



# elektor

## Grillez vos puces par USB

**ClariTy :**  
la classe... T

**Attila :**  
Bloque-pub TV

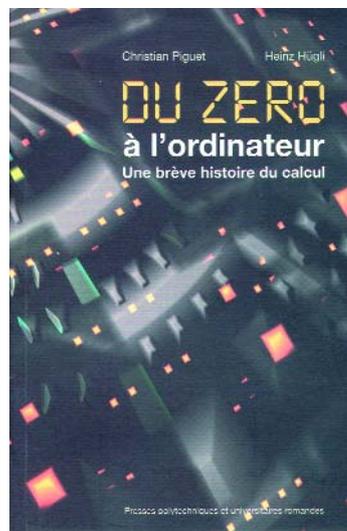
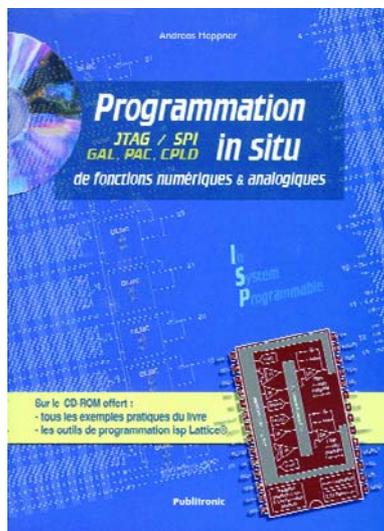
**DAB :**  
radio de qualité CD

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

N° 312 • JUIN 2004 • mensuel  
(B) 6,35 € • (CH) 10,90 FS (CAN) 8,10 \$Can.  
DOM surface 8,60 € • DOM avion 8,50 €  
ISSN 0181-7450

L 19624 - 312 - F: 5,75 €





## Programmation in situ

### JTAG / SPI, GAL, PAC, CPLD

Bien que cet ouvrage date du novembre dernier, il n'est est pas devenue d'actualité, ce qui en justifie une « critique ».

En tout état de cause, le sujet de ce livre n'est pas près de ne plus être d'actualité : la programmation in situ (ISP). Nous n'avons pas l'intention de vous apprendre qu'il existe des composants que l'on peut programmer bien qu'ils se trouvent sur une platine. Plus intéressant est le fait qu'il existe un ouvrage qui soit consacré aux possibilités offertes par les circuits intégrés logiques complexes programmables.

La rédaction avait eu vent de la publication de cet ouvrage. La raison qui nous fait en parler sont les réactions que nous avons eues à la série d'articles publiée dans les numéros de février et mars 2004 et consacrée aux CPLD (*Complex Programmable Logic Device*). Si le contenu de cet ouvrage recoupe bien le sujet abordé par la série, il ne s'arrête pas là, le livre s'occupant non seulement des circuits plus simples tels que les

GAL (*Generic Array Logic*) mais consacre une partie importante aux composants ISP analogiques. Il s'agit de circuits intégrés intégrant, par exemple, plusieurs amplificateurs opérationnels épaulés par des résistances et des condensateurs.

Cet ouvrage ne se contente pas d'aborder l'aspect théorique. Une bonne partie, celle qui ne manquera pas d'intéresser nos lecteurs amateurs de réalisations est consacrée à quelques platines d'étude pour l'ispGAL22V10, l'ispLSI-1032/2064, les LC4064/4128, le tout suivi par quelques expériences à faire soi-même.

En quelques mots, un ensemble équilibré associant la théorie et les informations pratique à la pratique sous la forme de cartes d'étude.

Nous ne voudrions pas passer sous silence la présence d'un CD-ROM reprenant tous les fichiers du livre et le fameux outil ispLEVER STARTER de Lattice ; à noter en outre la création d'un UserGroup baptisé ElektorISP chez Yahoo.

(040067F)

## Du zéro à l'ordinateur

### Une brève histoire du calcul

Christian Piguet & Heinz Hügli

L'histoire du calcul débute avec l'apparition des systèmes de numération, de Sumer à Rome, en passant par Thèbes et Athènes, sans oublier les Chinois et l'Inde et l'invention (très importante) du zéro (en Inde) ; elle se poursuit avec la conception des premières machines à calculer mécaniques (abaques, automates, Pascaline) et électromécaniques, puis aboutit enfin aux machines électroniques (dont les premières ne datent que du début des années 60 du siècle dernier). Ces dernières apparaissent d'abord sous la forme d'ordinateurs primitifs constitués de volumineux tubes à vides, avant d'évoluer vers les machines sophistiquées que nous connaissons aujourd'hui, composées de (dizaines de) millions de minuscules transistors à base de silicium.

Ce livre expose de manière claire et progressive la façon dont sont nées les idées qui ont conduit à la réalisation des ordinateurs et des micropro-

cesseurs actuels. Il offre au lecteur les clés lui permettant de mieux comprendre la nature et le fonctionnement de ces machines. Richement illustrée, proposant fils conducteurs et détails de fonctionnement, l'histoire exposée dans cet ouvrage intéressera tous ceux et celles curieux de connaître la genèse et l'avènement des ordinateurs.

Contenu : Préface - Avant-propos - Histoire des chiffres - Histoire des machines à calculer et des automates - Histoire des machines à calculer électroniques - Histoire des ordinateurs - Histoire de la programmation - Histoire des microprocesseurs. En résumé, un livre très bien documenté sur le sujet, que l'on lira avec plaisir et qui servira de référence le jour où il faudra déchiffrer des calculs d'intendants égyptiens ou de gérants sumériens, mais aussi lorsque l'on voudra se faire une idée sur ce qui nous attend dans les prochaines années.

2004, 186 pages, ISBN 2-88074-469-5  
PRESSES POLYTECHNIQUES ET UNIVERSITAIRES ROMANDES

(047099-1)

# D'Artagnan

Un pour tous et ...

Andreas Oyrer



**En règle générale, chaque programmeur est spécialisé aux contrôleurs d'un unique fabricant. D'Artagnan est lui en mesure, grâce à USB, de programmer rapidement, vite et élégamment, les contrôleurs mi-gamme (*mid-range*) les plus importants d'Atmel (ceux de Microchip ensuite) sans oublier les EEPROM sérielles.**

D'Artagnan a été conçu dans le cadre d'une utilisation « semi-professionnelle ». Il n'a pas été taillé pour programmer la totalité des microcontrôleurs d'un fabricant, mais plutôt les microcontrôleurs 8 bits à usage général dotés de périphériques standards et d'une mémoire de travail limitée. En contrepartie, il est en mesure de programmer des contrôleurs de différents fabricants (Atmel et Microchip pour l'instant) ainsi que des EEPROM sérielles, ce qui n'est pas monnaie courante vu les algorithmes de programmation et les tensions différents que cela implique mais pourrait être plutôt une exception.

Qui dit utilisation semi-professionnelle sous-entend du même coup que le programmeur est également un outil de développement, devant permettre un débogage. Il lui faut en outre être rapide pour éviter que la mise au point d'un programme impeccable ne devienne un supplice de Tantale.

La mise en oeuvre du programmeur doit être la plus simple possible. D'où le choix de l'approche USB. La compatibilité HID garantit la mise en oeuvre la plus simple sous Windows, à partir de la version 98SE. D'Artagnan requiert partant la présence d'une embase USB libre sur le PC; l'alimentation se faisant par le biais de l'interface, il n'est pas nécessaire de prévoir d'alimentation externe.

Comme le contrôleur ne possède pas de mémoire de programme propre, on a transféré, lors du branchement du câble USB, du progiciel (*firmware*) dans la mémoire de travail du programmeur. En cas de mise à jour du progiciel, il suffit de renouveler un fichier sur le PC.

Il est également possible de stocker le progiciel dans une EEPROM sur la platine du programmeur, composant auquel accède le contrôleur lors de la mise sous tension. La seule utilité de la connexion USB sera de permettre le

transfert des mises à jour du progiciel dans l'EEPROM.

Une spécificité de cet appareil est qu'il ne requiert pas impérativement que le contrôleur à programmer soit extrait du montage-cible et placé dans le support de programmation. Le programmeur comporte en effet 2 interfaces ISP, l'une pour les contrôleurs de Microchip, l'autre pour ceux d'Atmel.

## Contrôleur USB

Le coeur du matériel est IC1, un TUSB3210 de Texas Instruments. Ce microcontrôleur compatible 8052 avec interface USB (Full Speed 12 Mbits/s) possède 4 ports à 8 lignes chacun, un UART, un chien de garde et une interface I<sup>2</sup>C. Sachant que le TUSB3210 ne possède pas de mémoire Flash propre, cela implique un rechargement du progiciel après chaque coupure de la tension. Le progiciel est chargé, par le biais d'un programme de chargement dit *bootloader*, dans la mémoire RAM de 8 K x 8 (8 192 octets). Ceci peut se faire par le biais de l'interface USB. Autre option : le progiciel se trouve dans l'EEPROM sérielle, IC5, une 24LC64 reliée aux lignes I<sup>2</sup>C SDA et SCL du TUSB3210, d'où il est lu lors de chaque démarrage et recopié dans la mémoire RAM. L'approche USB requiert, sur le PC, un pilote servant à envoyer le progiciel vers le TUSB3210. Dès que la totalité du progiciel, en provenance soit de l'EEPROM soit du PC, est recopiée en RAM, le *bootloader* déconnecte le programmeur de l'interface USB. Le programme en RAM est lancé et réinitialise l'appareil au niveau de l'USB.

## Tensions de programmation

IC1 pilote l'ensemble de lignes de programmation et de tension par le biais de 32 broches disponibles. L'obtention

des différentes tensions de programmation requises par les différents types de microcontrôleurs se fait à partir de l'alimentation 5 V, tension rehaussée à quelque 13 V par le biais d'un régulateur-rehausseur. La tension de sortie du régulateur est normalement de 12 V, la diode prise dans la ligne de contre-réaction de IC3 la rehaussant de la valeur de la tension directe de la diode, de manière à permettre aussi la programmation des microcontrôleurs PIC.

Des FET à canal-p et canal-n commutent les tensions de programmation en fonction des besoins. Il est possible ainsi, de paramétrer sur les broches 1 et 31 du support de programmation une tension de 0, 5 ou 12 V. On dispose, sur le bornier ISP K3, utilisé pour la programmation des PIC, 13 V sur la ligne MCLR. Les diodes D5 et D7 servent à rabaisser ces 13 V à 12 V, vu que c'est là la tension de programmation (plus faible) requise par les contrôleurs d'Atmel.

La commande des FET se fait par le biais de portes TTL à sorties en collecteur ouvert (74LS07). Ce n'est qu'ainsi qu'il est possible de paramétrer une tension de 0 V entre la grille et la source, niveau garantissant un blocage des transistors. Le petit 3,3 V de la ligne de port se serait traduit par 10 V environ, de sorte que le transistor aurait été passant en permanence.

Certains contrôleurs requièrent un niveau de tension plus élevé sur les entrées Reset (initialisation) et XTAL (quartz). Ainsi, un 90S1200 requiert à l'entrée Reset une tension minimale de  $0,85 \cdot V_{CC} = 4,25 \text{ V}$  (pour une alimentation sous 5 V). L'alimentation du TUSB3210 se faisant sous 3,3 V seulement, le niveau haut maximal possédant partant la même valeur de 3,3 V, les autres portes de IC4 assurent un niveau haut supérieur à 4 V.

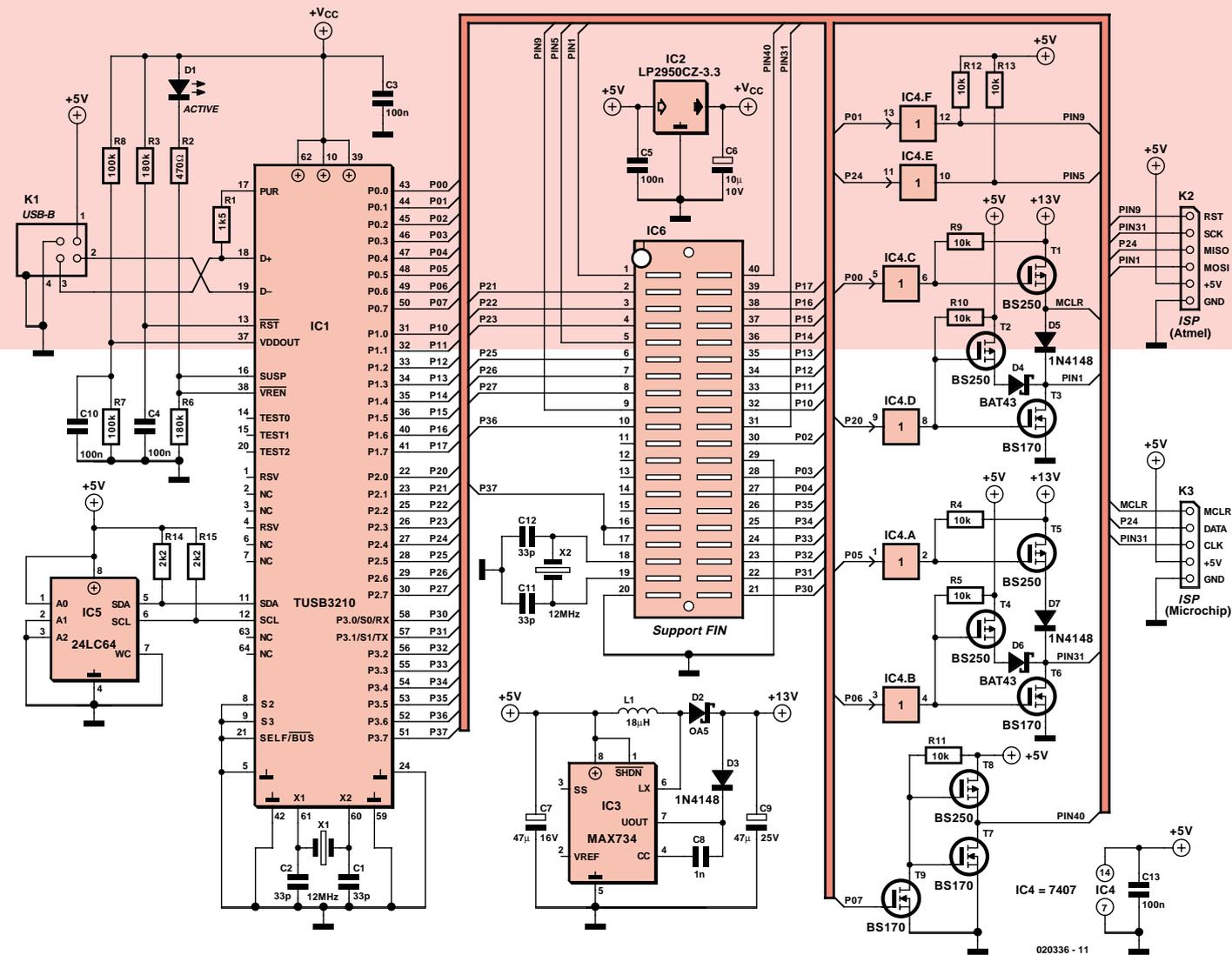


Figure 1. Grâce au TUSB3210 doté de I<sup>2</sup>C et d'USB, le matériel se limite à un convertisseur de niveau et à quelques FETMOS.

## Support de programmation

La plupart des composants peuvent être programmés directement par implantation dans le support IC5. Le quartz X2 constitue la source de signal d'horloge pour les contrôleurs 89Cxx et 89Sxx d'Atmel. De part le nombre limité de ses E/S, le TUSB3210 ne peut programmer qu'un nombre limité de contrôleurs. D'autres microcontrôleurs (plus « étoffés ») se laissent cependant programmer par le biais des lignes ISP. Un examen critique montre que la broche 10 ne constitue pas une vraie masse pour, par exemple, l'alimentation d'un 89Cx051. Comme, en cours de programmation, le courant de service n'est pas très élevé, on peut se contenter du potentiel de masse mis à disposition par la ligne P3.6 de IC1. Bien qu'il soit légèrement rehaussé, il reste dans les limites sûres.

## Logiciel du PC

Le programme tournant sur le PC a été écrit en Delphi 7. Il est possible, dans le Setup (figure 2), de choisir la langue de travail (anglais, allemand, français). Ce paramètre mémorisé dans le Registry est chargé automatiquement lors d'un nouveau démarrage du programme. Le menu Setup offre d'autres options de paramétrage, lecture ou non de (ou des) l'octet de signature, comparaison ou non après programmation (*verify after write*). Le choix du composant se fait dans le menu Device (figure 3). Il existe 2 options : Socket (support de programmation IC5) et ISP Connector (par le biais des embases K2 et K3). Pour le moment, seule est disponible l'option Socket. Il existe le choix entre Atmel MCU et EEPROM série. L'option Atmel MCU offre 3 séries de contrôleurs 89Cx051, 89C5x, 89Sx auxquelles s'ajoutent les 2 AVR 90S1200/90S2313.

Toutes les fonctions telles que bit de verrouillage (*Lock bit 1/2*) et programmation du fusible (*Fuse bit*) sont bien entendu supportées. À noter cependant qu'avec les contrôleurs 89C5x, la programmation des bits de verrouillage est limitée aux 2 premiers bits (*Lock bit 1 et 2*) vu que les lignes de port du TUSB3210 ne sont pas en nombre suffisant pour permettre la programmation du bit de verrouillage 3. En cas de choix des 90S1200/90S2313 il est proposé, lors de la configuration du second bit de fusible, 2 possibilités (*RCEN* et *FSTRT-Fuse bit*). Après détection du contrôleur, le texte change, affichant le fuse bit supporté par le contrôleur identifié. Lors du choix d'une mémoire EEPROM série on retrouve plusieurs fois le même final. 24AAxx et 24CxxC se différencient par la taille de la page, c'est-à-dire le nombre d'octets programmables en une seule opération (2 ms environ).

Plus la taille de page est importante, plus la programmation est rapide.

On peut commander toutes les fonctions de programmation telles que programmation (Écrire), comparaison (Vérifier), effacement (Effacer), lecture (Lire), programmation (Écrire) EEPROM, lecture (Lire) EEPROM, lecture bits fusible/verrouillage (Lire fuse bit) et identification composant (Détecter Device) directement par des boutons soit par le menu déroulant (figure 4). Si l'on procède au choix d'un contrôleur par auto-identification on pourra le vérifier par la fonction Détecter Device. On a alors lecture et traitement des octets de signature du composant et des données telles que taille de la mémoire, tension de programmation voire dénomination exacte en haut à droite sous Device.

En cas de choix d'une opération, on a lecture automatique des octets de signature du contrôleur avant exécution de l'opération en question. Il est possible de sauter ce test en n'activant pas la fonction Lire octet de signature. Ceci peut être nécessaire lorsqu'il est devenu impossible, suite à un problème du microcontrôleur, de lire l'octet de signature.

L'action Lire se traduit par une lecture de la totalité de la mémoire du composant, le nombre d'octets à lire étant dérivé de l'information des octets de signature ou de la fin dans le cas des EEPROM sérielles. En l'absence de lecture de l'octet de signature de contrôleurs, le système utilise la taille de mémoire maximale de la série dont fait partie le composant, ainsi 4 Koctets par exemple pour un 89C4051 vu que le contrôleur le plus « musclé » de la série, le 89C4051, possède une mémoire de 4 096 octets.

Sous le point de menu Édition on peut paramétrer la possibilité de modifier le tampon par le biais de l'éditeur hexadécimal (Éditer Tampon). On peut en outre décider, par l'option (Actualiser

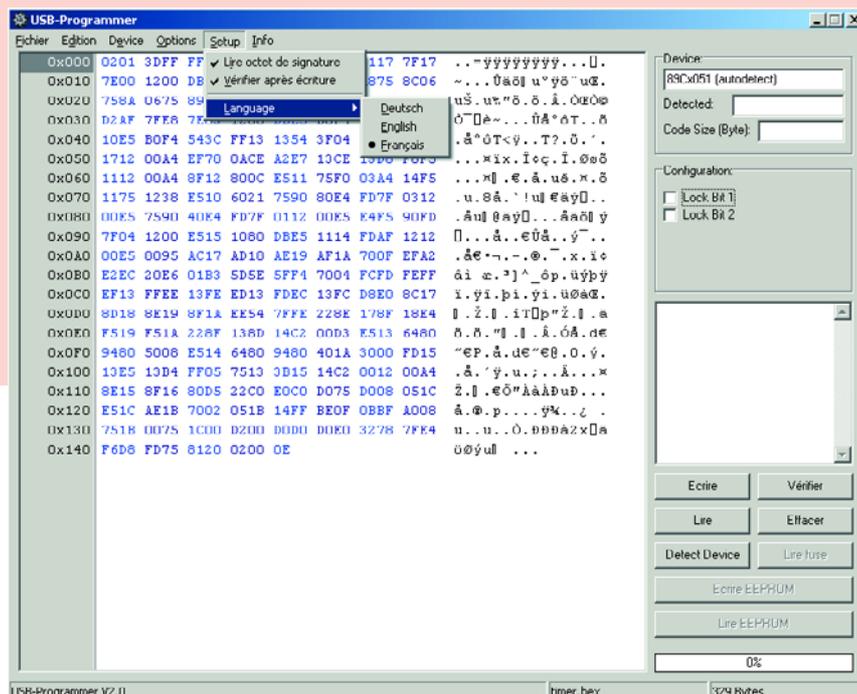


Figure 2. Menu Setup.

tampon <- fichier) s'il faut réactualiser le fichier ouvre auparavant avant chaque lancement d'une opération d'écriture ou de vérification.

## Le protocole

Il faut bien entendu que l'échange de données entre le PC et le programma-

teur soit synchronisé. Ceci implique l'utilisation d'un protocole. Le premier octet envoyé par le PC vers le programmeur comporte l'information de type de contrôleur ou de mémoire : la valeur 1 identifie le groupe 89C051, le 2 les groupes 89C5x et 89Sx. Le second octet identifie l'action choisie : 1 = lecture des octets de signature, 2 = effa-

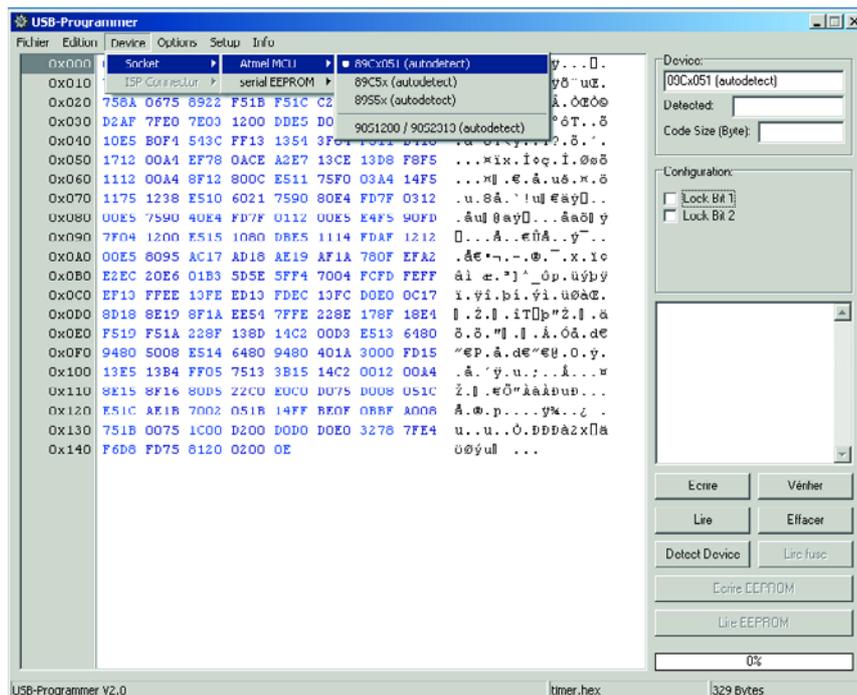


Figure 3. Choix du type de composant.

