:  mov     r1, 0
del1:  nop
    djnz  r1,del1
    djnz  r0,del2
    ret
    end
```

Le tableau expliquera mieux le fonctionnement du programme.

## Assemblage

Utilisez Notepad ou un autre éditeur de texte pour créer le programme ci-dessus que vous sauvegardez avec l'extension « .asm », FlashLed.asm par exemple. Une fois le programme écrit, il faut le convertir en un format que le processeur peut comprendre. Pour cela, il vous faudra utiliser l'invite de commande « > » si vous travaillez sous Windows, en DOS vous y êtes déjà.

Choisissez un répertoire où mettre les fichiers, FlashLed.asm, Asm51 et MOD8252 devraient se trouver dans le même

<b>&amp;MOD8252</b>	Tout d'abord ceci intègre tous les noms spéciaux associés avec ce processeur de sorte qu'il nous faut cela avec notre 89S8252.
<b>Cseg</b>	Dit à l'assembleur que ce qui suit est du code, il y aura un « dseg » correspondant pour les données.
<b>Org</b>	Est l'adresse d'origine du programme. TCB se termine approximativement à 1650 <sub>HEX</sub> , le début du programme ne peut être plus bas sous peine d'endommager TCB.
<b>setb p1.0</b>	Mise au niveau haut du bit 0 du port 1, tout comme le ferait pz1=1.
<b>Call</b>	Équivalent de gosub
<b>delay:</b>	Une sous-routine; à noter le « : », exigé par asm51. Il faut, en raison de la vitesse pure du processeur, 2 temporisations. Cela se traduit en fait par l'exécution de 61 000 instructions avant retour, ce qui se traduit par une temporisation de 0,25 s environ.
<b>R0, #0f0h</b>	R0 est un registre à usage général; il n'existe 8, R0 à R7. Il existe en fait 4 bancs, un seul de ces bancs pouvant être utilisé à la fois. Cette instruction place la valeur de F0 (hex) dans le registre R0. Le « # » est connu comme modificateur « immédiat ». Si on avait R0,0f0h, le contenu de l'emplacement de mémoire sis à l'adresse F0 serait mis dans R0 (indirect).
<b>Djnz</b>	Signifie Decrement and jump if not zero; ainsi dans la boucle imbriquée, R1 est décrémenté et qu'il n'est pas égal à zéro, le programme sautera au label « del1: ». À noter que le registre est décrémenté avant le test à zéro; ainsi, un registre démarrant avec une valeur 0, sera décrémenté à FF avant d'être testé.

répertoire (à moins que vous ne sachiez comment modifier les cheminements). À l'invite entrez « asm51 FlashLed » (sans guillemets). Une action sur Enter devrait vous donner quelque chose ressemblant à l'écran de la **figure A**. Notez qu'il vous faudra obtenir Asm51 ou un autre assembleur 8051 vu qu'il ne fait pas partie du logiciel de ce projet.

Le processus génère 2 fichiers de même nom d'extension différente : FlasLed.hex et FlasLed.lst. Le fichier .lst vous indiquera d'éventuelles erreurs, le fichier .hex étant un format spécial développé par Intel de manière à permettre le chargement de code dans les processeurs. Pour info, ce format est constitué de lignes de caractères précédées par un double point.

```
:10170000D29012170CC29012170C80F478F0790066
```

Les 2 premiers chiffres, « 10 », donnent la longueur du code en hexadécimal, les 4 chiffres suivants, « 1700 » l'adresse en mémoire, suivis par un type d'enregistrement presque toujours « 00 ». Le reste de la ligne est le code proprement dit, exception faite des 2 derniers caractères, qui constituent la somme de vérification (66) indiquant que la ligne a été reçue correctement.

## Chargement

L'étape suivante consiste à charger le programme dans l'espace de code. C'est là qu'entre en scène une fonction spéciale. Cliquez sur l'icône « 101 » et l'arrière-plan vire au noir. Nous nous trouvons maintenant en mode de programmation Flash.

Appuyez sur <P> et le texte deviendra jaune, ce qui signifie que nous allons utiliser de l'espace programme ou code (la zone de mémoire Flash). En entrant « v » « 88 » <enter> puis « 99 » <enter> vous pourrez voir le contenu de la mémoire

de code de l'adresse 0088 à 0099. C'est l'endroit où se trouve le message de bienvenue de TCB. Tous les caractères sont au format hexadécimal, cf. **figure B**.

Pour programmer notre programme dans cet espace entrez « h », ce qui produit l'ouverture d'une fenêtre de dialogue qui permettra de choisir le fichier FlashLed.hex. Il sera chargé en mémoire et vous verrez apparaître quelque chose ressemblant à la **figure C**. La programmation débutera automatiquement (il faut un certain temps).

### Exécution du programme

Commencez tout d'abord par retourner au mode terminal en appuyant sur « x » ou sur l'icône « 101 ». Pour exécuter ce programme, TCB connaît CALL. La syntaxe, CALL n, où n est l'adresse à laquelle on veut sauter. Nous avons choisi 1700h comme adresse de début. Par défaut, TCB travaille en décimal; pour passer en hexadécimal entrez « HEX ». Attention, à partir de maintenant, tout est en hexadécimal jusqu'au prochain reset ou que vous entriez DECIMAL. Ceci vaut pour les programmes ordinaires en BASIC, voir des numéros de ligne en hexadécimal est très étrange.

Entrez ce qui suit pour exécuter le programme :

```
HEX
CALL 1700
```

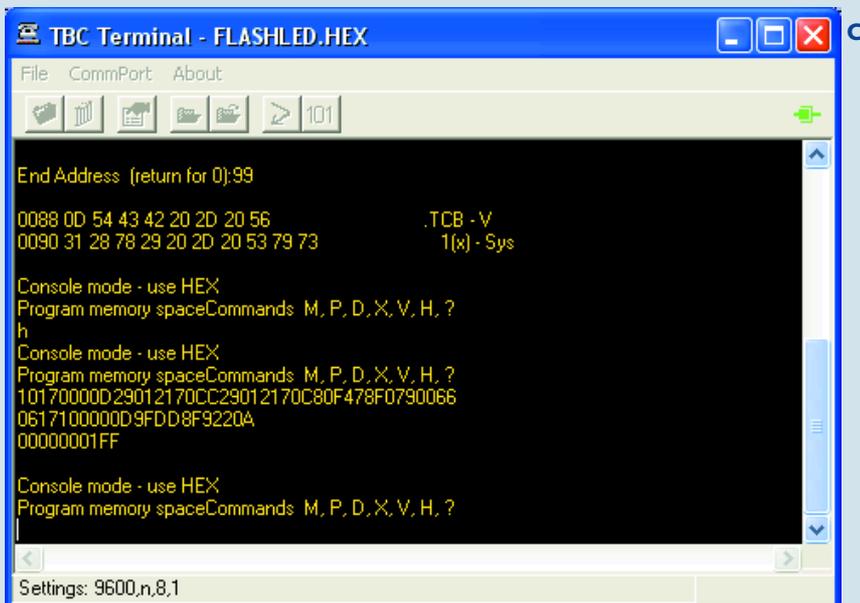
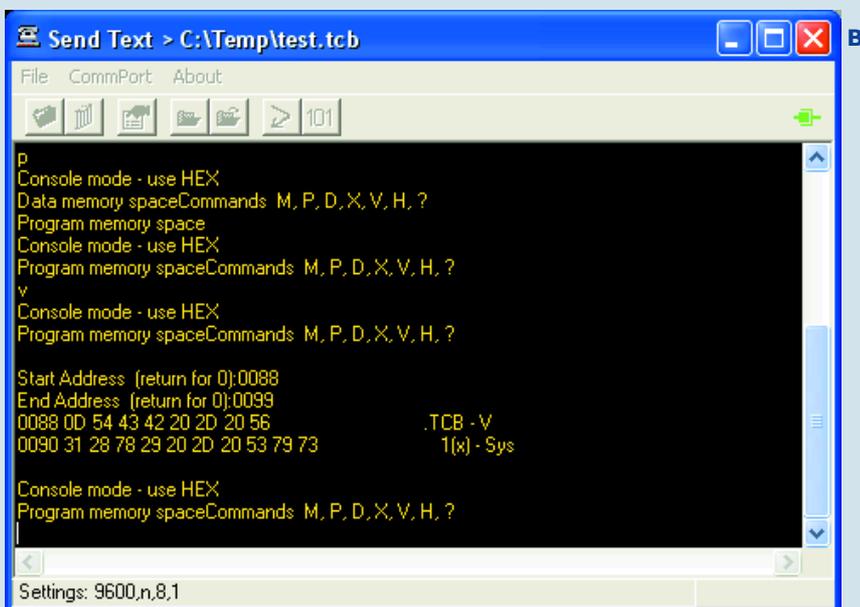
Si vous ne voulez pas travailler en HEX, CALL 5888 donne exactement le même résultat, 5888 étant l'équivalent décimal de 1700h.

La LED devrait alors clignoter signalant que l'exécution du programme assembleur est en cours. Comme il s'agit d'un programme rudimentaire, la seule façon de l'arrêter est d'appuyer sur Reset. Ceci pourra se faire en mode terminal en utilisant l'icône à côté de « 10 » ou en appuyant sur le bouton-poussoir de la platine principale. Une fois qu'il se trouve en mémoire Flash, le programme y restera même après coupure de la tension d'alimentation. En raison de la lenteur de la programmation sérielle, ce processus ne convient pas aux programmes de forte taille. Bien que cela soit possible, attendez-vous à ce que TCB ait besoin de quelque 25 minutes pour le fichier basic.hex. Tout n'est pas possible avec TCB seul, c'est ce qui donne à ce projet toute son universalité.

Vous pouvez fort bien utiliser un mélange de TCB et d'assembleur. Vous pouvez, par exemple, assembler le code à l'adresse 1700 et utiliser un programme BASIC simple de 2 lignes :

```
10 HEX
20 CALL 1700
```

Ceci produira le clignotement de la LED dès l'application de la tension d'alimentation à la carte. D'accord, une LED clignotante connaît ses limitations, mais cela vous a permis de vous faire une idée...



# Téléchargements gratuits

- Byvac-Terminal pour PC (fichier d'installation avec fichiers OCX d'environnement), TCB (fichier file), référence : 030448-11.zip
- 2 programmes de dé simples (fichier Word), référence : 030448-12.zip.
- Guide Quick Start Guide pour SAK (fichier Word), référence : 030448-13.zip.
- Manuel du Tiny Control BASIC (fichier .pdf), référence : 030448-14.zip.
- Dessin des pistes au format .pdf, référence : 030448-1.zip.

Allez sur [www.elektor.fr/dl/dl.htm](http://www.elektor.fr/dl/dl.htm) et choisissez le mois de publication.

d'espace 1 à 2 secondes après la mise sous tension.

Comme ce projet ne dispose pas de RAM externe, on se retrouvera par défaut dans l'espace de RAM interne « i> » à moins qu'il n'y ait une ligne 10 dans l'espace de ROM, auquel cas on aura exécution du programme se trouvant à cet endroit.

En cas d'action sur la touche d'espace dans les 2 secondes environ qui suivent une initialisation (reset) ou une mise sous tension, TCB détectera le taux de transmission et se présentera dans l'espace de RAM interne.

En fait, TCB n'efface pas toute la RAM, il se contente de placer le marqueur de fin de programme (FF) dans le premier octet. Vous pouvez le vérifier en entrant une instruction DUMP après une instruction NEW.

Si nous revenons au circuit du processeur, la LED D3 est connectée à la ligne de port P1.0 au travers de 2 tampons intégrés dans IC2. Par défaut, cette LED est éteinte. Ceci est dû au fait qu'à la mise sous tension toutes les lignes sont forcées au niveau haut.

Il suffit, pour faire s'allumer la LED, d'entrer : PZ1=254

```
i>pz1=254
OK
i>
```

PZ1 représente la variable port 1, toute valeur que vous lui donnez aura une influence sur le port 1. En mettant le port 1 à 254, ce qui se traduit en 1111 1110 en binaire, vous mettez à 0 la ligne 0 du port 1. Par convention, cette ligne est référencée sous le nom de p1.0.

Essayez :

```
ROM
10 FOR J = 1 TO 10
20 PZ1=254
30 GOSUB 200
40 PZ1=255
50 GOSUB 200
60 NEXT J
70 END
```

```
200 FOR K = 1 TO 50
210 NEXT K
220 RETURN
```

Le programme ci-dessus devrait se traduire par une vingtaine de clignotements de la LED.

Le **tableau 2** donne une description succincte du langage.

## En guise de conclusion

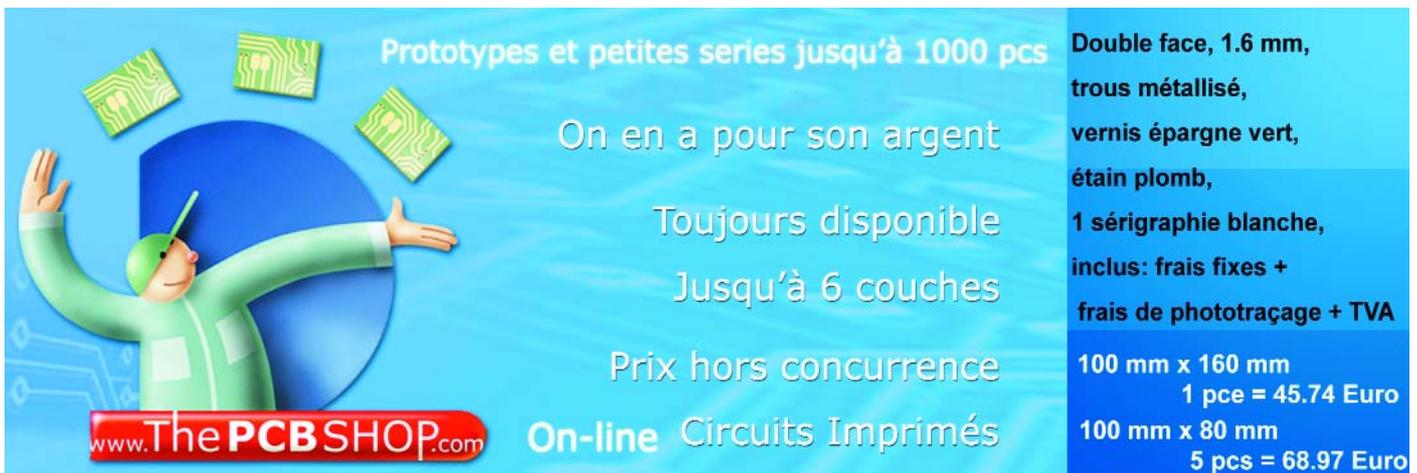
La quantité impressionnante de documentation fournie par Jim pour illustrer ce projet n'aurait pas de mal à remplir la moitié de ce magazine, ce qui explique que nous en ayons transféré une partie vers notre site d'où vous pourrez la télécharger gratuitement. Les documents (en anglais) en question incluent le Quick Start Guide (illustré) - Swiss Army Knife, le manuel Tiny Control BASIC et Simple Dice; il est temps de vous mettre à télécharger...

(030448-1)

### Adresses Internet

Pilotes USB FT232BM : [www.ftdichip.com](http://www.ftdichip.com)

Publicité



Prototypes et petites séries jusqu'à 1000 pcs

On en a pour son argent

Toujours disponible

Jusqu'à 6 couches

Prix hors concurrence

On-line Circuits Imprimés

www.ThePCBSHOP.com

Double face, 1.6 mm, trous métallisés, vernis épargne vert, étain plomb, 1 sérigraphie blanche, inclus: frais fixes + frais de phototraçage + TVA

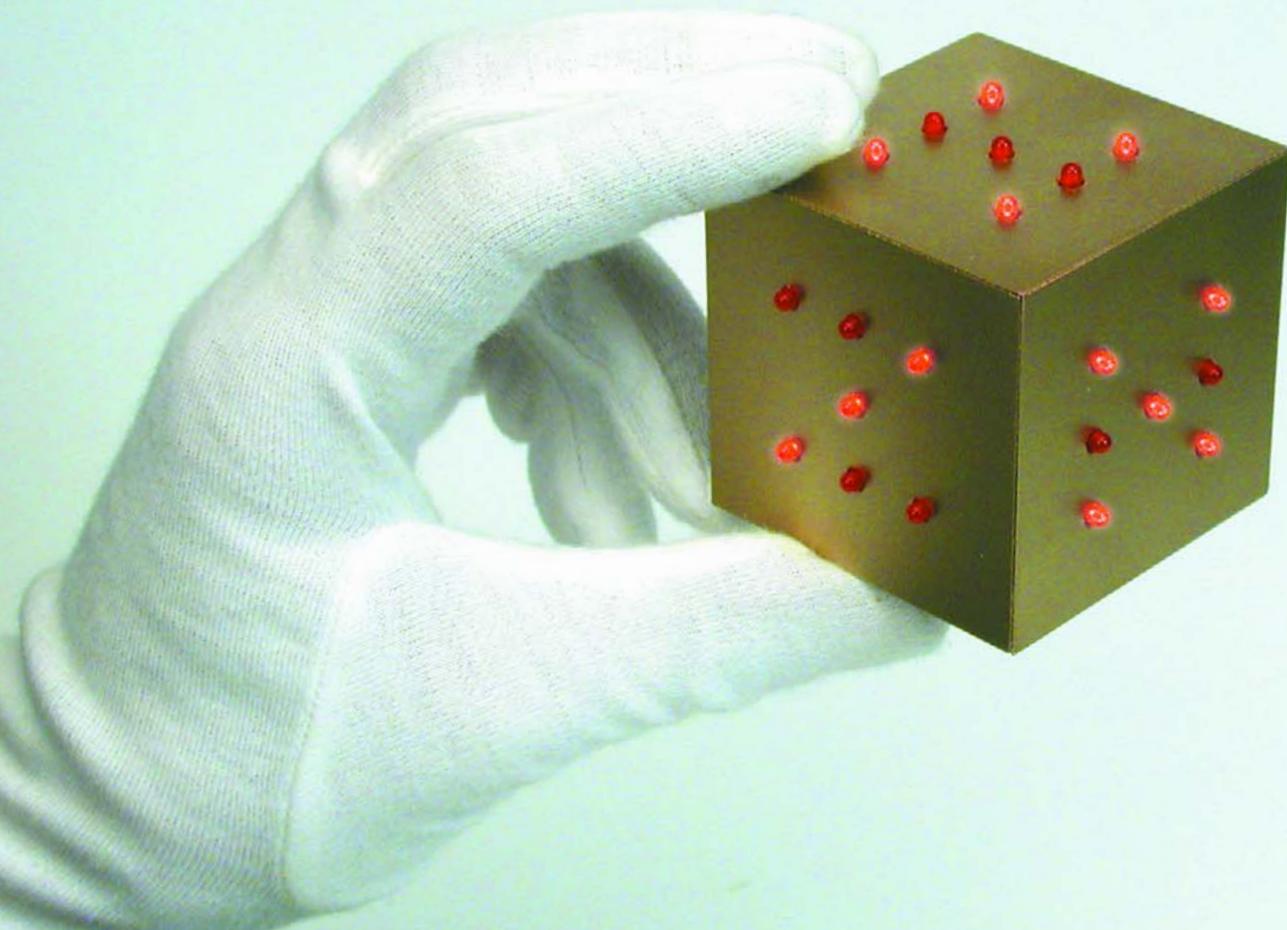
100 mm x 160 mm  
1 pce = 45.74 Euro

100 mm x 80 mm  
5 pcs = 68.97 Euro

# Dé tridimens

Paul Goossens

Nous avons eu l'occasion, de décrire des dés électroniques dans nos colonnes. De tailles et de modèles différents ils avaient un point commun, ils étaient en 2D, tout plats. Nous avons changé d'avis !



La quasi-totalité des dés électroniques sont réalisés sur une platine plane. Pour que le résultat soit « ressemblant », on utilise le plus souvent 7 LED disposées de manière à reproduire le chiffre en respect de la disposition classique d'un dé. C'était d'ailleurs l'approche que nous envisagions jusqu'à ce qu'un membre de la rédaction se souvienne brusquement qu'« un dé doit rouler... ». Un dé n'est pas né pour rester à plat sur la table.

« Mais c'est bien là précisément l'intérêt d'un dé électronique ! » direz-vous. Il n'est pas nécessaire de disposer de place pour lancer un dé qui tombe toujours de la table ! C'est également ce que dirent nos ingénieurs en réaction à cette proposition saugrenue, mais le rédacteur en question ne se laissa pas désarmer : ce dé doit rouler et roulera ! Heureusement que ce genre de situations est « attaqué » doucement par l'équipe d'Elektor et rapidement il s'avéra que le « Labo » voulait bien jeter un coup d'oeil au problème.

Il ne fallut pas longtemps avant de voir arriver les premières propositions sur papier. L'une des réalisations évidentes consiste à imaginer un dé sous forme de cube doté sur toutes ses faces de LED qui donnent la valeur sortie. Les LED, dotées de leur résistance de limitation de courant, sont reliées, via un interrupteur de mise en fonction (mais pas marche/arrêt), à une pile de 9 V et l'affaire est réglée !

Il apparut rapidement qu'il est extrêmement difficile, dans le cas d'un dé de fabrication-maison, d'arriver à centrer parfaitement le centre de gravité. Un tel dé pourra toujours être soupçonné de tricherie. Il y a plus de chances de retrouver la face la plus légère en haut. La valeur correspondante sortira partant plus souvent que celle d'une face « plus lourde ».

## Comment faire alors ?

Très rapidement, nous en arrivâmes à la conclusion que la meilleure option consistait à placer 7 LED sur chaque face. Après le jet du dé, chacune des 6 faces prendra une valeur aléatoire. Nous sommes assurés ainsi que le dé se comportera « honnêtement ».

Cette solution implique bien plus d'électronique. Il faut en effet commencer par détecter le lancement du dé. Nous utilisons pour cela un interrupteur à mercure. Cela permet aussi de « lancer » le dé en le secouant tout simplement.

L'électronique doit ensuite générer des chiffres aléatoires, qui seront affichés par les 6 faces du dé. Il va de soi qu'il va falloir faire appel à un microcontrôleur pour cette tâche. Ceci permet de réaliser un montage compact, logique lorsqu'il s'agit d'un dé direz-vous !

Autre point délicat, l'interconnexion des platines constituant les faces. Pas très joli d'avoir à effectuer, depuis la platine principale, des dizaines de liaisons vers les 5 autres circuits imprimés. Nous avons opté pour une liaison sérielle entre les platines pour limiter le nombre d'interconnexions. Sur chaque platine, un registre à décalage se charge de la commande des LED et il reste 4 liaisons à effectuer, 2 pour l'alimentation, une autre pour les données et la dernière pour le signal d'horloge.

La consommation de courant et l'interrupteur marche/arrêt furent un autre point méritant attention. Nous ne pouvons pas, ici, utiliser un interrupteur standard vu qu'il ne doit pas être protubérant. Le dé ferait sinon de bizarres cabrioles et risquerait même de se couper de lui-même. La solution : doter le montage d'un bouton-poussoir encastéré pour mettre le dé en fonction, son arrêt se faisant automatiquement.

## La platine principale

Toutes ces considérations nous ont donné le schéma final. Le coeur du circuit est IC1, un microcontrôleur d'Atmel (cf. **figure 1**). Il s'agit d'un AT89C2051 que nous avons déjà utilisé à plusieurs reprises et décrit dans le détail. Nous nous limiterons à signaler que le logiciel de ce projet est disponible au téléchargement gratuit sur notre site ([www.elektor.fr](http://www.elektor.fr)). Le contrôleur disposant de mémoire Flash interne, de la mémoire externe n'est pas nécessaire. Le quartz X1 fournit, épaulé par C1 et C2, le signal d'horloge. Via certaines de ses lignes d'Entrées/Sorties, IC1 attaque directement 7 LED. Il n'est pas nécessaire de prévoir de tampon additionnel vu qu'il s'agit de LED à faible courant.

Le schéma de l'alimentation peut paraître singulier. Cela tient au fait que nous voulons « allumer » le dé à l'aide d'un bouton-poussoir et charger le microcontrôleur de couper lui-même l'alimentation. Cette fonctionnalité est l'affaire de l'électronique centrée sur T1 et T2. K2 sert à la connexion de la pile 9 V. Lors d'une action sur le bouton-poussoir S1 de la platine 2, un faible courant se met à circuler du +9 V vers la masse en passant par la jonction base-émetteur de T1 et la résistance R8. T1 entre ainsi en conduction, permettant au courant de circuler vers le régulateur de tension IC2. Ce dernier alimente ensuite le reste du montage sous 5 V.

Lorsque la pression sur S1 cesse, il ne peut plus circuler de courant par R8. Il faut cependant que le circuit continue d'être alimenté. Il n'est pas évident de garder le bouton-poussoir enfoncé lors du jet du dé. Où donc est le confort d'utilisation ?

Pour résoudre ce problème, nous avons ajouté la résistance R9 et le transistor T2. Ce dernier est attaqué par le contrôleur. Après mise sous tension, T2

## Figure 3. Réalisation du dé 3D en 9 photos.

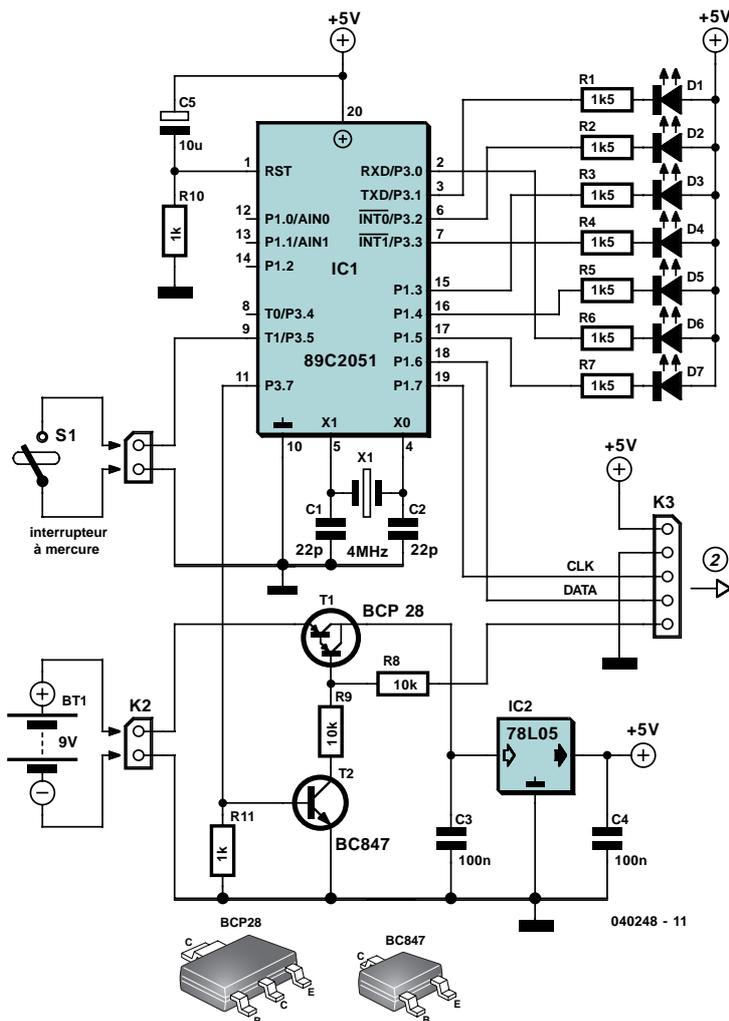
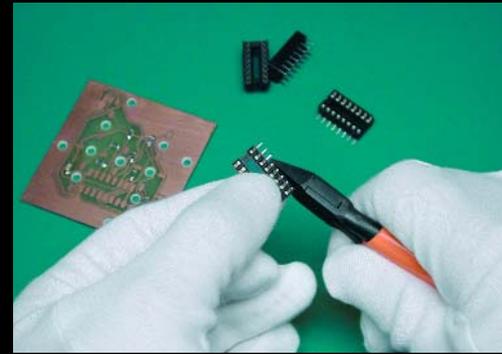
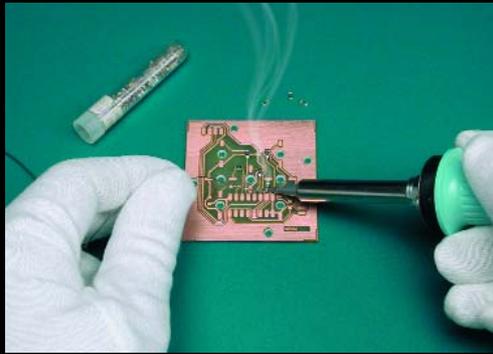


Figure 1. La platine principale est relativement complexe.

entre en conduction, ce qui permet la circulation d'un courant depuis la base de T1, vers la masse en passant par R9 et T2. L'alimentation ne dépend plus ainsi d'une pression sur l'interrupteur S1.

Si le processeur constate que le dé reste immobile un certain temps (quelques minutes), T2 cesse d'être

attaqué : le dé se met automatiquement hors-tension. Une pression sur S1 redémarre tout le processus. Lors de l'application de la tension d'alimentation C5 et R10 fournissent un signal d'initialisation (reset) correct au microcontrôleur.

La transmission des signaux CLK et DATA vers les platines suivées en aval se fait par l'embase K3. Cette dernière

comporte également des contacts pour la tension d'alimentation et l'interrupteur de mise en fonction (contact 5). L'interrupteur se trouve en effet, pour des raisons de place, sur la platine 2.

### Autres platines

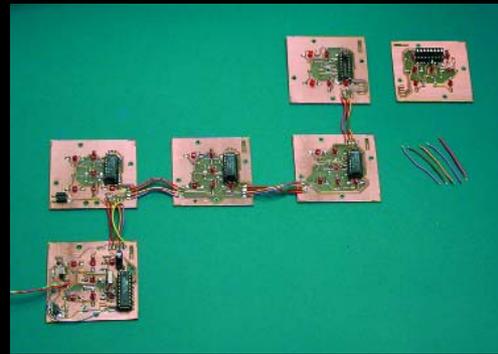
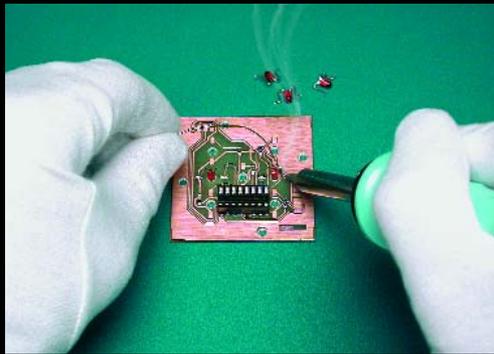
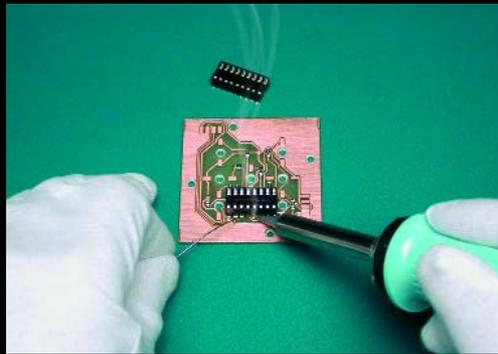
Les schémas des autres platines (2 à 6) sont quasiment identiques. Seul le schéma 2 diffère puisqu'il comporte un interrupteur qui n'existe pas sur les autres platines. La figure 2 donne le schéma 2 et celui des 4 dernières platines. On ne sera guère étonné de l'absence du connecteur de transfert de signaux K2 sur la dernière platine.

Le coeur de ces platines est le registre à décalage, un 74HCT4094. Ce circuit intégré possède un registre de sortie parallèle. Chaque flanc montant appliqué à l'entrée d'horloge se traduit, dans le circuit, par un décalage d'une position des 8 bits de données. Le dernier bit « poussé » vers l'extérieur est relié, par le biais de la broche 9, à l'entrée de donnée du registre à décalage de la platine suivante. De ce fait, le montage est une sorte de registre à décalage « monstre ».

### Réalisation

La réalisation de ces petites platines sort de la routine ordinaire. En certains endroits nous avons utilisé des composants standards que nous soudons comme des CMS. La raison est que nous ne voulions pas fils de connexion sur le dessous des platines. Cela se serait traduit par des points acérés qui auraient pu endommager une belle (et chère) table en bois lors du jet du dé. Nous avons, pour vous aider, fait une sorte de petit reportage photo de la réalisation de notre prototype (figure 3), une image parlant plus que mille mots (Confucius).

La figure 4 donne la sérigraphie de



toutes les platines du montage. Nous commençons par le soudage des composants CMS. Viennent ensuite les supports pour circuit intégré soudés comme s'il s'agissait de CMS. Il faut pour cela replier les pattes vers l'extérieur et les raccourcir à la bonne longueur (à vérifier sur la platine !).

C'est ensuite au tour des 42 (!) LED. Il faut commencer par mettre les LED en forme (cf. **figure 5**). Attention à veiller à mettre l'anode et la cathode à la position correcte sur la platine. Le boîtier de la LED doit passer dans l'orifice prévu à son intention. Les pattes sont ensuite mises à plat et soudées en place. Il n'est pas important, pour l'instant, que les LED soient parfaitement enfichées. Une fois les LED soudées, on pourra appuyer sur le boîtier des LED pour qu'elles sortent toutes très légèrement. Si l'on pose la platine à plat sur une surface plane il suffit d'appuyer sur la platine pour que toutes les LED se trouvent exactement à leur place.

Outre les circuits intégrés en boîtier DIL, nous utilisons aussi d'autres composants en version classique. Pour éviter que les pattes de ces composants et les soudures ne départent les faces extérieures du dé, il faudra les souder comme le seraient des CMS. Il s'agit, entre autres, du quartz et de l'interrupteur à mercure. Sur les dits composants les pattes seront raccourcies et repliées à 90 ° environ avant qu'elles ne soient soudées. Attention à ne pas couper trop court !

## Interconnexion des platines

Les différentes platines ayant été réalisées, il est temps de les interconnecter. La figure 4 montre comment s'y

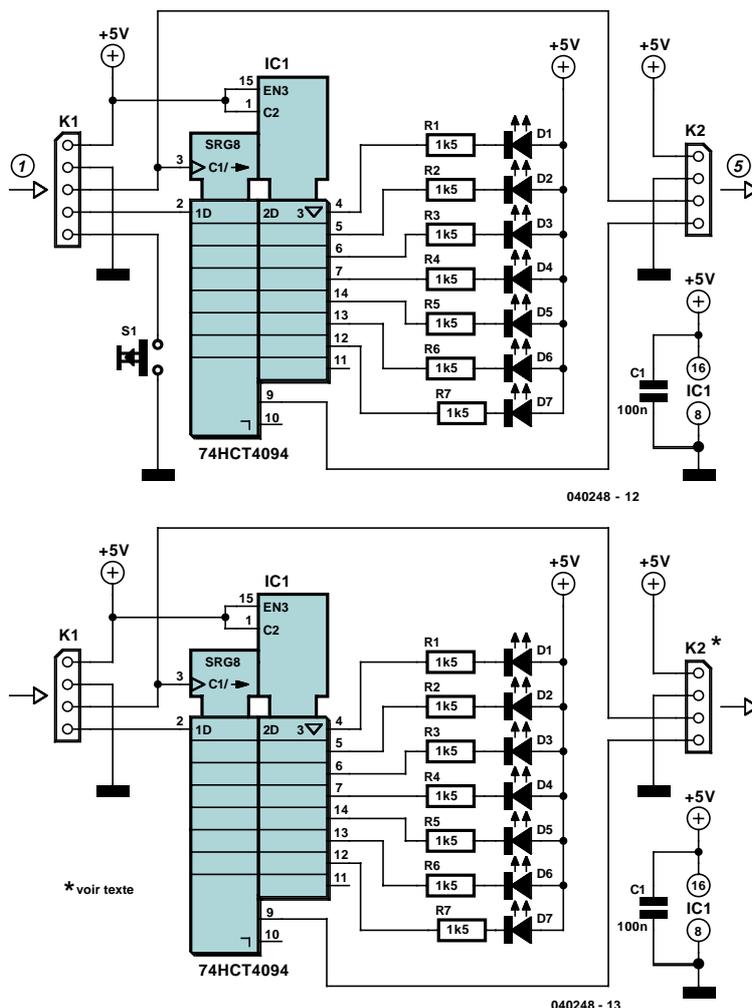
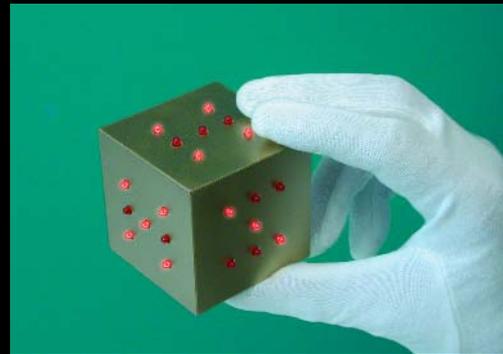
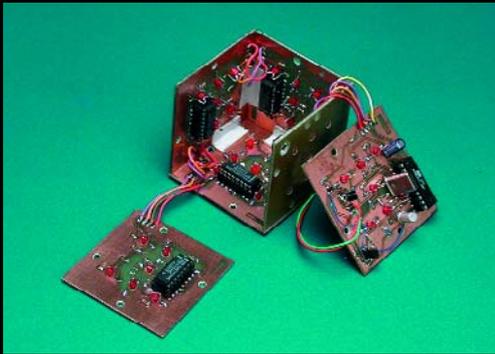


Figure 2. En haut, le schéma de la platine 2, en bas celui des 4 platines restantes.

prendre. On utilisera pour cela des morceaux de conducteur de 5 cm de long environ. L'utilisation de longueurs trop faibles compliquera ultérieurement inutilement l'interconnexion des platines. Vous courez en outre le risque que l'un ou l'autre fil lâche ! Dernière étape : insérer les circuits intégrés dans les supports. Attention à l'orientation !

L'aspect construction électronique étant terminé, il est temps de passer aux essais. Il est judicieux de les effectuer avant d'avoir terminé la construction mécanique du dé. L'électronique est accessible pour le moment et ne le sera plus le dé monté. Plus facile à éliminer un problème. Branchez, après vous être assuré de l'absence de court-circuit, la pile 9 V.



Une action sur l'interrupteur (de la platine 2) réveille le circuit et allume un certain nombre de LED sur les différentes platines.

Si vous secouez la platine 1, le dé « roulera » virtuellement. Les chiffres des différentes faces changent de valeur. Au bout de quelques secondes, la vitesse diminue et chaque face affiche un chiffre.

### 3D

Ce n'est pas parce que l'électronique est OK que l'on se trouve en présence d'un dé. Il faut maintenant monter les platines pour en faire un cube. On mettra la platine 4 à plat sur la surface de travail et on place les platines 2, 3, 5 et 6 à l'équerre par rapport à la platine de base (cf. **figure 6**).

Utiliser un profil en équerre de plastique (à acheter dans les magasins de bricolage) pour fixer les platines entre elles-mêmes. Découpez le profil en petits morceaux et collez-les sur les platines. Il nous faut pouvoir enlever le dernier circuit imprimé (platine 1) pour changer la pile, de sorte que l'on évitera de

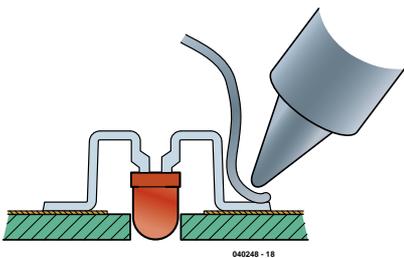


Figure 5. Repliez les LED comme indiqué ici.

## Liste des composants

### Résistances :

R1 à R7 = 1kΩ5 CMS  
R8,R9 = 10 kΩ CMS  
R10 = 1 kΩ CMS  
R11 = 100 kΩ CMS

### Condensateurs :

C1, C2 = 22 pF  
C3,C4 = 100 nF CMS  
C5 = 10 μF/16 V radial

### Semi-conducteurs :

D1 à D7 = LED 3 mm rouge faible courant  
IC1 = AT89C2051-12PI (programmé **EPS040248-41**)  
IC2 = 78L05  
T1 = BCP28 (Conrad 153225-8B)  
T2 = BC547B

### Divers :

X1 = quartz 4 MHz  
S1 = interrupteur au mercure (Farnell 178-338)  
pile 9 V avec connecteur à pression  
support pour circuit intégré 20 broches  
profil plastique en équerre  
2 x vis M3 (à tête fraisée)

### Liste des composants des platines 2 à 6 (pour chaque platine)

#### Résistances :

R1 à R7 = 1kΩ5 CMS

#### Condensateurs :

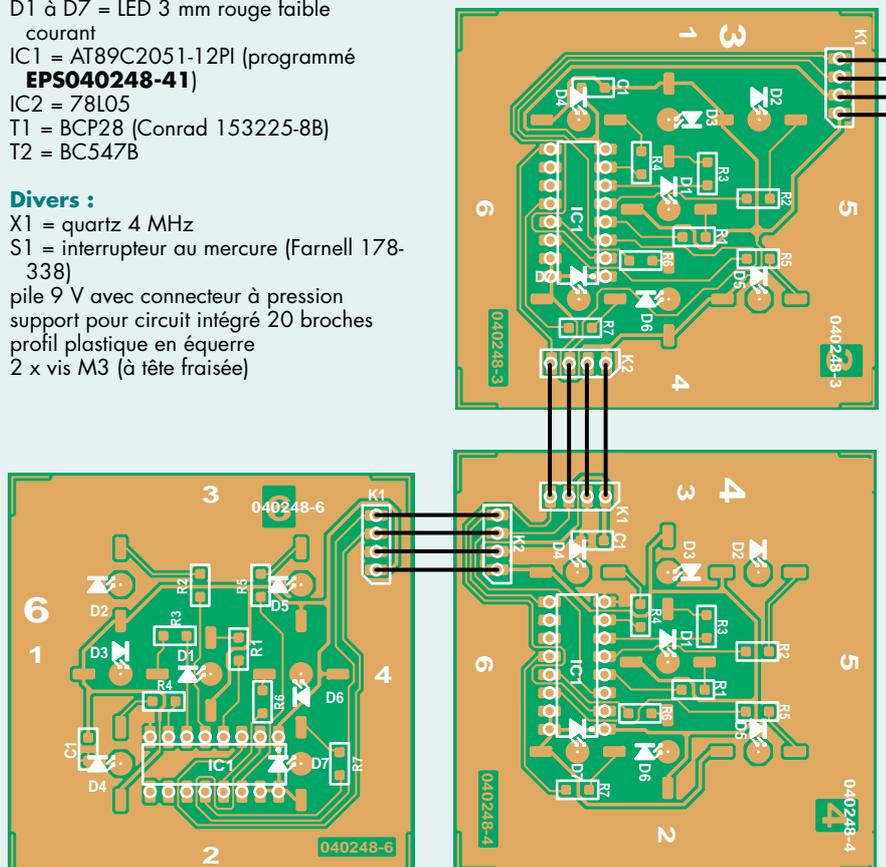
C1 = 100 nF CMS

#### Semi-conducteurs :

D1 à D7 = LED 3 mm rouge faible courant  
IC1 = 74HCT4094

#### Divers :

support pour circuit intégré 16 broches



# Étapes de construction

- Soudage des résistances CMS et condensateurs
- Replier les pattes des supports et les souder

- Souder les supports
- Replier les LED et les souder
- Interconnexion des platines
- Tester

- Fixer les platines entre elles
- Remplir de mousse et mettre la pile
- Monter la dernière platine
- Test final

S1 = interrupteur DTS61K (platine 2 uniquement)

Platine EPS 040248-1 (comporte 6 platines pour réaliser un dé complet)  
Disquette avec code source et hex : EPS040248-11

Dessin des pistes et logiciels sont également disponibles sur [www.elektor.fr](http://www.elektor.fr)

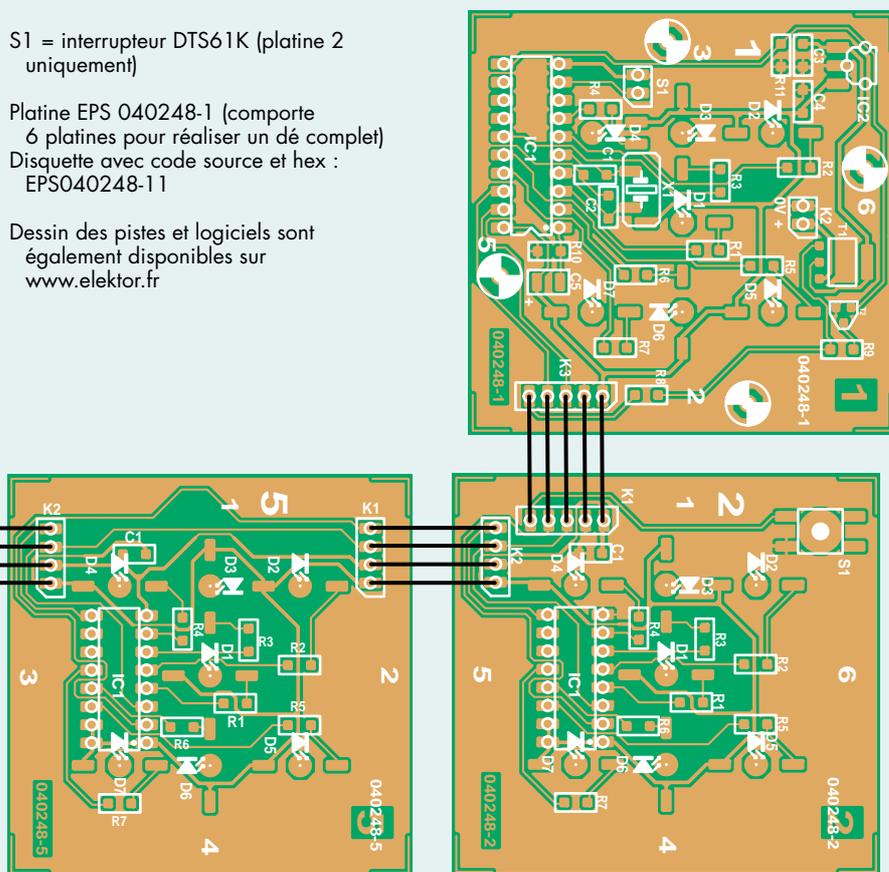


Figure 4. Sérigraphie des composants. 4 des 6 platines sont quasiment identiques.

le coller. Collez un petit morceau de profil en équerre sur les autres platines de manière à ce que les 2 orifices de 3 mm de la platine 1 reposent sur les profils. On perce ensuite avec précaution, un orifice de 2,5 mm aux endroits où se trouvent les orifices et on réalise un filetage de 3 mm. De manière à renforcer le total, on pourra coller une petite chute de profil sur le dessous et y percer un orifice doté d'un filetage. La platine 1 pourra alors être fixée à l'aide de 2 vis.

Remplissez l'intérieur du dé de coton, mousse de caoutchouc ou autre matériau d'amortissement de manière à éviter que la pile, enveloppée dans une gaine de plastique, ne puisse provoquer de court-circuit.

## En conclusion

Vous pourrez, comme nous l'avons fait pour notre prototype, doter votre dé de belles couleurs; nous vous laissons toute liberté à ce sujet. Bonne chance à tous les jeux que vous jouerez avec ce magnifique dé 3 D !

(040248-1)

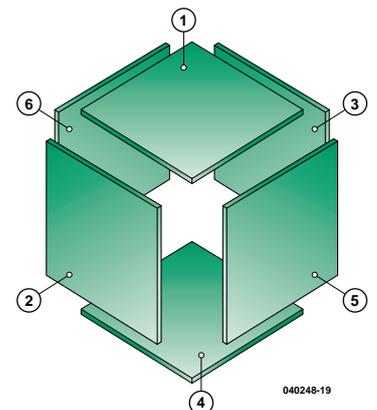


Figure 6. Les 6 platines constituent ensemble un cube.

# Feu arrière à m

## Pour votre bicyclette

Ralf Nolde

A close-up photograph of a black bicycle rear light mounted on a fender. The light has a red lens with a pattern of small triangles. The background is a bright, blurred yellow and white, suggesting a studio setting.

Pris dans leur bataille des prix, les fabricants et revendeurs de bicyclettes n'utilisent souvent que l'éclairage le moins cher possible, même pour le haut de gamme. Les cyclistes conscients de sécurité montent qui une lampe halogène à l'avant, qui un feu avec feu de position électronique très lumineux à l'arrière.

# mode « arrêt »

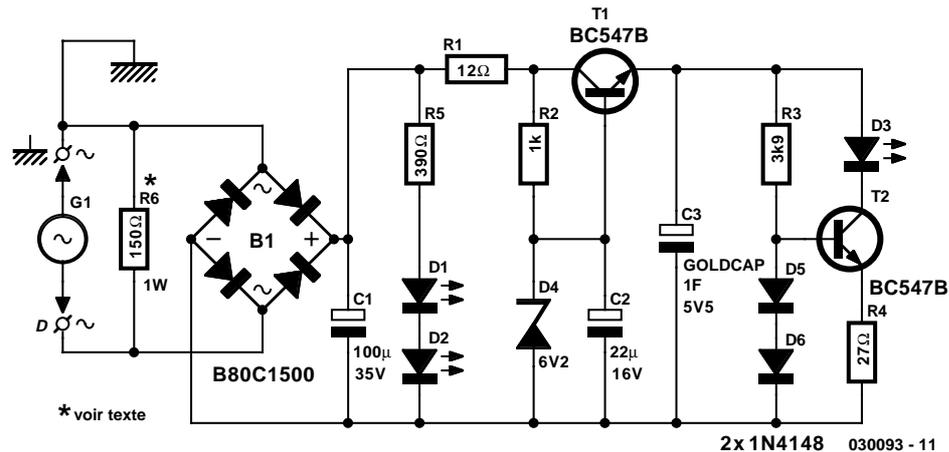


Figure 1. Le circuit du feu arrière rappelle l'électronique d'une alimentation régulée.

Il est possible, à moindre peine et peu de frais, de réaliser soi-même un tel feu arrière à LED. La solution décrite ici se caractérise par une luminosité importante, conservée, même à l'arrêt, et ce pendant quelque temps. Comme elle n'utilise pas d'ampoule pour l'éclairage ni de pile pour son alimentation, son entretien est réduit au minimum, ce qui ne peut qu'améliorer la sécurité de tout cycliste.

## L'électronique...

On retrouve, en **figure 1**, l'électronique de notre feu arrière permanent. La tension alternative fournie par la dynamo arrive, par le biais des contacts D (dynamo) et G1 (cadre) au pont de redressement B1; elle est convertie, par un redressement double alternance, en une tension continue. C1 lisse cette tension continue pulsée. En cours de trajet, la tension de dynamo redressée charge la GoldCap C3 au travers de la résistance R1 et du transistor T1. Pour éviter un dépassement de la tension nominale de la

GoldCap, 5,5 V, la diode zener D4 limite à 6,1 V (stabilisée par C2) la tension de base du transistor-série T1 et partant à un peu moins de 5,5 V sa tension d'émetteur. R1 limite le courant de charge à travers le transistor.

Au cours du trajet on n'a pas uniquement charge de la GoldCap C3 mais également alimentation de « l'éclairage principal », la LED D3. Il s'agit d'une LED de 8 mm de diamètre très lumineuse lorsqu'elle est parcourue par un courant de 26 mA. Elle est alimentée par un courant régulé très peu sensible à l'état de charge de la GoldCap : la tension de base de T1 reste constante à quelque 1,4 V; outre la tension base-émetteur, on a une chute de 0,7 V aux bornes de R4. L'auteur indique un courant de quelque 26 mA à travers la LED D3.

À l'arrêt, la LED est bien entendu alimentée par la GoldCap. Avec le dimensionnement du schéma, la charge de la GoldCap donne 2 minutes de lumière avant que la LED ne commence à s'assombrir. La durée de cet allumage dépend principalement de la valeur de

R4. Si l'on veut une durée plus longue au prix d'une luminosité moindre, on pourra faire passer la valeur de R4 à 33 voir 39 Ω. Ceci se traduit cependant par une luminosité de marche un peu plus faible.

Nous n'avons pas parlé de certains des composants. Si la luminosité de la LED est sensiblement supérieure à celle d'une ampoule miniature classique, l'angle de vision est plus limité. Ceci explique la présence, à gauche et à droite de D3, de 2 LED, D1 et D2 qui, en fonction de la tension de dynamo et donc de la vitesse de la bicyclette, s'allument aussi augmentant ainsi l'angle de rayonnement.

La consommation de courant totale du feu arrière à LED est de quelque 45 mA dans le cas d'une tension de dynamo alternative de 6 V, ce qui correspond à une puissance de 0,3 W, soit environ la moitié de celle d'une ampoule à incandescence de feu arrière classique. Ceci se traduit par une augmentation de la consommation de l'éclairage avant. Si ce dernier se manifeste alors par une luminosité

# Réglémentation

Certains pays, dont la RFA, envisagent de rendre obligatoire le feu arrière à feu de position électronique pour les nouveaux modèles de bicyclettes mises sur le marché. Cet éclairage n'a pas été homologué. A noter que nombre de pays européens interdisent la participation à la circulation des bicyclettes non dotées d'éclairage, comme les bicyclettes tout terrain MTB et BMX. La présente réalisation peut se targuer d'être supérieure, tant au niveau de la fonctionnalité que de la puissance d'éclairage et de la fiabilité à nombre de produits homologués vendus dans le commerce. L'auteur insiste cependant sur le fait qu'il s'agit ici d'un circuit non commercial à utiliser à des fins techniques d'expérimentation.

## Liste des composants

### Résistances :

R1 = 12  $\Omega$   
R2 = 1 k $\Omega$   
R3 = 3k $\Omega$   
R4 = 27  $\Omega$   
R5 = 390  $\Omega$   
R6 = 150  $\Omega$ /1 W\*

### Condensateurs :

C1 = 100  $\mu$ F/35 V radial  
C2 = 22  $\mu$ F/16 V radial  
C3 = GoldCap 1 F/5V5 (Selectronic 48.8607)

### Semi-conducteurs :

B1 = B80C1500 rond  
D1, D2 = LED rouge 3 mm (Kingsbright L-934SRC-J)  
D3 = LED super-rouge 8 mm (Kingsbright L-793SRC-F)  
D4, D5 = 1N4148  
D6 = diode zener 6V2/500 mW  
T1, T2 = BC547B

### Divers :

Platine 030093-1  
boîtier pour feu arrière (Busch & Müller 339, cf. [www.bumm.de/index-f.html](http://www.bumm.de/index-f.html))

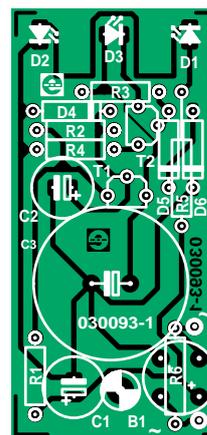


Figure 2. La petite platine a été dessinée pour s'encaster dans un feu arrière du commerce.

plus importante, une vitesse importante peut se traduire par une destruction rapide de l'ampoule. La solution à ce problème est la résistance R6 qui fait pratiquement passer la consommation du feu arrière à celle d'une ampoule classique. Si vous ne pédalez toujours que tranquillement, vous pouvez supprimer cette résistance.

### ... et sa réalisation

Notre feu arrière à feu de position n'utilise que des composants conventionnels. La petite platine **EPS030093-1** a été conçue pour être montée dans un boîtier de feu arrière de Busch & Müller. Bien qu'il s'agisse d'une simple face, il sera doté de composants des 2 côtés (à moins de monter R5 au-dessus de D1 et de D2 et R6 en-dessous du redresseur).

La mise en place des composants est facile bien que serrée. Attention à la polarité correcte des diodes, transistors et condensateurs électrochimiques. Pour un test de fonctionnement rapide, on connectera une pile de 9 V au circuit. Peu importe sa polarité, toutes les

LED doivent s'allumer très visiblement. Après une courte durée de chargement, on vérifie la tension sur la GoldCap (5,5 V environ) et la tension aux bornes de R4 (<0,75 V).

Si le test est satisfaisant, on protège la LED par du ruban papier autocollant et on applique un enduit en spray sur les 2 faces de la platine pour la protéger contre l'humidité et la poussière. Après avoir laissé sécher, on pourra substituer la platine aux composants se trouvant normalement dans le feu arrière. La platine sera montée dans la demi-coquille inférieure de manière que la partie au-dessus de D3 vienne s'encaster sous la languette de cuivre et qu'elle puisse être fixée en-bas à l'aide d'une vis pour tôle. Après connexion (bien assurée !) des câbles on fixera les composants et les câbles à la colle thermoplastique sur la platine et entre eux, pour une meilleure tenue aux vibrations. Il reste à placer, avec précaution, le capot de manière à ce que les LED viennent s'encaster aux endroits prévus. Vous voici en possession d'un feu arrière « High Tech ».

(030093-1)

## MISES AU POINT

### Orgue lumineux de poche,

Elektor juin 2004, page 70 et suivantes (030019-1)

IC1 identifié par erreur comme un LM385N dans le schéma et la liste des composants est en fait, comme permet de le vérifier un coup d'œil à la photo, un LM358N. C'est en effet le composant à utiliser, le prototype fonctionnant le confirmant, vu qu'il s'agit d'un amplificateur opérationnel. Le LM385N est une référence de tension qui n'a rien à faire dans un orgue lumineux de poche.

### Les Canon EOS passent au sans fil,

Elektor mai 2004, page 64 et suivantes (030432-1)

Sur le schéma, l'alimentation de l'émetteur prend la forme d'une pile 9 V. Il s'agit en fait de 3 piles de 1,5 V, des piles AAA (LR03) par exemple. Dans la liste des composants il faudrait partant lire : BT1, 3 piles 1,5 V LR03 + porte-pile pour 3 piles. On pourrait fort bien envisager d'alimenter l'émetteur à l'aide d'une pile de 9 V, mais cela impliquerait l'ajout d'un régulateur 5 V (7805) comme en comporte le récepteur.

**MAX734**
**Régulateur 12 V pour programmation  
de mémoires Flash**
**MAX734**
**Régulateur 12 V pour programmation  
de mémoires Flash**
**Fabricant :**

 Maxim  
[www.maxim-ic.com](http://www.maxim-ic.com)  
<http://pdfserv.maxim-ic.com/en/ds/MAX734.pdf>
**Caractéristiques :**

- Sortie +12 V régulée ( $\pm 5\%$ )
- Courant en sortie de 120 mA garanti
- Régulateur complet extrêmement compact
- Pilotage logique du mode de mise en sommeil (*shutdown*, 70  $\mu\text{A}$ )
- Rendement typique de 85%