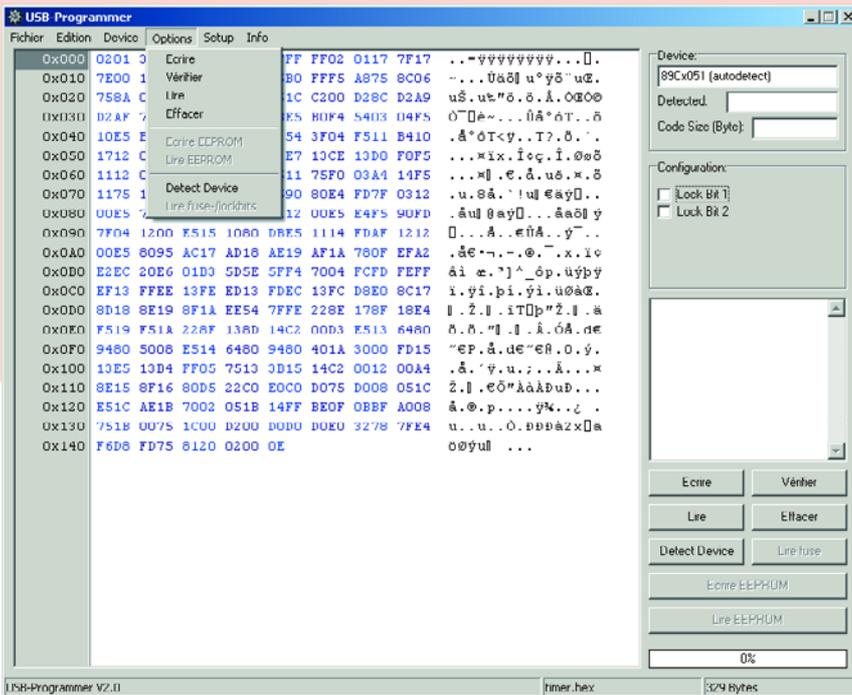


Figure 4. Sous Détecter Device on voit s'afficher en haut à droite les octets de signature et d'autres données.

cer, etc. Les octets suivants contiennent des informations additionnelles, la tension de programmation du 89C5x par exemple, ou encore la taille de page d'une EEPROM sérielle. Pour la programmation, un des octets sert à indiquer le nombre d'octets à envoyer par paquet. Un autre octet indique si le paquet envoyé était le dernier (octet à « 0 ») ou s'il est suivi par d'autres (octet à « 1 »).

Lors d'une lecture, le programmeur est informé du nombre d'octets à lire, taille souvent indiquée en Koctets ou Kbits. Les premiers octets permettent au programmeur de « comprendre » les données.

Après une action du programmeur, le PC reçoit plusieurs octets en retour qui lui permettent de savoir s'il lui faut

envoyer d'autres données de programmation ou s'il peut passer à l'opération suivante.

### Progiciel du programmeur

C'est le compilateur C  $\mu$ Vision2 de Keil qui a servi à la programmation du progiciel. On commence, dans la routine principale, par désactiver le chien de garde et, par la fonction ResetProgrammer(), mettre à 0 V toutes les broches du support de programmation, tous les registres USB étant ensuite définis.

En clôture du logiciel de *bootloader*, IC1 est déconnecté de l'interface USB par déconnexion du 3,3 V de la résistance de forçage au niveau haut (*pull up*) R1 par le biais de la sortie PUR. Le

progiciel chargé doit positionner le bit SDW du registre MCNFG réactivant du même coup cette sortie. Le programmeur est à nouveau activé sur le port USB.

Pour identifier le périphérique USB et en vue de le configurer, le PC (hôte) envoie plusieurs éléments (ou jetons = *token*) de paramétrage qui sont traités à la position Endpoint 0. On a alors, par exemple, transmission de l'adresse unique du périphérique par le biais de laquelle se fera l'adressage du programmeur. De plus, il est demandé, en cours de phase de paramétrage, des descripteurs qui donne à l'hôte des informations concernant l'appareil, ses fonctions et caractéristiques, dont le (*report*) descripteur qui identifie le programmeur comme étant un périphérique compatible HID.

Une fois tous les descripteurs envoyés au PC, l'appareil est initialisé sur le bus USB et prêt à fonctionner. On peut maintenant passer à l'échange des données entre le programmeur et le PC. Le sens des données est indiqué par des « jetons ». Si le TUSB3210 détecte un jeton « IN », les données requises sont transmises du programmeur vers l'hôte, des informations d'état par exemple, mais aussi des paquets de données. À l'arrivée d'un jeton « OUT » on a décodage d'un paquet de données

## HID

L'intérêt d'une initialisation en tant que périphérique compatible HID est qu'il n'est pas nécessaire à un pilote Windows spécial d'échanger des données entre le programmeur et le PC. Windows reconnaît ce standard à partir de sa version 98SE.

Le transfert de données avec un HID se fait par des rapports

(reports). Lors de l'énumération des périphériques (devices) branchés au port USB le PC (hôte) requiert quelques descripteurs. Le descripteur Device comporte des informations concernant l'identité du vendeur (Vendor-ID) et du produit (PID), voire la version USB du périphérique connecté. Le descripteur de configuration comporte des informations de consommation de courant du matériel et du nombre de Endpoints disponi-

bles. Le descripteur de rapport décrit la taille et le nombre de rapports échangés entre le PC et le programmeur. On y définit le nombre d'octets à envoyer et à recevoir, de même que la fonction du périphérique connecté (souris, clavier, manette de jeu, Memory Stick...). Pour plus d'informations concernant USB et HID, un petit tour à l'adresse : [www.usb.org/home](http://www.usb.org/home) s'impose.

envoyé au programmeur en vue de la programmation d'un composant (si tant est que l'on ait choisi alors l'opération « Programmation »).

### Routines de programmation

En vue de réaliser la synchronisation, les données entrantes sont décodées par la routine `DecodeProgrammerData()` dans le fichier `Prog.c`. Le premier octet du rapport qui en comporte 64, comporte l'information identifiant le contrôleur choisi, le second donne l'action à effectuer. Pour cela, il est fait appel à différents algorithmes de programmation en respect des fiches de caractéristiques des différents contrôleurs.

Chaque groupe de composants comporte une routine d'activation qui applique les tensions requises aux broches correctes et forcent les lignes de programmation utilisées aux niveaux définis. Une fois une opération terminée avec succès, on appelle la routine de réinitialisation du programmeur qui remet à 0 V toutes les lignes du support de programmation. Comme, lors d'une programmation, on a la plupart du temps besoin de plus de 64 octets, le PC doit envoyer de nouveaux paquets de données au programmeur dès que les données en cours ont été traitées. Celui-ci envoie un caractère défini au programme du PC pour lui indiquer qu'il peut envoyer le paquet suivant.

Le PC prépare le paquet suivant et l'envoie au périphérique. Le dernier paquet adressé au programmeur comporte un octet de valeur « 0 ». Lors de la lecture d'un composant les données sont en un rapport de 64 octets. Le premier octet y indique le nombre d'octets valides que comporte le paquet.

Pour de plus amples informations sur le programme nous vous recomman-

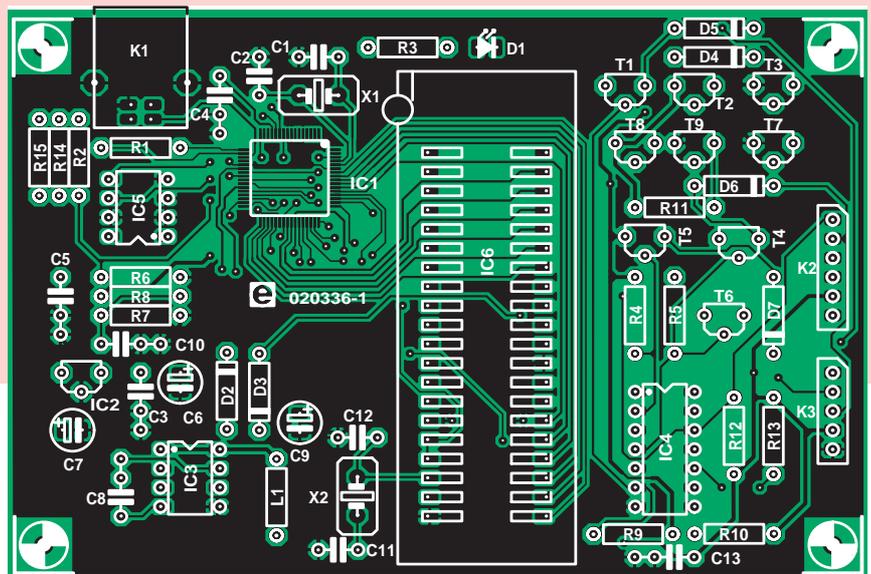


Figure 5. Sérigraphie de la platine double face.



Figure 6. Miniature et épouvantable (à souder) : le contrôleur USB en boîtier SPFP à 64 pattes.

ons d'examiner le programme mis à votre disposition sur notre site ([www.elektor.fr](http://www.elektor.fr)).

### Réalisation et mode d'emploi

Nous pourrions qualifier la réalisation de cette platine de jeu d'enfant s'il n'y avait pas cet épouvantable TUSB3210 « multi-pattes » en boîtier S-PFP-G64. Le soudage de ce type de composants requiert une main ferme et des nerfs à toute épreuve. Après fixation du circuit (à l'aide d'une goutte de colle) nous allons faire exactement ce qu'il ne faut jamais faire : non seulement sou-

der les pattes à leurs îlots, mais aussi entre elles, et cela le plus vite possible pour éviter que le composant ne chauffe trop. Après cette mise en court-circuit impériale et après laissé le temps au circuit de refroidir, nous allons éliminer cet excédent de soudure (les ponts de court-circuit) à l'aide de tresse à dessouder que l'on place perpendiculairement aux broches (ne pas utiliser de pompe à dessouder !). L'important est de faire attention à ce que le circuit intégré ne chauffe pas trop. Il reste à vérifier à l'aide d'une loupe et d'un multimètre que les pattes sont bien soudées et qu'elles ne sont pas en court-circuit avec leurs voisines.

# Composants supportés actuellement

Il est facile de réactualiser le progiciel du programmeur. Il suffit, lorsque cela est devenu nécessaire, de réécrire le nouveau progiciel dans l'EEPROM et de remplacer le programme sur le PC.

Actuellement, le progiciel permet la programmation des contrôleurs et EEPROM énumérées ci-dessous. Très bientôt, nous pensons avoir une mise

à jour pour les contrôleurs PIC et ATmega qui sera, comme les mises à jour ultérieures (programme pour le PC inclus) disponibles gratuitement sur le site d'Elektor sous la dénomination 020336-11.

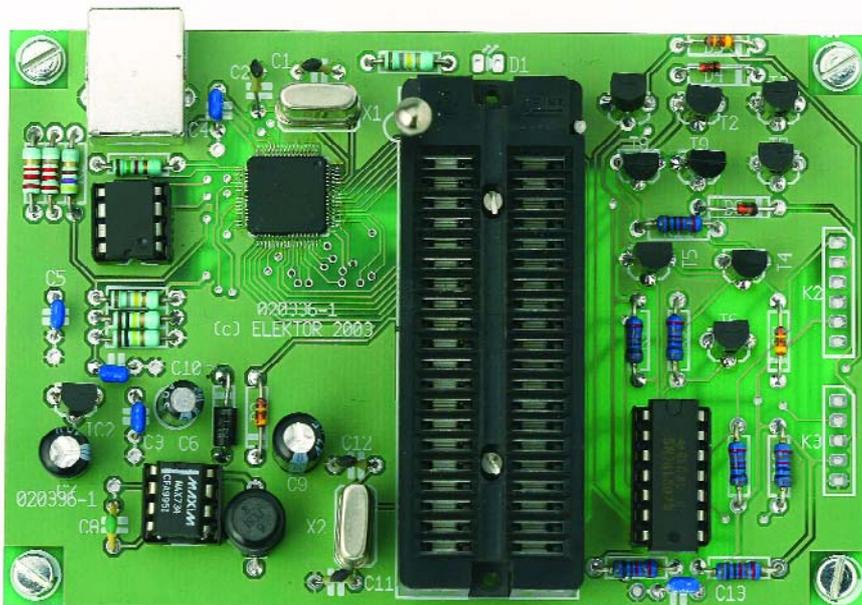
## Contrôleurs Atmel :

89C1051, 89C2051, 89C4051,

90S1200, 90S2313, 89C51, 89C52, 89C55, 89LV51, 89LV52, 89LV55, 89S53, 89S8252

## EEPROM :

24xx00, 24xx01, 24xx02, 24xx04, 24xx08, 24xx16, 24xx32, 24xx64, 24xx128, 24xx256, 24xx512



Une fois que vous avez soudé le TUSB3210 correctement sur la platine, nous pouvons vous qualifier d'expert et ne perdrons pas de temps à décrire le reste de cette étape. Tous les circuits intégrés, hormis le petit régulateur, seront mis sur support. Le support FIN (à Force d'Insertion Nulle) ne sera pas soudé directement sur la platine mais glissé dans des barrettes support.

Après inspection et vérification de son travail, il sera temps d'effectuer un premier essai. Il n'est pas besoin, en cas de présence d'une EEPROM programmée, de pilote pour Windows. Lors de la connexion du programmeur au port USB on devrait le voir apparaître dans le gestionnaire comme un périphérique compatible HID. Vous pouvez maintenant effectuer votre première programmation.

En l'absence d'EEPROM sérielle, il faudra transférer le progiciel par la liaison USB. Le *bootloader* du TUSB3210 se manifeste et Windows reconnaît le

nouveau périphérique requérant un pilote de Texas Instruments. Ce pilote ne se trouve ni sur la disquette ni sur notre site mais il est téléchargeable gratuitement depuis le site de Texas Instruments ([www.ti.com](http://www.ti.com)). On choisira le répertoire comportant le fichier **TUSB3210.inf**, installe le pilote puis le fichier **ApLoader.sys** du même répertoire. Il reste à entrer, lorsque le système le demande, le répertoire du progiciel. Il suffira de donner le cheminement du fichier **TUSB3210.bin**.

Celui-ci est alors, avec le fichier **Aploader.sys**, copié automatiquement dans le répertoire `/System32/drivers`. Lors d'une nouvelle connexion du programmeur, le pilote envoie le progiciel du fichier `/System32/drivers/TUSB3210.bin` vers le programmeur de façon à ce que ce dernier puisse traiter très rapidement le progiciel. D'Artagnan apparaît maintenant en tant que périphérique HID dans la liste USB.

(020336-1)

## Liste des composants

### Résistances :

R1 = 1k $\Omega$   
R2 = 470  $\Omega$   
R3, R6 = 180 k $\Omega$   
R4, R5, R9 à R13 = 10 k $\Omega$   
R7, R8 = 100 k $\Omega$   
R14, R15 = 2k $\Omega$

### Condensateurs :

C1, C2, C12, C13 = 33 pF  
C3 à C5, C10, C13 = 100 nF  
C6 = 10  $\mu$ F/16 V radial  
C7, C9 = 47  $\mu$ F/16 V radial  
C8 = 1 nF

### Semi-conducteurs :

D1 = LED rouge  
D2 = OA5 ou 1N5817 (Farnell 573-097)  
D3, D5, D7 = 1N4148  
D4, D6 = BAT43  
IC1 = TUSB3210PM  
IC2 = LP2950CZ-3.3 ou LE33CZ (Farnell 302-4568)  
IC3 = MAX734CP  
IC4 = 74LS04  
IC5 = 24LC64  
IC6 = support FIN à 40 broches (à fentes larges)  
T1, T2, T4, T5, T8 = BS250  
T3, T6, T7, T9 = BS170

### Divers :

K1 = embase USB en équerre encartable  
K2 = embase à 1 rangée de 6 contacts  
K3 = embase à 1 rangée de 5 contacts  
L1 = self 18  $\mu$ H  
X1, X2 = quartz 12 MHz  
Platine 020336-1  
Disquette 020336-11 (Logiciel PC+ Progiciel avec codes-source)

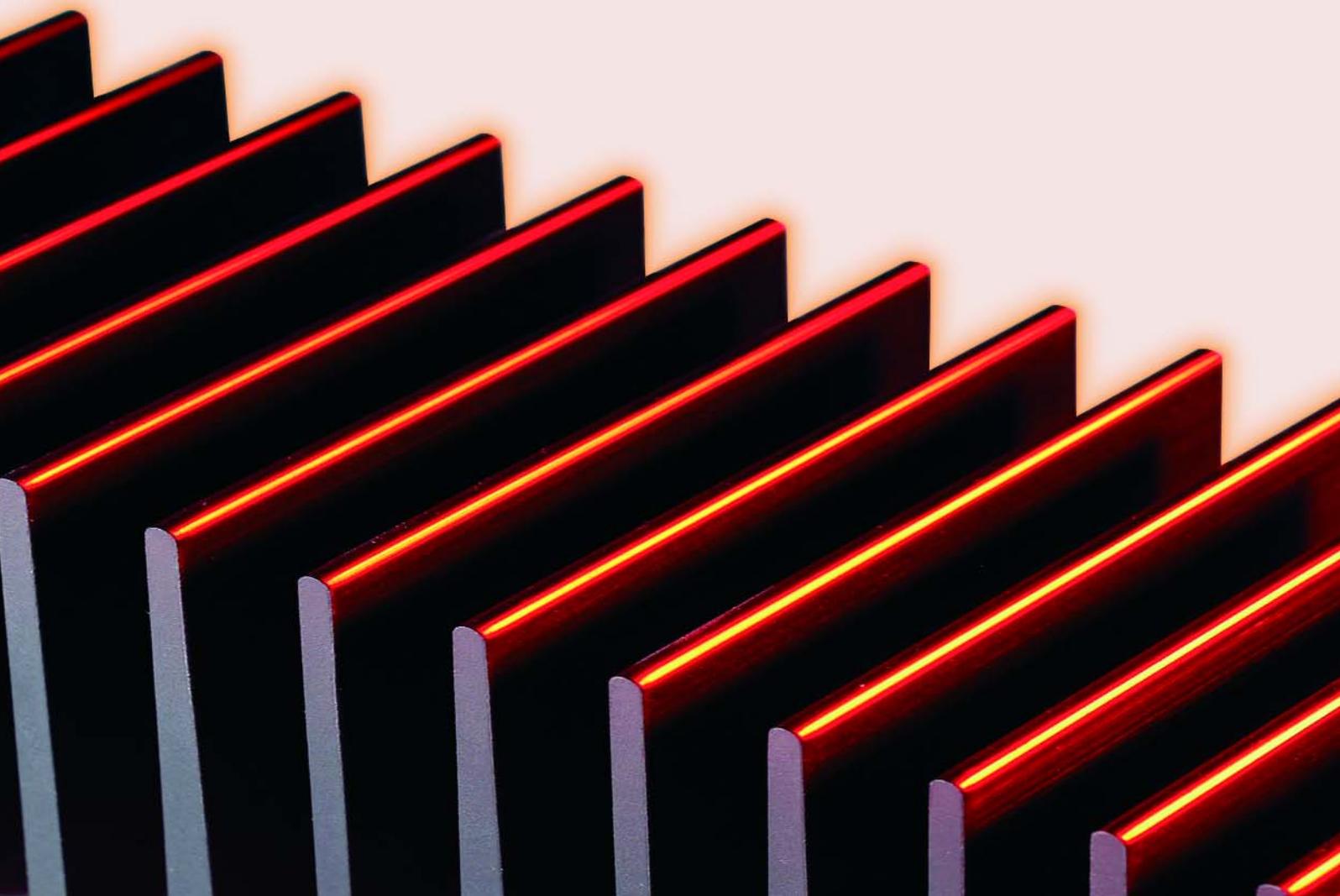
Le dessin de la platine et le logiciel sont également disponibles au téléchargement depuis le site [www.elektor.fr](http://www.elektor.fr), le dessin des pistes étant également proposé dans les pages LAYOUT-EXTRA

**Harry Baggen**

# *LA CLASSE...*

## **Amplificateurs audio de A à T**

L'amplificateur final constitue la source de puissance de toute installation audio. Il doit transformer, avec le moins de distorsion possible, une toute petite tension alternative en un fort signal pour attaquer les haut-parleurs. Au fil du temps, les concepteurs ont imaginé différentes techniques pour y parvenir. Mais tout commence par la classe A.



## Beaucoup de raffut à peu de watts

Pour beaucoup de gens, choisir les caractéristiques d'un amplificateur consiste avant tout à déterminer la puissance voulue (votre ampli fait 2 x 40 W ? Le mien en sort 2 x 70 !). Elle joue un rôle, c'est certain, mais moindre que ce que l'on pense souvent.

Vous pourriez faire un grand vacarme avec quelques watts, il suffit d'utiliser des enceintes acoustiques qui délivrent 86 dB pour 1 W (c'est une caractéristique communément indiquée dans les spécifications du fabricant). Avec 2,5 W, elles vous fourniront déjà 90 dB. Pour atteindre 100 dB, 25 W suffiront. C'est déjà singulièrement violent et dommageable pour nos oreilles.

Augmenter le niveau acoustique de 6 dB nous apparaît comme un doublement du son, mais pour l'obtenir, il faut quadrupler la puissance. Donc si vous voulez acheter un amplificateur final sensiblement plus puissant que le vôtre actuellement, il vous faudra en choisir un dont la puissance est au moins quatre fois supérieure.

Pas facile de délivrer beaucoup de puissance, quand on est un amplificateur ! Il faut un gain en tension et un gain en courant pour transférer une puissance suffisante vers les haut-parleurs. Les enceintes ne restituent sous forme sonore que quelques centièmes de la puissance reçue, il faut donc pas mal de watts pour atteindre, dans une pièce d'habitation, un niveau d'écoute acceptable. En concert ou lors des manifestations en plein air, il en faut encore bien davantage, on parle alors de milliers de watts.

Dans les dernières décennies, on a développé divers concepts pour produire, à l'aide de transistors ou de FET, un signal de haute qualité tout en améliorant le rendement de l'étage de sortie. Nous ne parlons pas ici d'amplificateurs à tubes.

Lors de l'étude d'un étage de sortie, le concepteur doit tenir compte des caractéristiques particulières des semi-conducteurs utilisés. Si nous pouvions travailler avec des transistors parfaits, qu'ils soient bipolaires ou à effet de champ, construire un bon amplificateur serait bien plus facile.

Mais malheureusement, les semi-

conducteurs souffrent tous de non linéarité en amplification du signal, ce qui cause de grosses difficultés lorsqu'on opère sur signaux analogiques. Mais une rétroaction précise nous permettra d'amenuiser ces inconvénients. D'autre part, des phénomènes désagréables se produisent parfois, selon la configuration choisie, comme la fameuse distorsion de croisement.

En outre, sur un gros amplificateur, il faut aussi envisager sérieusement la production de chaleur qui risque d'occasionner de graves dérives thermiques, comme le déplacement du point de fonctionnement et la distorsion par modulation thermique. On a coutume de répartir les amplificateurs de sortie en différentes classes selon la structure de l'étage final. Elle détermine pour une bonne part l'efficacité et la qualité, parce que c'est là que se déroule vraiment l'amplification en puissance. Chaque organisation d'amplification se désigne par une lettre de l'alphabet qui n'a aucune relation avec la méthode de fonctionnement. On a démarré, en ce temps-là, par la lettre A...

### Classe A

Nous commençons donc avec la plus simple, mais aussi une des meilleures configurations pour une reproduction de qualité, l'étage final en classe A. Dans sa version de base, nous pourrions y mettre en œuvre un simple émetteur suiveur (figure 1).

Le courant de repos du transistor est égal

Figure 1. Sur un étage final en classe A, le rendement est fort bas, mais il n'y a pratiquement aucune distorsion de croisement.