**Description :**

Le MAX734 est un régulateur à découpage rehausseur (*step-up*) fournissant +12 V en sortie. À une tension d'entrée de +4,75 V, il garantit un courant de 120 mA, l'idéal pour la programmation de mémoires Flash. Les seuls composants connexes requis par le MAX734 proposé en boîtier SO ou DIP, sont une diode, une self de 18  $\mu\text{H}$  et une paire de condensateurs de 33  $\mu\text{F}$ . Le régulateur possède une entrée de mise en sommeil « compatible logique » pouvant être attaquée directement par un microcontrôleur.

De par son rendement de 85%, sa consommation de courant de 1,2 mA (en fonctionnement) ou de 70  $\mu\text{A}$  (*shutdown*) seulement, le MAX734 convient idéalement aux appareils alimentés par piles. Il est même possible, en « jouant » sur la broche *shutdown*, de faire tomber la consommation de courant en fonctionnement à moins de 500  $\mu\text{A}$ .

Le régulateur travaille en MLI (**M**odulation en **L**argeur d'**I**mpulsion = PWM pour *Pulse Width Modulation*) du courant, de manière à réaliser une régulation de la sortie précise ne comportant que peu de bruit subharmonique. Un oscillateur travaillant à une fréquence fixe de 170 kHz permet un filtrage des ondulations en utilisant, comparativement, des condensateurs externes de faible capacité. On utilisera le MAX732 pour les applications requérant des courants élevés jusqu'à 250 mA.

À noter qu'il existe, pour le MAX734, un kit d'évaluation, baptisé MAX734EVKIT-SO.

**Applications 12 V typiques :**

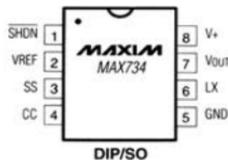
- Tension de programmation pour les mémoires Flash (+12 V)
- Alimentation PCMCIA
- Disques à semi-conducteurs (*solid-state*)
- Ordinateurs de poche (PDA)
- Alimentation compacte pour opamp

**Boîtiers et brochage :**

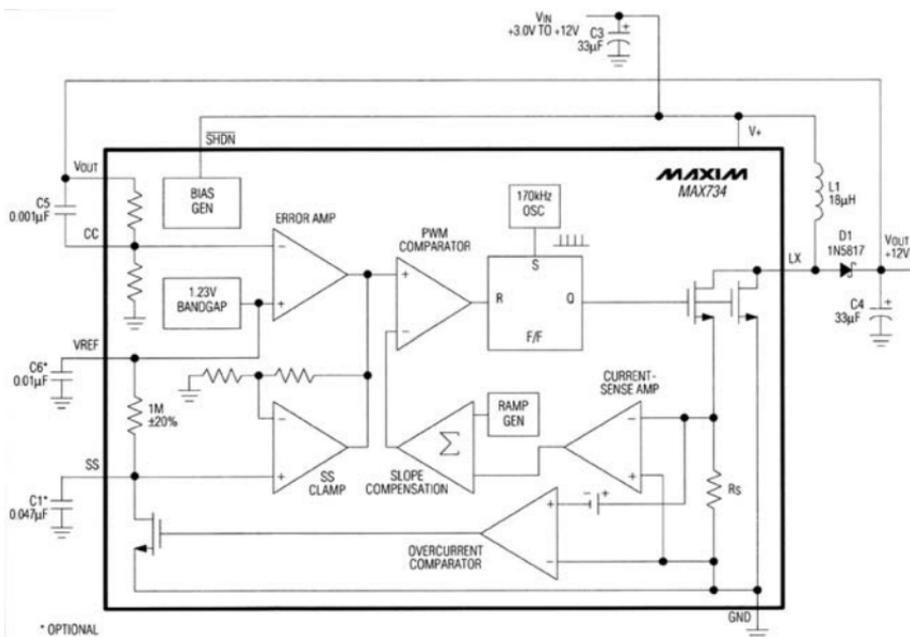
MAX734....	Plage de température [°C]
CPA	0 à 70
CSA	0 à 70
EPA	-40 à +85
ESA	-40 à +85
MJA	-55 à +125

**Boîtier**

Plastique DIP8  
 Small Outline SO8  
 Plastique DIP8  
 Small Outline SO8  
 Céramique CERDIP8



Broche	Description	Fonction
1	$\overline{\text{SHDN}}$	Entrée de mise en sommeil ( <i>shutdown</i> ) active au niveau bas – forçage à la masse = shutdown – le FET de puissance est déconnecté, fonctionnement normal si $\overline{\text{SHDN}}$ est forcé à V+
2	$V_{\text{REF}}$	Sortie de tension de référence (+1,23 V) – peut être chargé jusqu'à 100 $\mu\text{A}$
3	SS	Soft Start – Forçage à la masse par un condensateur donne démarrage en douceur et protection contre les courts-circuits
4	CC	Entrée – Condensateur pour la compensation externe de la boucle de feedback externe
5	GND	Masse
6	LX	Drain du FETMOS à canal-N de puissance interne
7	$V_{\text{OUT}}$	Entrée de suivi de la tension de sortie par boucle de feedback
8	V+	Entrée de la tension d'alimentation. Placer le condensateur de découplage le plus près possible !

**Synoptique avec composants externes :**


**Valeurs limites :**

Paramètre		Valeur
Tensions aux broches	V+, LX	+17 V, -0,3 V
	V <sub>OUT</sub>	±25 V
	SS, CC, $\overline{\text{SHDN}}$	-0,3 V à (V+ + 0,3 V)
Courant d'enclenchement de crête I <sub>LX</sub>		1,5 A
Courant de référence I <sub>VREF</sub>		2,5 mA
Dissipation (T <sub>A</sub> = 70 °C)	DIP plastique	727 mW
	SO	471 mW
	CERDIP	640 mW
Température de la couche limite		+150 °C
		+175 °C (MJA uniquement)

**Chronologie de démarrage en douceur (softstart) :**

V+	Durée de softstart en ms				
	sans C <sub>SS</sub>	0,047 μF	0,1 μF	0,47 μF	1,0 μF
5 V	1	29	55	260	500
7,5 V		18	27	83	162
9 V		6	10	47	78

**Caractéristiques électriques :**

 (Circuit de la figure 2,  $V_+ = 5\text{ V}$ ,  $I_{\text{LOAD}} = 0\text{ mA}$ ,  $T_A = T_{\text{MIN}}$  à  $T_{\text{MAX}}$ , valeurs typiques à  $T_A = 25\text{ }^\circ\text{C}$ , sauf mention contraire)

Paramètre	Conditions	Min.	Typ.	Max.	Unité	
Tension de sortie	$V_+ = 4,75$ à $12\text{ V}$ , $0\text{ mA}$ $< I_{\text{LOAD}} < 120\text{ mA}$	MAX734C/E	11,64	12,12	12,60	V
		MAX734M	11,40	12,12	12,60	
Courant de charge	$V_+ = 4,75\text{ V}$	120	150		mA	
	$V_+ = 4,5\text{ V}$		225			
	$V_+ = 3,0\text{ V}$		150			
Tension d'entrée maximale				$V_{\text{OUT}}$	V	
Régulation de ligne	$V_+ = 5$ à $12\text{ V}$		0,20		%/V	
Régulation de charge	$I_{\text{LOAD}} = 0$ à $120\text{ mA}$		0,0035		%/mA	
Rendement	$V_+ = 5\text{ V}$ , $I_{\text{LOAD}} = 120\text{ mA}$		83		%	
Courant d'alimentation	avec courant de commutation		1,2	2,5	mA	
Courant de veille ( <i>standby</i> )	$\overline{\text{SHDN}} = 0$ , ensemble du circuit		70	100	$\mu\text{A}$	
	$\overline{\text{SHDN}} = 0$ , vers $V_+$		6			
Seuil d'entrée pour Shutdown	$V_{\text{IH}}$	2,0			V	
	$V_{\text{IL}}$			0,25		
Courant de fuite de l'entrée Shutdown				1,0	$\mu\text{A}$	
Résistance de commutation LX	$I_{\text{LX}} = 500\text{ mA}$		0,5		$\Omega$	
Courant de fuite LX	$V_{\text{DS}} = 12\text{ V}$		1,0		$\mu\text{A}$	
Tension de référence			1,23		V	
Dérive de la référence	$T_A = T_{\text{MIN}}$ à $T_{\text{MAX}}$		50		ppm $^\circ\text{C}$	
Fréquence d'oscillateur			170		kHz	
Impédance broche de compensation			7500		$\Omega$	



## Trilogie des inductances

Heinz Zenker, Alexander Gerfer, Bernhard Rall

Nous avons la possibilité, ici, de vous présenter, dans le cadre de notre rubrique « grains de sel », un ouvrage écrit à l'origine en allemand et qui existe maintenant en anglais et même en français. Nous avons eu la version allemande en main. Nous avons également pu consulter quelques chapitres en français et devons reconnaître que la traduction est très correcte.

Ne rien comprendre aux inductances et à leur mise en oeuvre n'est pas l'apanage des électroniciens amateurs uniquement. Nombre de concepteurs de l'industrie ou de bureau d'ingénierie ne s'approche qu'à contre-cœur de ce type de composants passifs, serait-ce sous la forme de filtres CEM, de régulateurs à découpage, en HF et autres circuits analogiques. Cela tient à plusieurs raisons : à commencer par le fait que, longtemps, les inductances n'ont pratiquement pas joué de rôle (hormis en HF) et qu'elles étaient plutôt passées sous silence lors des études, qu'ensuite leur mise en oeuvre implique bien plus de paramètres que l'utilisation de

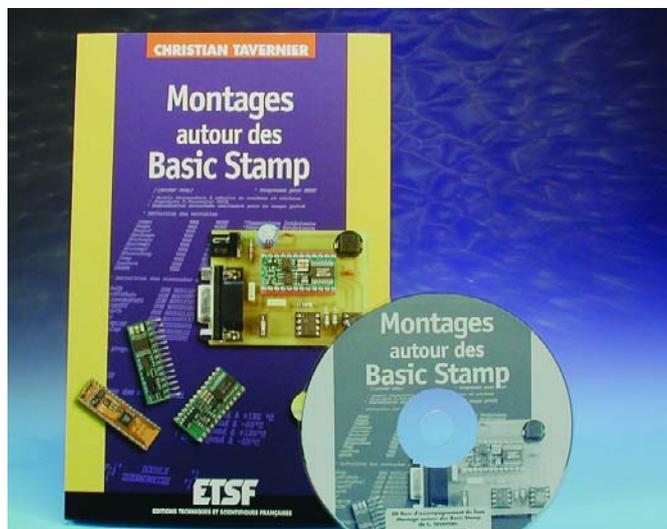
résistances ou de condensateurs par exemple et qu'il est, pour finir, extrêmement difficile à un concepteur semi-professionnel de mettre la main sur l'inductance dont il a besoin s'il n'a pas de (très) bons contacts avec l'industrie des inductances ou ses distributeurs.

Ce problème semble être en voie de résolution. Le département EMC & Inductive Solutions du groupe Würth offre également aux non-professionnels la possibilité d'obtenir des inductances en petit nombre voire à l'unité; de plus, ils proposent un ouvrage de base et de pratique de la mise en oeuvre des inductances et des ferrites dans différentes applications dont la première édition a suscité énormément d'intérêt dans le monde des professionnels.

Pour de plus amples informations concernant cet ouvrage, nous vous renvoyons à l'adresse Internet :

[www.wuerth-solergy.de/web\\_frames.php?parLANG=FR&parKAT=71](http://www.wuerth-solergy.de/web_frames.php?parLANG=FR&parKAT=71)

(040132-1)



## Montages autour des Basic Stamp

Christian Tavernier

Nous n'allons pas faire l'injure à nos lecteurs de leur présenter le(s) Basic Stamp. Ils savent tous que les Basic Stamp sont une famille de microcontrôleurs aux caractéristiques extrêmement attrayantes vu qu'ils comprennent le BASIC, se passent de circuit d'horloge et d'alimentation stabilisée et se laissent programmer depuis n'importe quel PC (un « vieux » 486 DX suffit à la peine) à l'aide d'un simple câble de liaison série. Parallax, la société où le Basic Stamp a vu le jour, ne s'est pas reposé sur les lauriers du succès foudroyant de leur premier-né, le Basic Stamp 1, lui donnant, à intervalle plus ou moins régulier, un nouveau petit frère baptisé qui Basic Stamp 2, Basic Stamp 2SX, 2E, 2P24, 2P40, les 2 premiers étant à base de PIC de Microchip, les autres de SX28 et SX48 de Scenix.

L'ouvrage présenté ici passe en revue les différentes versions de Basic Stamp, limitant la théorie au strict nécessaire, deux tiers étant consacrés à ce qui intéresse le plus nos lec-

teurs, la pratique. L'un des aspects les plus intéressants de ce livre est qu'il décrit la réalisation de ses propres modèles de Basic Stamp. Pas moins d'une vingtaine de réalisations allant de ses propres Basic Stamp à celle d'une horloge radio-pilotée en passant par une centrale d'alarme constituée l'ossature de cet ouvrage.

Un CD-ROM, comment avouons-nous pu vivre si longtemps avec la disquette 3"1/2, accompagne le livre. On y trouve les dessins des différentes platines et les programmes nécessaires aux différentes réalisations.

Cet ouvrage s'adresse aux utilisateurs passionnés de Basic Stamp qui aimeraient mettre à contribution les composants qu'ils possèdent déjà mais aussi à des utilisateurs potentiels qui n'attendent, comme les milliers d'inconditionnels de cette petite merveille, que l'occasion de passer à l'acte.

(037147-1)

# CONTRÔLEUR-

## Pour mémoires de masse mobiles

**Mark Schultz**

Senior Applications Engineer Cypress Semiconductor

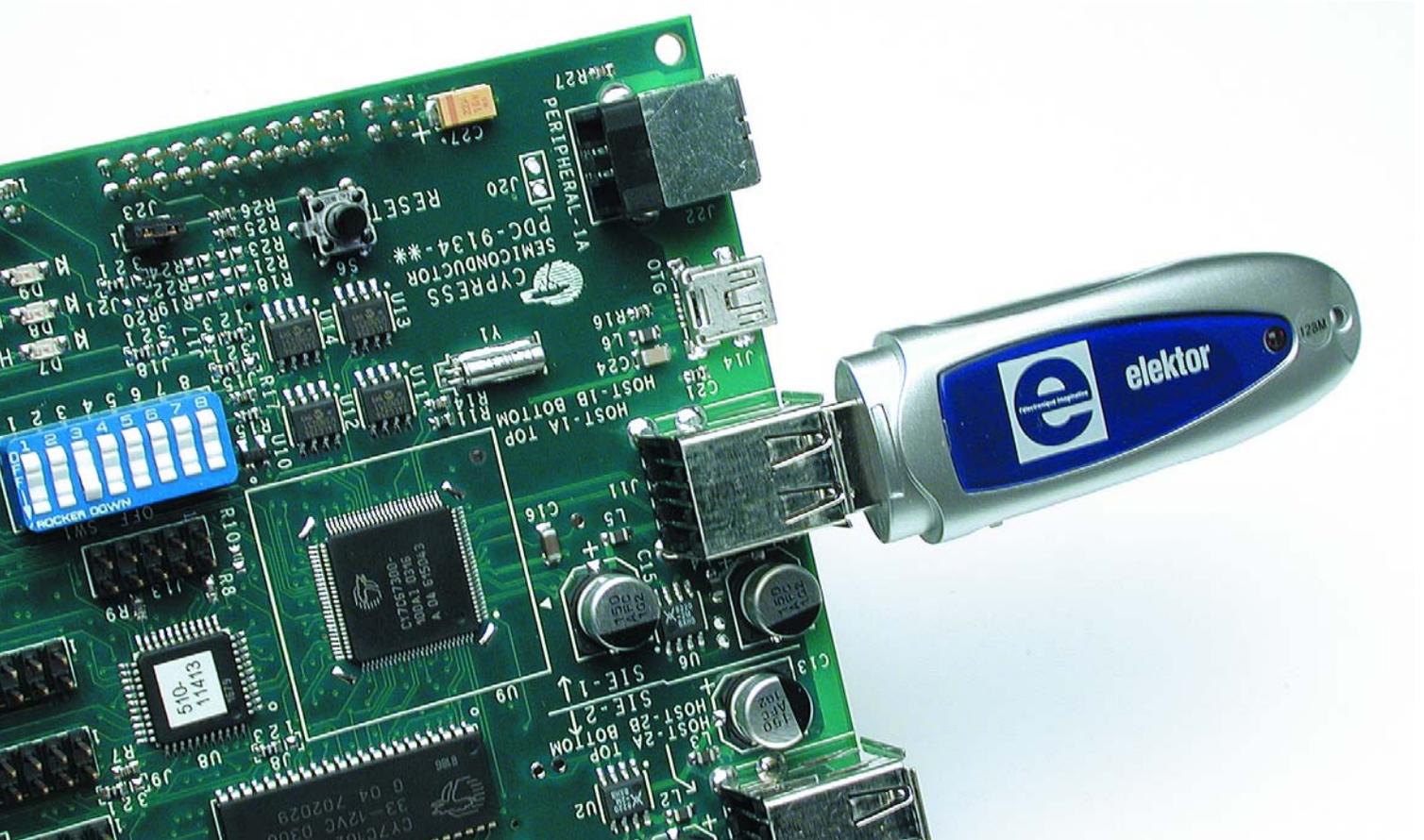
Le bus sériel universel (USB = Universal Serial Bus) n'était, par le passé, disponible que dans le monde du PC mais pas dans le monde des applications à base de systèmes enfouis (embedded systems). Les choses vont changer avec les nouveaux contrôleurs dit « Embedded Host » de Cypress.

Lorsque l'on parle de système enfoui, il est question d'un ensemble complexe matériel (*hardware*) et progiciel (*firmware*), « existant » en autonome ou faisant partie d'un système plus sophistiqué, le plus souvent associé à un système d'exploitation tel que Windows CE, VxWorks ou de fabrication-maison. Dans ces conditions, toute électronique dotée d'un processeur est un hôte-USB enfoui potentiel auquel il devient possible de connecter n'importe quel périphérique USB. Ceci permet d'augmenter ou de réduire dynamiquement à volonté la capacité de

mémoire d'une application.

Un bon exemple de ce genre d'appareil : un lecteur MP3 à hôte-USB intégré acceptant un transfert sans problème de fichiers MP3 soit par mémoire USB (*USB-stick*) soit tout autre média à support SB intégré. Ou encore la possibilité de mises à niveau (*update*) ou de télécharger des données importantes d'un système ne comportant pas de connexion réseau.

Les solutions existantes d'extension de systèmes enfouis requièrent, en règle générale, un ordinateur portable ou



# HÔTE ENFOUI

un autre système micro portable que l'on reliera au système enfoui par le biais d'un câble sériel. En dépit de sa simplicité et de son efficacité, cette solution ne peut se passer de PC et d'un câblage additionnel.

Le PC est l'exemple-type d'un hôte-USB, tous les autres appareils à interface USB étant à classer dans la catégorie des périphériques. Un exemple, l'appareil photo numérique conçu à l'origine comme périphérique USB. Il faut commencer par transférer les photos vers le PC avant de pouvoir les imprimer sur une imprimante USB. Si notre appareil photo avait la fonctionnalité d'un hôte-USB il serait possible de transférer directement les photos vers l'imprimante.

Cette fonctionnalité présente bien évidemment aussi des inconvénients. L'hôte-USB remplit un certain nombre de fonctions telles que l'énumération des périphériques connectés, la chronologie des tâches et l'attribution de bande passante. Il est intéressant et pratique, qu'ici, un « code-framework » se charge de ces tâches.

L'objet de cet article est la connexion d'un contrôleur hôte-USB à une mémoire de masse mobile. Après une introduction succincte à l'utilisation des instructions SCSI et leur relation par rapport aux systèmes de fichiers, nous examinerons les systèmes de fichiers FAT. L'article se terminera par un exemple de concept.

## Dites... mémoire de masse

Lors de la communication via l'interface USB, on utilise des instructions SCSI enfouies dans un *UFI packet*. La notion de système de fichiers sous-entend une structure logique pour différents paramètres du médium servant de mémoire.

### Le système de fichiers FAT

Le système supporte le système de fichiers FAT (*File Allocation Table*, tableau d'allocation de fichiers). Au coeur du système FAT réside un tableau comportant un numéro pour chaque partie de fichier et aussi cluster du disque. Ce tableau d'allocation est nécessaire de manière à pouvoir localiser les différents clusters d'un fichier vu que ceux-ci ne se trouvent pas nécessaire dans l'ordre et l'un après l'autre. Il existe outre FAT, d'autres systèmes de fichiers tels que NTFS (*NT File System*) et UFS (*Unix File System*) ces systèmes n'étant pas supportés par notre concept.

Au plan logique, le système de fichiers est représenté par une structure de données appelée *Volume*. Un *Volume* contient des informations sur la position du FAT, du répertoire-racine (*root*) et du domaine des données subdivisé en secteurs et clusters. Les mémoires de masse comportent toujours un enregistrement de démarrage, le *Boot Record*. On trouve à cet endroit non seulement du code utilisé pour le démarrage de l'appareil, mais également des informations utiles concernant l'appareil qui peuvent servir à comprendre les informations de volume. On trouve en outre un enregistrement de partition (*Partition Record*) pour chacune des partitions utilisables de l'appareil. Ultérieurement on se servira également de cette information. La **figure 1** illustre la structure du système de données.



Figure 1. Structure du système de fichiers.

Si l'on fait abstraction du domaine des répertoires et données, la mémoire de masse comporte toute une autre série de domaines. L'information de partition débute avec un offset de  $1BE16_{HEX}$ , calculé à partir du premier secteur physique du disque. Elle est suivie par plusieurs secteurs réservés ainsi que le tableau d'attribution des fichiers proprement dit. Le FAT travaille à l'aide de groupes de plusieurs secteurs connus sous la dénomination de clusters. Le FAT comporte des indications quant aux clusters physiques du disque, telles qu'une autorisation à l'écriture et où se trouve le cluster suivant du fichier concerné. Il existe, pour des raisons de redondance, 2 FAT. Dans le cas d'un FAT de type FAT16, le répertoire-racine comporte 512 entrées d'une longueur de 32 octets chacune. Partant, le répertoire-racine occupe 32 secteurs. Il est suivi par l'espace de données où se trouvent les fichiers et les sous-répertoires. En règle générale, on admet que l'espace de données commence au cluster 2, en réalité, il ne débute que plus de 512 secteurs après le début du disque. C'est aussi ce qu'illustre le **tableau 1** donnant les informations de volume d'une mémoire Flash USB. On trouve, dans l'espace destiné aux données, le domaine de partition, celui du boot, 2 copies du FAT ainsi que le répertoire-racine, informations utilisées ultérieurement pour la localisation des fichiers sur le lecteur.

### SCSI

Nombre de mémoires de masse utilisent des instructions SCSI pour la communication entre l'hôte et le périphérique. Dans le cas présent on utilisera des instructions SCSI pour

Tableau 1. Informations de volume		
Informations de volume	Taille données	Valeur HEX pour le lecteur utilisé ici
Taille du FAT	BYTE	FA
Nombre total de secteurs	DWORD	3E7E0
Nombre total de secteurs de données	DWORD	3E5c8
Nombre de clusters	WORD	F972
Nombre de secteurs du répertoire racine	BYTE	20
Premier secteur de données	WORD	235
Octets par secteur	WORD	200
Secteurs par cluster	BYTE	004
Premier secteur du FAT	BYTE	21
Premier secteur du répertoire	WORD	215

les opérations de fichiers. Il s'agit des instructions *Inquiry*, *Read Capacity* et *Read(10)*. Le tableau qui suit donne des informations concernant ces 3 instructions SCSI.

Chaque commande connaît un code opérationnel (*opcode*) et un LUN (*Logical Unit Number*), certaines instructions comportant un champ LBA (*Logical Block Address*) additionnel. Un bloc logique est la plus petite unité adressable souvent baptisée secteur. Nous verrons un peu plus loin comment utiliser un LBA pour l'accès aux données de fichier. L'instruction *Read(10)* comporte de plus un champ pour la longueur de transfert qui donne le nombre de secteurs à envoyer ou à recevoir. Pour éviter de compliquer les choses, nous ne travaillerons qu'avec une unité logique, de sorte que le champ LUN sera toujours à 0.

## UFI

Les mémoires de masse mobile à interface USB utilisent une

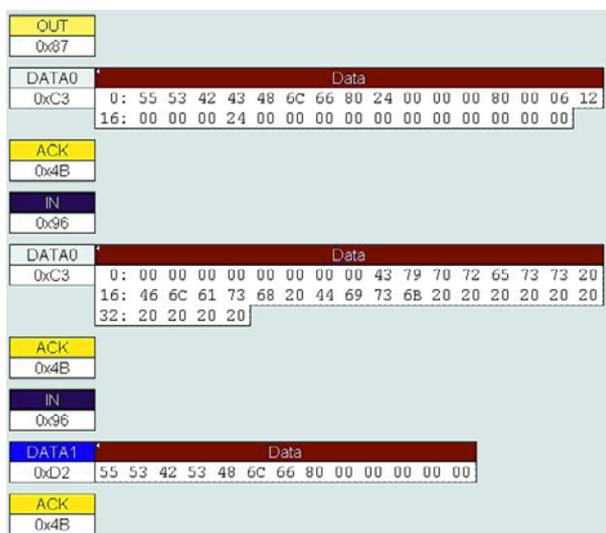


Figure 2. Transactions UFI à base de CBW et CSW.

interface uniforme baptisée interface UFI. Les instructions SCSI sont enfouies dans des paquets UFI. Un transfert UFI est exécuté en 3 phases (*Command*, *Data* et *Status*). La phase *Command* utilise un *Command Block Wrapper* (CBW), la phase *Status* au contraire un *Command Status Wrapper* (CSW). Le CBW de 31 octets se compose de 4 octets de signature, 4 octets de Tag, 3 octets de commande et indicateurs (*flags*) et 4 octets donnant la longueur de données à lire/écrire. Les 16 octets qui suivent sont le Command Bloc comportant les instructions SCSI proprement dites. La signature CW identifie le périphérique et, en respect des spécifications UFI, est fixée à 55 53 42 43. Il est intéressant de noter que la signature est, en format ASCII, USBC pour *USB Command*. Le Tag est une valeur à 4 octets. CSW utilise la valeur Tag reçue au cours de la phase CBW en vue de donner l'état du CBW. Le CSW est un paquet d'une longueur de 13 octets. Il ressemble au CBW vu qu'il comporte 4 octets de signature fixés eux aux valeurs 55 53 42 53 (en ASCII, USBS pour *USB Status*). À cela s'ajoute un tag de 4 octets, qui sera comparé au tag du CBW. Le CSW comporte aussi 4 octets qui donnent la longueur de transfert diminuée des longueurs des données déjà transférées. Pour finir, un octet additionnel donne l'état du transfert. La **figure 2** illustre le déroulement d'un transfert USB avec CBW et CSW.

CBW est la première transaction. Si les octets 0 à 3 comportent la signature, l'information de tag se trouvant elle dans les octets 4 à 7. Le tag est un nombre à 32 bits quelconque. Le CSW utilise la valeur du tag pour signaler la fin de l'instruction spécifiée dans le CBW. La valeur 24<sub>HEX</sub> de l'octet 8 correspond au nombre décimal 36 et donne la longueur effective des données transférées au cours de la seconde phase.

La seconde transaction concerne la phase de données avec les 36 octets de données lues depuis la mémoire de masse USB. Dans le cas présent les données sont une réaction à l'instruction *Inquiry* (1216), que l'on trouve dans l'octet 15 de la première transaction. Si l'on convertit au format ASCII les 36 octets de données commençant à l'octet 9, on obtient la chaîne *Cypress Flash Disk*. CSW constitue la troisième transaction. Les tags de CBW et CSW sont identiques. Les octets 8 à 11 indiquent qu'il n'y a plus de données à transférer. L'octet 12 donne l'état de la transaction (le « 0 » signalant ici son succès).

## Framework, connaît pas !

EZ-Host est un microcontrôleur RISC 16 bits de Cypress à 2 SIE *Serial Interface Engines*. Chaque SIE pilote 2 ports USB hôtes ou un port USB périphérique. EZ-Host intègre d'autres fonctions de périphériques telles que RS-232, SPI, HPI et HSS. Outre un espace de mémoire interne de 16 Koctets il existe des provisions pour une interface de mémoire externe.

Qui dit EZ-Host dit « Firmware-Framework » (cadre logiciel) qui englobe la plupart des détails d'un hôte USB. En raison d'un support total des différents types de transferts USB, les opérations de lecture et d'écriture se limitent, dans le cas des mémoires de masse USB, à donner certaines informations et à laisser au framework (le cadre) le reste du travail.

Le framework du EZ-Host englobe l'ensemble du progiciel pour l'implémentation de la fonctionnalité de l'hôte USB : la répartition des tâches (*task-scheduling*), l'énumération (enregistrement) des périphériques, l'attribution de bande passante et la gestion d'alimentation (*power management*). Dans le cas d'applications, il s'agit d'autre part du progiciel pour le pilotage de certains périphériques USB et du transfert des données correspondantes vers l'application en bout de chaîne.

Le noyau du framework est le processeur TD. Il traite une structure de données baptisée *task descriptor* (d'où le TD) et utilise ses informations pour la communication avec le matériel USB et surtout en particulier avec le SIE (*Serial Interface Engine*). Il est important que chacun des SIE pilote 2 ports et qu'il y ait un processeur TD pour chaque SIE. L'un des composants du code framework est un répartiteur (*scheduler*) simple et au fonctionnement cyclique. Celui-ci effectue une série de tâches telles que le test de la connexion de nouveaux périphériques ou inversement la déconnexion de périphériques branchés auparavant, l'interrogation de l'UART pour détecter l'arrivée de messages et l'exécution de la fonction « Run » pour chacun des pilotes installés.

On fait appel, pour le transfert des données entre l'hôte et le périphérique, et inversement, à des transferts USB élémentaires. C'est l'hôte qui décide des transferts, qui, par le biais de la fonction `usb_send_bulk_out`, transfère des données à une fonction périphérique. La fonction `usb_recv_bulk_in` cherche elle des données présentes au niveau de l'unité périphérique. Les 2 fonctions nécessitent des données additionnelles, à savoir un pointeur référant à la structure-appareil, le point USB final à utiliser, la longueur du transfert et un pointeur concernant le tampon d'émission ou de réception. Une fois ces détails spécifiés, le framework se charge du reste.

## Processus de traitement du framework

Après une initialisation à la mise sous tension (POR, *Power-On-Reset*) le microcontrôleur EZ-Host initialise les différents registres et compteurs ainsi que toutes les structures-appareil. Ensuite, il entre dans une boucle :

- Interroger le port-hôte USB à la recherche de changement (connexion ou déconnexion de périphériques).
- Test du processeur TD et saisie des messages d'état de tous les TD tournant sur les 2 SIE.
- Exécution de la fonction « Run » sur tous les pilotes de périphériques actifs listés.

Le test du port-hôte USB à la recherche de changements d'état se fait par l'interrogation d'une variable de changement. Cette variable est positionnée par le PCIH (*Port Change Interrupt Handler*), en cas de changement. Si le port a changé, on a exécution du code d'énumération. En cas de détection d'un nouveau périphérique, le code du framework tente de lui attribuer l'un des pilotes enregistrés. Il existe plusieurs possibilités pour cela. S'il s'agit uniquement de supporter un appareil donné, il suffit de prendre dans une TPL (*Targeted Peripheral List*) les VID

**Tableau 2. Instructions SCSI**

Instruction SCSI	Inquiry	Read Capacity	Read(10)
Byte #0	12 <sub>HEX</sub>	25 <sub>HEX</sub>	28 <sub>HEX</sub>
Byte #1	LUN	LUN	LUN
Byte #2	Page Code	LBA (MSB)	LBA (MSB)
Byte #3	Réservé	LBA	LBA
Byte #4	Allocation Length	LBA	LBA
Byte #5	Réservé	LBA (LSB)	LBA (LSB)
Byte #6	Réservé	Réservé	Réservé
Byte #7	Réservé	Réservé	Transfert Length (MSB)
Byte #8	Réservé	Réservé	Transfert Length (LSB)
Byte #9	Réservé	Réservé	Réservé
Byte #10	Réservé	Réservé	Réservé
Byte #11	Réservé	Réservé	Réservé

(*Vendor ID*) et PID (*Product ID*) du dit périphérique. Les VID et PID d'un nouveau périphérique sont ensuite confrontés à cette liste. L'une des méthodes les plus courantes d'association de périphériques et de pilotes fait appel aux classes d'appareil (*device-class*) et aux classes d'interface. Dans le cas des mémoires de masse, le *device-class* et l'*interface-class* ont tous deux la valeur 8, les sous-classes étant elles à 6. Ceci explique que, lors de l'écriture d'un pilote, ces valeurs sont à entrer dans les champs correspondants (**figure 3**).

## Conception d'une application

Si un concept d'USB enfoui simple doit piloter un lecteur Flash USB, il faut commencer par créer un pilote pour la mémoire de masse baptisé, par exemple, `mass_storage_driver`. Les champs les plus importants de la structure de données sont les champs *Class* et *Subclass*. Le framework utilise les valeurs trouvées à ces endroits pour attribuer au pilote des lecteurs Flash correspondants les mêmes valeurs de classe et de sous-classe. Autre possibilité, entrer des valeurs VID et PID spéciales dans la structure du pilote. On obtient ainsi que seul un appareil possédant ces identificateurs soit démarré après connexion. Il faut ensuite ajouter les fonctions Start, Stop et Run ainsi que les codes de classe pour l'appareil de manière à ce que le pilote puisse être découvert (**figure 4**).

```

CLASS_DRIVER mass_storage_driver =
{
    0x08,          /* uint8 class; */
    0x06,          /* uint8 subclass; */
    0x08,          /* uint8 if_class; */
    0x06,          /* uint8 if_subclass; */
    0x00,          /* uint8 protocol; */
    0x04b4,        /* uint16 vendor_ID; */
    0x4e02,        /* uint16 product_ID; */
    mscperphdvr_start, /* uint16 (*start)(USB_DEVICE *dov); */
    mscperphdvr_stop, /* uint16 (*stop)(void); */
    mscperphdvr_run, /* void (*run)(void); */
    mscperphdvr_ioctl, /* uint16 (*ioctl)(USB_DEVICE *,uint16,uint16,uint16); */
    MSCPERPHDVR_ID /* uint16 id. */
};

```

**Figure 3. Éléments d'un pilote de périphérique.**

```

uint32 ReadFile(USB_DEVICE *dev, uint32 Location)
{
    uint32 Sector, NextSector = 0;
    uint16 Size, Result;
    uint8 Index, Done = 0;
    uint16 I, J;

    for(Index=0; (Index<Volume->sectors_per_cluster) && !Done; Index++)
    {
        /* we have to run through this n times, n being the number of */
        /* sectors per cluster since we are getting passed in the */
        /* cluster number but are reading by sector. */
        /* this function also needs to calculate the absolute location */
        /* on the drive based on the start_sector returned by FindFile. */
        /* note that Volume->first_data_sector points to "sector #2" */
        /* on the drive. Use this when doing the sector math shown */
        /* on the next line */
        Sector = Volume->first_data_sector + Volume->root_dir_sectors +
            (Location-2)*Volume->sectors_per_cluster + Index;

        Result = SendCBW(dev, 0x200, 0x0a, 0x28, WordSwap(Sector), 0x100);

        Delay_mS(1);
        Size = FWX_SECTOR_BUFFER_SIZE;
        Result = usb_recv_bulk_in( dev, MSCPERPH_BULK_IN_PIPE,
            XferBuffer, 5Size );

        BytesRead += 0x200;

        if(BytesRead >= FileSize)
            Done = 1;

        /* send the file data to the external process */
        PutData(XferBuffer, Size);

        Size = 13; /* CSW */
        Result = usb_recv_bulk_in( dev, MSCPERPH_BULK_IN_PIPE,
            XferBuffer, 5Size );
    }

    NextSector = ReadFAT(dev, Location);

    return(NextSector);
}

```

Figure 4. Code ReadFile.

Il reste encore à entrer le nom de la fonction de pilote dans le fichier drvrlist.h.

```
define FWX_DRIVER_LIST { &mass_storage_driver }
```

La fonction de début (*start*) *mass\_storage\_driver* utilise le protocole CBW/CSW pour interroger le périphérique et en déterminer les caractéristiques. Ces fonctions comportent du code qui transmet ces données à la couche application du concept de manière à ce que les couches plus élevées de l'application puissent se charger des différen-

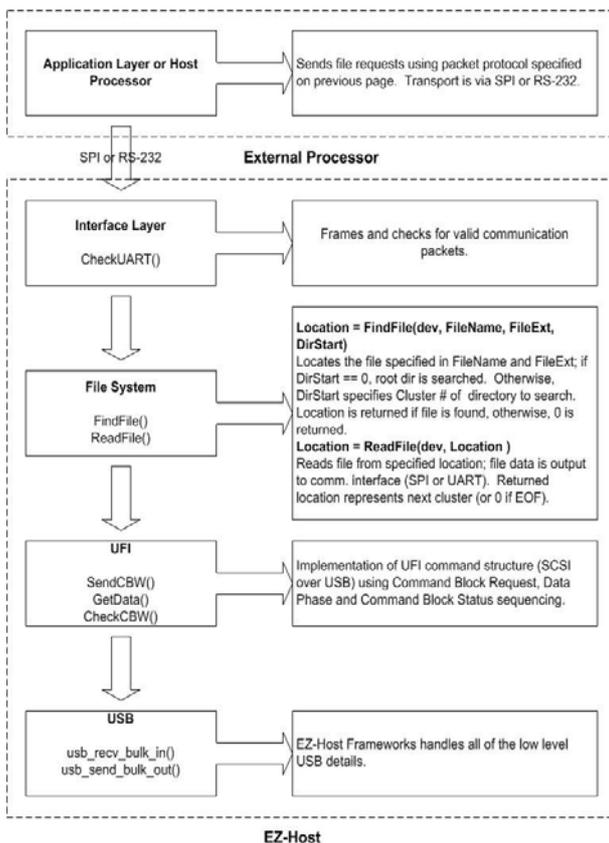


Figure 5. Pile USB-Host simplifiée.

tes tâches de gestion des fichiers. La fonction d'arrêt (*stop*) élimine toutes les opérations ou transferts peut-être encore à effectuer et en informe la couche application lui signalant la déconnexion d'un périphérique. La fonction d'exécution (*run*) sert, enfin, à voir si le code d'application émet l'une ou l'autre instruction.

## Framework lit un lecteur Flash USB

Il faut maintenant écrire du code application qui se traduit par l'exécution de fonctions utiles. Ce dont nous disposons pour l'instant est un pilote ainsi que de la possibilité d'attribuer ce pilote, de par les informations de classe et de sous-classe, à un périphérique donné. Il faut compléter le pilote de code assurant ses fonctions « start » et « run ».

Après son lancement, le pilote doit interroger le périphérique Flash pour recueillir des informations concernant son système de fichiers. Il faut en outre, périodiquement, par le biais d'une fonction d'exécution, voir si l'utilisateur désire lire l'une ou l'autre donnée de fichier.

Pour la saisie des informations concernant le système de fichier, la fonction *start* du pilote s'y prend ainsi :

- Exécution d'une instruction *SCSI Inquiry* pour demander des informations concernant le périphérique connecté
- Exécution d'une instruction *SCSI Read* pour demander des informations sur la taille du périphérique connecté
- Exécution d'une instruction *SCSI Read* (LBA étant à « 0 », pour chercher les données du premier secteur physique du périphérique Flash
- Examen de type des 16 octets, qui, avec un offset de 1BE16, commencent dans le premier secteur en s'aidant d'une *PRE (Partition Record Entry = entrée d'enregistrement de partition)*
- Localisation du *Start LBA*, c'est-à-dire du numéro du cluster comportant le *Boot Record*, en s'aidant de l'information d'enregistrement de partition (PR)
- Exécution d'une instruction *SCSI Read(10)*, dans laquelle la valeur LBA devient la valeur de *Start LBA* pour lire le secteur comportant le *Boot Record*. Ces informations sont entrées dans les informations de volume.

Il est possible, à l'aide des 3 instructions SCSI et des connaissances de la structure du lecteur, d'obtenir les informations données dans le tableau 1. Une fois que l'on dispose de ces informations, il est possible de lire n'importe quel fichier présent sur le lecteur. L'entrée de répertoire d'un fichier comporte le numéro du cluster de début qui, au champ de position (*location*), est transmis à la fonction de lecture de fichier. La fonction « ReadFile » lit chacun des secteurs du cluster et vérifie ensuite dans le FAT si le fichier comporte d'autres clusters. On trouve, en figure 4, le code d'exécution d'une lecture de fichier.

Il est possible de calculer la valeur de secteur à partir de la structure de volume. Pour chaque secteur du cluster on a exécution de la boucle File-Read. Après la lecture de tout un cluster, on cherche, dans le FAT, la valeur Next-Cluster. Pour la communication avec le lecteur, il est fait appel à des CBW et des CSW.

## Application Programming Interface (API)

La **figure 5** illustre la hiérarchie du progiciel de EZ-Host. Il est possible, à un processeur externe ou à une application de communiquer, via RS-232, au plan interface. Il est en outre possible d'écrire du code application qui travaille en API entre le plan interface et les fonctions *FindFile* et *ReadFile*.

Un plan interface est reproduit en dessous. On y a interrogation de l'UART quant à un code opératoire de 1 octet, suivi d'un nom de fichier avec extension et d'un DSL (*Directory Start Location*). Un code opératoire de 1 permet l'exécution d'une fonction *FindFile*, un code de 2 une fonction de *FileRead*.

Les champs *FileName* et *FileExt* du paquet donnent le nom du fichier à chercher ou à lire. Le champ suivant *DirStart* du paquet permet le placement des fichiers soit dans le répertoire-racine (*DirStart* = 0) soit dans un sous-répertoire (*DirStart* correspond au cluster de début du sous-répertoire).

Supposons, à titre d'exemple, qu'il faille, dans le répertoire-racine, lire le fichier *File1*. Le premier paquet UART envoyé au plan interface possède alors un paramètre *FileName* de *File1*, une valeur *DirStart* de 0 et un code opératoire de 2 (*ReadFile*). Le progiciel EZ-Host fournit alors le contenu du fichier *File1*.

Second exemple : on veut lire le fichier *File2* du sous-répertoire *SubDir1*. On commence par constituer un paquet UART comportant le paramètre *FileName* *File2*, un code opératoire 1 (*FileFind*) et une valeur 0 pour *DirStart*. La fonction *FindFile* transmet ensuite à l'application le cluster de début du sous-répertoire *SubDir1*. On a ensuite, émission d'un autre paquet UART comportant cette fois le paramètre *FileName* *File2* et le code opératoire 2 (*FileRead*), la valeur *DirStart* utilisée étant la valeur de *Start-Cluster* du sous-répertoire *SubDir1* fournie par la fonction *FindFile*. Le progiciel de EZ-Host fournit alors le contenu de *File2*.

## Conclusion

Il est possible, avec le concours d'un contrôleur USB-Host enfoui, de doter tout concept de mémoires de masse mobiles de manière à télécharger des données importantes ou d'effectuer, via USB, des mises à niveau (*update*) progicelles.

On peut implémenter un USB-Host enfoui à l'aide d'une unique puce additionnelle, et ce sans logiciel étranger. Certaines de tâches de traitement élémentaires sont prises en compte par un Framework USB-Host enfoui. EZ-Host de Cypress peut effectuer les transferts en masse à pleine vitesse (*full-speed-bulk-transfert*) requis par la communication avec les mémoires de masse. Pour le moment, la taille du code est de l'ordre de 13 Koctets. Il est possible, pour le support de périphériques « spécifiques-application », d'ajouter sans problème le support du système de fichier correspondant.

(040238)