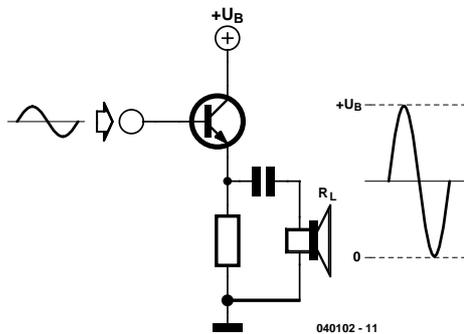


Figure 2. Sur un montage en classe B, chaque transistor conduit pendant une demi-période de la sinusoïde. L'ennui se situe autour du passage par zéro.

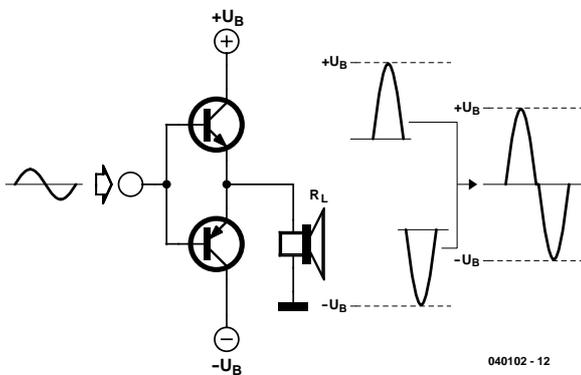


Figure 3. La classe G fait appel à une alimentation glissante dont la tension se règle en permanence sur l'amplitude du signal.

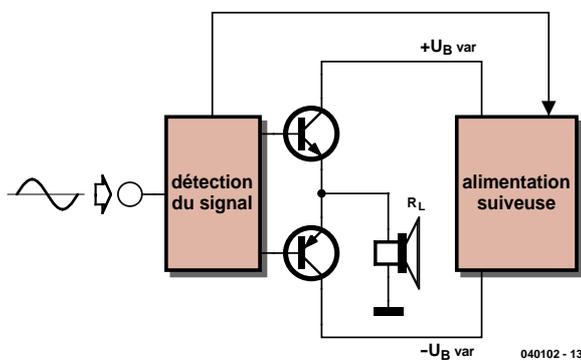
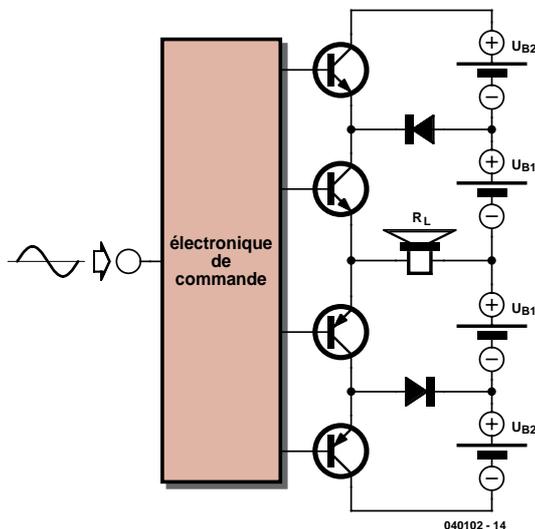


Figure 4. La classe H se contente de commuter entre plusieurs tensions d'alimentation, deux dans ce cas-ci.



au maximum du courant alternatif de sortie, le transistor se situe alors au milieu de sa plage de fonctionnement et se contente de conduire plus ou moins selon la tension alternative appliquée. Son rendement est bas, 25 % au maximum, lorsque le signal est maximum, encore moins dans les autres cas. Dans un montage symétrique à deux transistors, le rendement s'améliore, mais il ne dépassera jamais les 50 %.

## Classe B

La configuration en classe B fait appel à deux transistors qui conduisent exactement pendant la moitié de la période. Au repos, aucun courant ne circule dans les transistors. Le rendement d'un étage final polarisé en classe s'élève à environ 78 %, mais son principal inconvénient réside dans la distorsion de transition qui survient au moment où la conduction doit passer d'un transistor à l'autre. La cassure nette au pied de la courbe caractéristique empêche les deux alternances du signal de se raccorder exactement, il s'ensuit de la distorsion de croisement bien connue, une altération très audible de la forme du signal. Pour résoudre ce problème, on a combiné les classes A et B pour donner la classe AB. Il s'agit d'une classe B dans laquelle un petit courant circule au repos, ce qui fait qu'en régime de petits signaux, l'étage final travaille en classe A. La plupart des amplificateurs de puissance actuels utilisent encore cette méthode, avec de nombreuses variantes. Le rendement reste voisin de celui de la classe B.

## G et H

Hé là ! On a sauté des classes ! Oui, mais c'est à dessein. Les classes C, E et F existent aussi, mais ne conviennent guère que pour des applications à haute fréquence, elles sortent quelque peu du cadre de l'audio. Et le concept de la classe D s'écarte tellement de A et B que nous le traiterons séparément plus loin. Intéressons-nous d'abord aux classes G et H qui ont un point commun important. Sur les deux, la tension d'alimentation s'adapte à l'amplitude du signal de commande. En classe G (figure 3), la tension d'alimentation se règle continuellement sur la grandeur du signal de sortie désiré. Depuis l'avènement des alimentations modernes à découpage, il est assez simple de forcer la sortie à calquer une tension d'entraînement, à condition évidemment d'aménager la régulation pour que l'alimentation réagisse suffisamment vite au signal de commande de l'étage final.

En classe H (figure 4), c'est le même processus qui est utilisé, sauf qu'ici l'alimentation ne varie pas en permanence, mais on commute entre différentes tensions, souvent deux. De cette façon, il est possible de réduire considérablement la dissipation à forte puissance dans l'étage de sortie.

## Classe D

Le « D » n'a rien à voir avec « digital », c'est pure coïncidence. Pourtant, l'amplificateur, dans cette technologie, va commuter au rythme d'une modulation en largeur d'impulsion (figure 5). Un comparateur évalue l'écart entre le signal d'entrée et une onde triangulaire, en fonction de quoi il commute l'étage final entre le positif et le négatif de la tension d'alimentation. La fréquence de récurrence est très élevée, souvent 10 fois, voire davantage, que la largeur de bande, donc à 200 kHz ou plus.

La largeur des impulsions varie ainsi en relation avec le signal d'entrée. Si nous intercalons un filtre passe-bas derrière l'étage de sortie, il intégrera les impulsions du signal et donnera un signal analogique conforme à celui d'entrée, mais fortement amplifié.

Comme le seul travail de l'étage final se résume à une commutation, le rendement est très élevé. Mais cette méthode pêche par certains défauts. Il n'est pas simple de préserver de toute distorsion la forme du signal, il faut absolument un solide filtre de sortie et des mesures drastiques pour assourdir les signaux parasites. Obtenir un faible taux de distorsion exige dans tous les cas l'emploi d'une rétroaction, analogique ou numérique.

## Les classes S et T

Bien que le principe de l'amplificateur en classe D totalise déjà quelques décennies d'ancienneté, il ne s'est jamais vraiment imposé en Hi-Fi. En cause surtout, le manque de bons semi-conducteurs (FET de puissance rapides) et l'excès de distorsion. Entre-temps, plusieurs constructeurs y sont allés de leurs variations sur le thème, tout en les accompagnant de leurs propres dénominations. Ainsi Crown a imaginé la classe I, Sony a développé sa technologie S-master et Tripath a conçu ses amplificateurs de classe T. Et le bel ordre alphabétique s'est évanoui dans les abréviations choisies par les fabricants.

Dans la technique S-master, Sony a combiné plusieurs méthodes pour rendre la classe D apte à figurer dans la chaîne Hi-Fi de salon. La conversion en un train d'impulsions dont la largeur reflète la forme du signal incident s'appelle ici C-PLM, modulation de largeur d'impulsion complémentaire. Elle accorde un maximum d'attention à l'affaiblissement de l'instabilité temporelle, par l'emploi d'un signal d'horloge extrêmement précis et d'un circuit (*clean data cycle*) qui corrige, si nécessaire, la position des impulsions de sortie (cf. **figure 6**). Une particularité de l'approche Sony, c'est certainement le réglage de volume. En classe D normale, on retrouve la totalité de l'onde carrée en sortie sous une amplitude de 50 à 100 V (pic à pic). Mais en signaux faibles, il devient très difficile d'éliminer par filtrage les résidus d'impulsions. Dans le système Sony, le volume se règle par adaptation de la tension d'alimentation de l'étage final, la variation peut atteindre 50 dB. De cette façon, il n'y a aucune perte d'information à craindre à bas niveau.

La firme Tripath a développé une technologie qui, selon ses dires, combine la qualité des classes A et AB tout en affichant un haut rendement de 80 à 90 %. Elle associe des circuits

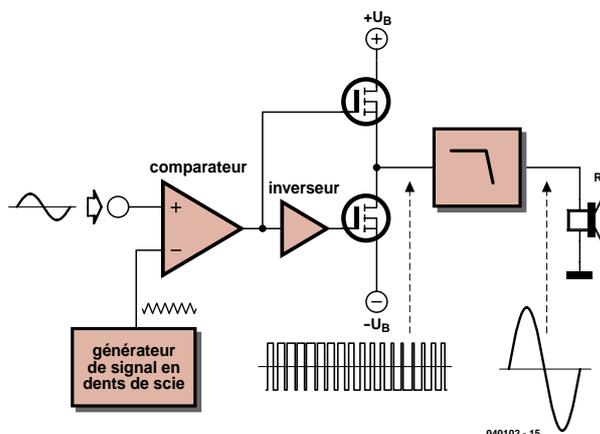


Figure 5. La classe D est constituée d'un modulateur en largeur d'impulsion à sortie de puissance et d'un filtre passe-bas.

analogiques et numériques ainsi que des algorithmes numériques pour moduler le signal incident par des salves modélisées à haute fréquence. Les algorithmes que Tripath a façonnés dérivent de ceux déjà utilisés en télécommunication, adaptatifs et prédictifs.

Sur l'amplificateur Tripath, la majeure partie de l'électronique numérique et analogique est embarquée dans une seule puce, entourée ou non de transistors de sortie, selon la puissance demandée. Le **figure 7** en reproduit le bloc-diagramme. Le signal incident est d'abord tamponné par l'étage d'entrée. Il est ensuite appliqué au bloc *Digital Power Processing* qui contient le processeur de signal adaptatif, un convertisseur numérique, le circuit de silencieux, la protection contre les surcharges et la détection d'erreur. L'étage final de puissance est commandé par la logique de qualification et le haut-parleur y est connecté à travers un filtre de sortie.

Grâce à ses algorithmes spéciaux, le processeur de l'amplificateur en classe T est en mesure de tenir compte des caractéristiques spécifiques des transistors de puissance employés : les imperfections de commutation, le défaut de symétrie des transistors complémentaires de sortie, la déformation due aux temps morts et les résidus de commutation dans la bande audio.

La fréquence de commutation en classe T s'adapte sans arrêt à la grandeur du signal d'entrée. Aux faibles niveaux, elle est relativement élevée, de l'ordre de 1,2 MHz. C'est tout profit pour la qualité du signal. Lorsque le niveau d'entrée s'élève, la fréquence baisse pour améliorer le rendement. À pleine puissance, la commutation s'opère aux alentours de 200 kHz. Il y a aussi une sorte de réduction de bruit sur les sommets du

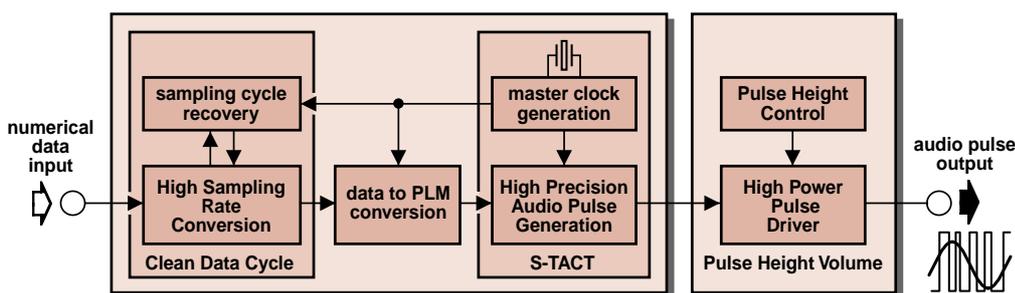
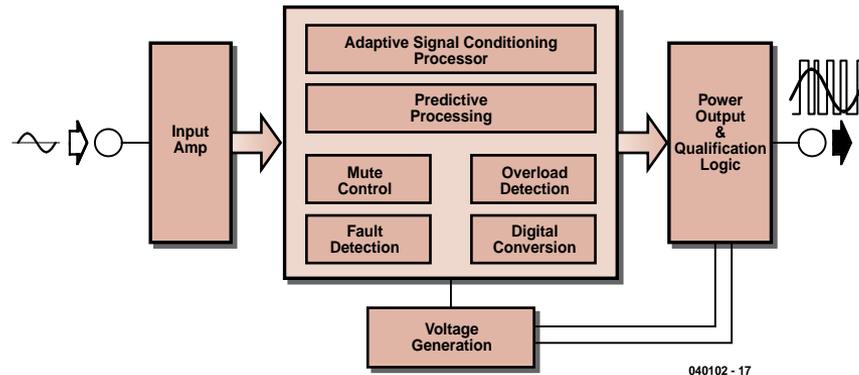


Figure 6. Bloc-diagramme d'un amplificateur Sony S-master.

Figure 7. Chez Tripath, la classe T est une ingénieuse variante de la classe D dans laquelle un processeur surveille en permanence le signal d'entrée et corrige les écarts de commutation.



signal de sortie pour en améliorer la forme. Toutes ces dispositions concourent à donner à l'audiophile la sensation que son amplificateur en classe T reproduit le son comme les modèles analogiques.

## L'avenir

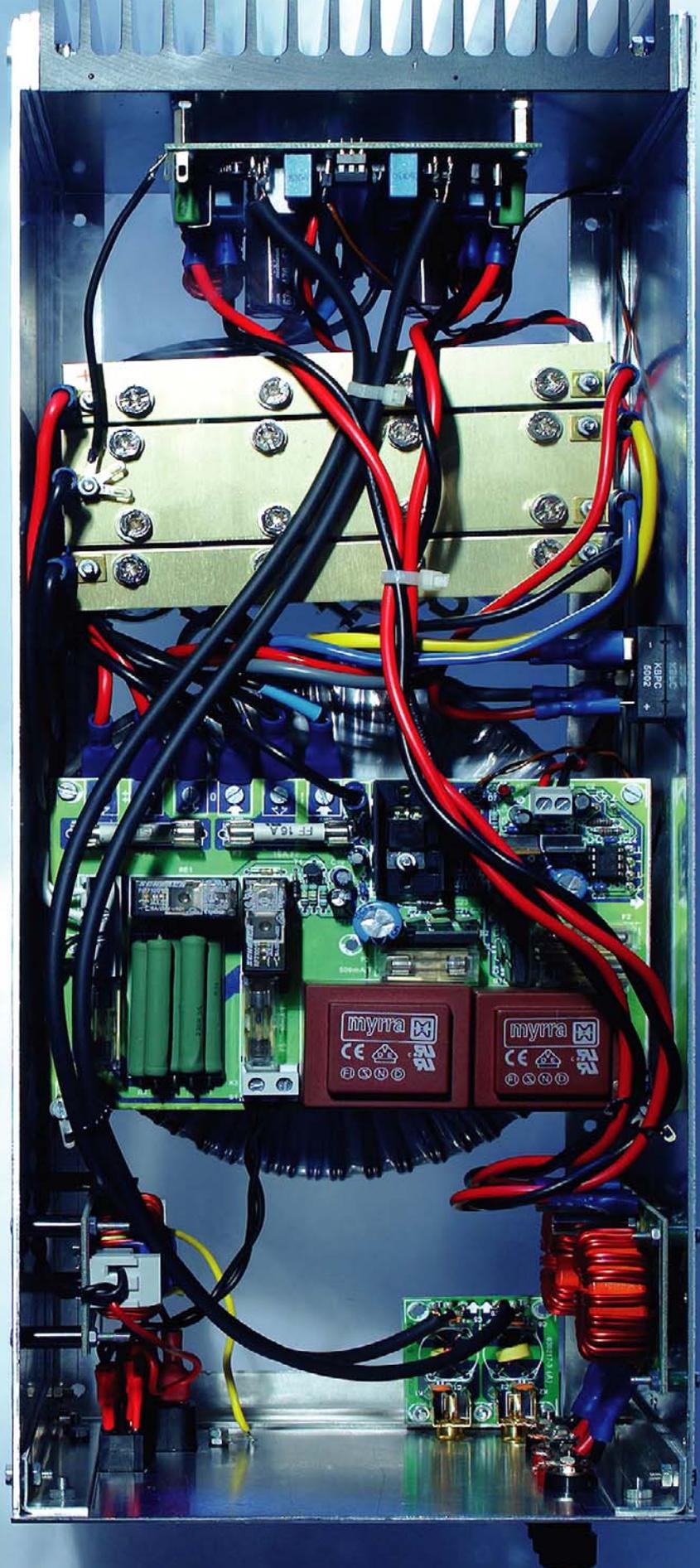
Les progrès de l'audio numérique amèneront finalement les variantes numériques des amplificateurs de puissance à

concurrencer les autres dans le domaine de la grande consommation. Les raisons en sont le meilleur rendement, la compacité et le prix inférieur. Vous dire s'ils perceront également dans le haut de gamme serait prématuré. Pour l'instant, les amplificateurs numériques de qualité sont encore rares sur le marché. Mais peut-être voudrez-vous en juger par vous-même en construisant l'amplificateur Clarity proposé dans ce numéro ?

# Clarity 2x300W

Ton Giesberts

**Cet amplificateur de pointe prouve que grande puissance ne rime pas forcément avec encombrement et poids lourd. Haut rendement, donc compacité et cependant des spécifications qui surpassent sans peine nombre de modèles conventionnels.**



# en classe T

N'allez pas imaginer non plus que l'amplificateur construit sera léger comme la plume au vent, nuançons quelque peu. Deux fois 300 W de puissance réelle, il faut les lui fournir et l'alimentation sera la seule partie comparable avec les autres amplificateurs. Mais par une modulation en largeur d'impulsion bien pensée, cet amplificateur a un rendement tel que le radiateur, et donc le boîtier aussi, gardera des dimensions réduites. En outre, il ne s'agit pas simplement d'un amplificateur à « largeur d'impulsion », il est bâti sur le pilote audionumérique TA3020 en classe T de Tripath, qui offre d'excellentes caractéristiques et peut rivaliser avec des amplificateurs de haut de gamme. Si vous désirez en savoir plus sur les étages audio de puissance, allez donc jeter un œil à l'article « La classe... » dans cette même édition.

Le projet se base dans une large mesure sur l'application standard et les références du fabricant. Sa géométrie détermine largement la qualité d'ensemble. Mais il faut y ajouter, en raison de la nature du montage (haute fréquence de commutation et forts courants), beaucoup d'attention dans le choix de certains composants. Nous devons effectivement utiliser des condensateurs de découplage et des électrolytiques spéciaux. Même pour assurer le couplage thermique des transistors de sortie, nous n'utiliserons pas une simple feuille de mica ou un film de polyamide (Kapton), mais bien un matériau céramique épais de plusieurs millimètres. La puce réclame encore deux tensions auxiliaires, pour lesquelles nous avons conçu une platine séparée, avec un retard à l'enclenchement du secteur pour le transformateur principal et deux bons gros fusibles sur la première tension d'alimentation.

En plus, pour affaiblir les parasites (compatibilité électromagnétique oblige) nous avons ajouté des filtres aux entrées et sorties. Cela vous donne déjà une idée de ce qui vous attend, mais cette première partie ne couvrira que les fonctions essentielles du CI Tripath.

Nous pouvons constater à la **figure 1** que le circuit intégré se compose de trois blocs par canal, un étage d'entrée analogique (amplificateur inverseur), l'unité de traitement et de modulation et enfin les étages pilotes (adaptateurs de niveau) pour les MOSFET de puissance externes. Il y a aussi une protection en courant, contre les sous et surtensions ainsi qu'un silencieux qui, ensemble, déterminent si les sorties de l'amplificateur doivent être actives ou non.

## L'étage d'entrée

L'entrée analogique consiste en un amplificateur inverseur parce qu'il permet plus commodément de définir le gain et la largeur de bande. Il faut savoir que le signal nécessaire à la commande du modulateur ne peut pas dépasser  $4 V_{pp}$ . Nous avons fait en sorte que, pour une sensibilité de  $1,13 V_{eff}$  à plein débattement, il y ait  $3,2 V_{pp}$  derrière l'étage d'entrée. Le rapport entre R3 et R2 (R24 et R23 sur l'autre canal) définit le gain de l'étage d'entrée, il vaut donc un, comme on le voit dans le schéma de la **figure 2**. C2 (ou C15) améliore la stabilité de l'amplificateur de tête et amoindrit les parasites HF par limitation de la bande passante à quelque 240 kHz. C1 (ou C14) détermine la coupure basse qui se situe ici aux environs de 2,5 Hz. On garantit ainsi la meilleure rectitude possible de la caractéristique de fréquence dans tout le spectre audio. C1

et C14 sont des modèles MKT parce que nous essayons d'éviter par principe la présence d'électrolytiques sur le trajet du signal.

Les résistances de R4, R5 et P1 associées au condensateur de découplage C3 (R25, R26, P2 et C16) concourent à minimiser le décalage de polarisation en sortie de l'amplificateur.

## Le modulateur

Le modulateur remonte le signal de l'étage d'entrée jusqu'au niveau de sortie. Il opère donc la deuxième partie de l'amplification globale ou plus exactement constitue le véritable étage à gain. Le processeur confectionne une trame d'impulsions en fonction du niveau de signal et de la fréquence. En l'absence de signal, la fréquence de commutation moyenne avoisine les 700 kHz et peut varier au maximum entre 200 kHz et 1,5 MHz. Deux pilotes à transistors MOSFET complémentaires avec décalage de tension déplacent le signal jusqu'au niveau approprié pour polariser les grilles des MOSFET de puissance. Ils sont alimentés par VN10 sous 10 V, qui profite principalement du découplage offert par C39 (ou C38). C'est pourquoi ces condensateurs se situent aussi près que possible des broches correspondantes de la puce. Côté négatif, LO1COM (la source de T2) et VN10 forment les raccordements d'alimentation du pilote. Côté positif, pendant que la tension de sortie est négative, le condensateur auto-élévateur C7 (C20) se charge via D5 (D12) jusqu'à une tension d'une dizaine de volts qui glisse en concordance avec la sortie lorsque T1 se met à conduire. Cette tension est appliquée à VBOOT qui, avec HO1COM (la source de T1 et aussi la sortie du pont), constituent les

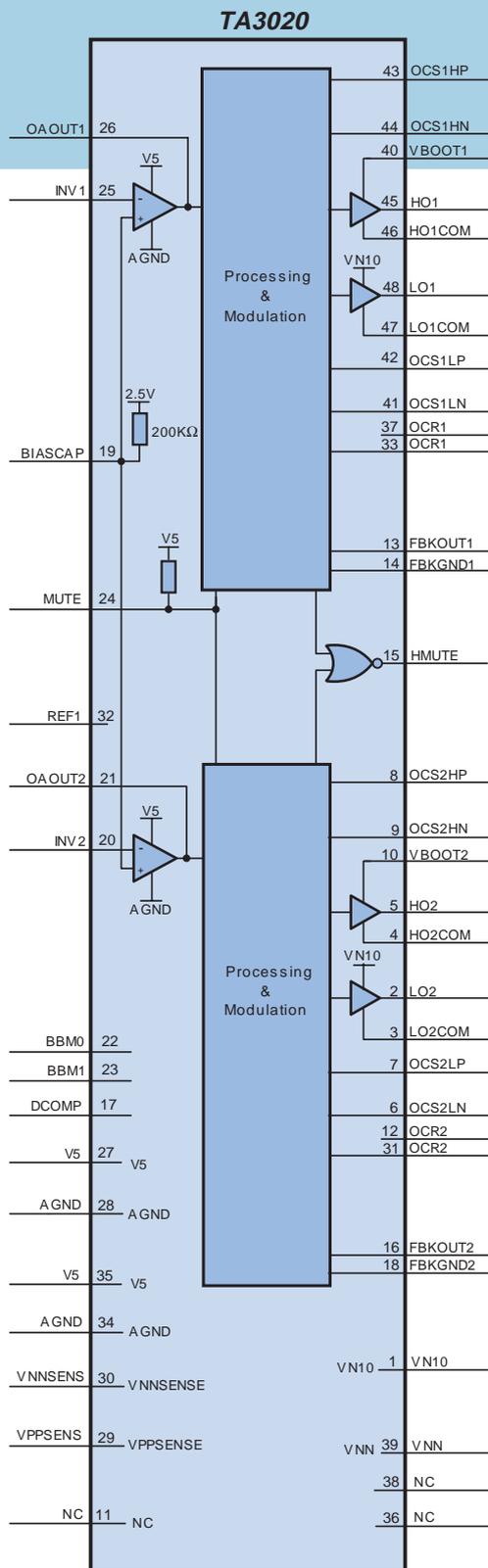


Figure 1. Les entrailles du TA3020 de Tripath.

bornes de raccordement d'alimentation du pilote. En régime d'écrêtage, C8 (C21) constitue un tampon supplémentaire parce qu'alors la fréquence de commutation est plus basse. La résistance R13 (R34) limite le courant de charge de C8 au moment de la mise sous tension de l'amplificateur.