```
uint32 ReadFile(USB_DEVICE *dev, uint32 Location)
{
    uint32 Sector, NextSector = 0;
    uint16 Size, Result;
    uint8 Index, Done = 0;
    uint16 i, j;

    for(Index=0; (Index<Volume->sectors_per_cluster) && !Done; Index++)
    {
        /* we have to run through this n times, n being the number of
        /* sectors per cluster since we are getting passed in the
        /* cluster number but are reading by sector.
        /* this function also needs to calculate the absolute location
        /* on the drive based on the start sector returned by FindFile.
        /* note that Volume->first_data_sector points to "sector #2"
        /* on the drive. Use this when doing the sector math shown
        /* on the next line
        Sector = Volume->first_data_sector + Volume->root_dir_sectors +
        (Location-2)*Volume->sectors_per_cluster + Index;

        Result = SendCBW(dev, 0x200, 0x0a, 0x28, WordSwap(Sector), 0x100);

        Delay_mS(1);
        Size = FWK_SECTOR_BUFFER_SIZE;
        Result = usb_recv_bulk_in(dev, MSCPERPH_BULK_IN_PIPE,
        XferBuffer, &Size);

        BytesRead += 0x200;
        if(BytesRead >= FileSize)
            Done = 1;

        /* send the file data to the external process */
        PutData(XferBuffer, Size);

        Size = 13; /* CSW */
        Result = usb_recv_bulk_in(dev, MSCPERPH_BULK_IN_PIPE,
        XferBuffer, &Size);
    }

    NextSector = ReadFAT(dev, Location);

    return(NextSector);
}
```

Figure 6. Exemple d'implémentation du plan interface.

## Informations additionnelles

On trouvera d'autres informations concernant le système de fichiers FAT sous :

<http://www.microsoft.com/hwdev/download/hardware/fatgen103.pdf>

Pour d'autres informations concernant la spécification UFI Command voir ici :

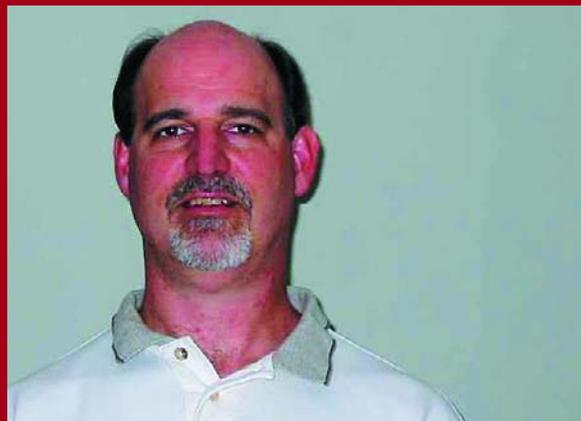
[http://www.usb.org/developers/devclass\\_docs/usb-mass-ufi10.pdf](http://www.usb.org/developers/devclass_docs/usb-mass-ufi10.pdf)

Le lien ci-après donne des informations additionnelles au sujet de USB Mass Storage Class Bulk Only Transport Specification :

[http://www.usb.org/developers/devclass\\_docs/usb-massbulk\\_10.pdf](http://www.usb.org/developers/devclass_docs/usb-massbulk_10.pdf)

## Mark Schultz

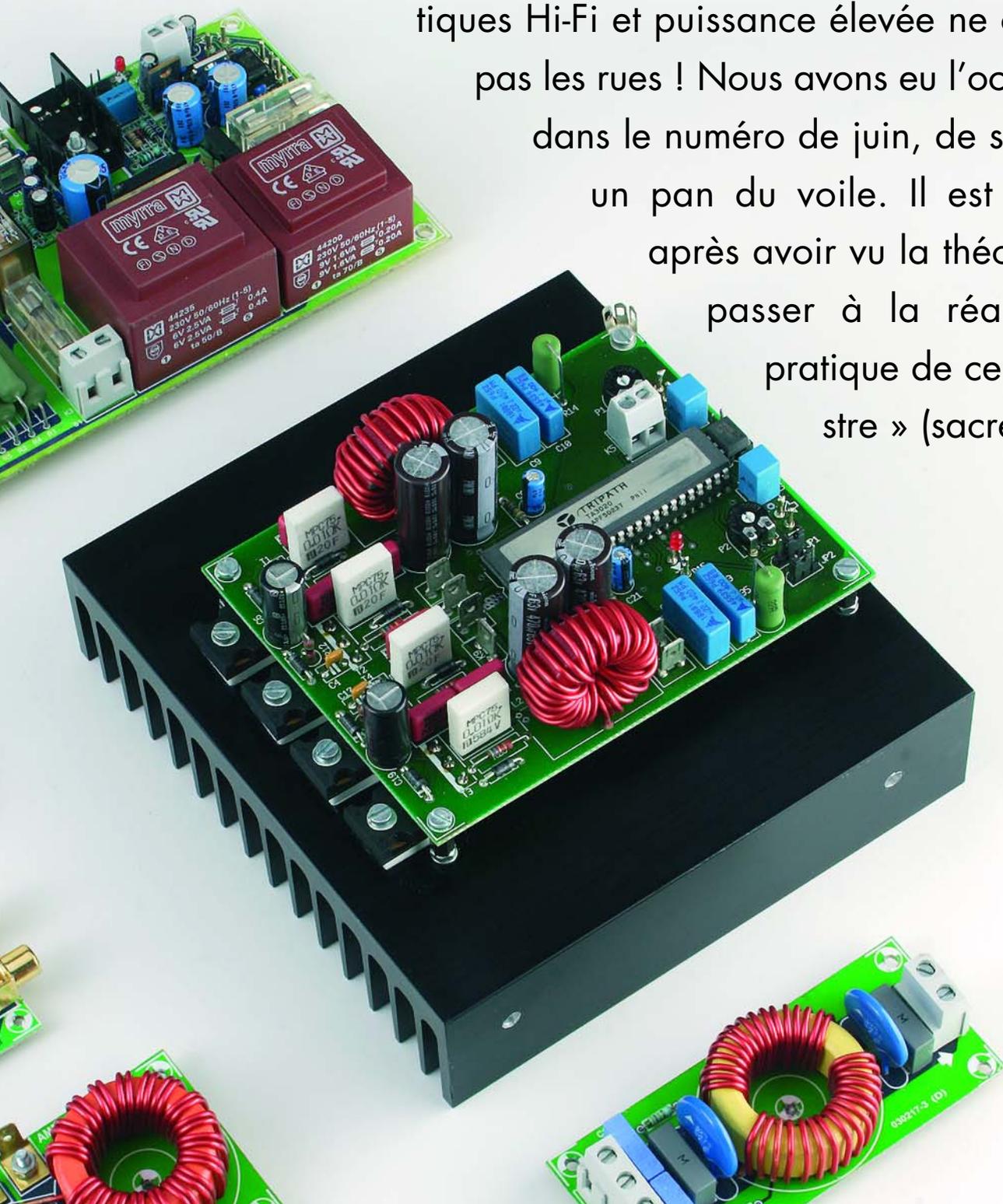
est Senior Customer Design Center Engineer de la société Cypress Semiconductor. Après avoir terminé avec succès des études d'informatic au College de New Jersey comme Bachelor of Science. Avant d'arriver chez Cypress, il a travaillé chez Princeton Instruments dans la conception de cartes d'interface de bus pour appareils photos à CCD. Schultz, un membre de l'IEEE possède une expérience de plus de 20 ans dans l'industrie de l'électronique.



# Clarity 2x300W en

Ton Giesberts

Les étages de sortie MLI associant caractéristiques Hi-Fi et puissance élevée ne courent pas les rues ! Nous avons eu l'occasion, dans le numéro de juin, de soulever un pan du voile. Il est temps, après avoir vu la théorie, de passer à la réalisation pratique de ce « monstre » (sacré ?).



# classe T (II)

## Réalisation de la platine de l'ampli

L'amplificateur MLI évoqué quelques lignes plus haut, met à contribution le circuit de commande TA3020 de chez Tripath. Si l'acronyme MLI (**M**odulation en **L**argeur d'**I**mpulsion) peut faire naître l'une ou l'autre association négative, cela est totalement injustifié. Les spécifications sont superbes et l'appareil n'a rien à envier à nombre de ses congénères du segment commercial du haut de gamme.

De manière à simplifier au maximum la réalisation par nos lecteurs, nous proposons, pour le présent projet, une platine déjà dotée de tous les composants CMS, et ceci à un prix très abordable ! Rappelons-le, une platine comporte un amplificateur stéréo complet de 2 x 300 W.

### Composants spéciaux

Permettez-nous, avant que vous ne vous mettiez à la recherche des composants nécessaires, d'insister sur le fait que l'on ne réalise pas un amplificateur de 2 fois 300 W à l'aide de composants récupérés dans son tiroir de surplus ! Même l'alimentation doit, à de tels niveaux de puissance, répondre à un cahier des charges sévère. Nous avons de plus affaire à un amplificateur à découpage. Le dessin de la platine et la qualité des composants sont ici des facteurs primordiaux. Il est partant nécessaire, pour éviter des problèmes, que nous signalions, avant de nous lancer dans la réalisation de l'amplificateur, les exigences auxquelles sont confrontés les différents composants. Il est possible de trouver chez Farnell la plupart des composants, sachant qu'il existe certaines sources, hors de l'Hexagone malheureusement, proposant cet amplificateur en kit. Certains des composants doivent impérativement être du type CMS (à **M**ontage en **S**urface) en raison de leurs caractéristiques et des trajets de signal raccourcis qu'ils permettent. Cela ne devrait cependant pas vous poser de problème

vu que les CMS sont déjà montés sur la platine. De même, les noyaux ferrite des selfs (2) sont également fournis avec la platine. À noter que le bobinage d'un fil de cuivre émaillé de 1,5 mm de diamètre requiert des doigts solides, point auquel nous reviendrons.

### Découplage

Le point requérant le plus d'attention sur cet amplificateur sont les parasites qui peuvent naître du découpage de courants importants. Le dessin de la platine est tel que les pistes véhiculant des courants élevés sont découplées le plus possible du reste du circuit. De plus, il a été effectué un découplage local des tensions d'alimentation de manière à éviter le mieux possible la naissance de boucles dans les différents sous-ensembles. Les composants les plus importants dans ce contexte sont les condensateurs C5, C18, C32, C33, C36 et C37 au niveau des transistors de sortie. Les condensateurs de découplage utilisés sont du type MKT à tension de service de 250 V vu qu'ils supportent mieux les courants de commutation élevés.

C6 et C19 méritent une attention particulière. Il est important que ces 2 condensateurs électrochimiques aient la self-induction et la résistance de perte la plus faible possible ainsi qu'une bonne résistance aux effets thermiques.

Le réseau d'amortissement (*snubber*) constitué par les paires C4/R12 et C17/R33, sert à éviter des suroscillations (*overshoots*) HF. Pour gagner de la place, les résistances R12/R33 sont montées verticalement. Veillez, lors du montage de ces résistances, à réduire le plus possible leur boucle en vue de limiter au strict minimum la self-induction parasite. Nous avons prévu un diamètre plus important vu qu'il s'agit de versions 1 W (cf. **figure 1**). Les condensateurs C4 et C17 sont des céramiques

200 V. La tension maximale aux bornes de ces condensateurs peut en effet atteindre quasiment la pleine tension d'alimentation (près de 110 V entre le positif et le négatif de l'alimentation) voire même plus en cas d'ondulations !

### Suppression de pics inductifs

En raison des dimensions physiques des composants on se trouve inévitablement confronté à des self-inductions parasites et autres ondulations. Il est possible d'en limiter les effets, ceux des crêtes inductives des selfs des filtres de sortie en particulier par l'utilisation de diodes Schottky de pinçage à temps de récupération ultra-rapide. C'est là la fonction de D3, D4, D6, D7, D10, D11, D13 et D14. Pour D3, D4, D10 et D11 nous avons monté des diodes en boîtier DO-15 (MUR120) sur la platine. D6, D7, D13 et D14 sont des CMS en boîtier DO-214AA (MURS120T3) prises entre la source et le drain côté « pistes » sur le dessous du circuit intégré (**figure 2**). Ces 2 types de diodes supportent une charge allant jusqu'à 1 A et 200 V, leur temps de récupération n'étant que de 25 ns.

### Pilotage des FETMOS

Lors du dessin nous avons particulièrement fait attention aux risques de boucles dans les lignes allant des sorties de commande (*driver*) du circuit intégré vers les grilles des FETMOS. Il est important de limiter au maximum ces boucles. La boucle de commande du MOSFET supérieur de chaque canal prend la forme du pilote HO, de la résistance de grille, de la capacité grille/source et de la connexion de commande (HO1COM/HO2COM).

Dans le cas du FETMOS inférieur, cette boucle devient le pilote LO, la résistance de grille, de la capacité

# Liste des composants

Cette liste ne reprend aucun des composants CMS pré-montés. Vous pouvez, si vous le désirez, télécharger la liste comportant les composants pré-montés depuis notre site : [www.elektor.fr](http://www.elektor.fr). Pour les composants délicats nous donnons l'une des sources connues.

## Résistances :

R6,R11,R27,R32 = 0Ω01 au pas de 9 mm, MPC75-E01 (H.O.D, Bürklin)  
 R7,R9,R28,R30 = 5Ω6/1 W au pas de 15 mm (max.), PRO1 BCComponents (Farnell 337-584, 10+)  
 R12,R33 = 15 Ω/1 W PRO1, BCComponents (Farnell 337-638, 10+)  
 R13,R34 = 240 Ω  
 R14,R35 = 22 Ω/5 W (vertical)  
 P1,P2 = ajustable 10 kΩ

## Condensateurs :

C1,C14 = 3μF3/50 V MKT au pas de 5/7,5 mm

C4,C17 = 220 pF/200 V COG au pas de 5 mm, dipped radial multilayer ceramic, Multicomp (Farnell 747-075, 1+)  
 C5,C18,C32,C33,C36,C37 = 100 nF/250 V au pas de 7,5/10 mm, 1 x L = 6 x 13 mm (max.), Wima MKS4 (Farnell 148-888, 1+)  
 C6,C19 = 47 μF/160 V radial au pas de 5 mm, diamètre 10 mm (max.), 105 °C, Panasonic EEUED2C470 (Farnell 383-6400, 1+)  
 C8,C21,C38 = 47 μF/25 V radial  
 C9,C22 = 220 nF/400 V MKP au pas de 15 mm, BxL = 8,5 x 18 mm (max.), Epcos B32652-A4224-J (Farnell 400-

grille/source et de la connexion de masse (LO1COM/LO2COM). Les ondulations naissant de la commutation des FETMOS sont limitées par les résistances prises dans les lignes de grille. Il faut inévitablement accepter un compromis entre le retard naissant de la mise en blocage de l'un des FETMOS et de la mise en conduction de l'autre. Nous recommandons, sur le présent amplificateur, des résistances de limitation de 5,6 Ω. Cela se traduit effectivement malheureusement par la dissipation dans une résistance d'une certaine puissance qui l'aurait été sinon dans le circuit de commande.

Un autre compromis se situe au niveau de la charge maximale de l'amplificateur et du choix des FETMOS qui en résulte. Malheureusement, le courant de drain et la capacité grille/source sont intimement liés ( $C_{iss}$  max 3 800 pF). La self-induction parasite de la grille joue, elle aussi, un rôle : plus elle est faible, plus rapides sont l'établissement et la disparition de la charge au niveau de la grille. Nous avons pris une diode en parallèle sur les résistances de grille de manière à accélérer la mise en blocage. Il s'agit à nouveau des diodes à récupération ultra-rapide en boîtier « standard » (MUR120). De par leurs dimensions

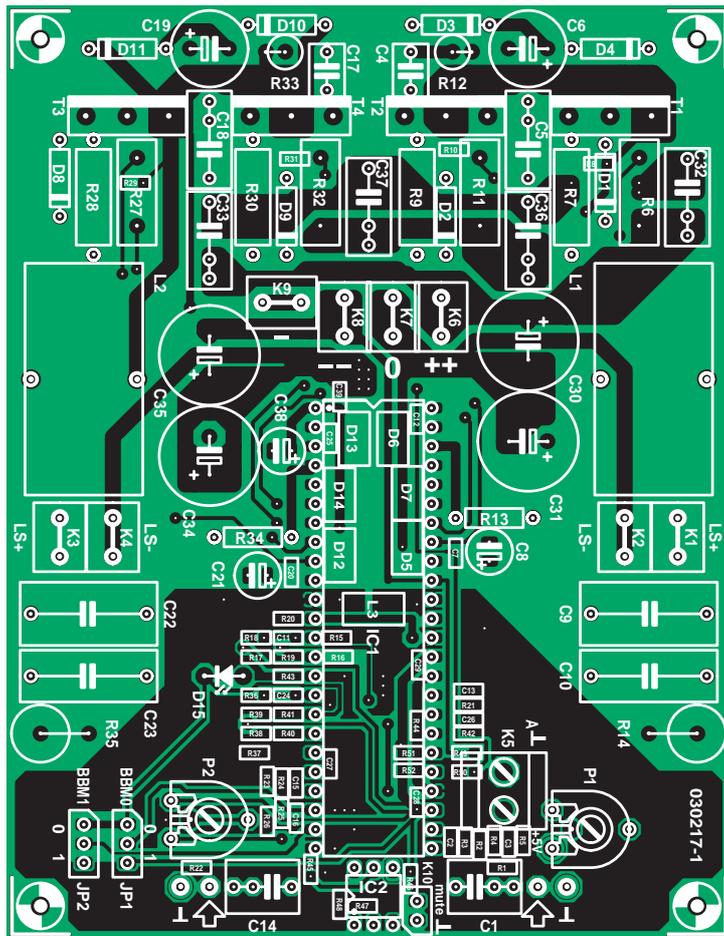
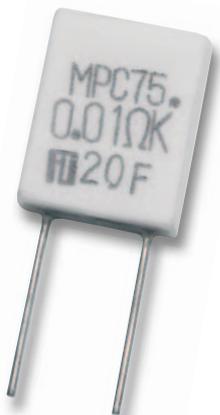


Figure 1. Implantation des composants (dessus de la platine de l'amplificateur).



3755, 1+)  
 C10, C23 = 100 nF/400 V MKP au pas de 15 mm, l x L = 7 x 18 mm (max.), Epcos B32652-A4104-J (Farnell 400-3731, 1+)  
 C30, C31, C34, C35 = 470 µF/63 V radial au pas de 5 mm, diamètre 13 mm (max.), 105 °C, Nichicon UPM1J471MHH (Farnell 415-3030, 5+)

#### Selfs :

L1, L2 = 11 µH3, 29 spires de fil de cuivre émaillé de 1,5 mm sur noyau T106-2 Micrometals (noyaux fournis avec la platine, bobinage à faire soi-même)

#### Semi-conducteurs :

D1 à D4, D8 à D11 = MUR120  
 1 A/200 V ultra fast ON Semiconductor (Farnell 930-994, 1+)  
 D15 = LED rouge haut rendement  
 T1 à T4 = STW38NB20 boîtier TO-247 200 V/38 A ST (Farnell 323-9408, 1+)  
 IC1 = TA3020 (Tripath)  
 IC2 = CNY17-2

#### Divers :

JP1, JP2 = embase autosécable à 3 contacts + cavalier  
 K1 à K4, K6 à K9 = connecteur languette encartable vertical  
 K5 = bornier encartable à 2 contacts au

pas de 5 mm  
 K10 = embase autosécable à 2 contacts  
 Support 48 broches (support DIP à contacts en tulipe écartement des rangées 0,6" (15,24 mm) Farnell 416-8653, 1+)  
 4 x isolation céramique AOS220SL, Fischer 14 x 18 mm, épaisseur 4,5 mm (Huijzer-Avera)  
 radiateur 0,6 °C/W, 160 x 150 mm Marston 938SP01500A200 (Farnell 526-794, 1+)  
 Platine EPS 030217-91. Tous les CMS y sont déjà montés + noyaux destinés à L1, L2 disponible sur notre site

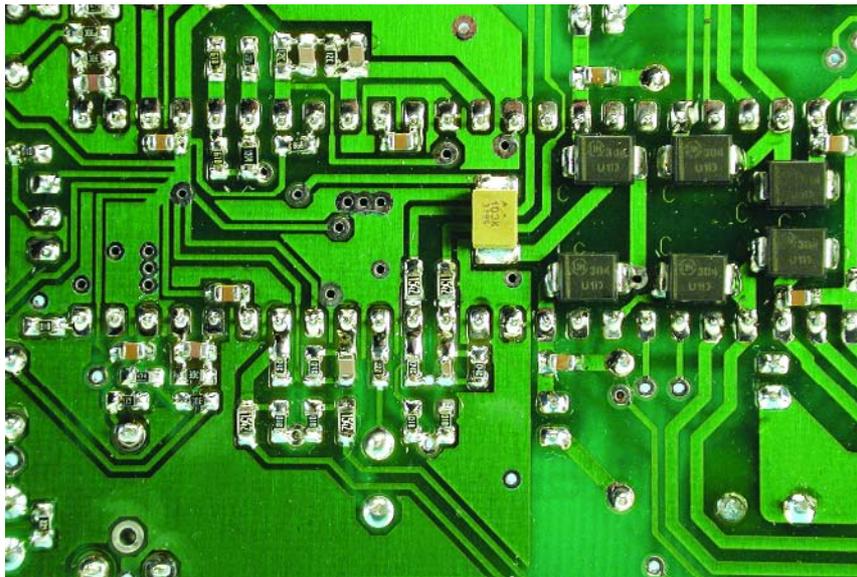


Figure 2. Le dessous de la platine. Les CMS y sont déjà montés.

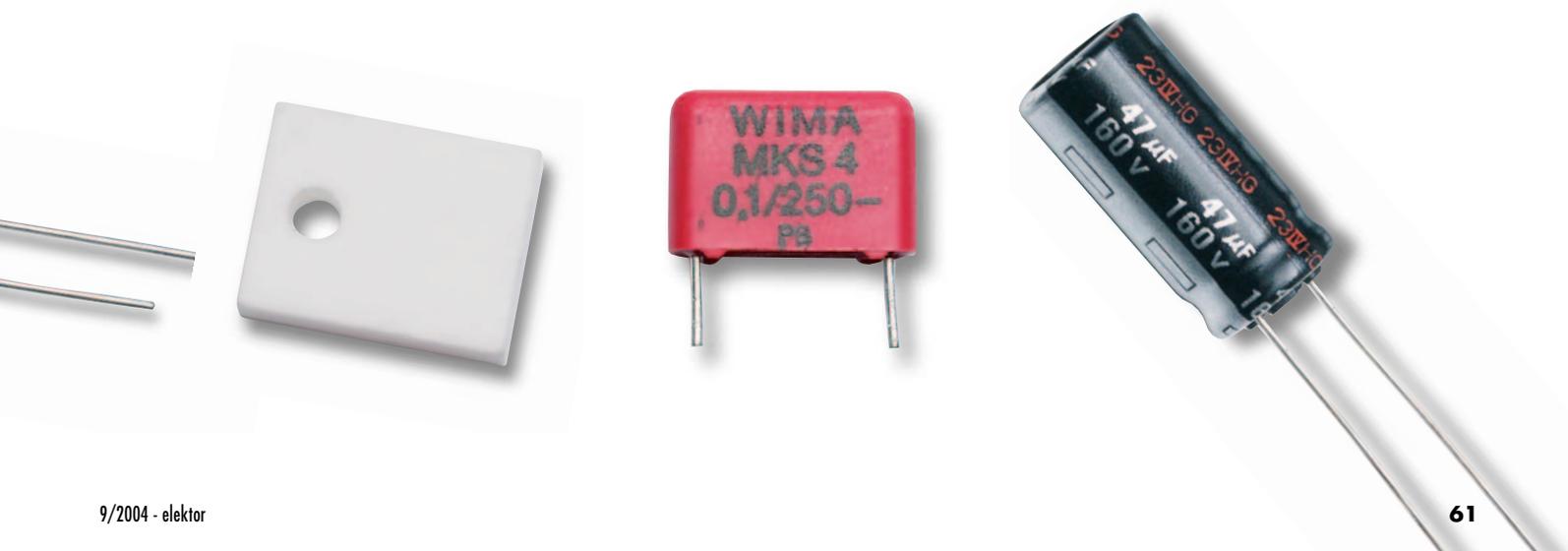
plus importantes elles permettent de « ponter » certaines pistes plus larges. L'inconvénient de cette approche est une augmentation de la dissipation au niveau du circuit de commande. En raison de la dissipation, les résistances de grille sont des 1 W. Nous avons opté pour des variantes extrêmement compactes de la série PR-01

de BCComponents. De par la constitution des résistances à film métal, dite *helical groove*, on se trouve, par définition, confronté à une self-induction parasite, mais parallèlement l'impédance de ces types reste quand même relativement constante jusqu'à 10 MHz. On pourrait, ce serait une bonne alternative, utiliser des résistan-

ces compound qui présentent, de par leur construction, une self-induction bien plus faible. Nous avons, sur la platine, prévu cette option.

### Alimentation et masse

Les connexions de l'alimentation principale se fait, au niveau de la platine, à l'aide de languettes plates (de type automobile). Elles permettent le passage de courants très importants et partant une liaison aisée de la platine à l'alimentation. Nous avons pris, en parallèle sur ces connexions de l'alimentation, des condensateurs électrochimiques spéciaux destinés à découpler les pics de courant HF les plus dramatiques. Nous avons essayé de donner aux dites connexions une forme en étoile mais leur forme s'est inévitablement aplatie de manière à éviter d'accroître inutilement la distance entre le circuit intégré et les FETMOS. Les condensateurs de découplage C30, C31, C34 en C35 proviennent de la série de Nichicon qui combinent un excellent rapport entre la capacité et les dimensions et des résistance-série et self-induction faibles. Le composant optimal est le condensateur de 470 µF



# Connexions Kelvin

En 3 endroits sur l'amplificateur nous utilisons des connexions Kelvin (connues également, en fonction de l'application, sous la dénomination de mesure 4 points) en vue d'éliminer l'influence de diverses résistances de transfert et des self-inductions parasites. Il s'agit de connexions réalisées sans dérivation additionnelles permettant de mesurer avec précision la tension aux bornes de certains des composants. Il s'agit ici de la tension aux bornes des résistances de shunt servant à la détection de surintensité, à la contre-réaction des connexions des enceintes et à la masse d'entrée.

Il est évident que la détection de courant

requiert une mesure 4 points. La résistance de shunt vaut 10 mΩ seulement. Les lignes de détection de surintensité sont dérivées directement aux bornes des résistances. Les dites résistances sont des modèles verticaux en boîtier céramique (MPC75 de Fukushima Futaba Electric Co. Ltd.) ne possédant pratiquement pas de self-induction. S'il vous est impossible de les trouver, il faudra essayer de trouver des résistances à faible induction de modèle similaire. Il n'est pas possible d'utiliser des résistances axiales sachant que le simple fait de les monter verticalement produit une self-induction trop importante. Les lignes de signal de

chaque paire sont amenées serrées vers les points de soudure du circuit intégré. Ceci vaut également pour la contre-réaction des lignes des enceintes (vers les résistances correspondantes). Au niveau de la masse d'entrée cela signifie que toutes les connexions de masse sont amenées, par le biais de leur propre piste, à la masse commune (le « 0 » de l'alimentation +5 V analogique) du circuit intégré, ce qui est bien visible côté « pistes » de la platine de l'amplificateur. Vu le nombre de pistes se rencontrant, ce point nodal prend inévitablement un certain embonpoint.

à tension de service de 63 V. C'est en fait lui qui détermine la tension d'alimentation maximale admissible dans l'amplificateur de sortie.

Depuis le point en étoile, la piste va à la self, L3, chargée d'assurer la séparation entre les masses analogique et numérique du circuit intégré. Cette self est une CMS (boîtier 1812A) montée côté « pistes » du circuit intégré (figure 2). Il s'agit en l'occurrence d'une inductance de la série SIMID de Epcos d'une valeur de 10 μH. Cette self présente une résistance de perte inférieure à 1 Ω et peut supporter un courant supérieur de 300 mA.

On dérive également depuis le point en étoile les lignes vers les enceintes de manière à ce que les courants des enceintes retournent le plus directement possible vers l'alimentation principale. Cela évite toute influence sur d'autres sous-ensembles de l'amplificateur.

Les plans de masse présents sur la platine servent uniquement à une protection contre les parasites. Ils sont tous reliés au point AGND de l'alimentation analogique (broche 28). Il n'y a pas d'autre liaison dérivée des plans de masse vers l'amplificateur. L'alimentation analogique de +5 V est elle aussi reliée, au travers d'un point en étoile, aux potentiomètres, cavaliers et ponts de résistances servant au paramétrage des modulateurs. On retrouve tout ceci au niveau du dessin des pistes côté « composants » de la figure 1.

## Construction

En dépit de tout le soin pris à la conception du dessin des pistes, le niveau de parasites produit par l'étage de puissance reste relativement élevé.

Cela signifie qu'il va falloir prendre des mesures additionnelles pour minimiser les effets négatifs sur la partie analogique, ce que nous réalisons en réduisant au strict minimum la surface nécessaire à la mise en place des composants passifs. La seule manière d'y arriver est de n'utiliser que des CMS et de la placer le plus près possible du circuit intégré sur le côté « pistes » (cf. figure 2). Les seuls composants ne respectant pas cette règle sont les 2 potentiomètres de l'offset en CC et les condensateurs d'entrée. C3 et C16 éliminent par filtrage d'éventuelles parasites que pourraient capter les potentiomètres. Les parasites captés par C1 et C14 sont filtrés par C2 et C15. Certains CMS doivent avoir, en raison de leur place sur le schéma, une tension de service supérieure à 50 V. Il s'agit de R15, R16, R36, R37 et R51. R8, R10, R29, R31 et certains des condensateurs de découplage sont des CMS sachant que cela cadrerait mieux tant d'un point de vue fonctionnel que d'encombrement. Les îlots de soudage ont été conçus de manière à pouvoir utiliser des boîtiers tant 0805 que 0603 pour tous les condensateurs et résistances CMS, ce que ne manqueront pas d'apprécier ceux qui tiennent à souder leurs propres CMS (bien qu'il soit possible d'acquérir la platine dotée de ces CMS) en vue de modifier la sensibilité ou, par exemple, changer la tension d'alimentation. Ceci explique que leur taille soit supérieure à la taille standard pour les opérations de soudure à la vague. Il est possible ainsi, lors de la soudure, de placer sur l'îlot la pointe d'un fer à souder très fin, ce qui simplifie très notablement le soudage des CMS.

Les condensateurs des filtres de sortie sont des polypropylènes 400 V qui

conviennent excellemment aux applications ayant affaire à des charges impulsives extrêmes. Nous avons utilisé ici encore une version compacte. Le pas de ces condensateurs est de 15 mm « seulement » de manière à réduire la self-induction parasite. Il y a suffisamment de place pour de variantes plus larges au cas où vous choisiriez un autre type (polyester par exemple ou un autre fabricant).

## Bobinage

Le bobinage de la self de sortie n'a rien de bien sorcier mais cette opération requiert le respect d'une méthode. Le fil possède une épaisseur de 1,5 mm. Il n'est pas possible, sur le noyau utilisé, de placer les 29 spires en une seule couche. De manière à réduire la capacité propre de la self, on procède à un bobinage « progressif » de quelque 7 spires. Ceci signifie qu'après les 3 premières spires, la quatrième est bobinée sur la 3<sup>ème</sup>, la 5<sup>ème</sup> reprend place à côté de la 3<sup>ème</sup> sur le noyau, ainsi que les spires 6 et 7, la 8<sup>ème</sup> reprenant place par-dessus la spire 7, la 9<sup>ème</sup> étant à nouveau placée à côté de la 7<sup>ème</sup> et ainsi de suite (cf. figure 3). La section relativement importante du fil ne simplifie pas cette opération. Il se peut, en fonction de la qualité du bobinage (bien serré) il peut se faire qu'il vous faille placer une spire par-dessus une autre plus tôt que prévu.

## TA3020 et transistors de puissance

La plupart d'entre vous n'aimeront sans doute pas souder le circuit intégré directement sur la platine; nous

# Platine avec composants pré-montés

En raison du caractère spécial de cet amplificateur (les fréquences de commutation élevées en particulier) le choix des composants présente une importance capitale et l'utilisation de composants CMS en divers endroits est incontournable.

La majorité des réalisateurs d'amplifica-