### Les MOSFET

Dans chaque canal, les deux MOSFET à canal N, T1 et T2 (T3/T4) forment la moitié d'un pont. Les sorties des polariseurs placent donc à tour de rôle les MOSFET en conduction. On se réserve un temps mort (absence de chevauchement) de quoi s'assurer que deux transistors ne puissent être conducteurs simultanément, pour éviter le court-circuit. Ce délai s'ajuste à l'aide de deux cavaliers, JP1 et JP2. Il n'est pas recommandé d'exécuter le réglage par expérimentations successives, mais plutôt de s'en tenir au réglage standard conseillé ! Il n'y a que pour des MOSFET d'une capacité de grille sensiblement plus basse, et de plus faible puissance, que l'on peut éventuellement raccourcir le temps mort, mais pas dans cette application-ci. Les résistances de grille R7 et R9 (R28/R30) déterminent la vitesse de montée et du même coup le dépassement qui survient lors de la commutation. Elles limitent aussi, dans une certaine mesure, la puissance qui serait sinon perdue dans les pilotes (il s'agit de résistances de 1 W) à cause de la charge et décharge de la capacité de grille de chaque MOSFET. Les diodes D1 et D2 (D8/D9) permettent une décharge plus rapide des grilles, ce qui réduit le temps de descente et le risque d'une conduction simultanée de T1 et T2. Nous avons ajouté R8 et R9 (R29/R31) par prudence, elles déchargent les grilles des MOSFET s'il n'y a pas de CI installé. Les ôter exposerait à des conséquences désastreuses dues aux

résistances de fuite et aux tensions parasites. R6 et R11 (R27/R32) sont des modèles à basse inductance pour la sécurité en courant, nous allons y revenir. R12 et C4 (R33/C17) constituent un réseau d'amortissement qui se place le plus près possible de T2 (T4). Les diodes D6 et D7 (D13/D14) sont branchées entre source et drain des transistors pour réduire les dépassements de tension causés principalement par la bobine du filtre de sortie quand de forts courants y circulent. Ces diodes-ci sont des modèles CMS, elles se soudent également au plus près des broches concernées pour assurer la protection de la puce. Les diodes D3 et D4 (D10/D11) sont positionnées entre drains respectifs ainsi qu'entre sources des MOSFET et contribuent encore à atténuer les dépassements de tension. Toutes ces diodes (D1 à D14) doivent présenter un temps de recouvrement extrêmement faible. Les condensateurs C5 et C6 (C18/C19) constituent le découplage du demi-pont, ils ont pour mission de réduire l'ondulation de la tension d'alimentation, ce qui améliore d'autant le fonctionnement des transistors et du circuit intégré. Encore une fois, il faut les installer au plus près des bornes des MOSFET. Le condensateur C6 doit être un électrolytique caractérisé par une très basse RSE (résistance série équivalente) et d'excellentes propriétés en HF. Il ne faut pas choisir un autre type que celui proposé, à moins que ses spécifications soient égales ou supérieures. Un condensateur électrolytique ordinaire risque fort d'exploser ou de n'avoir qu'une très courte vie. Le signal modulé en largeur d'impulsion rejoint les bornes de sortie à travers le filtre LC L1/C9 (L2/C22).

## Le filtre de sortie

Du fait que la fréquence des impulsions est élevée, nous n'aurons besoin ici que d'un filtre du 2<sup>e</sup> ordre avec un

pôle relativement élevé (fréquence de résonance de 101 kHz). Pour amortir le facteur Q du filtre, ce qui est surtout nécessaire en l'absence de charge, nous avons placé un réseau Zobel à la sortie, sinon les courants à la résonance et les ondulations qui s'y produisent risqueraient de compromettre la fiabilité de l'amplificateur. Comme le pôle du filtre se situe plus haut que dans la classe D conventionnelle, l'influence du système de haut-parleurs branché est sensiblement moindre. En raison des grands courants, une simple bobine antiparasite peut être insuffisante. Pour préserver une faible distorsion et un haut rendement, il vaut mieux faire appel à un noyau en matériau spécial. Nous vous en dirons davantage dans la suite de cet article à paraître en septembre.

## Le réglage du gain

Le gain du modulateur se règle par la rétroaction R15 (R36) et le diviseur de tension R18/R20 (R39/R41) et est choisi en fonction de la tension d'alimentation maximale utilisée. La raison est qu'ainsi l'amplificateur est indépendant du comportement de l'alimentation (comme les fluctuations dues au signal ou les variations du secteur). La résistance R16 (R37) et le diviseur de tension R17/R19 (R38/R40) forment une rétroaction supplémentaire destinée à se prémunir contre les rebondissements par la ligne de masse. Les deux cellules doivent être identiques ! Le calcul des résistances est très simple. Pour R17 et R18, on prend une valeur typique de 1 k $\Omega$  et les autres résistances ne dépendent plus que de la tension d'alimentation VPP (en supposant que l'alimentation est réellement symétrique) et de la valeur choisie pour R17 :

$$R19 = R17 \cdot VPP / (VPP - 4)$$

Cela fournit une grandeur de résis-

tance dans la série E96 de 1,07 k $\Omega$ . Cette valeur ne varie pratiquement pas avec la tension d'alimentation, il faut dépasser 51 V de tension maximale pour arriver à 1,10 k $\Omega$ .

Et finalement, la valeur de R15 déterminera le gain du modulateur :

$$R15 = R17 \cdot VPP / 4$$

Nous avons supposé une tension d'alimentation maximale de 62 V, parce que les condensateurs électrolytiques spéciaux pour le découplage sur la platine d'amplificateur sont des modèles de 63 V. On trouve alors pour R15 la valeur de 15,4 k $\Omega$ . On peut aussi calculer le gain du modulateur comme pour tout amplificateur non inverseur suivant la relation :

$$A_{\text{modulateur}} = R15 / R_p + 1$$

dans laquelle  $R_p$  est la résultante de la mise en parallèle de R18 et R20.

C11 et C24 sont les condensateurs qui filtrent et retardent le signal de rétroaction du modulateur. Leurs valeurs ne sont pas égales, question d'éviter que du bruit à très haute fréquence ne s'infilte dans la rétroaction et ne pousse la fréquence de commutation des deux modulateurs à se superposer. On échappe de la sorte à des interférences entre modulateurs. Les valeurs sont choisies de manière à ce que la différence entre eux se monte à 40 kHz.

## La protection

Comme mesure de sécurité, le CI pilote surveille les tensions d'alimentation et les courants dans les transistors. L'entrée VPPSENSE s'occupe de la détection de sous- et surtension de l'alimentation positive. Le même contrôle sur la branche négative de l'alimentation, c'est l'affaire de VNSENSE. Si l'une des alimentations s'écarte des limites permises, l'étage de sortie se voit

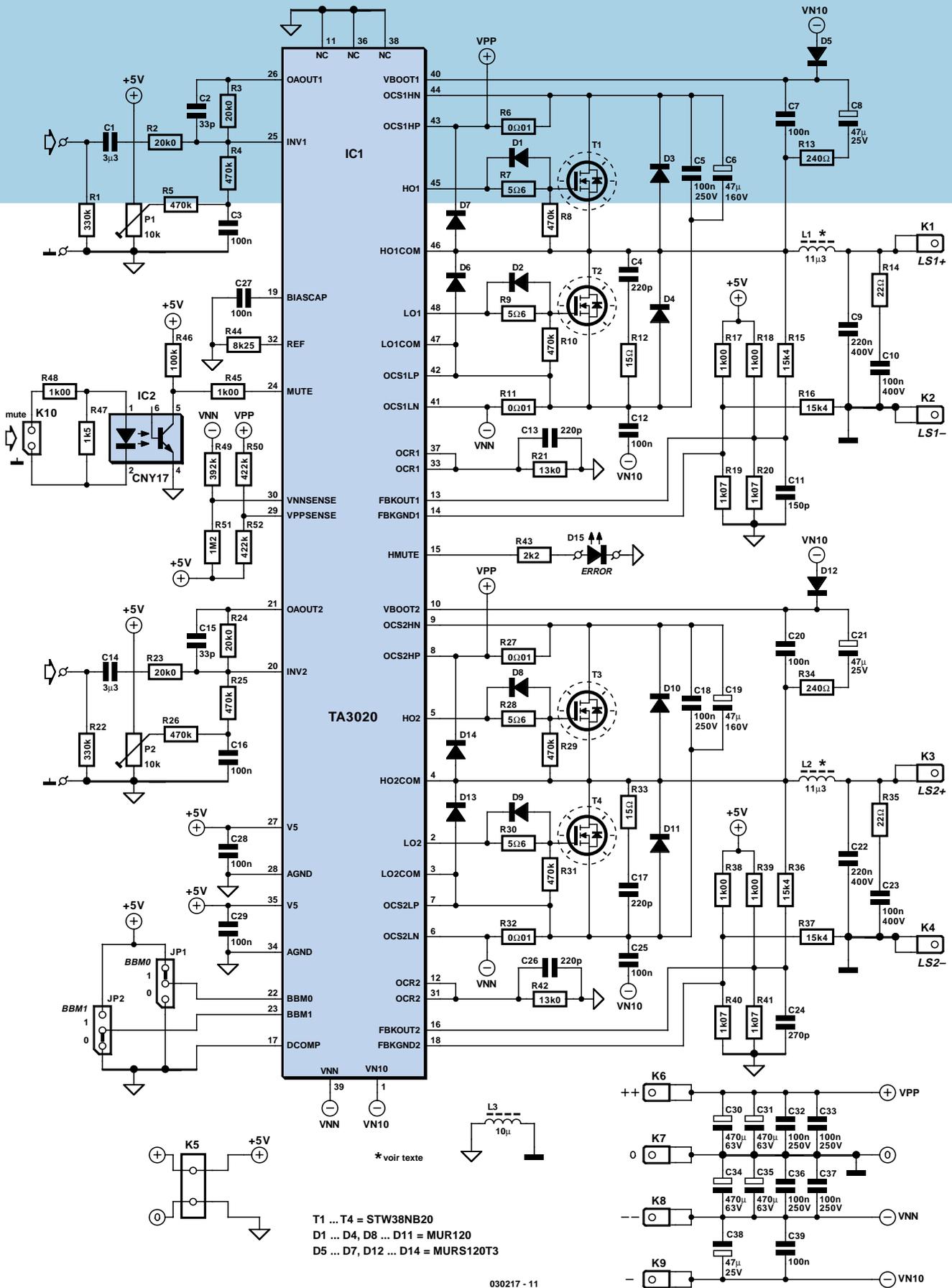


Figure 2. Le montage autour de la puce amplificatrice.

débranché, il passe en mode silencieux jusqu'à ce que la tension revienne à la normale. Pour le calcul de ces composants, voyez le feuillet de caractéristiques. Théoriquement, il se peut que l'amplificateur reste bloqué dans un état de détection de surtension, mais c'est fort peu probable, parce qu'il faut facilement 68 V sur chacune des entrées de détection avant qu'elles ne réagissent. Il s'agit donc ici plus particulièrement d'une protection de la puce elle-même, n'oublions pas que les différents condensateurs électrolytiques de l'alimentation ne sont que des modèles de 63 V...

Le calcul de la protection en courant se révèle bien plus intéressant que celui de la détection de tension, du fait qu'il définit la charge minimum de l'amplificateur en situation d'excursion totale. Puisqu'il est question d'une sortie commutée, ce sont les MOSFET utilisés qui déterminent les possibilités de charge de l'amplificateur. Comme transistors de sortie, nous avons sélectionné un des types les plus costauds de chez Microelectronics, à savoir le STW38NB20. Ce transistor sous boîtier TO-247 peut charrier jusqu'à 38 A, alors que la tension maximale drain-source se monte à 200 V. La résistance de canal pour une  $U_{GS}$  de 10 V affiche tout au plus 0,065  $\Omega$  ( $I_D = 19$  A). L'inconvénient de ce genre de MOSFET, c'est que sa capacité d'entrée  $C_{iss}$  est relativement élevée, elle atteint ici 3 800 pF. Cela explique que les circuits de commande dans la puce doivent pouvoir délivrer du courant en quantité pour arriver à commuter les MOSFET à une vitesse suffisante. Nous avons engagé ces transistors surtout pour éviter les surprises face à des systèmes de haut-parleurs à l'impédance assez imprévisible. Bien sûr, avec d'autres modèles de transistors qui présentent une plus basse capacité de grille vous pouvez diminuer le délai avant la mise en conduction, ce qui procure

aussi une réduction de la distorsion. Mais alors il faut aussi se montrer plus prudent avec les impédances basses. La détection de courant se compose des deux résistances à faible inductance R6 et R11 (R27/R32) utilisées comme shunt en série avec les transistors, R6 pour la moitié positive, dans le drain de T1 et R11 en série avec la source de T2 pour la moitié négative. La limite pour laquelle la sécurité va réagir se règle en combinaison avec R21. Le CI mesure directement aux bornes du shunt et produit un courant dans R21. La tension sur R21 est comparée à la tension de seuil  $V_{TOC}$  de détection de surcharge, ce qui détermine le courant de sortie maximum. C13 (C26) filtre la tension redressée. On peut déterminer l'influence de ces composants par la relation :

$$I_{max} = 3580 \cdot (V_{TOC} - I_{bias} \cdot R21) / (R21 \cdot R6)$$

$$R21 = (3580 \cdot V_{TOC}) / (I_{max} \cdot R6 + 3580 \cdot I_{bias})$$

$V_{TOC}$  représente ici la tension de seuil de détection d'excès de courant (nominal 0,97 V) et  $I_{bias}$  vaut 20  $\mu$ A.

On peut facilement retrouver à l'aide de la première équation la valeur d'un composant. Quant à la seconde, elle vous permet de déterminer la valeur de R21 (ou R42). Nous avons choisi un courant de sortie maximum de près de 20 A, de manière à ce qu'une impédance légèrement inférieure à 3  $\Omega$  n'enclenche pas encore le mode silencieux.

On ne peut sortir du mode silencieux qu'en basculant l'entrée « mute » ou en débranchant momentanément l'amplificateur. Quand le mode silencieux est actif, la sortie HMUTE est haute et attaque une LED à placer en face avant, par exemple. Mieux vaut y consacrer une LED rouge à haut rendement, parce que diminuer R43 surchargerait la sortie.

## L'alimentation

L'énergie nécessaire à la platine de l'amplificateur provient d'une autre carte. On y trouve entre autres le +5 V analogique, VN10 et les fusibles pour les alimentations principales. Vous y verrez aussi la génération d'un signal « un-mute » destiné à retarder l'enclenchement pour éviter les bruits disgracieux à cet instant. Il faut empêcher la formation d'une boucle de masse qui occasionnerait la circulation d'un courant d'ondulation sur la masse d'entrée. C'est donc un photocoupleur, IC2, qui appliquera le signal « mute » à la puce. Il a été prévu sur la platine d'amplification, mais il lui faut un signal de validation pour pouvoir y actionner le mode silencieux.

Les tensions principales, VPP et VNN, du TA3020 sont découplées dans les meilleures conditions possibles par une batterie de condensateurs électrolytiques spéciaux : C30, C31, C34 et C35, avec des MKT pour la HF : C32, C33, C36 et C37 en doublure. Le découplage de la tension de 5 V pour les amplificateurs d'entrée reste simple. La masse analogique et celle du modulateur (et de tout le reste) sont séparées et ne sont couplées qu'en un seul point, situé sous la platine, par l'intermédiaire d'une inductance en technologie CMS, pour tâcher d'affaiblir au mieux les éventuels parasites du circuit de sortie.

## Le tracé

Nous vous le disions en commençant, sans une géométrie astucieuse, l'amplificateur a peu de chance de fonctionner correctement. La platine est une composante essentielle de l'ensemble amplificateur. Tripath insiste pour que nous utilisions le tracé de référence parce qu'en haute fréquence, les courants peuvent occasionner des effets inattendus. Il y a naturellement des

composants que nous avons choisis et dont les dimensions diffèrent, par exemple. Nous avons effectivement porté toute notre attention sur la disponibilité des composants, de préférence par quantité limitée. Nous avons dû allonger quelque peu certaines liaisons et ajouter d'autres composants, mais le résultat est atteint. Si vous comparez la photo publiée dans le feuillet de caractéristiques à notre prototype, la similitude vous apparaîtra immédiatement. Certains composants que Tripath a montés sous la platine, nous les avons

replacés sur la face supérieure. Raison pour laquelle, dans notre version, la platine se positionne en parallèle avec le radiateur et les transistors en dessous. Il en ressort un module compact et bien solide, mais nous en reparlerons au moment de la construction. Ce que nous tenons à vous dire maintenant, c'est que, si le module a l'air tout simple à première vue, en y regardant de plus près, vous aurez l'impression que certains composants manquent. C'est qu'une grande partie d'entre eux sont en exécution CMS et donc soudés sous

la platine, le meilleur remède contre les perturbations et l'embonpoint qui guettait cet amplificateur de 2 x 300 W. La plupart des CMS sont au profil 0603, ce qui risque de donner le vertige à certains d'entre vous. Aussi, pour vous aider, allons-nous essayer de livrer ultérieurement la platine déjà équipée de ses CMS.

Et puis la prochaine fois (en septembre) nous reviendrons en détail sur la construction de cet amplificateur hors du commun.

(030217-1)



Paul Goossens

# Forgez votre puce

## Partie 2 - La pratique des CPLD

Après la description du matériel dans l'article du mois dernier, nous allons passer aux choses sérieuses avec la carte d'expérimentation. Nous supposons le logiciel d'Altera installé et le didacticiel lu.



La conception de circuits numérique se résume souvent à une dissection de la fonction souhaitée en petites solutions à des problèmes et ce jusqu'à ce que le concept résultant soit doté des diverses fonctions logiques de base. Ceci sert alors de base à un circuit électronique dont on développera éventuellement un circuit imprimé.

## Langages descriptifs

L'utilisation d'un langage descriptif simplifie la conception de circuits numériques. Le but d'un tel langage est de pouvoir décrire (d'où son nom) une fonctionnalité donnée. Un logiciel intelligent est en mesure de concevoir un circuit électronique qui satisfasse la description de spécifications qu'en a fait le concepteur.

Il existe actuellement différents langages descriptifs. Deux d'entre eux sont sans lien fabricant et fort bien supportés par nombre de fabricants. Il s'agit de Verilog et de VHDL.

Nous allons, dans le cadre de cet article, utiliser Verilog comme langage descriptif

## Verilog

Nous avons opté pour Verilog vu qu'il nous semble que ce langage est un peu plus clair que VHDL. Sous de nombreux aspects, ces 2 langages présentent de fortes similitudes. La grande différence se situe au niveau de la description. De ce fait, nombre des remarques, des pièges, etc. évoqués dans cet article sont également utilisables directement avec VHDL.

Un concept en Verilog peut présenter une structure hiérarchique. Ceci signifie qu'il peut être subdivisé en concepts plus restreints. Ces sous-concepts peuvent à leur tour être subdivisés en nouveaux modules, car c'est très précisément le terme qu'utilise Verilog pour les désigner.

La subdivision d'un projet en différents modules présente un autre avantage, celui de permettre une réutilisation de ces modules pour un autre projet. Un compteur, par exemple, est un module souvent utilisé; il est partant recommandé d'en faire un module spécifique.

## Exemple 1

Rien de tel que de mettre soi-même la main à la pâte. Nous débutons partant par un exemple. Il vous faudra commencer par télécharger les exemples depuis le site Internet ([www.elektor.fr](http://www.elektor.fr)). Vous les trouverez dans le mois de juin sous la dénomination **030385-11**. Il vous suffira de décompresser le fichier .zip.

L'exemple 1 se trouve dans le dossier *Ex1*. Il suffit d'un double clic sur le fichier *ex1.quartus*. Le logiciel de développement est démarré automatiquement. On voit alors sur le schéma que les diverses broches d'E/S du composant sont reliées à un bloc dans lequel sont mentionnés un certain nombre de signaux. Les dits signaux sont tous des entrées et des sorties de ce bloc. Comme vous avez pu le lire dans le didacticiel (*tutorial*, vous l'avez bien lu, n'est-ce pas ?) un double clic sur le bloc se traduit par l'ouverture du code-source correspondant. Dans le cas présent il s'agit de Verilog. Sous Verilog, le texte en vert représente des commentaires sans influence sur le résultat final. Quartus utilise ces lignes pour stocker de l'information. Il est recommandé de ne pas supprimer ces lignes de commentaire.

## Structure

La structure du code-source sous Verilog est toujours la même. Il commence par une déclaration du module. Cette partie débute par le mot « module » suivi par le nom. Il comprend un ensemble d'entrée et de sorties prises entre parenthèses, les éléments étant sépa-

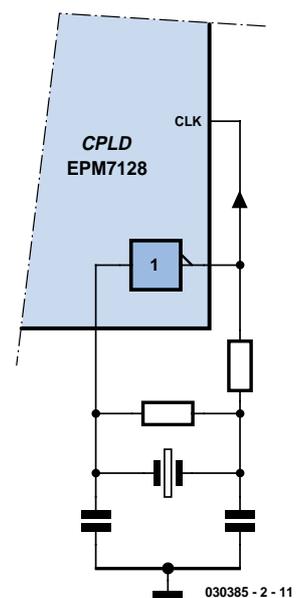


Figure 1. Schéma d'un oscillateur à quartz classique.

rés par des virgules. Le total se termine par un point-virgule. Dans notre exemple il s'agit des lignes 30 à 35.

Il nous faut ensuite définir les directions des signaux (ports), mentionnés dans la déclaration du module. Ici, les lignes 39 à 45. Il existe, pour chaque signal, 3 options : entrée (*input*), sortie (*output*) et bidirectionnel (*inout*). Nous n'avons ici que des entrées et des sorties. Chaque ligne se termine par un « ; ». On voit, sur la première ligne, qu'il est possible de définir plusieurs signaux sur une même ligne à condition qu'ils soient séparés par une virgule.

Les sorties requièrent une spécification additionnelle. En cas de description, dans un *procedural statement* (rassurez-vous, nous verrons de quoi il retourne) d'une sortie, celle-ci doit être du type reg (*register*, sortie d'une bascule). Dans la ligne 47 les signaux D2,

**Tableau 1. Fonctions booléennes**

&	=	AND
~&	=	NAND
	=	OR
~	=	NOR
~	=	NOT
^	=	XOR
~^	=	XNOR

**Tableau 2. Opérateurs mathématiques**

+	=	Ajouter
-	=	Soustraire
*	=	Multiplier
/	=	Diviser
%	=	Modulo

**Tableau 3. Opérateurs relationnels**

>	=	supérieur à
>=	=	égal ou supérieur à
<	=	inférieur à
<=	=	inférieur ou égal à
=	=	égal à
!	=	différent de (non égal à)

D3 et D4 sont définis comme registres. Ayant rempli nos « devoirs administratifs », nous pouvons nous attaquer au projet proprement dit. Notre exemple est, comme se doit d'être tout premier exemple, extrêmement simple. Nous allons démontrer comment décrire des signaux à l'aide d'algèbre booléenne. Sous Verilog, cela peut se faire de 2 façons : à l'aide d'une déclaration (*statement*) *assign* ou par le biais d'une attribution procédurale (*procedural assignment*). Le **listage 1** donne un exemple des 2 méthodes.

La ligne 50 illustre la première méthode. On y décrit que le signal D1 est le résultat d'une fonction ET (AND) des signaux S1 à S4. Cela signifie que D1 ne sera actif qu'à condition que S1 à S4 sont eux aussi actifs (« 1 »). Dans tous les autres cas, D1 sera inactif (« 0 »). Le symbole « & » représente donc une fonction ET. Le **tableau 1** récapitule toutes les fonctions booléennes que connaît Verilog.

### Procedural statement

Les sorties restantes, D2 à D4, sont décrites dans un *procedural statement*. Ces déclarations (*statements*) sont toujours précédées par le mot-clé *always*. Nous l'approfondirons dans un prochain exemple.

Comme dans le langage de programmation Pascal, il est possible d'associer plusieurs déclarations pour en faire une entité encadrée par les mots-clé *begin* et *end*. Toutes les déclarations prises à l'intérieur de ces mots-clé sont considérées comme constituant une seule et même déclaration. Si, maintenant, nous cliquons sur la ligne 54, nous voyons que le signal D4 est décrit comme étant un signal devenant actif lorsque soit S1, soit S2, soit S3, soit encore S4 est actif. C'est à dessein que nous avons choisi le mot « devenant » et non pas « est ». Le double caractère <= signifie « devient ». La règle est que l'on utilise, dans un

*procedural statement*, ce double signe plutôt que le signe « = ».

Un coup d'oeil en cachette au tableau 1 aura vite fait de trahir les fonctions des signaux D3 et D4.

Nous verrons dans le prochain exemple, l'intérêt que présente le placement de signaux dans une déclaration procédurale plutôt que de travailler avec une attribution (*assign*).

Pour clôturer, le mot-clé *endmodule* indique que nous en sommes arrivés à la fin de la description du module.

## Compilation

Il est temps maintenant de remonter les manches. Il va nous falloir commencer par compiler le projet. Le compilateur sait déjà exactement quel signal est relié à chaque broche de la CPLD. Nous l'avons déjà fait à votre place. La compilation en devient ainsi un jeu d'enfant; tout ce que vous avez à faire est de cliquer sur l'option *Start Compilation* du menu *Processing* pour que le programme se mette au travail.

Après apparition de quelques messages à l'écran, on verra quelques mouvements. Au bout d'un certain temps, le programme annoncera que la compilation a réussi. Cela signifie qu'il a été créé un fichier de programmation qui pourra servir à programmer notre puce.

## Programmation

Comme nous le disions dans le premier article, la programmation requiert de disposer du programmeur JTAG du numéro double d'Elektor de 2002. Si vous disposez d'un ByteBlaster d'Altera original, parfait. Nous supposons que vous avez connecté le programmeur au port imprimante du PC et que l'embase JTAG est reliée à l'embase K2 de notre carte d'expérimentation. Mettez maintenant la carte d'expérimentation sous tension.

Dans Quartus nous sélectionnons le menu

Tools puis l'option Programmer. Il apparaît une nouvelle fenêtre. Assurez-vous que le programmeur est paramétré pour JTAG et que c'est l'interface (ByteBlaster) qui est sélectionnée.

Nous découvrons, dans une fenêtre, une ligne avec, sous la colonne *Device*, un EPM7128SLC. Sur la même ligne, sous la colonne *File*, devrait se trouver le fichier de programmation *ex1.pof*. Tout est paré dans ce cas-là pour la programmation de la CPLD. Il nous reste à indiquer au programme que nous voulons programmer le composant en cochant la case sous la colonne *Program/Configure*.

Il nous reste, pour terminer, à cliquer sur le bouton *Start Programming* que l'on peut identifier tout en haut à sa forme rappelant celle d'un bouton PLAY.

## Tester

Dès que la programmation est effectuée, la CPLD est active, ce qui signifie que le projet programmé est immédiatement fonctionnel. Assurez-vous que les cavaliers JP1 et JP2 sont bien en place.

La vérification du projet est simple. La LED D1 doit uniquement être « allumée » lorsque tous les inverseurs se trouvent en position « 1 ». Dans tous les autres cas, la LED doit être éteinte. La LED D4 doit réagir à l'inverse. Lorsque D1 est allumée, D4 doit être éteinte et vice-versa.

Nous avons décrit D2 comme une fonction OU, ce qui signifie que dès que l'un des inverseurs (ou plusieurs) se trouve(nt) en position « 1 », cette LED doit s'allumer.

La LED D3 doit s'allumer dès que S1 ET S2 se trouvent à l'état « 1 » OU que S3 et S4 se trouvent dans le même état. Il est facile de vérifier ces fonctions par le biais des inverseurs.

Essayez donc de modifier le projet sous Quartus de manière à ce que la LED D1 s'allume dès que S1 est à l'é-

### Listage 1. Algèbre booléenne

```
50 assign D1 = S1 & S2 & S3 & S4;
51
52 always
53 begin
54     D2 <= S1 | S2 | S3 | S4;
55     D3 <= ( S1 & S2 ) | ( S3 & S4 );
56     D4 <= ~(S1 & S2 & S3 & S4);
57 end
```

tat « 1 » et que S2 se trouve à « 0 ». Les positions des autres inverseurs n'ayant aucune importance. Bonne chance !

### Exemple 2

L'exemple 1 était, nous le disions, un exemple ultra-simple. La puissance de Verilog est que le concepteur dispose d'une approche plus descriptive pour ses conceptions. L'algèbre booléenne peut être très pratique en Verilog, mais il n'est bien évidemment pas question d'avoir à saisir des projets complexes uniquement sous la forme d'algèbre booléenne.

Nous allons voir à quoi cela ressemble au travers de l'exemple 2, où nous allons concevoir 2 bascules (*flipflop*) et un verrou (*latch*).

Les fichiers de ce second exemple se trouvent dans le répertoire *ex2*. Nous allons y ouvrir *ex2.quartus*, le logiciel ouvrant automatiquement tous les autres fichiers nécessaires.

Nous voyons, sur le schéma (*ex2.bdf*), que les interrupteurs S1 à S4 sont reliés à un bloc fonctionnel appelé *flipflop*. S1 est relié à l'entrée d'horloge (CLK), etc. Un double clic sur ce bloc se traduit par l'ouverture du code-source Verilog correspondant.

### Always @

Rien de neuf jusqu'à la ligne 46. Sur cette ligne, le mot-clé *always* est doté d'un suffixe, un @ suivi d'une équation. Le listage 2 donne la partie de code correspondante. Le caractère @ indique que les déclarations procédurales correspondant à cette déclaration *always* ne pourront être évaluées que si la condition définie par l'équation en aval est respectée. Dans le cas présent cela signifie que les déclarations qui suivent ne sont d'actualité que lorsque le signal d'horloge (CLK) OU le signal de remise à zéro (RAZ = reset) OU le signal de positionnement (SET) pré-

sente un flanc montant (*posedge*). Pour éviter toute ambiguïté, un flanc montant est le passage d'un niveau logique bas « 0 » à un niveau logique haut « 1 ».

*Verilog connaît aussi le terme « negedge », c'est-à-dire flanc descendant.*

Si l'une de ces conditions est remplie, on a évaluation du petit morceau de code. On commence par tester le signal de RAZ pour voir s'il est à « 1 ». Dans ce cas-là OUT est inactivée « 0 » et cette partie du code est exécutée. Sinon, on teste à « 1 » le signal SET. Si oui, la sortie OUT passe à « 1 » et c'est la fin de l'exécution du code.

Outre les valeurs « 0 » et « 1 », chaque signal peut aussi prendre les valeurs « x » (inconnu) ou « z » (haute impédance).

Si ni RESET ni SET ne sont à « 1 », CLK doit présenter un flanc montant sachant que sinon il n'y aura pas évaluation du code. Le but de la manoeuvre est que lors d'un flanc montant du signal d'horloge la sortie de la bascule prenne la valeur présente à l'entrée.

Que se passe-t-il au niveau de la sortie si ni CLK ni RESET ni SET ne présente de flanc montant ? La réponse est simple : rien du tout. Dans la ligne 45 le signal OUT est défini comme registre, ce qui signifie que la dernière valeur attribuée à ce signal doit être mémorisée. Tant qu'aucun code n'est actif, la valeur de ce registre ne changera partant pas. L'ajout du caractère « @ » à la déclaration « always », nous pouvons indiquer dans quelles conditions on peut avoir évaluation d'un morceau de code. Le reste du temps, la sortie pilotée par ce bloc, ne change pas d'état.

Un examen du code montre à l'évidence que l'entrée RESET possède une priorité supérieure à l'entrée SET. Cela ne signifie cependant pas qu'en cours de fonctionnement la CPLD va évaluer successivement ces 2 signaux. La CPLD

réagira à la même vitesse à un signal RESET qu'à un signal SET. L'ordre ne présente d'importance que pour le compilateur. Ce dernier évalue un petit morceau de code et détermine ce qui doit se passer au niveau de la sortie pour chacune des combinaisons possibles des signaux d'entrée. À partir de cette évaluation, le compilateur « conçoit » un bloc de logique numérique qui réagit comme le requiert le code.

### Variations sur un même thème

Le schéma (*ex2.bdf*) comporte une seconde bascule, baptisée, comme il se doit, « flipflop2 ». Le fichier Verilog correspondant ressemble beaucoup au fichier Verilog de la première bascule. La seule différence est qu'ici on teste d'abord l'état du signal SET puis ensuite celui du signal RESET. Cela signifie que pour cette bascule, l'entrée SET a une priorité supérieure à celle de l'entrée RESET. La sortie passera de ce fait à « 1 » lorsque tant l'entrée RESET que l'entrée SET se trouvent à « 1 ». Avec la première bascule, cette situation se serait traduite par une mise à « 0 » de la sortie.

### Verrou

Le dernier bloc de ce schéma est un verrou. Un verrou est une « brique » que l'on rencontre souvent dans des projets numériques. La fonction d'un verrou est on ne peut plus simple : tant que l'entrée d'horloge se trouve à « 1 », la sortie doit présenter le même niveau que l'entrée. Dès que l'état de l'entrée change, la sortie doit suivre. Si, au contraire, le signal d'horloge est inactif (« 0 »), le dernier état connu de la sortie doit être conservé quels que soient les changements au niveau de l'entrée. Le fichier Verilog *latchexample.v* montre comment décrire ce processus en

## Listage 2. Une bascule cadencée

```
46 always @ (posedge CLK or posedge RESET or posedge SET)
47 begin
48     if (RESET)
49         OUT <= 1'b0;
50     else if (SET)
51         OUT <= 1'b1;
52     else
53         OUT <= D;
54 end
```

Verilog. Le signal de sortie **peut** changer lorsque l'état du signal d'horloge change OU que l'état de l'entrée de données change. Ceci peut se faire tant sur un flanc montant que sur un flanc descendant. Ceci explique que nous trouvions, après le caractère @, « (posedge CLK or D) ». On notera que le signal D n'est pas accompagné de « posedge ». Ceci signifie que le code doit être évalué lors de tout changement d'état du signal D.

Nous pouvons voir, dans le code correspondant à cette déclaration *always*, que le compilateur commence par s'intéresser à l'état du signal d'horloge. Si ce dernier est actif (« 1 »), la sortie est mise à l'état de l'entrée. Sinon, il ne se passe rien et la sortie garde son état actuel. Nous pouvons compiler et programmer cet exemple comme nous l'avons fait avec l'exemple précédent. Vérifiez ensuite, à l'aide de la carte d'expérimentation que le projet fait bien ce que l'on en attend. Vous pouvez ensuite, à titre d'exercice, essayer de voir si vous pouvez doter le latch d'une entrée SET et d'une entrée RESET. C'est en forgeant que l'on devient forgeron dit le proverbe !

## Mathématiques

Nous avons vu, dans le précédent exemple, comment décrire une fonction sans avoir à nous occuper de portes logiques, d'algèbre booléenne, etc. Après avoir pris Verilog en main, tout concepteur saura apprécier la situation. C'est le compilateur, et non plus le concepteur, qui se charge des processus les plus courants.

Dans l'exemple n°3 nous allons voir que Verilog n'a pas de problème avec les maths. Nous allons, cette fois, utiliser des compteurs. Ces compteurs requièrent un signal d'horloge; c'est par lui que nous allons commencer.

La **figure 1** montre un oscillateur à quart classique. Hormis l'inverseur, tous les autres composants se trouvent sur la carte d'expérimentation. Par la prise d'un inverseur entre les broches 71 et 81 de la CPLD nous nous donnons un oscillateur à quartz.

Si nous ouvrons l'exemple 3, nous découvrons sur le schéma cet inverseur pris entre 2 broches du circuit intégré. Après programmation de la CPLD nous disposons partant d'un oscillateur dont la sortie (broche 81 de la platine) est reliée à la broche 83 de la CPLD. Cette entrée est prévue spécialement pour faire office d'entrée d'horloge.

## Matrices

Nous supposons que vous connaissez le système binaire. Vous savez alors sans doute que les nombres sont souvent représentés par un train de signaux. Il est possible, en Verilog, de grouper un certain nombre de signaux dans ce que l'on appelle des matrices (*array*).

Dans le fichier Verilog du bloc *Count* nous découvrons, en ligne 40 : « output [7:0] D; ». Cette expression spécifie une matrice de 8 signaux, à savoir D[7] à D[0]. Tous les signaux réunis sont appelés « D ».

Il nous faut parler de la ligne 47 pour éviter toute ambiguïté. On y déclare un nouveau signal. Ce signal n'apparaît pas dans la déclaration du module ce qui signifie que ce signal n'est pas visible de l'extérieur (du module). Ce signal a une utilisation purement interne.

La ligne intéressante suivante est la ligne 51 : « temp=temp+1; ». Il nous est donc possible de compter en utilisant le caractère « + ». Le **tableau 2** mentionne d'autres opérateurs (mathématiques) utilisables avec les matrices sous Verilog.

La ligne 52 nous propose une compa-

raison (« == » signifie vérifier si le terme gauche est égal au terme droit). Le **tableau 3** donne tous les opérateurs relationnels (tels que « == ») que connaît Verilog. Le nombre 24'd4000000 peut paraître étrange au premier abord. C'est ainsi que l'on écrit des nombres sous Verilog. Le premier nombre indique le nombre de signaux concernés (24 dans le cas présent). Le « d » indique que le système utilisé pour la constante est le système décimal. 4000000 représente la constante proprement dite.

Il nous faut ici utiliser 24 signaux vu que le registre *temp* est constitué de 24 signaux. 24 zéros en Verilog n'est pas la même chose que 23 zéros ! Faites donc bien attention qu'il y ait autant de signaux sur la gauche du terme « == » que sur sa droite !

## Compteur

Si nous analysons le code Verilog, nous voyons qu'à chaque impulsion d'horloge la valeur du registre *temp* est incrémentée. Dès que ce registre a atteint la valeur de référence .000.000, le registre D est incrémenté (augmenté de « 1 », temps repasse à « 0 » et la sortie *SLOW* passe à « 1 ». Si le registre *temp* n'a pas encore atteint la valeur 4.000.000, *SLOW* se voit attribuer la valeur « 0 ».

Dans le cas d'un signal d'horloge de 4,000 MHz, cela signifie que la valeur du registre D est incrémentée une fois par seconde moment auquel la sortie *SLOW* passe brièvement à « 1 ». *SLOW* est partant un signal d'horloge de 1 Hz.

## Compteur BCD

Le signal d'horloge de 1 Hz attaque l'entrée d'horloge d'un compteur BCD. On le retrouve sur le schéma sous la dénomination de « BCD\_counter ».

### Listage 3.

#### Combiner le statement case et des signaux en une matrice (array)

```
59 always @ (COUNT)
60 begin
61 case (COUNT)
62 4'd0 : SEG=7'b1111110;
63 4'd1 : SEG=7'b0110000;
.....
71 4'd9 : SEG=7'b1111011;
72 default : SEG=7'b0000001;
73 endcase
74 end
75
76 always @(SEG)
77 {SEGA,SEGB,SEGC,SEG D,SEGE,SEGF,SEGG} = SEG;
```

Le code Verilog du compteur BCD comporte pas moins de 3 blocs distincts commençant par la déclaration « always ». Il nous reste à signaler qu'il y a eu déclaration d'une paire de registres pour utilisation interne, à savoir *SEG* et *COUNT*.

La première fonction, à compter de la ligne 52, est un compteur qui fait en sorte que le registre *COUNT* compte de 0 à 9. Lorsqu'il atteint la valeur 10, *COUNT* est rechargé de la valeur 0.

La seconde fonction est exécutée à chaque fois que *COUNT* change. Dans la ligne 60 nous trouvons une nouvelle déclaration baptisée *case*. Le **listage 3** en donne un exemple. Les programmeurs en C connaissent. La déclaration *case* a 1 argument, *COUNT* en l'occurrence. Les lignes suivantes se traduisent alors ainsi :

```
"4'd0 : SEG=7'b1111110;
4'd1 : SEG=7'b0110000;"
```

que l'on peut traduire :

```
if (COUNT==4'd0) SEG=7'b...
else if (COUNT==4'd0) SEG=...
etc.
```

La ligne commençant par *default* est exécutée si la valeur actuelle de *COUNT* n'apparaît pas dans la liste.

Le dernier morceau de code Verilog débute en ligne 75. Comme on peut le voir, il est évalué lorsque le registre *SEG* change de valeur.

La ligne 76 illustre une possibilité intéressante de Verilog. Les caractères ASCII « { » et « } » permettent d'associer plusieurs signaux pour en faire une matrice. Comme *SEG* est une matrice et que les sorties des afficheurs 7 segments sont déclarées comme signaux « distincts », il nous faut coupler ces signaux à une matrice. Autre solution : coupler chacun de

signaux à une partie de la matrice, par exemple :

```
"SEGA = SEG[6];"
```

Cette méthode fonctionne tout aussi bien, mais rallonge le code en le rendant partant moins lisible.

### Tester

Il nous faut, dans le cadre du test de cet exemple, évoquer une imperfection de la carte d'expérimentation.

Lors de la programmation de la CPLD, toutes les sorties sont désactivées. Immédiatement après la programmation, la CPLD est active. Ceci pose des problèmes de démarrage de l'oscillateur. Il vous faudra donc, une fois la programmation effectuée, procéder à un bref reset. Vous pourrez le faire par le biais d'un bouton-poussoir à contact travail à prendre entre les broches 2 et 20 de l'embase K6. Vous pouvez aussi couper brièvement l'alimentation avant de remettre le montage sous tension.

### En guise de conclusion

Verilog offre bien plus de possibilités que nous ne pouvons en évoquer dans le cadre de cet article. Nous nous sommes limités aux fonctions les plus utilisées de Verilog. Celles-ci permettent déjà à un concepteur de concevoir des fonctions numériques très complexes. Internet fourmille de site proposant les projets Verilog les plus divers. L'étude et la simulation de ces projets aura vite fait de vous donner une certaine expérience dans ce langage ô combien intéressant.

Il est un point important que nous n'avons pas pu aborder dans cet article : la simulation d'un projet sous Quartus. Le didacticiel de Quartus pourra beaucoup vous apprendre, sans même par-

ler de la fonction d'Aide.

À noter qu'il est judicieux de subdiviser un projet en petits morceaux que l'on simulera l'un après l'autre pour vérifier qu'ils fonctionnent bien comme prévu. Nous attendons avec impatience vos applications intéressantes à base de notre carte d'expérimentation ! Bonne chance !

(030385-2)

#### Liens intéressants :

[www.altera.com/support/examples/verilog/verilog.html](http://www.altera.com/support/examples/verilog/verilog.html)

[www.asic-world.com/verilog/index.html](http://www.asic-world.com/verilog/index.html)

### Trucs & Astuces

**- Un module peut comporter plusieurs morceaux de code qui sont tous précédés par la déclaration « always ». La limitation de cette approche est qu'il n'est possible de définir une sortie que dans un morceau de code. Un concepteur pourra, pour contourner cette limitation, définir 2 signaux (wire T1,T2; par exemple) et piloter ces 2 signaux par 2 morceaux de code distincts. La sortie proprement dite pourra être définie comme une fonction booléenne de ces 2 signaux.**

**- Veillez à ce que toutes les bascules du projet réagissent soit sur un flanc montant soit sur un flanc descendant. Ceci se traduit par un projet plus efficace (lire : rapide).**

**- Tenez compte du fait que le compilateur doit évaluer le code. Ceci est sensiblement différent d'un processeur ayant à exécuter un morceau de code. Si un projet comporte, comme c'est le plus souvent le cas, plusieurs segments précédés par un mot-clé « always », ceux-ci sont en fait exécutés simultanément !**



On l'avait presque oublié, les premiers jeux électroniques se branchaient sur la télé. Il s'en est fallu de peu qu'on ne les voie jamais sortir. En 1951, Ralph Baer, un technicien en télévision se dit qu'il serait agréable d'utiliser l'écran de la télé pour un jeu électronique. Son patron ne voyait aucun avenir dans cette idée, qui resta sans suite. Des années plus tard, le projet recommençait à le démanger et Baer entama en 1966 la construction d'un prototype. Le jeu vidéo était né.

Mais venons-en au jeu qui nous occupe ici : le Pong d'Atari, dérivé d'une autre trouvaille de Baer. C'est en 1974 que commence la conception des premières versions de jeux domestiques. Contrairement aux autres fabricants, Atari trouva un filon prospère : développer pour Pong une puce spéciale, un ASIC (*application specific integrated circuit*). Grâce à elle, le prix de revient restait très bas, alors que les possibilités offertes, comme l'affichage numérique du score et les effets sonores, étaient inégalées dans la concurrence. Pong fut lancé en 1976 en version domestique. Aujourd'hui, 28 ans plus tard, nous en remettons une couche, mais cette fois-ci, plus sur la télé, mais avec une matrice de LED en guise d'écran.