teurs audio n'ont pas ou peu d'expérience de soudage de ces composants miniatures, ce qui explique que nous ayons choisi de vous proposer cette platine déjà dotée de ses CMS. Il vous suffira de monter le circuit intégré et les composants « ordinaires ». Nous proposons

en outre les noyaux des 2 selfs de sortie en raison de leur matériau spécifique. Le prix de l'ensemble n'est que de 49 E ! Comme il s'agit d'une version bi-canal, il vous suffira d'un seul circuit imprimé pour réaliser un amplificateur stéréophonique !

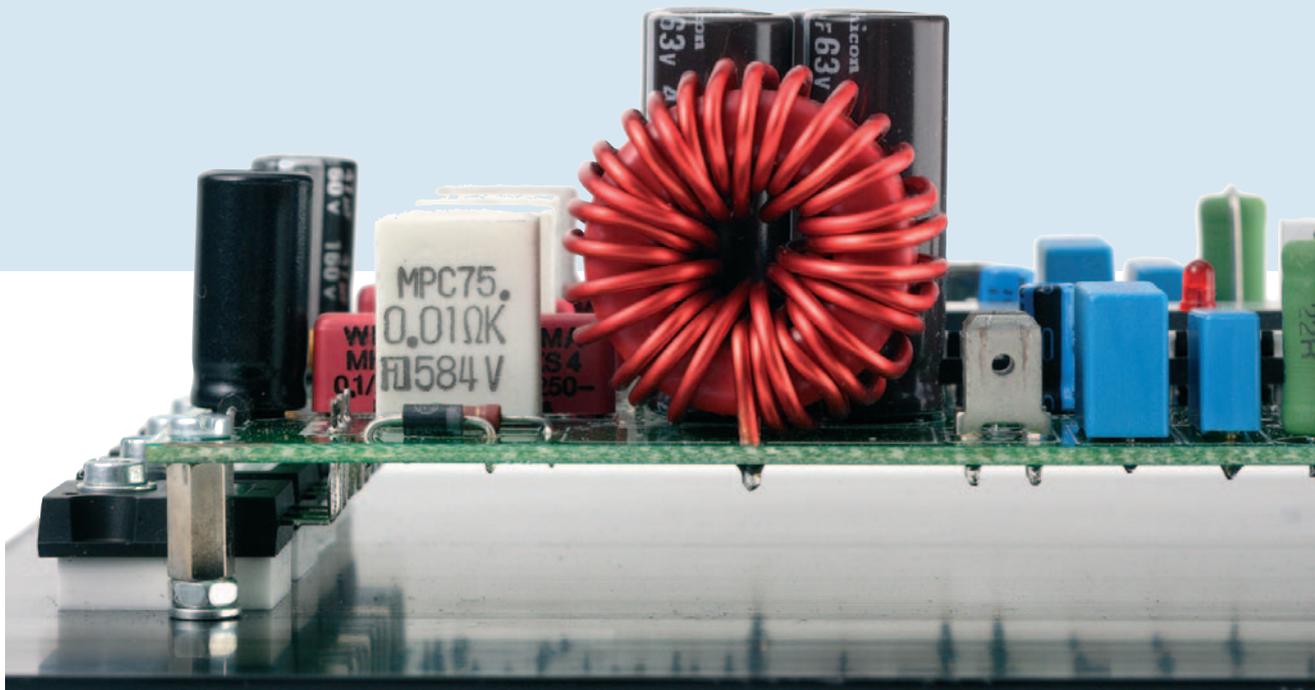


Figure 3. Certaines de spires des selfs de sortie viennent se superposer à d'autres. Les transistors de puissance sont montés par le dessous de la platine et isolés du radiateur à l'aide de plaquettes spéciales.

avons partant opté pour un support à 48 contacts d'excellente qualité à contacts en tulipe dorés à 30 microns supportant un courant jusqu'à 3 A.

Un détail très important du montage des transistors de puissance concerne le matériau servant à l'isolation électrique. La surface de dissipation des transistors étant reliée au drain (T2 et T4), cette caractéristique combinée à des plaquettes de mica, de caoutchouc ou de mousse siliconé voire de kapton, constituerait une capacité trop importante par rapport au radiateur relié lui à la masse.

Nous avons essayé les différentes options et avons constaté, à pleine modulation, la circulation de courants parasites importants impossibles à éliminer. La seule solution acceptable est l'utilisation de matériau céramique ( $Al_2O_3$ ) de quelques millimètres d'épaisseur. Les plaquettes que nous avons utilisées (AOS220SL de Fischer) ont une épaisseur de 4,5 mm, étant en fait prévues pour des boîtiers TO-220 et non pas pour des TO-247 plus grands.

Bien que les plaquettes soient légèrement trop petites, elles recouvrent parfaitement la surface métallique pour le refroidissement du transistor. Cette technique permet en outre d'avoir une capacité parasite extrêmement faible. En ce qui concerne le radiateur nous avons opté pour un exemplaire d'une surface suffisante pouvant être monté parallèlement à la platine. Le type de Marson choisi a une largeur de 160 mm et une hauteur de 150 mm, ce qui laisse un petit jeu sur les bords. On pourra, dans le dos de 10 mm, d'épaisseur percer 8 orifices au filetage de 3 mm servant à la fixation de la platine et des 4 transistors de puissance. Il est recommandé de commencer par centrer la platine et de marquer les 4 orifices des coins. On replie ensuite les pattes des transistors de puissance à l'endroit précis où elles deviennent plus fines, ce qui permettra de déterminer les points de fixation des transistors. Les entretoises (version métal filetées) servant à la fixation de la platine au radiateur, pourront avoir une lon-

gueur de 10 mm environ. Il est fort probable que leur filet soit trop long pour être vissé totalement dans l'orifice percé dans le radiateur. On pourra dans ce cas-là placer d'abord un écrou et une rondelle sur le filet avant de le visser dans le radiateur. La hauteur de la platine est alors parfaite et les pattes des transistors de puissance tombent quasiment d'eux-mêmes dans les orifices prévus à leur intention présent sur la platine.

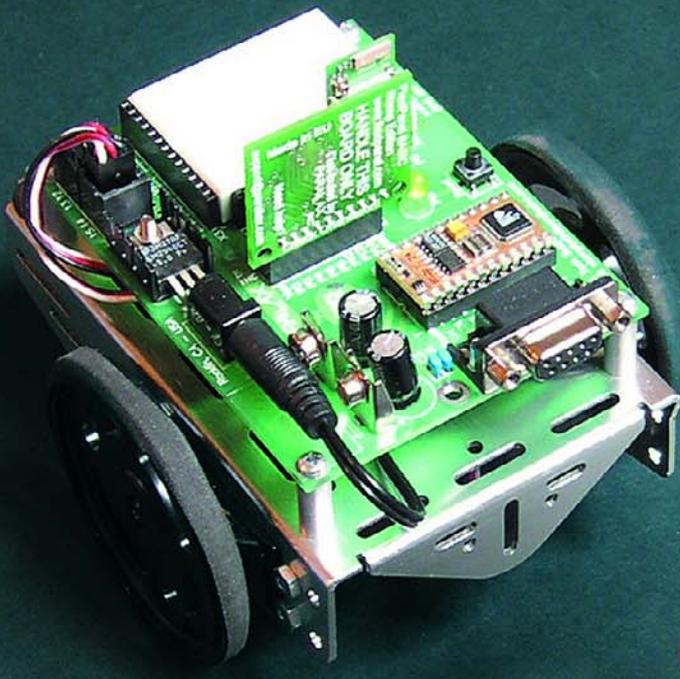
## Le mois prochain

Il nous reste à parler de l'alimentation et du plan de câblage. Nous aborderons ces points dans l'article du mois prochain. Nous vous proposerons aussi quelques résultats de mesure d'un exemplaire fonctionnel de Clarity. D'autres aspects tels que filtres d'entrée et de sortie et les problèmes de CEM feront l'objet d'un article séparé publié lui aussi dans le prochain numéro.

(030217-2)

# Télécommande B

Richard Hoptroff



Bluetooth est un immense succès commercial, mais jusqu'à présent les amateurs et les concepteurs n'avaient pas accès à des modules pré-programmés pour ce superbe nouveau moyen de communication. La raison en est simple : les constructeurs préfèrent traiter leurs produits comme des boîtes noires et ne s'intéressent qu'à leurs lignes de production intensives en Extrême-Orient ou en Chine.

# Bluetooth depuis votre PDA ou votre téléphone portable

Nous opposons à cette attitude la publication de l'article suivant, rédigé selon la devise : des modules Bluetooth pour tout le monde ! Un robot pisteur fournit la réponse.

Compulsez n'importe quel magazine d'électronique – professionnel ou amateur – et vous verrez un large éventail de platines mono-cartes pour ordinateurs ou micro-contrôleurs. Pour de nombreuses applications, elles rendent le développement de produits beaucoup plus simple qu'il y a, par exemple, 5 ou 10 ans. Rajoutez quelques composants auxiliaires et un panneau de commande, écrivez le programme informatique et c'est terminé. Ce qui rend le processus plus simple, c'est que la carte est programmable, permettant à un composant disponible dans le commerce d'être affecté à de nombreuses tâches. Ce concept peut-il être poussé encore plus loin ? Il faudra toujours inclure des composants auxiliaires dans n'importe quel produit, mais quid du panneau de commande (interface utilisateur) ? Ne pourrait-on utiliser un composant programmable du commerce pour remplacer le panneau de commande livré sur des unités comme les PDA, les téléphones GSM, ou aussi sur des systèmes micro-contrôleurs déjà montés ?

Le module « FlexiPanel BASIC Stamp Edition » de FlexiPanel Ltd pourrait fournir la réponse. Dans cet article, nous nous intéresserons à la version Bluetooth du FlexiPanel, cf. l'encadré « Brochage de l'unité ». En utilisant la liaison radio Bluetooth (à 2,4 GHz), le module demande à une unité située à une distance adéquate – disons un téléphone portable, un ordinateur bloc-note, PDA ou autre ordinateur de poche – de créer le panneau de commande requis (ou, si vous préférez l'interface utilisateur) sur son écran. Le module dispose d'une radio de classe 1, de sorte que l'unité mobile peut être située jusqu'à 100 m. Il fonc-

tionne dans des niveaux TTL (logique transistor-transistor), et on nous a informé qu'une version autonome RS-232 sera aussi bientôt en production. Un utilisateur placé dans le rayon d'action du module FlexiPanel-Bluetooth peut se connecter à tout moment à l'appareil en utilisant n'importe quelle unité compatible Bluetooth. Cette dernière affichera le panneau de commande demandé, mais son apparence peut varier selon l'unité utilisée. Nous verrons quelques exemples par la suite. Le logiciel situé sur les unités mobiles est le même pour chaque application et n'a pas besoin d'être personnalisé ou réinstallé. Il est téléchargeable gratuitement depuis le site [www.flexipanel.com](http://www.flexipanel.com). Au moment où nous rédigeons cet article, les logiciels Pocket PC, Windows PC et Smartphones (par exemple le SPV E200 d'Orange) sont supportés. Le logiciel pour le système d'exploitation Palm OS et les téléphones Java supportant le standard JABWT (par exemple le Nokia 6600 et le Sony Ericsson P900) est lui aussi disponible.

## Projets avec commande Bluetooth

Ne nous laissons pas distraire par des technologies nouvelles et accrocheuses comme celles des téléphones GSM et autres PDA compatibles Bluetooth. Avec un exemple, nous allons démontrer que des applications, suffisamment simples pour être à la portée de débutants relatifs, peuvent être développées pour la combinaison FlexiPanel-Bluetooth. L'exemple, un simple robot pisteur, utilise une version spécialement adaptée de la platine **BoE** (*Board of Education*) d'Elektor, publiée à l'origine dans le numéro de septembre 1999 et la carte BS2p BASIC Stamp de Parallax, de réputation mondiale. Parallax Inc., qui fournit la carte Basic Stamp, distribue aussi le module FlexiPanel via ses

distributeurs agréés (en F : Selectronic). Au cas où vous ne le sauriez pas, la carte BASIC Stamp peut être programmée en langage Basic sur n'importe quel PC utilisant un câble série. Le même lien est utilisé pour programmer le panneau de commande dans le module FlexiPanel-Bluetooth utilisé comme partie du projet : un pilote de robot avec relèvement de route.

Les programmes Basic et les fichiers de description du FlexiPanel sont disponibles chez Elektor en téléchargement gratuit. Les deux autres projets sont un système de contrôle d'accès et un enregistreur de températures, qui ne sont pas examinés dans cet article.

## BoE modifiée

Le schéma du circuit de la BoE (BoE = *Board of Education*, platine d'éducation) est présenté en **figure 1**. Les « cerveaux » sont une carte BS2p (BASIC Stamp 2) insérée dans le support à 24 voies. Comme son prédécesseur de 1999, cette BoE dispose d'un connecteur RS-232 (K2) et d'une zone de prototypage enclose entre K7, K3, K6, K8, K4 et K5. On utilisera ultérieurement cette zone pour y installer un couple de composants dont le robot a besoin pour pouvoir vous indiquer sa position.

Le module FlexiPanel-Bluetooth est branché directement dans son support dédié, K9.

La BoE a son propre régulateur de tension, IC2, qui, lorsqu'il est utilisé seul (par exemple, durant les sessions de programmation), peut être alimenté par une batterie de 9 volts. Les programmes utilitaires et l'éditeur de Basic sont disponibles chez Parallax Inc., cherchez « Board of Education » sur leur site [www.parallax.com](http://www.parallax.com) et vous trouverez une masse d'informations (gratuites).

Le plan de montage des composants de la nouvelle BoE est présenté en **figure 2**. Des braves à cette occasion pour la platine simple face.

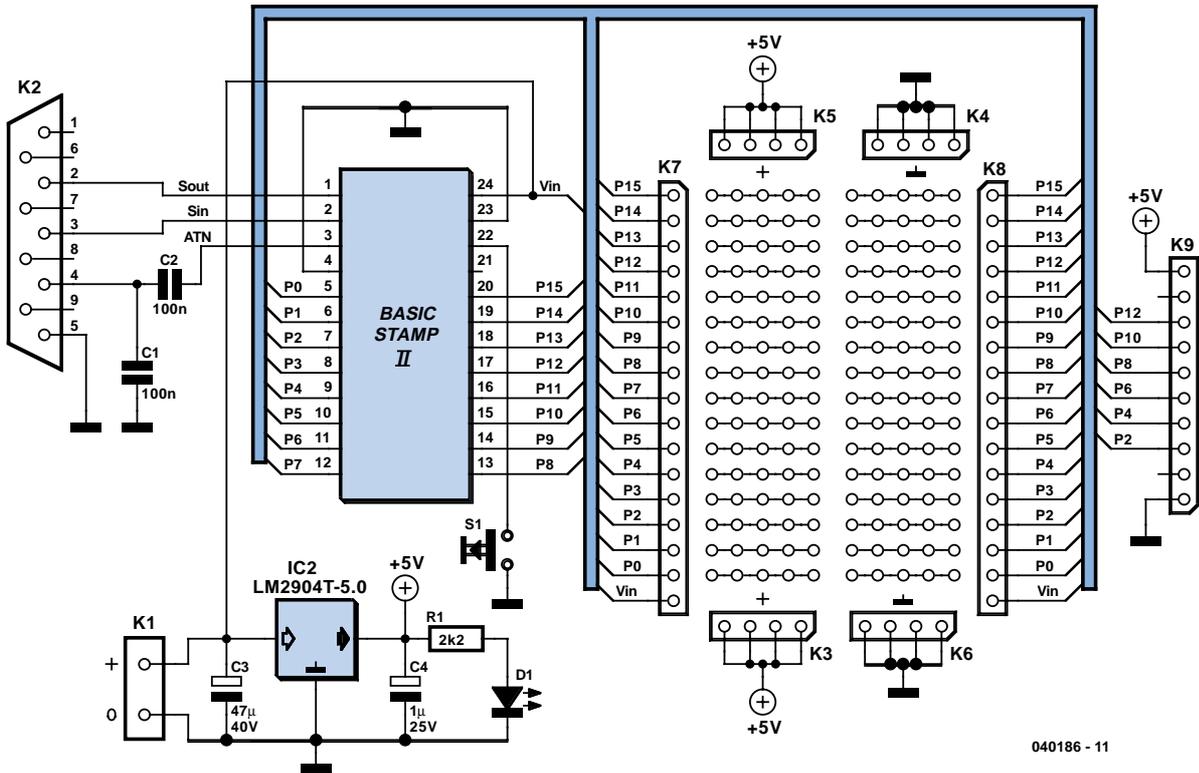


Figure 1. Schéma du circuit de la platine d'éducation (BoE), adaptée pour le module FlexiPanel-Bluetooth de Parallax.

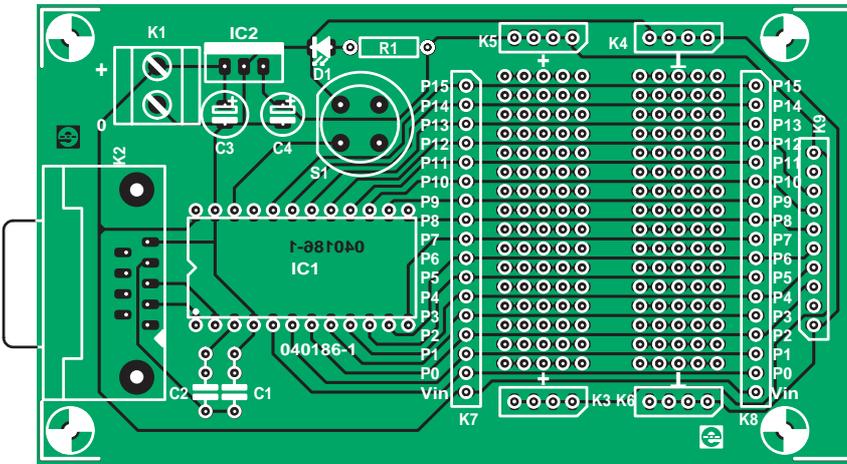


Figure 2. Platine de commande pour la BoE modifiée.

## Liste des composants

### Résistances :

R1 = 2kΩ

### Condensateurs :

C1, C2 = 100 nF  
 C3 = 47 µF/40 V radial  
 C4 = 1 µF/25 V radial

### Semi-conducteurs :

D1 = LED rouge faible courant

IC1 = Basic Stamp (BS2, BS2sx, BS2e ou BS2p) (Parallax Inc, Selectronic)  
 IC2 = LM2940T-5.0

### Divers :

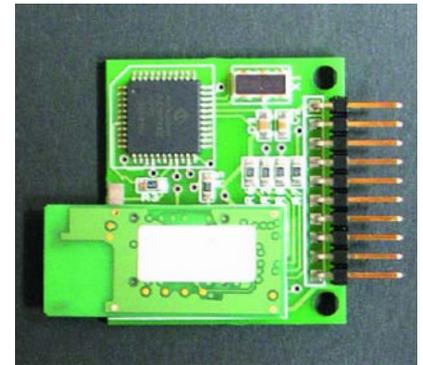
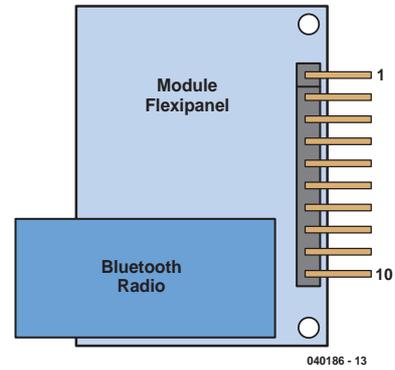
K1 = bornier encartable à 2 contacts au pas de 5 mm  
 K2 = embase sub-D à 9 points en équerre femelle encartable  
 K7, K8 = embase autosécable à 1 rangée de 17 contacts femelle (mâle ou femelle)  
 K9 = embase autosécable à 1 rangée de 10 contacts femelle  
 S1 = bouton-poussoir unipolaire à contact travail encartable, D6R par exemple

## Robot pisteur

Si vos amis font la grimace à l'idée d'encre un autre petit robot mobile, genre buggy, dites-leur que cette télécommande est bien différente car elle est capable de renvoyer l'information à l'unité mobile en transmettant des données sur une liaison radio. En utilisant un compas électronique monté sur le robot, un relèvement du chemin est enregistré et transmis à l'unité mobile. La **figure 3** présente ce qu'il faut ajouter à la BoE pour l'adapter à notre expérience. En premier lieu, il y a le module combiné FlexiPanel-Bluetooth relié à la BS2p par 5 lignes. Ces deux unités utilisent une communication série bilatérale avec synchronisation. Comme vous le soupçonniez, le FlexiPanel est aussi un système microcontrôleur (et même un système intelligent, en plus !) Le BS2p exécute un programme capable d'envoyer des commandes de recherche ou de modification de valeurs, de contenu ou d'état du FlexiPanel. De plus, le FlexiPanel peut requérir l'attention du BS2p lorsqu'une unité client a modifié une commande via Bluetooth. Ceci est réalisé par l'intermédiaire d'une sorte d'interruption transmise par la ligne de données qui, dans notre cas, est surveillée par une diode LED. Le compas électronique de type CMPS03 est une unité I<sup>2</sup>C de Devan-

# Brochage de l'unité

Broche	Nom	Utilisation
1	Vss	Connectée à 0 V
2		Non connectée
3	RxD	Entrée de données série depuis BASIC Stamp pour opérations serout
4	TxD	Sortie de données série vers BASIC Stamp pour opérations serin
5	RTS	Sortie de commandes de flux série vers BASIC Stamp pour opérations serout
6	CTS	Entrée de commandes de flux série depuis BASIC Stamp pour opérations serin
7	Mod	Voir texte
8	Data	Sortie de données en position haute lorsqu'une commande a été mise à jour par un client FlexiPanel.
9		Non connectée
10	Vdd	Connectée à +5 V.



En insérant le module, faire attention à le mettre du bon côté de l'encoche 'AppMode' et selon la bonne orientation.

Assurez-vous que la broche Vss est connectée à la broche Vss, et Vdd à Vdd (et non à Vin !). En cas contraire, le module pourrait être endommagé.

tec. Il est disponible entre autres chez Milford Instruments.

La BoE, avec son circuit d'extension tassé dans la zone de prototypage, est monté sur la superstructure BoE-Bot du robot disponible chez Parallax Inc. Elle dispose de roues motorisées pouvant être pilotées par MLI (Modulation en Largeur d'Impulsion directement depuis la carte BASIC Stamp, comme indiqué sur la figure 3. Le robot pisteur, prêt à entamer son périple, est présenté en **figure 4**. Notez que la photo montre la BoE de Parallax.

## La programmation de la FlexiPanel et de la BS2p

En utilisant le *FlexiPanel Designer*, un générateur de programmes fourni par Hoptroff, on peut créer une interface utilisateur contenant :

- l'affichage du relèvement du compas ;
- des boutons verrouillables pour stop, en avant, en arrière, gauche et droite ;
- un traceur de route : table affichant le chemin relevé par le robot.

L'utilitaire *FlexiPanel Designer* crée un programme pour PBasic qui permet de programmer l'électronique du FlexiPanel. En utilisant le PC et l'éditeur du Basic Stamp, ce programme est téléchargé sur le Stamp où il est exécuté.

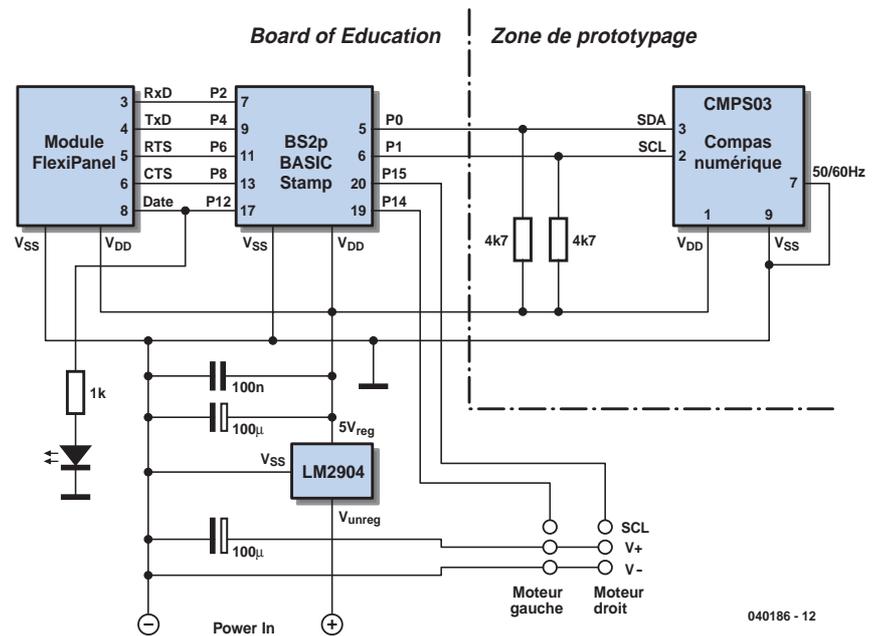


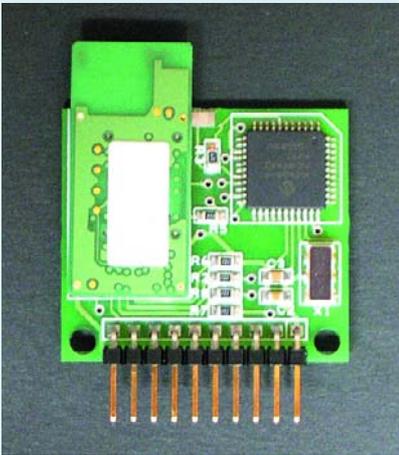
Figure 3. Le circuit additionnel pour le robot pisteur est un compas électronique.

Il peut arriver que le FlexiPanel doive être d'abord réinitialisé, par exemple lorsqu'un essai antérieur de programmation n'a pas marché. Dans ce cas, la tension d'alimentation du circuit doit être coupée et remise en fonction - le bouton de réinitialisation situé sur la BoE agissant seulement sur le Stamp et pas sur le FlexiPanel ! Lorsque le

circuit est à nouveau en fonction, la programmation va recommencer automatiquement - le programme étant toujours présent dans l'EEPROM, il n'a pas besoin d'être rechargé depuis le PC. Il est à noter que le FlexiPanel mettant environ 10 secondes à s'initialiser, vous avez le temps de touiller votre thé ou votre café avant que la

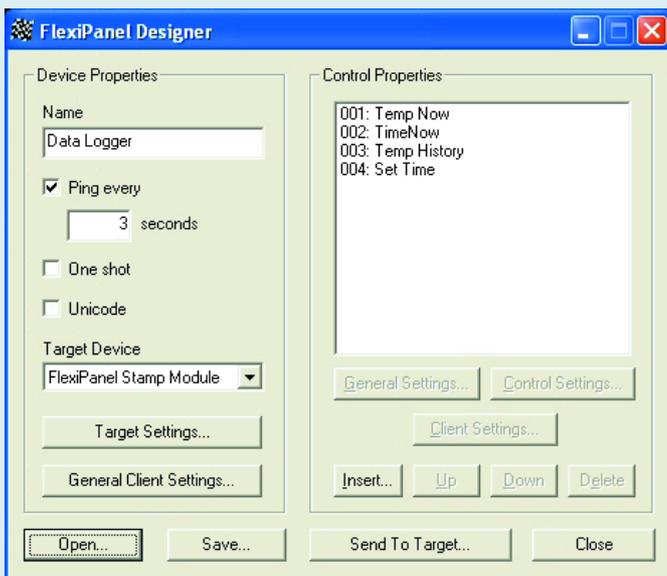
# Que puis-je en faire ?

Le propos de cet encart est de fournir des motivations pour le développement de projets utilisant le module FlexiPanel-Bluetooth de Parallax. Regardez ce que vous pouvez faire avec lui ! Les documentations et logiciels sont disponibles en téléchargement gratuit.

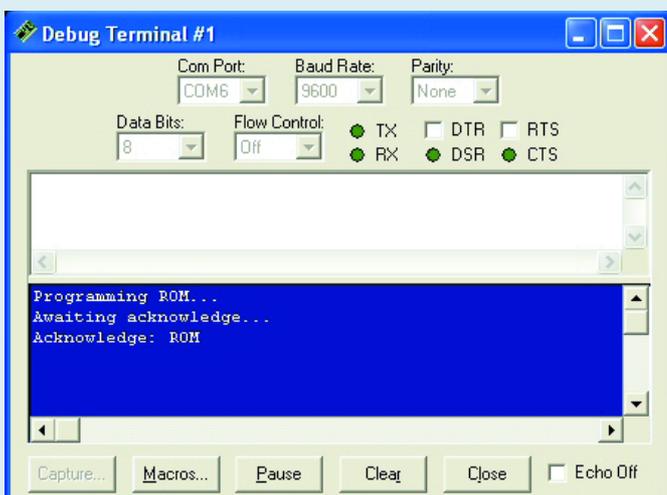


*FlexiPanel avec la sous-platine radio Bluetooth montée sur son dos. Vous avez le schéma de brochage et les tableaux de données – rien ne vous empêche d'y brancher un microcontrôleur.*

*Interface utilisateur d'un robot pisteuseur sur un Pocket PC.*

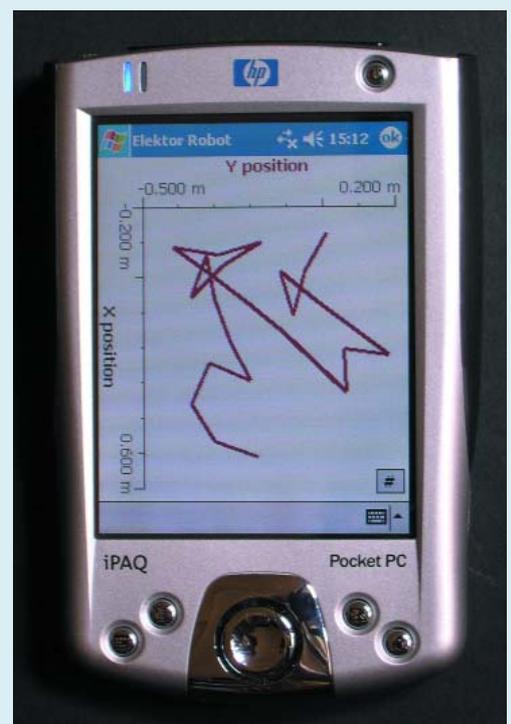


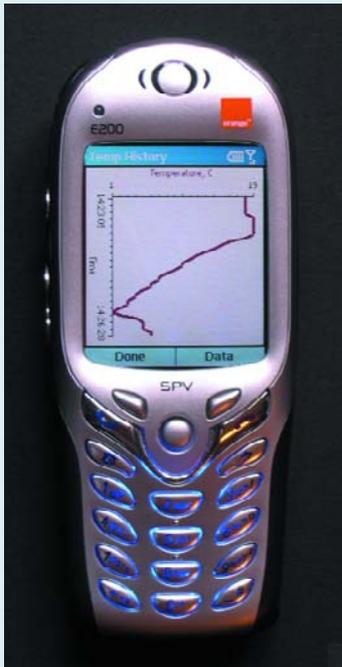
*L'utilitaire de programmation FlexiPanel Designer pour un projet d'enregistreur de données. Concevez vos propres boutons de commande tels qu'ils apparaissent sur les unités Bluetooth.*



*L'éditeur BASIC Stamp occupé à programmer le module FlexiPanel (une ancienne version, aujourd'hui remplacée par un glisser-déposer).*

*Je suis sûr que nous avons loupé un tournant quelque part, chérie ! Le chemin relevé par le robot pisteuseur sur un Pocket PC.*





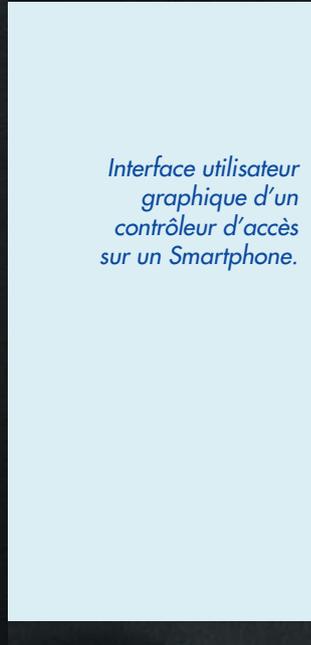
Historique de température sur un Smartphone.



Nous sommes perdus, n'est-ce pas ? Relevé du chemin parcouru affiché sur un Smartphone.



Interface utilisateur graphique d'un enregistreur de température sur un Smartphone.



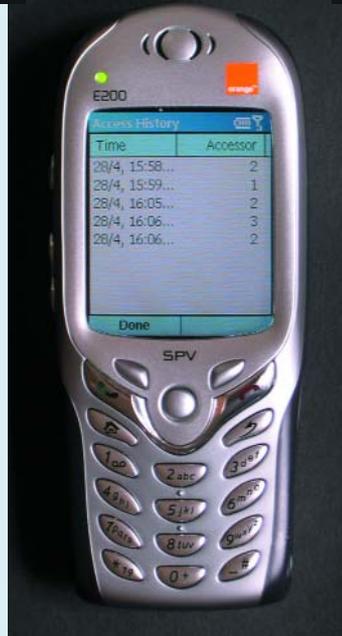
Interface utilisateur graphique d'un contrôleur d'accès sur un Smartphone.



Enregistrement d'un contrôleur d'accès affiché sur un Smartphone.



Interface utilisateur graphique d'un robot pisteur sur un Smartphone.



# Le protocole Bluetooth

Bluetooth est un protocole de radiocommunication numérique à 2,4 GHz développé par et sous licence d'Ericsson. Actives dans le « personal area network » (réseau de domaine personnel), les unités Bluetooth peuvent entrer et sortir à volonté. Par contraste, le protocole WiFi, qui opère à la même fréquence, est plus adapté aux infrastructures sans fil plus permanentes, chaque noeud individuel se voyant assigné une adresse IP (Internet Protocol) fixe.

Grâce aux écouteurs Bluetooth, ce protocole est maintenant fermement enraciné dans le marché des téléphones portables. Intel prévoit d'incorporer Bluetooth dans sa puce Centrino 2 qui doit être lancée à l'automne 2004. Ceci non seulement permettra aux PC de se connecter sans fil aux imprimantes, etc., mais va aussi donner un coup de fouet à la croissance des communications VoIP (Voice over Internet Protocol, voix par le protocole internet), c'est-à-dire la téléphonie par internet.

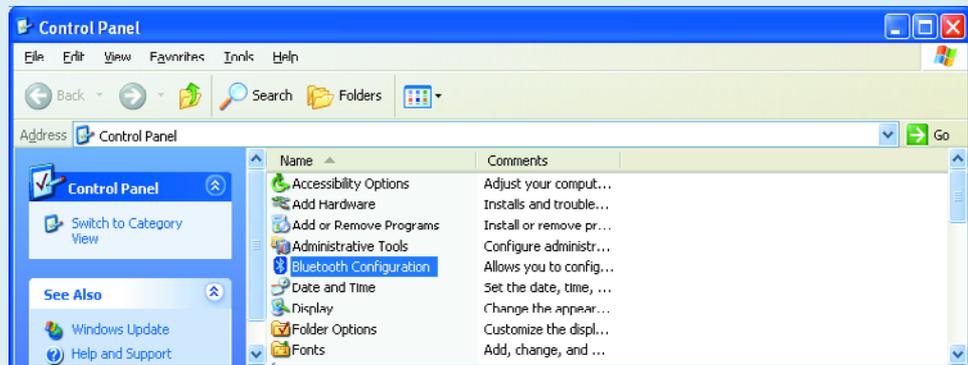
Le standard Bluetooth fournit des interfaces pour un large éventail de protocoles de communication, depuis un simple port série jusqu'à l'audio. A l'instar de nombreux protocoles de haut niveau, tel l'échange de fichiers OBEX, FlexiPanel s'installe au sommet de la couche émulation de port série dans la pile du protocole Bluetooth. Il ne fait pas partie du standard « officiel » Bluetooth. Ce dernier est toutefois relativement ouvert, en ce qu'il permet à tout un chacun de développer des logiciels pour les unités à distance, et les composants additionnels de produits comme le FlexiPanel sont fabriqués sous licence, comme n'importe quel module radio Bluetooth. Les premiers produits FlexiPanel étaient des bibliothèques de logiciels destinés à la commande à distance d'applications Windows et de systèmes embarqués haut de gamme.

En tant que produit électronique, le module FlexiPanel est un périphérique fournissant des services d'interface utilisateur graphique. Il tient à jour une liste des composants requis par le projet, ainsi que le statut actuel des commandes. Le produit peut mettre à jour une commande à tout moment et, à l'inverse, être averti si un utilisateur modifie une commande !

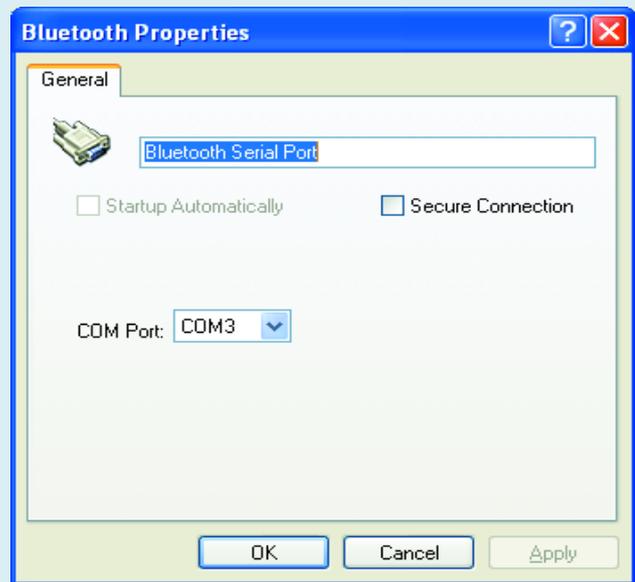
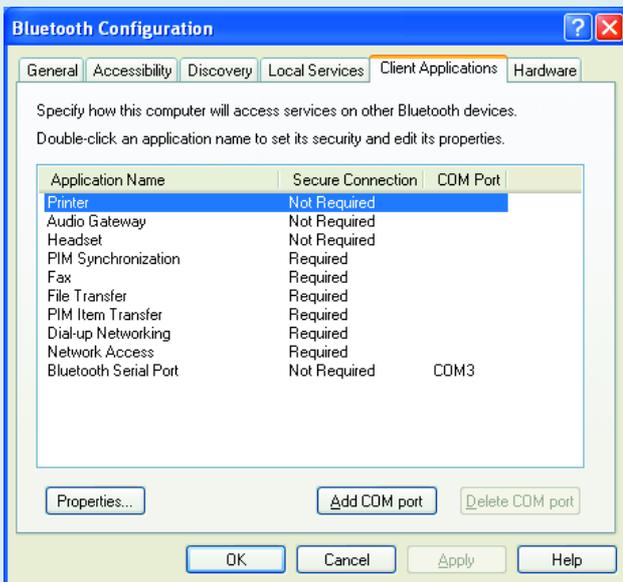
## Installer un adaptateur Bluetooth

Les PC de bureau utilisent généralement des adaptateurs Bluetooth externes pour communiquer avec des unités comme le Flexipanel. D'habitude, l'installation de tels adaptateurs est facile en utilisant le logiciel fourni, toutefois l'installation du FlexiPanel nécessite quelques réglages fins.

Pour ces adaptateurs, une série de ports COM est installée pour contrôler la liaison avec d'autres unités Bluetooth.



Un double clic sur cette option vous envoie à la fenêtre suivante (tabulation : Applications client)



Vérifiez les propriétés du port COM et assurez-vous que l'option Connexion Sécurisée n'est pas cochée.

Ces liaisons sont normalement protégées contre toute intrusion dans le PC non autorisée par le logiciel d'installation. Cependant, le FlexiPanel n'utilise pas de telles liaisons protégées et c'est pourquoi vous avez besoin de figoler le paramétrage des ports COM du PC. Commencez par le panneau

de configuration où vous trouverez la configuration Bluetooth.

Pour ces détails et d'autres lancinants sur Bluetooth, consultez la FAQ à l'adresse [www.flexipanel.com/comports/index.htm](http://www.flexipanel.com/comports/index.htm)

## Listage 1. Code BS2p pour le robot pisteur (extrait)