## Le circuit

Or en fait, nous allons marcher dans les pas d'Atari, en développant un CI spécifique pour cette application. Heureusement, nous ne devons pas concevoir un tout nouveau circuit à faire graver ensuite. Nous partons bien sûr d'un microprocesseur que le logiciel dressera à exécuter fidèlement ce que nous voulons. Nous travaillons ici avec un PIC18F452, une puce à 40 pattes avec 32 Koctets de mémoire de programme et, parmi d'autres, un convertisseur A/N à 10 bits embarqué. Ce processeur ne restera pas solitaire

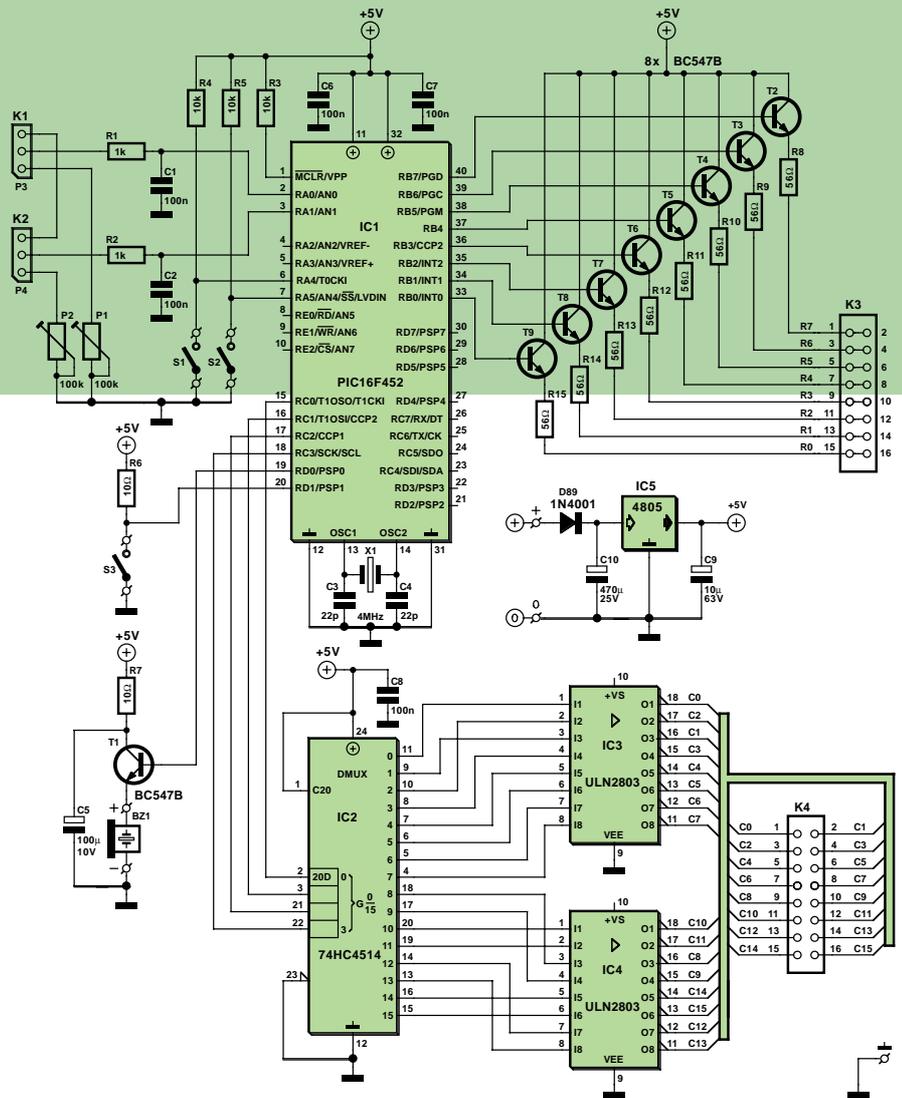


Figure 1a. Le microcontrôleur en juge-arbitre.

dans le montage, comme l'illustre la figure 1. IC2, un décodeur de 4 vers 16, assure, de concert avec IC3 et IC4, la direction de l'afficheur. Celui-ci se compose de deux parties, les chiffres à 7 segments LD1 et LD2 pour indiquer le score et une matrice de LED (D1 à D88) comme terrain de sport. Sur les bords gauche et droit du champ, on déplace les raquettes ou les palettes virtuelles pour faire rebondir la balle et la renvoyer dans l'autre camp. BZ1 produit les effets sonores. Il s'agit d'un vibreur acoustique continu attaqué par T1. C5 et R7 assurent le découplage de la tension d'alimentation. La mise en œuvre de IC5, un régulateur de tension fixe, est tout ce qu'il y a de classique. D89 sert de protection contre l'inversion de polarité. Le plus simple est de faire usage d'un bloc adaptateur sur secteur entre 9 et 12 V. Le PIC consomme à peine quelques milliampères, la puissance requise dépend essentiellement du courant

dans les LED. Mais le multiplexage fait chuter sérieusement la consommation moyenne, qui se situe aux alentours de 35 mA.

## Affichage

Les cathodes des LED constitutives des chiffres à 7 segments ainsi que de celles de la matrice se branchent sur des tampons ULN2803 (IC3 et IC4). Les anodes sont raccordées au processeur par transistors (T2 à T9). En principe, on aurait pu s'en passer, puisque les sorties du PIC peuvent délivrer jusqu'à 25 mA. Avec des LED à haut rendement, cela suffirait, mais à ce courant, les modèles ordinaires risquent de manquer de clarté et le confort du jeu en pâtirait. La valeur de résistance donnée de 56 Ω pour R8 à R15 fait circuler un courant d'environ 27 mA dans les LED. Mais cette valeur peut toujours être adaptée sans difficulté. On la calcule aisément par la célèbre loi

# Liste des composants

## Résistances :

R1, R2 = 1 kΩ  
 R3 à R6 = 10 kΩ  
 R7 = 10 Ω  
 R8 à R15 = 56 Ω  
 P1, P2 = ajustable 100 kΩ

## Condensateurs :

C1, C2 = 100 nF

C3, C4 = 22 pF  
 C5 = 100 μF/10 V radial  
 C6 à C8 = 100 nF  
 C9 = 10 μF/63 V radial  
 C10 = 470 μF/25 V radial

## Semi-conducteurs :

D1 à D8 = LED 5 mm rouge à haut rendement tel que, par exemple, HLMP-

D101 de HP (Farnell-nr. 323-044)  
 D89 = 1N4001  
 LD1, LD2 = LTS4301E LiteOn  
 T1 à T9 = BC547B  
 IC1 = PIC18F4521/P  
 IC2 = 74HC4514 (74HCT4514 ou 4514)  
 IC3, IC4 = ULN2803  
 IC5 = 4805

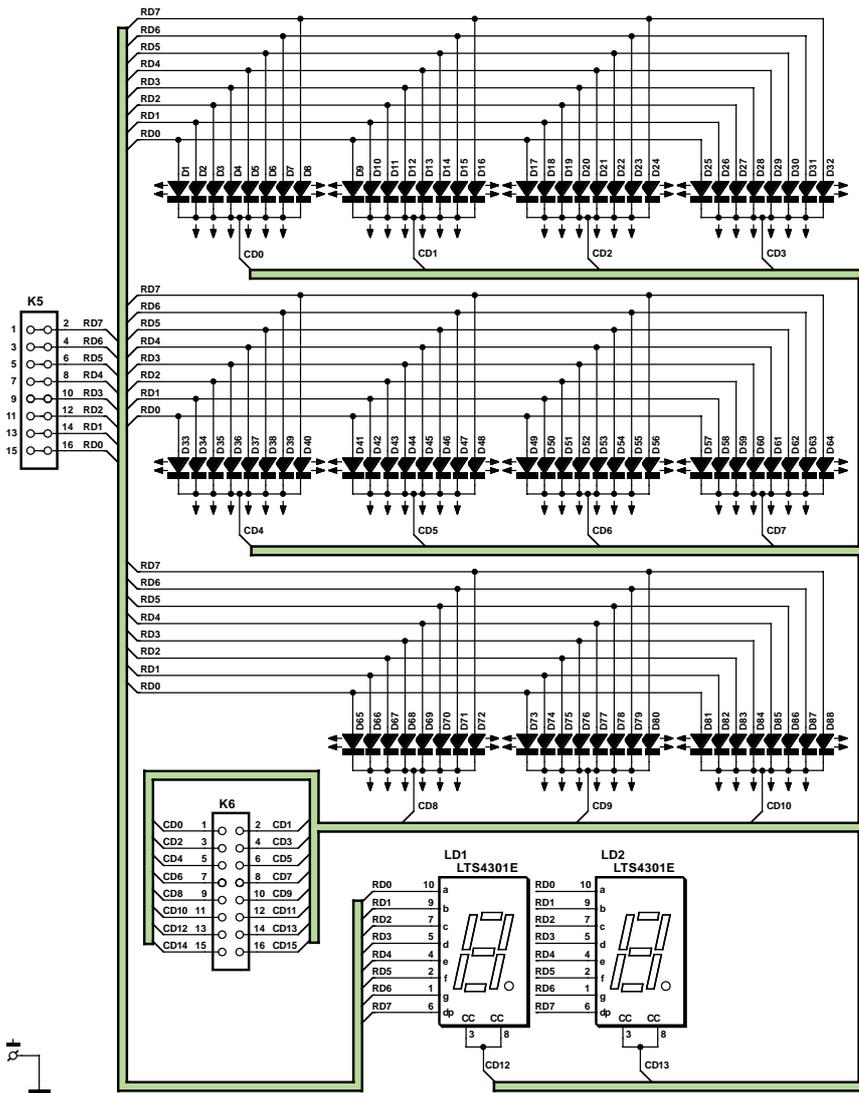


Figure 1b. Une nuée de LED remplace désormais l'écran de la télé.

d'Ohm. Partant d'une tension d'alimentation de 5 V, on retire la chute collecteur/émetteur (0,7 V) et la tension aux bornes de la LED (quelque 1,8 V pour une rouge) ainsi que la chute sur l'étape Darlington du tampon ULN2803 (à peu près 1 V), il reste 1,5 V sur la résistance. Si vous désirez par exemple un

courant de 10 mA, suivant la loi  $V = I \cdot R$  ou  $R = V / I$  on obtient  $R = 1,5 / 0,01$  soit 150 Ω.

## Mode d'emploi

Le jeu se manipule à l'aide de deux potentiomètres et deux interrupteurs.

S1 sert à lancer la balle. S3 sélectionne la vitesse de la balle. S'il est fermé, la balle va plus vite, le degré de difficulté s'en trouve plus élevé. Il y a encore place pour un troisième interrupteur (S2), mais on ne l'emploie pas dans ce jeu-ci. Pour P3 et P4, on peut prendre des potentiomètres rotatifs ordinaires, pour changer l'emplacement des raquettes, mais des modèles à glissière se révèlent peut-être plus pratiques. Il est naturellement encore plus agréable de faire usage d'un vrai manche de jeu.

Un joystick analogique pour PC comporte d'habitude deux potentiomètres, un pour modifier la position dans le sens horizontal, l'autre pour le sens vertical. La plupart du temps, il s'agit de potentiomètres de 470 kΩ dont on n'utilise que la plage de 0 à 120 kΩ. Comme dans ce montage, c'est une valeur nominale de 4,7 kΩ qui a été prévue, si vous y raccordez un manche de jeu, prévoyez deux résistances en parallèle (entre +5 V et broche 2 et entre +5 V et broche 3 du processeur) pour atteindre une valeur de résistance plus basse. La résultante d'une mise en parallèle se calcule comme suit :  $1/R_{\text{totale}} = 1/R1 + 1/R2$ .

Si vous désirez atteindre une valeur de 4,7 kΩ et que sur le manche :  $R1 = 120 \text{ k}\Omega$ , appliquez la formule :  $1/R2 = 1/R1 - 1/R_{\text{totale}}$  avec pour résultat  $R2 = 4,89 \text{ k}\Omega$ .

La valeur normalisée la plus proche dans la série E vaut 4,7 kΩ, pas de souci donc. Sur le connecteur sub-D à 15 broches du port de jeu, les broches 1 et 3 servent au raccordement de la direction horizontale, les broches 1 et 6 pour la direction verticale.

Les cellules R1/C1 et R2/C2 servent à éliminer le bruit produit par les potentiomètres.

### Divers :

K1 = P3 = potentiomètre 4k $\Omega$ 7 linéaire, mono, + embase SIL à 3 contact  
K2 = P4 = potentiomètre 4k $\Omega$ 7 linéaire, mono, + embase SIL à 3 contact  
K3, K4 = embase HE-10 à 2 rangées de 8 contacts  
K5, K6 = connecteur pour câble plat à 16 conducteurs encartable

S1 = bouton-poussoir unipolaire à contact travail  
S2 = n'existe pas  
S3 = interrupteur unipolaire à contact travail  
X1 = quartz 4 MHz  
BZ1 = résonateur piézo CC 5 V

## Construction

La platine (**figure 2**) se compose de deux parties à relier entre elles par du câble plat. Le plus gros du travail consistera à assembler la matrice de LED. La besogne n'est pas compliquée, à condition de faire bien attention à la polarité des LED avant d'avoir soudé, systématiquement à l'envers, la quatre-vingt huitième ! La plupart du temps, la cathode, la barre du symbole, correspond à la plus grande surface à l'intérieur de la LED et à la patte de raccordement la plus courte. N'empêche, mieux vaut en contrôler au préalable la polarité à l'aide d'un testeur de continuité, c'est très simple et tellement rassurant !

L'orientation des chiffres à 7 segments vous paraît peut-être incorrecte, mais si vous les installez sur la platine comme l'indique la **figure 2**, ils fonctionneront comme il faut.

Il faudra encore vous soucier de la polarité des transistors et des condensateurs électrolytiques et les CI doivent aussi se monter dans le bon sens sur la platine. Le processeur, contrôlez-le plutôt deux fois qu'une, c'est lui qui coûte le plus cher !

Il est conseillé d'incorporer le montage dans un boîtier capable d'héberger les interrupteurs, potentiomètres, raccords d'alimentation et diodes électroluminescentes. Un filtre rouge devant la découpe pour la matrice et l'affichage du score sera du plus bel effet.

Vous pouvez éventuellement remplacer l'alimentation secteur par quatre piles bâton. En ce cas, prenez la consommation des LED en considération et peut-être serait-il bon d'augmenter la valeur de R8 à R15 pour épargner les piles.

## Le logiciel

Il subsiste la question du logiciel à charger dans le PIC18F452. Comme

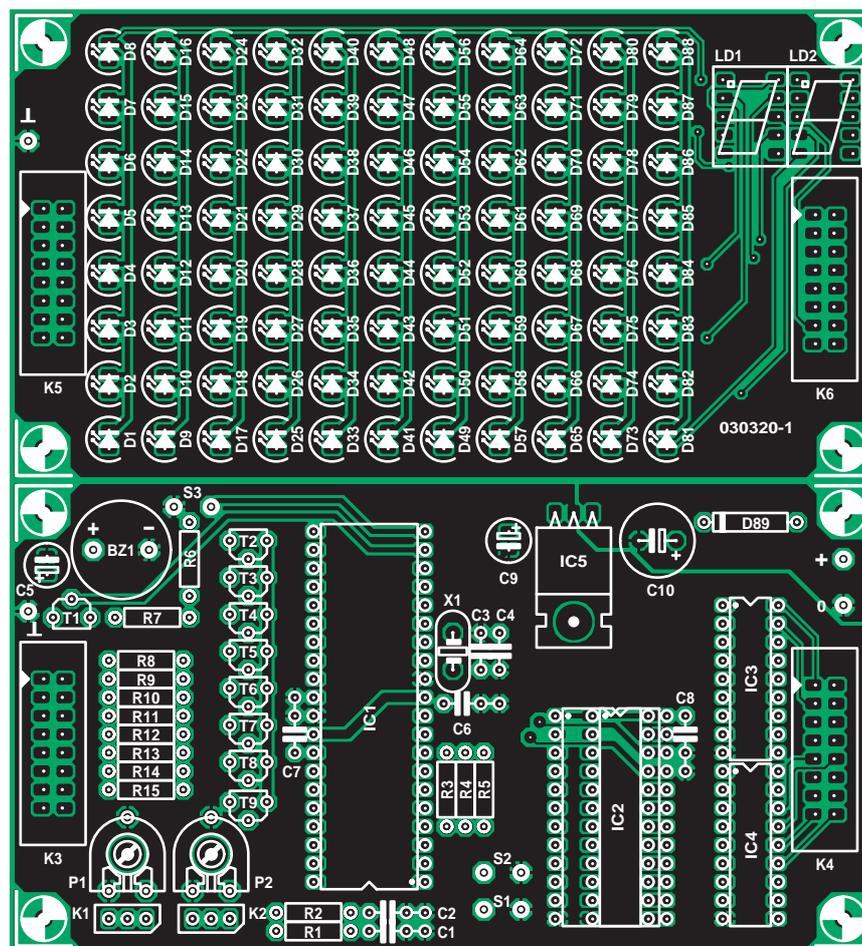


Figure 2. La platine est en deux parties, raccordées l'une à l'autre par deux câbles en nappe.

nous avons affaire à un jeu dont le prix se doit de rester modeste, nous avons choisi un logiciel très simple que chacun pourra très facilement programmer à partir de son PC. Un autre article de ce même numéro vous fournira toute indication utile à ce sujet. Et comme nous en avons l'habitude à

présent, ce logiciel est disponible gratuitement sur le site Internet d'Elektor (**EPS030320-11**).

(030320)

Projet: Andy Morell  
Rédaction: David Daamen



Michael Schulze Dipl.-Ing.

# Attila – Bloque-pub TV

## Fin des enregistrements de pubs

Ce montage sophistiqué élimine, lors d'un enregistrement, les blocs publicitaires, ô combien gênants.

Ce problème, vous le connaissez tous : lors de l'enregistrement d'une émission d'une chaîne commerciale sur magnétoscope ou graveur de DVD, on a le choix soit de suivre l'émission en « *stand-by* » et d'appuyer sur la touche *STOP* (en veillant à ne pas oublier d'appuyer à temps sur le bouton *RECORD* lorsque le programme reprend), soit on élimine la pub à grand renfort de technique, soit encore on l'accepte tout simplement à contre cœur.

Mais on peut également se résigner à passer à la caisse : pour une chaîne de Pay-TV sans pub, soit pour un service de suppression de celle-ci, demandé à la « *fée-télévision* », notre approche.

## Grâce à la législation

Le bloqueur de pub que l'auteur a baptisé « *ViConti* » (*VideoContinue* ou *ViConti* sont des marques déposées) peut uniquement fonctionner grâce au contrat état-chaînes d'émission du 31 août 1991 (en allemand : [www.swr.de/imperia/md/content/rundfunkgebuehren/rfg\\_staatsvertrag.pdf](http://www.swr.de/imperia/md/content/rundfunkgebuehren/rfg_staatsvertrag.pdf)). Les paragraphes établissent en effet une différenciation nette entre les émissions publicitaires et les contenus rédactionnels, ce qui se traduit par la situation de fait que les contenus rédactionnels (dont font partie également les films) sont émis dotés du logo de la chaîne alors que la pub ne le comporte pas.

Une électronique reposant sur cette constatation doit :

- Reconnaître la présence ou non du logo de la chaîne dans l'image de télévision,
- Surveiller l'image de télévision et, au cas où le logo disparaîtrait,

- Interrompre, au niveau de l'appareil enregistreur (magnétoscope, enregistreur de DVD) l'enregistrement par une commande par infrarouge,

- Continuer de surveiller l'image TV et, lors de la réapparition du logo, redémarrer l'enregistreur.

De nos jours, le traitement d'image en temps réel n'a plus rien d'exceptionnel. Les PC et des processeurs de signal spécialisés sont des outils prédestinés pour ce faire. Ceci à un coût (relativement) élevé il est vrai. Les microcontrôleurs « ordinaires » sont visiblement trop lents et le chapitre bloqueur de pub serait ainsi rapidement clos, s'il n'existait pas un minuscule microcontrôleur, le SX28 de Scenix (devenu Ubicom) qui, en raison de son incroyable vitesse, constitue, dans des cas spécifiques du moins, une alternative aux processeurs de signaux.

Si à l'origine le SX28 était spécifié pour 50 MHz des essais ont montré qu'il tournait sans le moindre problème jusqu'à 80 MHz au moins. Actuellement 75 MHz sont devenus standard, une version 100 MHz si annoncée depuis un certain temps n'est pas encore disponible.

Puisque nous en sommes à parler du contrôleur pourquoi ne pas jeter un coup d'oeil rapide sur le schéma de la **figure 1**. Outre le contrôleur, les composants les plus importants s'appellent LM1881, IC9, un séparateur de synchro vidéo et ADC1175, un CAN (Convertisseur Analogique/Numérique). On retrouve 2 périphériques pour microcontrôleur, IDT6698, IC5, une RAM 64 K x 4 servant à la mémorisation des informations d'image et une EEPROM I2C. Le contrôleur pilote la RAM en mode sériel au travers de compteurs rapides, IC3 pour les points d'image par ligne (pixels) et IC8 pour les lignes d'image. Le multiplicateur d'horloge ICS502 dérive du quartz de

20 MHz l'horloge de 80 MHz nécessaire au contrôleur. Le CAN est lui cadencé à 20 MHz.

L'ampli T1/T2 rehausse le signal vidéo à traiter à 2 V environ et en filtre la chroma (porteuse couleur) par le biais de L1, C5 et C6. Le CAN, IC2, numérise sur 8 bits le signal amplifié, le résultat attaquant le port RC du contrôleur.

Le séparateur de synchro extrait du signal vidéo d'une part le signal de ligne (BP) utilisé pour la synchronisation du contrôleur et le pilotage du circuit de verrouillage, IC7.C et T3 de façon à ce que le signal vidéo couplé en alternatif garde le niveau requis. Le logiciel a en outre besoin du signal Pair/Impair (*Odd/Even*) pour pouvoir reconstituer correctement les deux demi-images mémorisées.

Il faut tout d'abord que la télécommande IR « apprenne » au contrôleur le signal servant à arrêter l'enregistreur lors d'une pub. D'où la présence d'un récepteur IR du type SFH203A. À l'origine, le récepteur attaquait directement une entrée à trigger de Schmitt, mais une porte NON-ET (*NAND*) classique de IC7.D remplit ici la même fonction. Il n'est pas nécessaire de prévoir d'amplification vu que la sensibilité de l'ampli doit être, à dessein, faible (quelques cm d'écart de fonction seulement). Depuis le trigger de Schmitt le signal est, si besoin était, transféré à l'entrée RB1 du contrôleur par le biais du circuit multiplexeur que constituent IC7 et T4.

Les 2 signaux IR (enregistrement et pause) reçus par le magnétoscope sont mémorisés dans l'EEPROM sérielle IC6, une 24C08, d'où ils sont, en cas de besoin, extraits et émis par le biais de la diode d'émission IR LD1, une LD271H.

Si le bloqueur de pub *ViConti* ne possède pas d'affichage pour visualisation de son état, il n'en comporte pas moins 2 LED, D3 et D4. Pilotées par une seule ligne elles peuvent visualiser 4 situa-

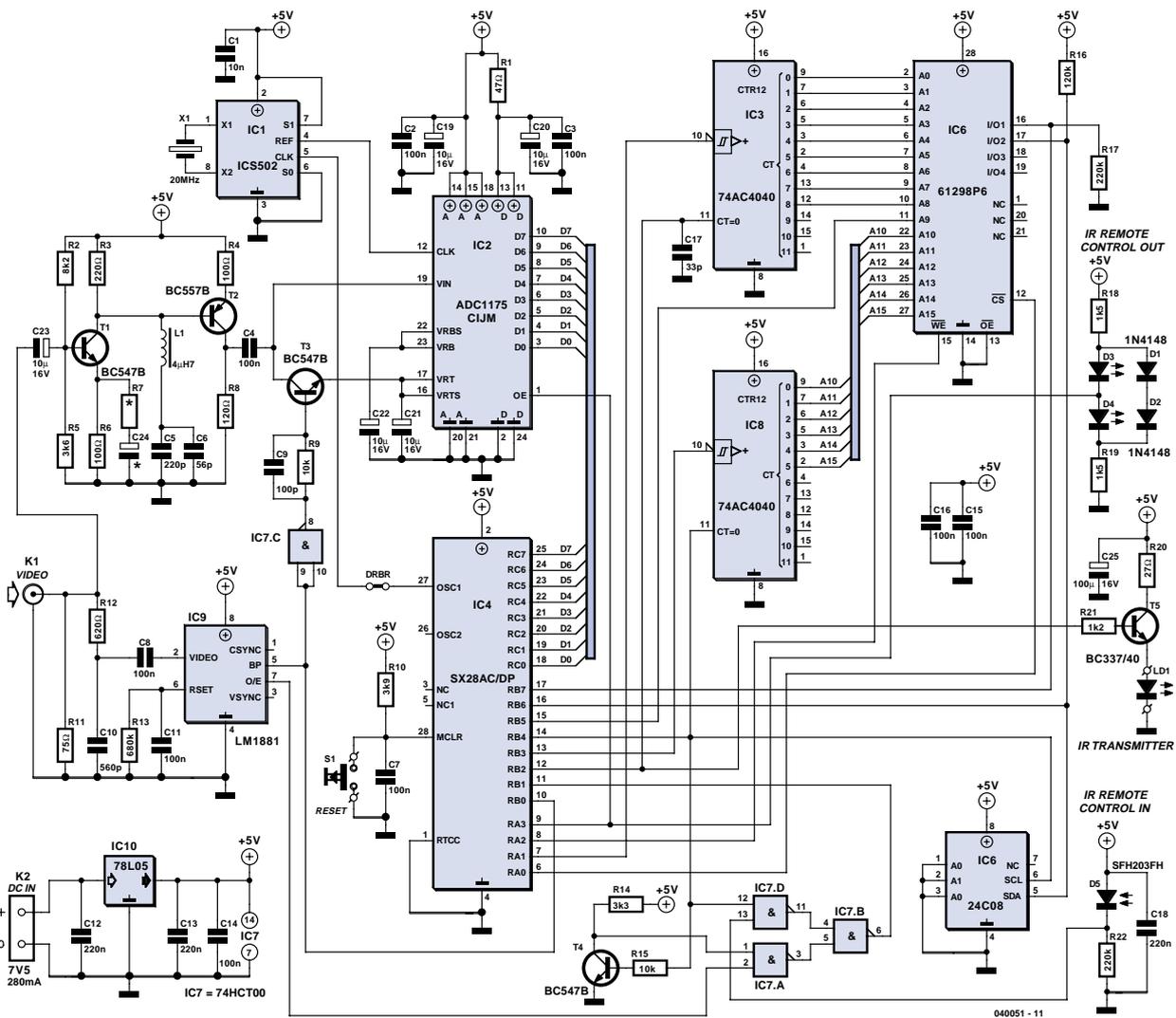


Figure 1. Le coeur du bloqueur de pub ViConti est un microcontrôleur de Scenix surcadencé.

tions (éteintes, vert allumé, rouge allumé et allumées).

### Logo: 3 phases

À quoi un être humain reconnaît-il un logo de chaîne ? Un coup d'oeil aux hebdomadaires donnant les programmes et on aura reconnu les caractères et graphiques... Un contrôleur lui ne lit pas de journal et ne comporte pas d'entrée de « déjà vu ». Comment fait-il alors pour reconnaître le logo d'une chaîne ?

Le logo de chaîne est, en règle générale, le seul contenu d'image statique alors que le reste de l'image change plus ou moins rapidement. Il faut partir voir à quel endroit de l'image TV

le contenu ne change pas, exception faite du moment où le logo disparaît changement signalant le début d'un bloc de pub.

Le logo étant également détectable sur les écrans N&B, on pourra se contenter d'un traitement de l'image Noir&Blanc. À l'endroit où se trouve le logo, les pixels conservent la même luminosité. Le pixel numérisé conserve la même valeur, les pixels environnant présentant des variations de luminosité plus ou moins importantes.

#### Phase 1

Si l'on attribue au logo de chaîne un niveau de luminosité minimal de, disons, 20%, on pourra attendre que les pixels environnants passent, en fonc-

tion du type d'émission, tôt ou tard, mais espérons-le avant l'arrivée du bloc de pub, au minimum une fois en dessous indiquant ainsi qu'ils ne font pas partie du logo. C'est ce qu'il faut détecter de façon à établir une « cartographie » en N&B de l'écran (noir = 0, blanc = 1 = logo). La figure 2 montre une partie d'écran avec logo et le logo filtré.

Le contrôleur prend en compte sur le port RC les pixels présentés sous la forme d'un nombre à 8 bits et les compare à une valeur de seuil et place, lorsqu'ils sont reconnus comme ne faisant pas partie du logo, dans la mémoire d'image un pixel noir (= premier bit, couleur de base = blanc) à l'emplacement correspondant.

En raison de la taille limitée de la RAM interne du SX28, 136 octets seulement, il faut le doter d'une mémoire externe. Il a fallu, pour conserver les lignes de port précieuses, intercaler une circuiterie auxiliaire constituée de 2 compteurs rapides chargés du pilotage des lignes d'adresse de la mémoire. Avec un SX28 travaillant à 80 MHz, le programme en assembleur requiert très exactement 9 cycles-machine (instructions) à 12,5 ns chacune pour le traitement d'un pixel. (Cf. EXTRAIT DE LISTAGE 1)

Dans ces conditions on pourra, sur une longueur de ligne visible de l'image TV de 52  $\mu$ s, traiter un nombre approximatif de 460 pixels par ligne. La synchronisation du programme avec les lignes d'images fournies se fait ligne par ligne par une interruption externe générée par le signal lignes du LM1881.

Le traitement se limite à une bande de 128 lignes dans le tiers supérieur de l'image, vu que c'est l'endroit de prédilection des logos de chaîne. À intervalle régulier on vérifie s'il se trouve des pixels blancs (faisant partie du logo) dans un rectangle réservé typiquement au logo dans le coin en haut à gauche ou à droite de l'écran. Si tel est le cas, c'est le début de la

## Phase 2

Vu qu'il faut, en cas d'absence du logo, réagir rapidement, en quelques dixièmes de seconde dans la situation idéale, il nous faut des critères permettant de détecter, indépendamment d'un mouvement aléatoire du contenu de l'image, la présence ou non du logo identifié. Ici encore on peut se contenter de l'image N&B. Le logo constitué de quelques centaines de pixels possède une luminosité moyenne par pixel typique qui se différencie normalement fort bien de la valeur des pixels environnants. Les paramètres typiques cor-

## EXTRAIT DE LISTAGE 1

```

mov W,#$02 ;Compteur de pixels: 460 pixels par ligne!
mov $1E,W
mov W,#$CD
mov $1D,W ;Boucle pour le test d'une ligne d'image

:loop1 setb RA.0 ;Remise à zéro impulsion écriture mémoire ext.
      clrb RA.1 ;Remise à zéro horloge compteur pixels ext.
      mov W,#$33 ;Seuil de recherche de logo
      mov W,RC-W ;Si W < RC positionner Carry-Flag
      rl RA ;Impulsion écriture mémoire ext. si Carry pas positionné
          ;Horloge compteur pixels ext.
      decsz $1D
      jmp :loop1
      decsz $1E
      jmp :loop1

```



Figure 2. On voit en haut une partie de l'image d'origine en N&B, en dessous le logo en ayant été filtré.

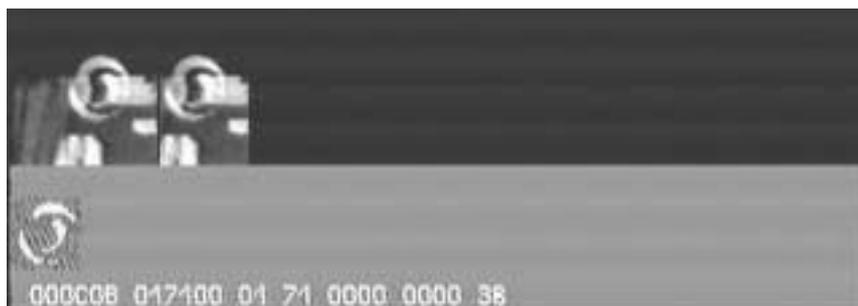


Figure 3. Partie de l'image d'origine, à côté partie d'image stockée ligne par ligne qui sera comparée au logo en dessous. Les pixels blancs représentent le logo, les noirs sont les pixels de référence, les gris sont neutres. Les nombres donnent les valeurs de seuil et de référence.

respondant au logo présent et à son environnement sont déterminés au cours de cette phase avant le début de l'étape de surveillance.

La taille du logo est déterminée par un examen ligne par ligne des pixels en mémoire à la recherche des pixels blancs « les plus extérieurs ». Ajoutons-y 3 pixels tout autour et les coordon-

nées de la plage de surveillance du logo (le rectangle du logo) sont fixées ! On compte les pixels blancs présents dans le logo et on distribue alors aléatoirement exactement le même nombre de pixels noirs dans le rectangle du logo. Les pixels restants sont neutralisés (peints en gris) par le biais du second bit dans la mémoire d'image.

## EXTRAIT DE LISTAGE 2

```
mov W,$1D ;Contient largeur du rectangle du logo
mov $04,W

:loop4 setb $04.4
nop
mov W,RC ;xxx octets lire l'image du CAN dans la mémoire
RAM
mov $00,W
nop
incsz $04
jmp :loop4
```

## EXTRAIT DE LISTAGE 3

```
mov W,$1D
mov $04,W

:loop5 inc RA ;Lire xxx octets de la mémoire RAM et les
additionner
setb $04.4
mov W,$00
snb RB.6 ;Bit de mémoire ext. de la mémoire d'image
jmp :grau
sb RB.7 ; Bit de mémoire ext. de la mémoire d'image
jmp :schwa
add $09,W ;Additionner blancs
snb C_Flag
incsz $0A ;Somme blancs dans $0B, $0A, $09
dec $0B
inc $0B
dec RA
incsz $04
jmp :loop5
ret

:schwa add $0C,W ;additionner noirs
snb C_Flag
incsz $0D ;somme noirs dans $0E, $0D, $0C
dec $0E
inc $0E
dec RA
incsz $04
jmp :loop5
ret

:grau jmp :x3 ;dummy pour pixel gris
:x3 jmp :x4
:x4 dec RA
incsz $04
jmp :loop5
ret
```

Note : langue d'origine du fichier assembleur est l'allemand.

### Phase 3

Suivi du programme en temps réel, c'est-à-dire image par image. On détermine, pour chaque image, la moyenne de luminosité des pixels du logo et d'un nombre identique de pixels repré-

sentatifs ne faisant pas partie du logo. Lorsque la différence entre les 2 valeurs passe à plusieurs reprises en deçà d'une valeur de seuil prédéterminée, cela constitue un indice signalant la disparition du logo.

La **figure 3** montre, en haut, un extrait de l'image originale, à côté l'extrait d'image mémorisé ligne par ligne comparé au logo en dessous. On voit à cet endroit les pixels blancs du logo et les pixels de référence noirs. Les pixels gris sont neutres, les nombres représentant les valeurs de seuil et de référence.

Le patron de bit stocké en EEPROM de la télécommande IR pour l'interruption de l'enregistrement vidéo est émis par l'émetteur IR dans l'espoir que l'enregistreur le détectera et fera en sorte d'interrompre l'enregistrement.

La surveillance du programme se poursuit, sachant que l'on relève quelque peu la valeur de seuil de la différence de luminosité moyenne pour la reconnaissance du retour du logo de chaîne. Dès que le logo a été identifié avec certitude, l'émetteur IR émet en direction de l'enregistreur le signal pour la reprise de l'enregistrement.

Pendant la phase de surveillance la position du logo dans l'image TV est connue. Maintenant, on mémorise dans la RAM interne du microcontrôleur, ligne par ligne, la série des pixels constituant le logo.

Pour obtenir la même résolution qu'en phase 1, on travaille à nouveau à 9 cycles-machine par pixel (Cf. EXTRAIT DE LISTAGE 2)

Au cours de l'intervalle de temps avant la fin de ligne on détermine si le pixel mémorisé à cet instant fait partie du groupe des pixels du logo (blanc), des pixels de référence (noir) ou des pixels neutres (gris) tout en faisant une moyenne de la somme de luminosité à cet instant (Cf. EXTRAIT DE LISTAGE 3)

Une fois que toutes les luminosités de pixel ont été additionnées, on crée la différence des sommes de luminosité du logo et des pixels de référence et on procède, en fonction de la taille du

# Remarque :

Il est important, pour le fonctionnement du bloqueur de pub que :

- Le logo de la chaîne se trouve dans le tiers supérieur de l'image et à une position fixe (ce qui est le cas avec nombre de chaînes françaises et la majorité des chaînes étrangères)
- Le signal vidéo soit de très bonne qualité. Si l'image ne synchronise pas à 100%, le logo frétille (ce qui n'est pratiquement pas visible à l'œil nu) ce qui empêche une identification correcte du logo.

logo, à une comparaison avec une valeur de seuil adaptée. Si, successivement, on trouve des images TV présentant une différence trop faible, on en déduit que *le logo a disparu* et on envoie la commande par infrarouge d'arrêter l'enregistrement.

Scenix propose un module logiciel de pilotage du bus I<sup>2</sup>C baptisé *Virtual Peripheral*; nous l'avons modifié pour la commande de l'EEPROM en vue de la mémorisation du code de télécommande IR de l'enregistreur à télécommander.

Comme il n'est nécessaire d'envoyer à l'enregistreur que 2 commandes IR, l'intégration d'une fonction d'apprentissage tombe sous le sens. Les patrons d'impulsions enregistrés par interruption subissent, en mode d'apprentissage, une quantification sur une chrono-trame, les valeurs obtenues étant stockées définitivement en EEPROM. En cas de besoin, les données en EEPROM sont transférées en RAM de sorte que le train d'impulsions pourra être reconstruit, indépendamment du code utilisé avant d'être émis par l'émetteur IR.

## Mise en oeuvre

Pour un fonctionnement correct il faut disposer d'un très bon signal vidéo. Si l'image ne se synchronise pas à 100%, le logo frétille (ce que l'on ne détecte pas à l'œil nu) mais qui empêche une identification correcte.

Après application de la tension d'alimentation ou une action sur la touche d'initialisation (*Reset*), le programme se déroule en 4 étapes. Au cours de la première on peut choisir les 2 commandes IR servant au pilotage de l'enregistreur.

### APPRENTISSAGE DU CODE IR

Allumer la LED verte  
Attendre le signal IR

On dispose alors de 4 secondes environ pour actionner la touche requise (REC par exemple). Il faudra placer l'émetteur IR de la télécommande à quelques centimètres seulement de la diode IR de réception du bloqueur de pub. S'il ne reçoit pas de signal IR au cours de ces 4 s, le programme passe à l'étape suivante : RECHERCHE DU LOGO, sinon il passe aux points suivants :

Détection d'un signal IR  
Clignotement de la LED verte  
Lecture du signal IR  
Mémoriser le code en EEPROM  
Extinction de la LED verte  
Allumage de la LED rouge  
Attendre un signal IR

Le processus d'apprentissage du code PAUSE est similaire à ceci près que l'on ne dispose que de 2 s. En l'absence de réception de signal IR au cours de ces 2 s, le programme passe à l'étape suivante : RECHERCHE DU LOGO, sinon il passe aux points suivants :

Détection d'un signal IR  
Clignotement de la LED rouge  
Lecture du signal IR  
Mémoriser le code en EEPROM  
Extinction de la LED rouge

La saisie des commandes d'enregistrement (*Record*) et d'arrêt (*Pause*) –il est bien entendu possible de programmer d'autres

commandes de télécommande– ne se fait qu'une fois, lors de la première utilisation ou lorsque l'on veut modifier les commandes. Sinon il suffit de mettre l'appareil sous tension !

### RECHERCHE DU LOGO

Allumer les LED rouge et verte

Voici la procédure exécutée successivement pour la première et la seconde demi-image :

Attendre une interruption de début d'image  
Attendre les interruptions de ligne  
Traitement des lignes 33 à 96

S'il est impossible de détecter le logo dans la mémoire d'image externe, le programme reste dans cette boucle, sinon il passe à l'étape

### ANALYSE DU LOGO

Nous avons examiné ce processus dans le détail dans le paragraphe intitulé Phase 2.



# Liste des composants

## Résistances :

(toutes à film métallique, 0,25 W/5%)

R1 = 47 Ω  
 R2 = 8kΩ  
 R3 = 220 Ω  
 R4,R6 = 100 Ω  
 R5 = 3kΩ  
 R7 = non implantée  
 R8 = 120 Ω  
 R9,R15 = 10 kΩ  
 R10 = 3kΩ  
 R11 = 75 Ω

R12 = 620 Ω  
 R13 = 680 kΩ  
 R14 = 3kΩ  
 R16 = 120 kΩ  
 R17,R22 = 220 kΩ  
 R18,R19 = 1kΩ  
 R20 = 27 Ω  
 R21 = 1kΩ

## Condensateurs :

C1 = 10 nF/63 V NPO  
 C2 à C4,C7,C8,C11,C14 à C16 =

100 nF/63 V X7R  
 C5 = 220 pF/63 V NPO  
 C6 = 56 pF/63 V NPO  
 C9 = 100 pF/63 V NPO  
 C10 = 560 pF/63 V NPO  
 C12,C13,C18 = 220 nF/63 V X7R  
 C17 = 33 pF/63 V NPO  
 C19 à C23 = 10 μF/16 V E2,5-5  
 C24 = non implanté  
 C25 = 100 μF/16 V E2,5-6

## Semi-conducteurs :

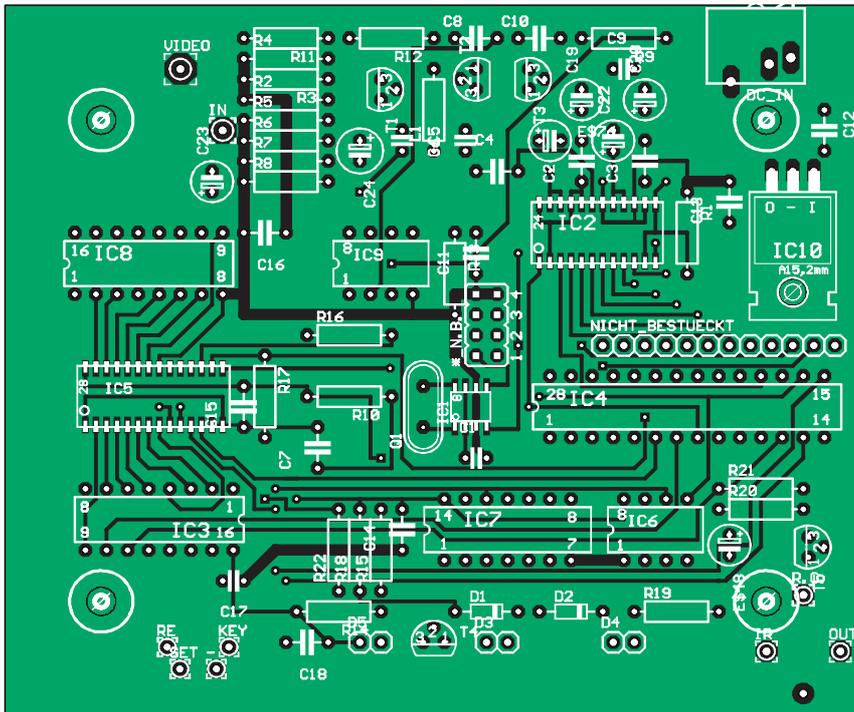


Figure 4. La sérigraphie de la platine double face montre que le « client » source de problèmes potentiels est la SRAM en boîtier SOJ à 28 pattes.



## SURVEILLANCE DU LOGO

La procédure suivante est exécutée successivement pour la première et la seconde demi-image :

- Attendre une interruption de début d'image
- Attendre les interruptions de ligne
- Traitement des lignes 33 à 96

Après traitement, le logiciel prend une décision :

Logo présent  
 La LED verte clignote périodiquement  
 Retour à la procédure SURVEILLANCE DU LOGO

À nouveau pas de détection de logo  
 La LED rouge clignote périodiquement  
 On passe à ÉMISSION DU CODE IR 1

Le logo est à nouveau détecté  
 La LED verte clignote périodiquement  
 On passe à ÉMISSION DU CODE IR 2

D1, D2 = 1N4148  
 D3 = LED 5 mm rouge (faible courant)  
 D4 = LED 5 mm verte (faible courant)  
 D5 = SFH203FA (Infineon)  
 T1, T3, T4 = BC547B  
 T2 = BC557B  
 T5 = BC337/40  
 LD1 = LD271-H (Infineon/Osram) \*  
 IC1 = ICS502M (ICS)  
 IC2 = ADC1175CIJM (National)  
 IC3, IC8 = 74AC4040  
 IC4 = SX28AC/DP (Scenix, devenu

Ubicom)  
 IC5 = 61298P6 SOJ28-3 (IDT)  
 IC6 = NM24C08N08E (Fairchild)  
 IC7 = 74HCT00  
 IC9 = LM1881N08E (National)  
 IC10 = 78L05

#### Divers :

X1 = quartz 20 MHz (HC49U-H)  
 L1 = self 4 $\mu$ H7  
 K1 = embase Chinch  
 K2 = embase basse tension 2 mm

K3 = mini-embase jack  
 S1 = bouton-poussoir à contact travail (tel que, par exemple, Schurter 1301.9502 sans capuchon)  
 alimentation secteur 7,5 à 9 VCC/300 mA  
 boîtier

#### Tête d'émission IR

jack mâle (tel que, par exemple, Lumberg KLS2SL)  
 diode d'émission IR LD271-H (Infineon)

Figure 5. Le cavalier placé ainsi sur l'embase de programmation sert au transfert de l'horloge vers le microcontrôleur.

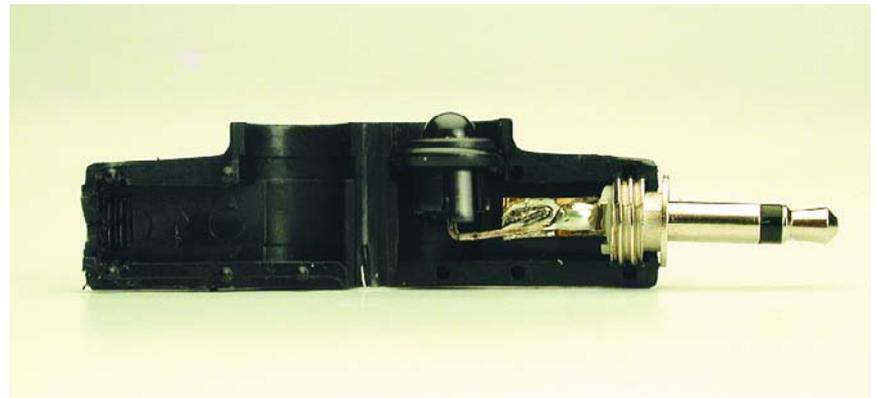
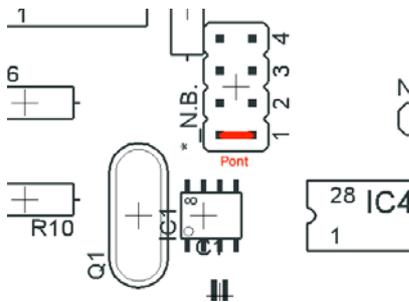


Figure 6. La tête d'émission IR est constituée d'un jack auquel on soude la diode au lien d'un câble.

#### ÉMISSION DU CODE IR 1

Chercher le premier code IR appris dans l'EEPROM et l'émettre 3 fois  
 Revenir à la procédure SURVEILLANCE DU LOGO

#### ÉMISSION DU CODE IR 2

Chercher le second code IR appris dans l'EEPROM et l'émettre 3 fois  
 Revenir à la procédure SURVEILLANCE DU LOGO

### Construction

Les dimensions de la platine double face à trous métallisés du bloqueur de pub sont telles qu'il se glisse exactement dans le coffret mentionné sans qu'il ne soit nécessaire de câbler de composants. L'implantation des composants appelle quelques remarques vu que le multiplicateur d'horloge, le CAN et la RAM sont des CMS. Si le soudage des 2 premiers composants ne requiert qu'un fer à souder à pointe fine et une main assurée, la mémoire SRAM elle se présente sous la forme d'un boîtier SOJ

dont les broches se trouvent sous le composant. On utilisera une méthode « brutale » mais efficace :

1. Fixer le circuit en 2 endroits diamétralement opposés en soudant ses connexions extérieures.
2. On soude aussi rapidement que possible, sans tenir compte des ponts de soudure pouvant être occasionnés par cette opération, toutes les pattes. On préférera un fer à souder normal (pas de pointe en aiguille) les choses allant plus vite.
3. On positionne devant les pattes soudées un morceau de tresse à dessouder imbibée de flux de désoudage et on fait passer un fer à souder bien chaud le long des pattes. Avec un rien de chance, on aura pu éliminer ainsi tout excédent de soudure, les soudures paraissant fort honnêtes. Il va bien entendu falloir vérifier qu'il n'existe plus de pont de soudure. L'important est que la soudure contienne suffisamment de flux de soudage et que les choses aillent vite.