```
CheckCompass:
' only check every 50 pulses
CmpCount = CmpCount - 1
IF CmpCount > 0 THEN GoBack
CmpCount = 50

' Get compass direction in binary radians and in tenths of a degree
I2CIN SerPt, CmpIn, 1, [brad, degs.HIGHBYTE, degs.LOWBYTE]

' Send degrees value to bearing control (code generated by FlexiPanel Designer)
SEROUT TxPin\CTSPin, BaudM, [SetData, ID_Bearing, degs.LOWBYTE, degs.HIGHBYTE, 0, 0]

' Calculate position with Send binary radians value to bearing control
' (code generated by FlexiPanel Designer & cut'n'pasted)

IF FwRvSp = 1 THEN
  xloc = xloc + COS( brad )
  yloc = yloc + SIN( brad )
ELSEIF FwRvSp = 2 THEN
  xloc = xloc - COS( brad )
  yloc = yloc - SIN( brad )
ENDIF

' if moving, send to trace
IF NOT FwRvSp = 0 THEN
  SEROUT TxPin\CTSPin, BaudM, [AddRow, ID_Route_trace, yloc.LOWBYTE, yloc.HIGHBYTE,
  xloc.LOWBYTE, xloc.HIGHBYTE ]
ENDIF

' return to motor control
GoBack:
RETURN
```

programmation ne commence. Pour avoir au moins une idée de ce qui se passe pendant la phase d'initialisation, une diode LED à basse intensité peut être connectée entre la ligne de données et la terre, sans oublier une résistance d'1 k $\Omega$  – cf. figure 3. Après un rebranchement de l'alimentation, ou une réinitialisation normale du module (qui se déclenche automatiquement après toute programmation), la diode LED s'allume pendant quelques secondes puis s'éteint lorsque l'initialisation est terminée. Elle s'allume aussi brièvement lorsque le FlexiPanel reçoit une commande par Bluetooth (requête d'interruption).

Le résultat de l'utilisation de *FlexiPanel Designer* est présenté dans l'encart Que puis-je en faire ? : boutons simples à presser sur un PC de poche ou un téléphone GSM, et carte renvoyée par le robot vous indiquant où il est allé ! Lorsque l'interface utilisateur a été programmée dans le module FlexiPanel, le BASIC Stamp est chargé avec le programme exécutable dont un extrait est donné dans le **listage 1**. Après l'initialisation, le programme cherche quel type d'impulsion de commande du moteur il est censé fournir. Puis il lit le compas et écrit le relèvement sur l'écran et le traceur de route.

## Plus sur le logiciel du projet

Pour vous lancer dans ce projet, vous aurez besoin des outils suivants :

1. Windows 2000 ou ultérieur
2. le kit de développement logiciel du FlexiPanel (SDK = *Software Development Kit*)
3. l'éditeur du Basic Stamp (BSE = *Basic Stamp Editor*)
4. le FlexiPanel.

Quelques notes : le kit de développement logiciel du FlexiPanel SDK (connu précédemment sous le nom de kit de développement du FlexiPanel BASIC Stamp) ne marche que sous les versions récentes de Windows.

Le kit SDK permet à l'interface utilisateur d'être conçue sur un PC, c'est-à-dire que c'est vous qui décidez des commandes nécessaires pour le produit final, et de leur apparence sur l'unité Bluetooth, cf. FlexiPanel Designer plus haut.

Au sujet de l'éditeur du BASIC Stamp, BSE, seuls les Bs2, Bs2sx, Bs2e et BS2p sont supportés. La version -p est préférée à cause de sa vitesse et de sa taille mémoire. Notez que le Bs2p40 à 40 broches ne fonctionnera sur aucune BoE où seul un support à 24 broches est disponible.

Une version du FlexiPanel pour Windows 95/98 est disponible en téléchar-

gement gratuit, de même que des versions pour les unités MS Smartphone, Pocket PC, Java JABWT et le système Palm OS.

## Notes finales sur le projet robot pisteur

Le BoE-Bot est alimenté par quatre piles AA qui, dans la pratique, ne durent que quelques dizaines de minutes avant de s'épuiser ou de ne fournir que des tensions trop basse pour que l'électronique puisse continuer à fonctionner.

Le module FlexiPanel peut générer un signal indiquant la nouvelle connexion d'une unité mobile. Le robot pourrait s'arrêter automatiquement lorsqu'il sort du rayon de liaison avec l'unité mobile. Le fonctionnement du compas électronique peut être significativement influencé par la proximité de métal et de courants directs. Ces effets peuvent être compensés par un calibrage local du compas.

(040186-1)

### Liens vers des sites Web

Parallax : [www.parallax.com](http://www.parallax.com)  
FlexiPanel : [www.flexipanel.com](http://www.flexipanel.com)  
Selectronic : [www.selectronic.fr](http://www.selectronic.fr)  
Milford Instruments :  
[www.milinst.demon.co.uk](http://www.milinst.demon.co.uk)

# DRM SUR LES

## DREAM version 1.06

### B. Kainka

Les choses n'arrêtent pas de bouger dans le monde de la radio numérique sur les Ondes Courtes. De nouvelles stations naissent et le logiciel de réception DREAM évolue lui aussi.

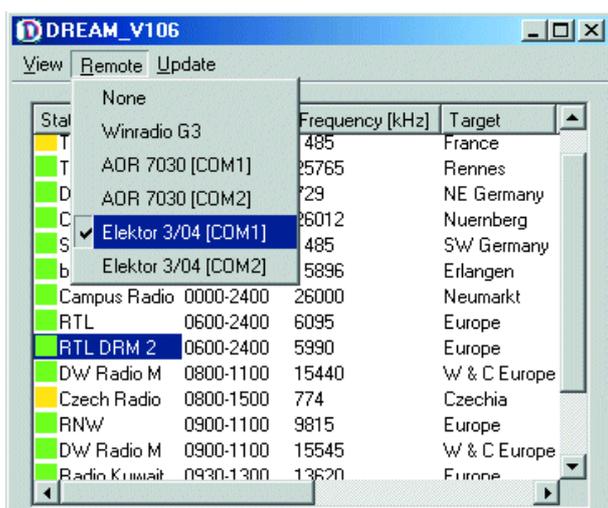


Figure 1. La liste d'émetteurs sous DREAM.

À compter de sa version 1.06, DREAM supporte directement la syntonisation du récepteur DRM d'Elektor. Il est en outre possible, par le biais de *Stations/Dialog/Remote* de sélectionner également Winradio G3 et le AOR 7030. Il est possible de télécharger, via Internet, une liste complète des sta-

tions d'émission avec nom, horaires, fréquences, puissances et autres informations (figure 1). Sur le site de Klaus Schneider ([www.drm-dx.de](http://www.drm-dx.de)) vous trouverez une liste bien à jour des stations actuelles. La dernière adjonction est, par exemple, RTL DRM-2 sur 5 990 kHz.

Il est possible, au choix, de n'afficher que les stations actives. Une action sur un bouton, il n'en faut pas plus pour accorder le récepteur sur une station. Il est possible, comme on garde une bonne vue d'ensemble, d'identifier rapidement les nouvelles stations et celles qui émettent peu. Lors de la syntonisation, le port sériel est ouvert brièvement puis refermé, de sorte que l'on peut également utiliser le programme DRM.exe pour un accord « libre ». Ceci est pratique en particulier lors de la réception AM (que DREAM supporte aussi !).

En outre, les capacités multimédia de DREAM se sont améliorées. Il est possible maintenant de recevoir des photos et le service Journaline. La BBC, entre autres, a commencé par « l'émission » d'images que DREAM pourra stocker sur le disque dur.

### Meilleure réception AM

Le mode AM a, dans la version 1.06, été amélioré, lui aussi. Il est devenu possible d'indiquer si l'on ne veut recevoir que la bande latérale supérieure ou inférieure. Il est possible ainsi, bien souvent, d'éviter les parasites des stations adjacentes. Il est en outre possible d'adapter la largeur de bande passante. Normalement, on utilise 5 kHz, mais dans certains cas, une largeur plus faible permet d'améliorer le rapport signal/bruit.

Il est souvent possible de trouver plusieurs émetteurs dans le spectre et de les décoder. Il est même possible, avec le filtre 12 kHz intégré, de sélectionner jusqu'à 3 stations AM sans avoir à faire de syntonisation matérielle ! On pourra, pour utiliser la largeur de bande totale de la carte-son, utiliser un filtre FI plus large. Un essai avec un CFW455C (largeur de bande nominale de 25 kHz) s'est fait avec succès. Il nous a été possible, sans perte de qualité, de capter jusqu'à 3 stations AM simultanément (figure 2). Ceci nous donne une petite idée de ce que l'avenir nous réserve. Il n'est pas exclu que très rapidement, les antennes attaquent directe-

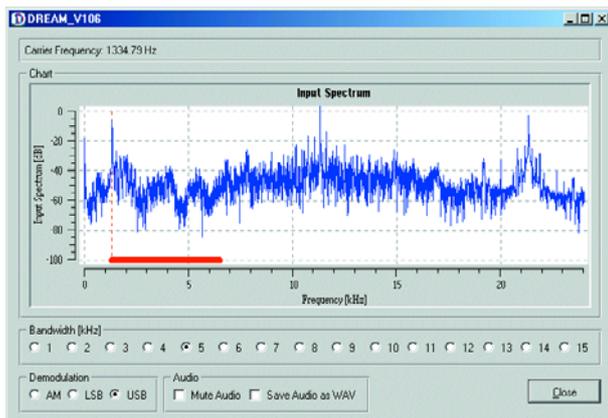


Figure 2. DREAM en mode AM, avec filtre FI large.

# RAILS

ment un convertisseur analogique/numérique (CAN) rapide. Une transformée rapide de Fourier (FFT) permet de déterminer le spectre et d'en extraire facilement la station de radio souhaitée.

## Émetteur de mesure DRM

DREAM permet non seulement la réception de signaux DRM, mais également d'en produire. Il faut pour cela démarrer le programme avec le paramètre -t. On voit alors apparaître, sur la sortie de la carte-son du PC, un signal de bande DRM. Il n'est pas encore possible d'envoyer de l'audio, mais il est déjà possible d'envoyer des données multimédia.

On pourra, à titre de test, interconnecter la ligne line-out directement à l'entrée de la carte-son. En démarrant un second programme DREAM il est possible de recevoir le signal. Vous constaterez que le résultat est un signal parfait dont le rapport signal/bruit est supérieur à 50 dB.

On pourra aussi utiliser le signal DRM pour tester d'autres récepteurs. Tout ce dont on a besoin pour compléter l'émetteur est un mélangeur. On pourrait, en principe, mettre à contribution un second récepteur DRM Elektor. Comme le mélangeur en anneau de trouve directement à l'entrée, il est possible de le monter en tête-bêche. Si vous ne disposez pas d'un second récepteur, vous aurez vite fait de réaliser un mélangeur. La **figure 3** donne le schéma d'un mélangeur à base de diode au germanium, mais il est également possible d'utiliser une diode Schottky. La sortie du mélangeur doit attaquer directement l'entrée d'antenne du récepteur.

On devrait voir apparaître à la sortie du récepteur le signal DRM de bande de base. Avec un système idéal, le rapport signal/bruit devrait à nouveau être de 50 dB, mais comme il y a toujours, dans la pratique, des pertes et des distorsions, ce nombre sera toujours un peu plus faible. Cela nous permet de déterminer la qualité d'un récepteur. En utilisant un récepteur DRM Elektor tant en récepteur qu'en émetteur, nous avons relevé des rapports signal/bruit supérieurs à 30 dB, un excellent résultat ! L'utilisation du mélangeur à diode en émetteur se traduit par un 29 dB (cf. **figure 4**).

À noter que le mélangeur de la figure 3 peut également servir de (bon) modulateur AM. On peut lui appliquer directement de l'audio qui sera rendue audible par un récepteur AM. En mode DRM on voit la porteuse à 16 000 kHz, le signal sur 16 012 kHz et la bande latérale inférieure à 15 988 kHz avec un signal DRM inversé. Il est également intéressant d'expérimenter avec d'autres largeurs de bande. Jusqu'à présent, l'émission en O.C. se fait à 10 kHz et à 9 kHz sur les P.O. Les normes permettent, pour les émissions DRM, une largeur de bande comprise entre 4,5 et 20 kHz. Il devient possible ainsi d'utiliser, dans le récepteur, un filtre de fréquence intermédiaire (FI) plus large (type CFW455C). DREAM convient fort bien pour ce genre d'expérimentations, pour, par exemple, l'émission d'i-mages (**figure 5**).

## DRM pour le radio-amateur

On parle d'utiliser DRM avec une largeur de bande moindre pour des applications de radio-amateurisme. Cette nouvelle technique n'est cependant pas encore autorisée. Le mini-modulateur décrit ici fonctionne aussi sans connexion câblée

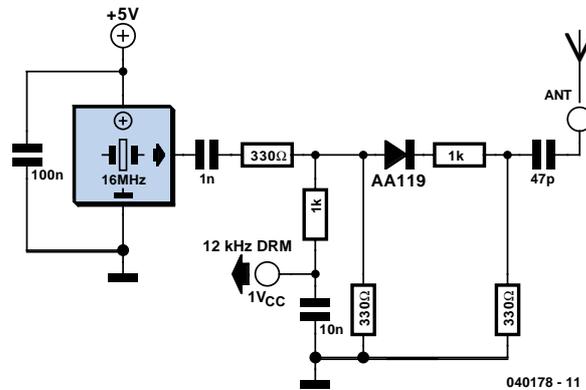


Figure 3. Mélangeur/modulateur à diode au germanium.

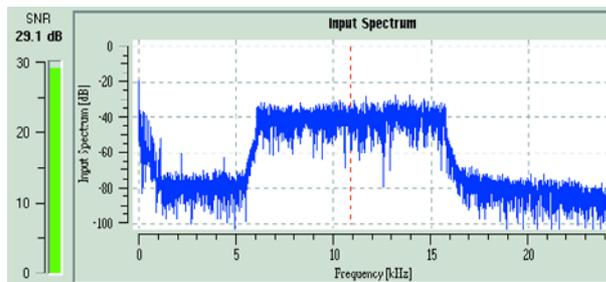


Figure 4. Le signal du mélangeur à diode reçu avec DREAM.

directe; en utilisant quelques longueurs de câble comme antenne il devient possible de réaliser une liaison sans fil. Il reste cependant déconseillé d'utiliser de « vraies » antennes, il n'est pas question que vos expérimentations puissent être taxées d'émissions illégales.

(040178)

Plus d'informations sur Internet :

[www.elektor.fr/drmrfaq](http://www.elektor.fr/drmrfaq)

(Site de support pour le récepteur DRM d'Elektor)



Figure 5. Cette image est envoyée par le biais de DREAM.

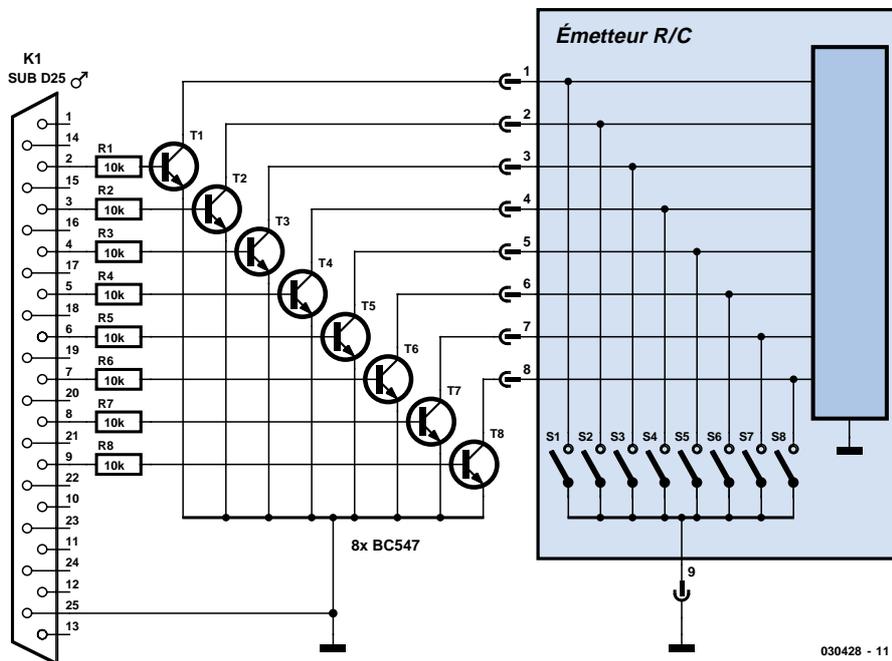


Figure 1. 8 résistances et autant de transistors n'ont pas de problème à trouver place dans le capot du connecteur.

# Interface pour télécommande

## Interface parallèle pour systèmes de télécommande

Dipl.-Ing. Karl Köckeis

Les systèmes de télécommande connaissent une popularité sans cesse croissante pour la télécommande de nombre d'appareils domestiques. Pourquoi ne pas envisager de les utiliser aussi pour le pilotage et l'automatisation au travers d'un PC. Il nous faut pour cela une interface adéquate assurant la liaison du PC vers le système de télécommande.

C'est très précisément la fonction de l'interface présentée ici on ne peut plus simple et partant réalisable facilement et à peu de frais. L'interface parallèle d'un PC, de moins en moins utilisée de nos jours, met à disposition 8 bits de données (D0 à D7) utilisés pour le pilotage de l'interface. Il s'agit des contacts 2 à 9 de l'embase sub-D 25 points. Un transistor par ligne remplit, dans l'interface, la fonction de commutateur électronique relié à un émetteur radio standard (vendu dans les maga-

sins spécialisés). Un programme génère de courtes impulsions de commutation qui, par le biais de l'étage de commutation dans l'interface, déclenchent un ordre radio dans le système de télécommande et exécute la fonction de commande souhaitée.

Les impulsions de commutation d'une longueur de 0,5 s sont transmises par les bits de données D0 à D7 de l'interface parallèle aux 8 étages de commutation constitués des résistances R1 à R8 et des transistors cor-

respondants T1 à T8. Ces derniers, qui travaillent en collecteur ouvert, sont reliés, par le biais d'un connecteur, aux entrées de touche de la puce de codage se trouvant dans la télécommande. Les transistors de commutation se trouvent de ce fait en parallèle sur les touches de l'émetteur. Les impulsions de communication produites par le logiciel de commande produisent ainsi un ordre radio exactement comme le ferait une action sur l'une des touches de la télécommande.

La liaison pourra prendre la forme d'un câble pour imprimante du commerce classique doté d'un connecteur pouvant être ouvert (non embouti). Il est possible, à condition d'utiliser des composants miniatures, d'intégrer la totalité de l'électronique dans le connecteur sub-D du câble. Pour ce faire, il faut ouvrir le capot du connecteur, dessouder les conducteurs, raccourcir les lignes non utilisées de manière à disposer de la place requise pour le câblage, en l'air, des résistances R1 à R8 et des transistors T1 à T8. Le connecteur de l'autre extré-

mité du câble est dessoudé voire purement et simplement sectionné, le connecteur à 9 contacts étant alors soudé aux lignes concernées. Une fois le fonctionnement contrôlé, on pourra refermer le capot du connecteur.

L'embase à 9 broches est encastree dans la télécommande. Ainsi, tant que l'interface n'y est pas connectée, la télécommande pourra être utilisée comme d'habitude. Le câblage du connecteur se fait en respect du schéma. Pour cela, on ouvre la télécommande et établit une ligne vers la masse et 8 lignes vers les commutateurs S1 à S8. Les conducteurs sont soudés aux points correspondants des composants de la platine. Le câble et l'embase sont mis en place et le boîtier est refermé. L'embase est alors accessible par le biais du compartiment destiné à la pile ouvert. Le matériel est prêt. Nous n'entrons pas, dans le cadre de cet article, dans le détail du logiciel, ce que nous ferons dans un prochain article publié dans l'un des prochains numéros d'Elektor !

(030428-1)

# TRICK.e DU



## L'auteur

Comme le montre une photo de jeunesse de Martin Oßmann (né en 1959), le virus de sa carrière future l'a pris très tôt. Après des études d'électrotechnique à la RWTH d'Aix-la-Chapelle (diplômé en 1984), un tour comme assistant scientifique à la chaire de Mathématiques de la même RWTH (promotion 1988), on le retrouve comme ingénieur de développement, assistant scientifique au laboratoire de développement de Philips et depuis 1999, comme professeur d'Électrotechnique et de Technologie de l'Information à Aix. Martin Oßmann lit Elektor (RFA) depuis 1972 et l'un de nos auteurs depuis 1989. Son premier article dans Elektor, « logiciel de décodage de signal R.D.S. pour PC (n° 138, décembre 1989).

Nombre d'entre nous gardons dans leur tiroir une grosse quantité de résistances de même valeur achetées en vrac. On pourra, pour se faire la main, les souder sous la forme d'une grille quadrilatère bidimensionnelle (**figure 1**). Un nombre infini de résistances constitue une grille infinie d'où la première question de cette nouvelle rubrique à plusieurs épisodes : TRICK.e.

### Voici la Question :

**On réalise, à l'aide de résistances identiques de  $R = 1 \text{ k}\Omega$ , une grille quadrilatère s'étendant à l'infini. Quelle résistance mesure-t-on, cf. figure 1, aux bornes de l'une des résistances ?**

### Comment expliquer cette valeur ?

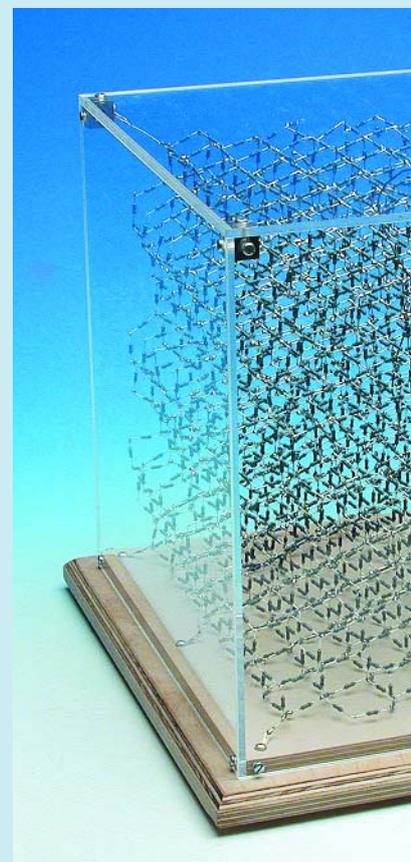
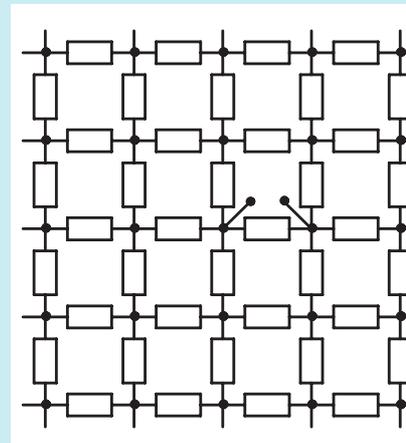
Si vous le voulez, vous pouvez bien évidemment souder quelques (!! ) résistances et mesurer pour obtenir une valeur approximative. Une simulation peut également aider. Mais attention : La question n'est pas uniquement la valeur mais aussi une explication simple et fondée. Cette explication comportera le moins de maths possible ! On pourra également s'essayer à une grille tridimensionnelle (**figure 2**). Bonnes cogitations !

### Question subsidiaire

Si vous n'êtes pas certain à 100% que votre explication de la valeur de résistance mesurée en figure 1, est la plus simple et la meilleure, la question subsidiaire de la figure 2 vous donne une seconde chance de gagner. Dans le cube que l'on y voit, les résistances constituent une grille tridimensionnelle correspondant à la structure cristalline d'un diamant.

### La question subsidiaire:

**Combien de résistances comporte le cube de la figure 2 ?**



# U PROF OSSMANN

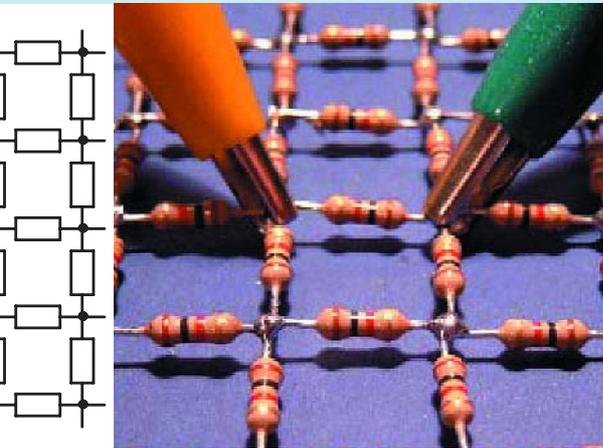
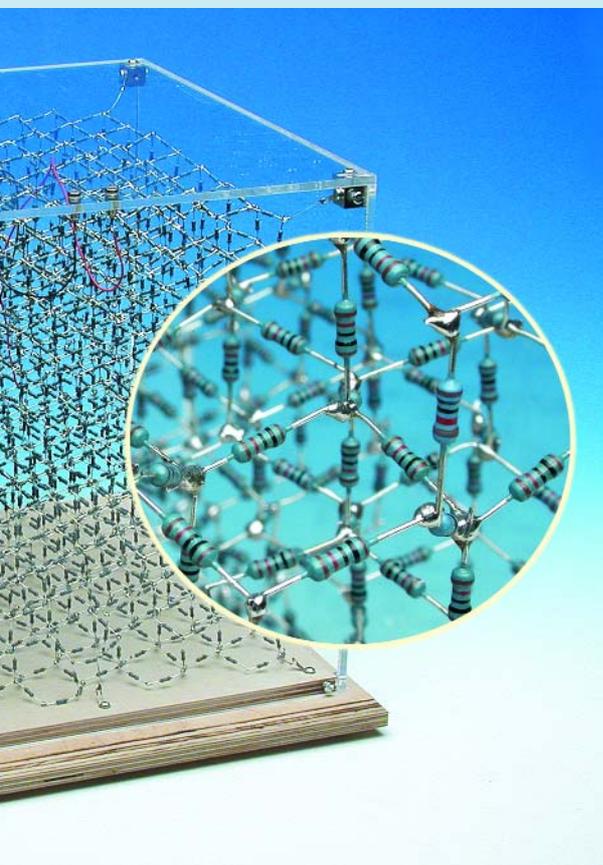


Figure 1. Un réseau de résistances s'étendant à l'infini.

Figure 2. Oßmann's Cube : un réseau de résistances tridimensionnel.



## Gagnez avec TRICK.e !

Vous pouvez, en donnant la meilleure réponse à la question posée dans TRICK.e, cet **oscilloscope USB Voltcraft** de **Conrad Electronic** d'une valeur de 200 € !



L'examen des réponses sera fait par le Professeur Oßmann épaulé par la rédaction d'Elektor; le jugement est sans appel. En cas de plusieurs réponses « identiques » nous procéderons à un tirage au sort. Les 5 lecteurs ayant donné la réponse (la plus) exacte gagnent une mémoire USB Elektor (pour la description, cf. les pages centrales du magazine).



### Règles de participation :

Envoyez votre réponse à la question Trick.e et/ou à la question subsidiaire par Courriel, Télécopie ou Courrier à

Rédaction Elektor, SEGMENT B.V./ELEKTOR  
chez WWS, 4, Rue Caroline, 75017 Paris,  
Télécopie 01.42.61.18.78,  
Courriel : redaction@elektor.fr

**Date limite de réception des envois :  
le 23 septembre 2004.**

Tout recours légal est exclu. Cette opération est ouverte à tous nos lecteurs, exception faite des personnels de SEGMENT B.V., société dont fait partie Elektor.