La mise en place du reste des composants n'appelle pas de remarque particulière. Attention à la polarité correcte des diodes, condensateurs, transistors et circuits intégrés.

L'embase au centre de la platine sert uniquement lors d'une programmation en circuit (*In-Circuit-Programming* à l'aide de la SX-Key de Parallax) du contrôleur. Pour une utilisation normale, il suffit, comme le montre la **figure 5**, d'implanter un cavalier en position 1, de manière à fournir l'horloge au contrôleur.

La **figure 6** montre, en guise de conclusion, la technique de réalisation de l'émetteur IR. On soude une diode IR à un jack, l'ensemble étant placé dans un support pour jack en équerre. La diode est verrouillée à l'aide d'une rondelle en plastique. Il suffira ensuite d'orienter la LED pour qu'elle soit parfaitement dirigée sur la diode de réception de l'enregistreur.

(040051-1)

# 10 ANS PLUS TARD

## La DAB en Europe

*Hans Weber*

-  Operational Services
-  Pre-Operational Services

Cela fait plus de 10 ans que l'on travaille en Europe à l'introduction commerciale de la radiodiffusion numérique, le DAB. Si dans certains pays d'Europe de l'Ouest et Centrale, le réseau se développe selon les plans, il n'en va pas de même en ce qui concerne la vente des récepteurs. Depuis fin 2002, les choses ont changé en Grande-Bretagne et les premiers signes d'un retournement de tendance se font jour sur le reste du Continent.



Il est un point sur lequel tous les stratèges et les techniciens des organismes de radiodiffusion européens sont d'accord : les systèmes analogiques trafiquant en FM voire même en Petites Ondes sont techniquement dépassés, la qualité de réception n'est pas optimale et les fréquences disponibles ne sont plus suffisantes. Dans les 15 prochaines années la radiodiffusion analogique devra être complètement numérisée. Et pourtant la DAB (*Digital Audio Broadcasting*) est resté coincée dans ses *starting-blocks*. La France, et l'Allemagne aussi, constituent un exemple type d'une stratégie ratée.

### **Un peu de patience, S.V.P.**

Depuis quelque temps déjà, tous les 2 ans, le Salon de la Radio de Berlin prévoyait la percée définitive de DAB. Les plans de la réalisation d'un réseau d'émetteurs voyaient le jour, les producteurs de programmes d'état et commerciaux annonçaient leur intention de passer très bientôt à l'âge de la radiodiffusion numérique, certains fabricants n'hésitaient même pas à présenter des prototypes, puis récemment des appareils de série, dont le prix ne répond cependant pas aux impératifs d'un marché de masse.

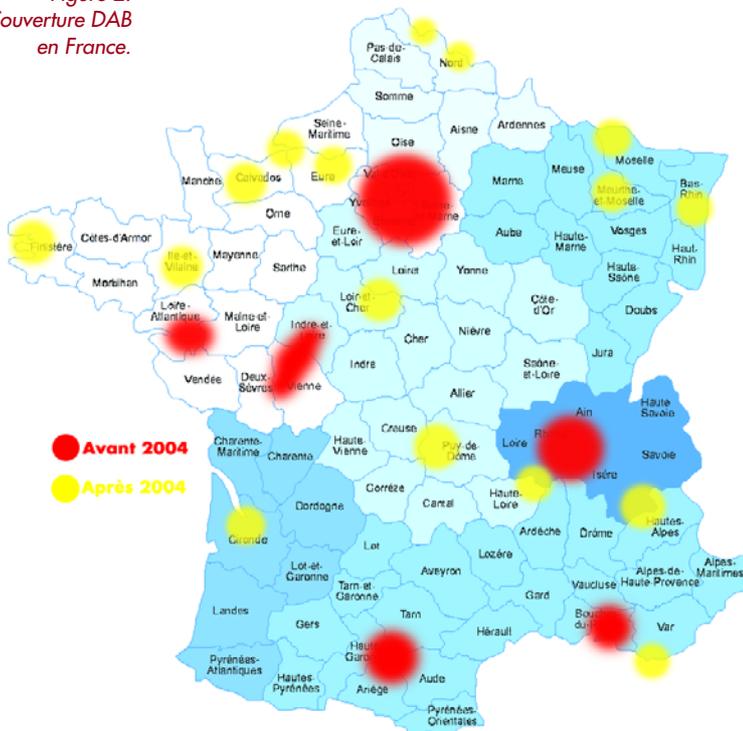
Dans les couloirs des organisations ayant trait à la radiodiffusion on se mit à parler d'approche synthétisée, de répartition des fréquences. Les experts ne sont pas encore d'accord quant à la réponse à donner à la question de savoir s'il faut proposer à l'auditeur ses programmes habituels en numérique cette fois (en « Simulcast ») ou s'il faut lancer de nouveaux programmes en vue d'attirer une nouvelle clientèle. La radio classique suffit-elle comme application ou seront-ce de nouveaux services de données qui produiront la percée finale ? Mais il se fait également jour une sorte de « guerre des religions » : DAB ou

TNT (Télévision Numérique Terrestre) ? voire même DRM ? L'an dernier, la décision semble avoir été prise : le successeur numérique de la radio FM ne pourrait être que DAB. Il est utopique, d'imaginer un réseau d'émetteurs TNT couvrant effectivement la totalité de l'Hexagone dans les prochaines années. Qui le financerait ? Il ne serait pas rentable d'imaginer un multiplex TNT réservé à la radiodiffusion, sauf pour certaines zones à forte densité de population, de sorte que les programmes de radio se verraient toujours voyager en *piggy-back* sur les programmes de TV. Une troisième forme de radio numérique terrestre, DRM (*Digital Radio Mondiale*) ne convient pas, sous sa forme actuelle, pour une distribution régionale et encore moins locale. En conclusion, DAB, TNT-R et DRM ont été conçus pour 3 domaines d'applications différents; ces systèmes peuvent coexister voire se compléter, mais pas se substituer l'un à l'autre.

Tout le monde était d'accord sur un point, la mise en place du réseau DAB ne pouvait se faire qu'avec une aide importante de l'État. Dans le domaine technique les choses avancent grâce à des fonds importants. Si la France est en retard par rapport à la plupart des pays européens, 25% de la population seulement a accès à DAB, alors que ce pourcentage atteint plus de 80% en Allemagne, de 85% en Angleterre avec un choix de services doubles de celui de la FM, 98% en Belgique, 75% au Danemark, 50% en Espagne, 70% au Portugal, 50% en Norvège et plus de 40% en Finlande, il semblerait que l'on mette les bouchées doubles maintenant que la décision d'installer DAB a été prise. Les plus optimistes prévoient une couverture numérique de la France achevée en 2006. Le temps presse. Mais d'autres sources sont bien moins optimistes... En effet, hormis d'anciens sites de diffusion pilote dont les autorisations sont à ce jour caduques, il n'existe pas encore,

Figure 1.  
Couverture DAB  
dans le monde  
(source World DAB Forum).

Figure 2.  
Couverture DAB  
en France.



à l'écriture de ces lignes, d'autorisation de diffusion officielle du DAB en France.

Pour nous faire une idée de ce qui pourrait nous attendre, prenons l'exemple de la RFA. Le nombre de programmes proposés ne cesse de croître, les partisans de l'approche « nouveaux programmes » comme la Bavière, jouent la carte de nouveaux programmes impossibles à écouter en FM, se trouvant confrontés à ceux qui préfèrent la philosophie Simulcast, principalement pour des raisons de coûts.

### Récepteur à 99 €

Les choses se présentent tout autrement en Grande-Bretagne : Le marché a pris vie fin 2002. On s'y était rendu compte que le succès des récepteurs DAB serait inévitablement lié à leur prix de revient. On y considérait que la « barrière magique » se situait en deçà des 100 livres. L'initiative de proposer un « récepteur pour cuisine à 99 livres » n'avait pas pour origine les fabricants, mais les producteurs de programmes. La BBC et d'autres stations privées dont Digital One en particulier, mirent des moyens à disposition pour le développement d'une puce, « Chorus », dont la fabrication coûtait si peu qu'il devenait possible de franchir cette barrière magique, avec quelques subventions au départ il est vrai. Juste avant Noël 2003, «PURE Digital Evoke-1» fut, soutenu par une campagne publicitaire, proposé dans différents magasins sélectionnés et vendus en quelques heures seulement. Ce « pré-allumage » a incité d'autres fabricants à se lancer dans la bataille avec d'autres offres intéressantes. L'impact de cette multiplication d'offres fut une chute des prix, le récepteur le moins cher ne coûtant plus que 70 £ (105 €), et une prise de conscience par le grand public de l'existence de ces produits, même si le prix est encore (relativement) élevé.

Le nombre des récepteurs DAB vendu passa de quelque 50 000 mi-2002 à plus d'un demi-million et ceci avec une couverture territoriale similaire à celle de l'Allemagne voire devenue même moindre depuis.

À noter au passage qu'en Grande Bretagne la situation de départ était autre que celle existant en RFA : les stations trafiquant en Petites Ondes y étaient bien plus courantes de sorte que la différence de qualité par rapport à DAB était sensiblement plus évidente qu'elle ne l'est par rapport à la FM. De plus, la DAB s'est vu attribuer des programmes impossibles à capter en analogique fort bien soutenus par les médias.

Autre avantage par rapport à la RFA : en Grande Bretagne il n'existe qu'une plage de fréquences pour DAB, la bande III (de 174 à 240 MHz). En RFA on émet également en bande L (1 452 à 1 492 MHz), ce qui accroît la complexité et limite la portée des émetteurs.

### Europe de l'Ouest : différences

Comme nous le disions plus haut, la Belgique peut se targuer d'une couverture presque complète, avec 98%, ce qui situe ce pays en tête de DAB.

Nous en venons à la situation en France, où, dès le départ, on s'est concentré sur les régions à forte densité de population. Comme le montre la carte plus haut, Paris, Lyon, Marseille, Toulouse et Nantes furent les premières métropoles régionales à pouvoir capter des programmes DAB. Les Pays-Bas furent le dernier pays de la Communauté Européenne (dans son état d'avant le 1er mai 2004) à avoir, le 27 février dernier, décidé

Figure 3. Tuner DAB  
haut de gamme  
EDAB de Restek.



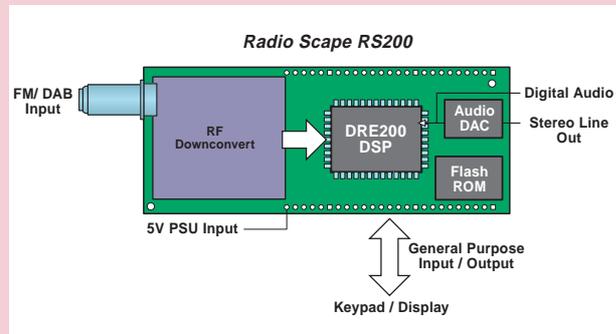
Figure 4. Il existe  
déjà des récepteurs  
DAB compacts avec  
gamme L pour  
moins de 200 €  
(www.thiecom.de).



## Au coeur de DAB

Lors d'une recherche de composants pour récepteur DAB on bute inévitablement sur 2 fabricants anglais : FrontierSilicon ([www.frontier-silicon.com](http://www.frontier-silicon.com)) et RadioScape ([www.radioscape.com](http://www.radioscape.com)). Le Chorus FS1010 constitue le produit de base de la palette de FrontierSilicon. Il s'agit d'un processeur multi-media à forte intégration doté d'un CAN en entrée, d'un noyau de processeur DSP, d'un coprocesseur DAB, de RAM embarquée et de mémoire cache, sans oublier une palette impressionnante de périphériques. Il ne comporte pas de CNA de sorte que l'utilisateur potentiel peut compliquer les choses à loisir, avec ou sans CNA pour écraser les coûts. FrontierSilicon propose également des modules complets à l'intention des fabricants de récepteurs, modules comportant déjà le tuner (front end) et quelques autres périphériques. Avant de disposer d'un récepteur complet il faudra au fabricant ajouter sa propre interface utilisateur (processeur de commande, affichage, touches), la partie BF (CAN, tampon, embases/Toslink, etc.) et bien entendu le coffret. RadioScape va encore plus loin, proposant, avec sa minuscule carte-module RadioScape RS200, sur laquelle règne le circuit DAB DRE200 de Texas Instruments, qui intègre la totalité des fonctions d'un récepteur DAB/FM numérique. Comme le montre le croquis, il reste à ajouter quelques touches, un encodeur rotatif, un affichage LCD standard (2 x 20 caractères) et une alimentation, pour disposer d'un tuner DAB/FM. Les dimensions de ce module permettent de réaliser des récepteurs de poche à base de module Radioscape.

On ne sera guère étonné d'apprendre que la plupart des récepteurs DAB sont dotés de tels modules. La photo montre l'intérieur du tuner DAB de la série « Audio Mini Modul » de Restek.



Proche du Plug&Play : le module tuner DAB/FM RS200 de RadioScape ([www.radioscape.com](http://www.radioscape.com)).



Au coeur du tuner DAB de la série « Audio Mini Modul » de Restek ([www.restek.de](http://www.restek.de)).

de se lancer dans la DAB après avoir procédé à des tests ayant duré pas moins de 5 ans. Ici aussi, on a opté pour l'approche « forte densité de population » de sorte que 40% de la population est actuellement « à portée » de DAB.

Le point commun de ces derniers pays est que le nombre de récepteurs vendus se situe bien en retrait de l'évolution du réseau d'émetteurs DAB. Cela peut tenir au fait que ce marché-niche n'est pas encore assez « juteux » pour le moment.

### Les appareils du marché

Le marché grand public se subdivise en 4 parties importantes. Les auto-radios, considérées comme le segment de marché le plus important, ce qui explique l'entrain mis à couvrir les axes routiers importants (celui de Tours à Poitiers a servi de zone de test). Les prix restent relativement constants. Le leader du marché Blaupunkt propose son « Woodstock 53 » pour 579 €. Le Allixx de Grundig est un peu plus abordable à

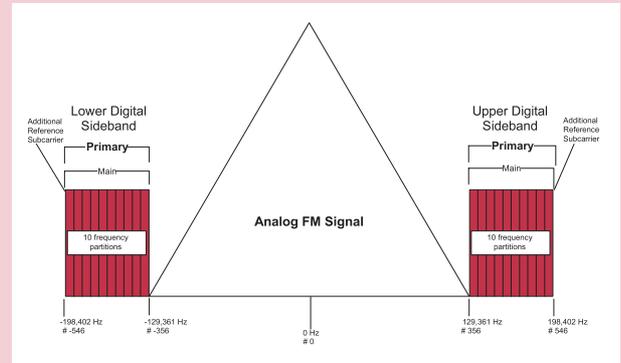
399 €. Les prix en varient guère d'un pays à l'autre. Seule la Grande Bretagne est, avec une offre inférieure à 200 £ (300 € pour le « Goodmans »), en avance sur le reste de l'Europe.

Il en va autrement pour les récepteurs mobiles ou portatifs : au Royaume Uni ce sont les petits récepteurs bon marché (tout est relatif) qui ne connaissent que la bande III, qui constituent l'offre la plus intéressante. Il est possible de trouver des récepteurs portatifs pour de l'ordre de 70 £ (105 €). En Allemagne, les prix sont, jusqu'à présent, restés coincés au-delà de 200 €. Jusqu'à présent, car au Salon CeBIT 2004 TechniSat a présenté des récepteurs meilleur marché : le DAB-Man 1 pour 169,99 € et comme récepteur de salon, le Digit-Radio DAB pour 159,99 €. À noter que c'est à CeBIT que fut présenté le premier récepteur tant pour DAB que pour DRM, le Starwaves Prelude. Ce récepteur fabriqué par une petite société de Hanovre ne sera disponible qu'en fin d'année : son prix, un « nombre d'euros à 3 chiffres haut de gamme ».

## HD-Radio au lieu de DAB

Dans l'article « DAB, la Radio Numérique » du numéro de mars 1998, nous disions que la NBA (Organe représentatif de nombre de stations de radio) était contre DBA, lui préférant la Radio numérique In-Band pour la FM et les Petites Ondes, sachant que l'on n'a pas encore trouvé de solution pratique jusqu'à présent. Ceci est dépassé : sous la dénomination « HD-Radio », iBiquity Digital Corporation ([www.ibiquity.com](http://www.ibiquity.com)) a mis au point un procédé permettant une transmission numérique sur les fréquences FM et MA existantes par le biais des émetteurs disponibles et ceci parallèlement aux programmes MA et FM actuels qui continuent d'être émis sur les mêmes fréquences. Il suffit, pour jouir des nouveaux services de radio et de données de HD-Radio, de disposer d'un récepteur supportant HD-Radio. Si les programmes en FM atteignent alors la qualité CD, les programmes en MA arrivent au niveau de la FM actuelle. On dispose dans les 2 cas d'une réception sans parasites dotée de fonctions de données additionnelles. La plupart des récepteurs HD-Radio sont en mesure de recevoir les programmes en FM et MA normaux (lire non HD-Radio).

Avant même la mise sur le marché du premier récepteur, iBiquity avait donné des licences à plus de 280 stations de radio dans 37 états. Des fabricants de composants tels que ALPS, (tuner, CAN et CNA, processeurs de signal) et Philips (SAF3350, processeur HD Radio), ont l'intention de prendre le train en marche au courant de cette année.



Signal HD-Radio hybride : porteuse principale modulée en FM avec bandes latérales HD-Radio à modulation numérique.

Les tuners DAB pour installation audio Hi-Fi sont encore l'apanage de fabricants haut de gamme tels que, par exemple, Restek, ([www.restek.de](http://www.restek.de)). Il se peut que l'on ait, si l'on veut acheter un tel tuner DAB, à déboursier 2 700 €, alors que l'on trouve en Grande Bretagne un tuner DAB/FM-Tuner pour son installation audio pour de l'ordre de 150 €. Il n'existe pratiquement pas encore de récepteur DAB piloté Par PC. Le seul système distribué dans les différents pays est le DR Box 1 de TerraTec (399 € ou 270 €).

Pour un panorama des récepteurs disponibles on pourra faire un tour sur Internet, sur, par exemple, [www.digitalradio.de](http://www.digitalradio.de) pour l'Allemagne et sur [www.digitalradionow.com](http://www.digitalradionow.com) pour la Grande Bretagne.

### L'avenir ?

Il y a eu une époque où DAB avait été jugé « moribond ». La situation actuelle mérite une approche plus positive. On peut s'attendre à ce que les prix baissent encore à l'avenir. La société anglo-israélienne Sonarics propose son module DAB CSM 1 pour 25 dollars et un DSP Blackfin d'Analog Devices, permettant une implémentation logicielle des fonctionnalités DAB coûte, en grosses quantités, 5 dollars à peine. Finalement, les grands fabricants, Sony en particulier, se disent prêts à entrer sur le marché « très prochainement ».

La situation est complexe, vu qu'il n'est pas question d'une révolution du type tout ou rien, mais d'une évolution du type fusée à 3 étages comme le dit VRN (Vivement la Radio Numérique) : une couverture significative, de nouvelles radios

et de nouveaux contenus, des récepteurs disponibles à des prix abordables.

(040101-1)

### Littérature :

**DAB, la Radio Numérique, Elektor 03/1998, page 62 et suivantes et 04/1998, page 20 et suivantes**

### Liens Internet:

[www.worlddab.org](http://www.worlddab.org)

[www.VRN.com](http://www.VRN.com)

[www.radiofrance.fr/chaines/radio-france/diffusion/dab.php](http://www.radiofrance.fr/chaines/radio-france/diffusion/dab.php)

[www.tdf.fr/article/articleview/701/7/21/](http://www.tdf.fr/article/articleview/701/7/21/)

[www.digitalradio.de](http://www.digitalradio.de)

[www.radionumerique.org](http://www.radionumerique.org)

[www.pure-digital.com](http://www.pure-digital.com)

[www.drdb.org](http://www.drdb.org)

[www.digitalradionow.com](http://www.digitalradionow.com)

[www.restek.de](http://www.restek.de)

[www.sonarics.com](http://www.sonarics.com)

[www.thiecom.de](http://www.thiecom.de)

[www.technisat.de](http://www.technisat.de)

[www.radioscape.com](http://www.radioscape.com)

[www.frontier-silicon.com](http://www.frontier-silicon.com)

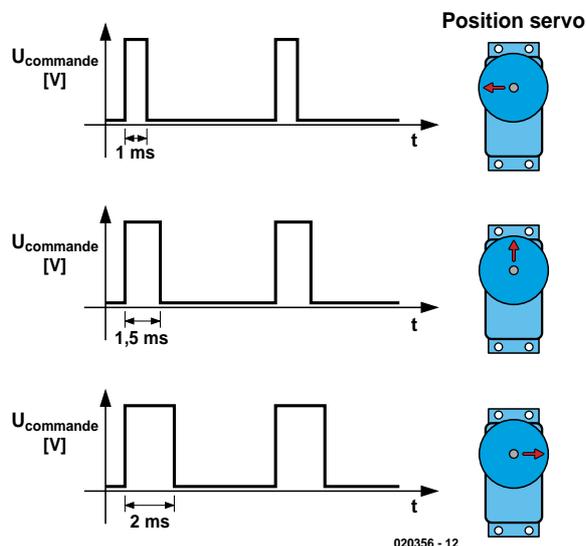
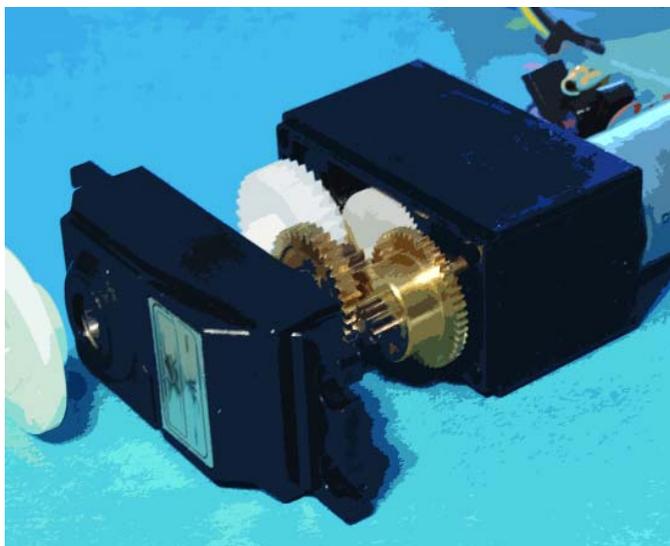


Figure 1. La largeur d'impulsion détermine la position.

# Motricité numérique

## Pilotage de servos intelligent

Giel Dols

Tout compte fait, les servo-commandes analogiques sont déjà des condensés d'ingéniosité. Il est bien entendu possible de faire encore mieux... passer au numérique ! Nous allons voir, en prenant une servo analogique comme référence, les avantages du numérique.

Une servo est une combinaison de moteur à courant continu, alternatif ou sans balais et d'un « circuit de détection de position ». Les modèles les plus utilisés en modélisme et en robotique sont les servos à courant continu trifilaires. Une telle servo se compose d'un moteur à courant continu, d'un engre-

nage pour la démultiplication et d'une paire de butées mécaniques pour éviter que l'axe n'aille trop loin. La servo possède en outre un potentiomètre servant à coupler la position de l'axe de sortie à de l'électronique.

Cette électronique constitue le maillon entre le moteur, la

position mesurée et le monde extérieur.

Une triplette de fils fournit à la servo la tension d'alimentation, la ligne de masse et le signal de commande. Ce dernier –fourni par le récepteur radio en modélisme– définit la position à prendre par l'axe de sortie de la servo. Si des éléments extérieurs (la pression

de l'air sur une gouverne de profondeur d'un avion par exemple) risquent de modifier la position de l'axe de sortie, une réaction de l'électronique de commande la maintient en position. Tant que le signal de commande ne change pas, la servo maintiendra la position instantanée de l'axe de sortie (la corrigeant au besoin). Ce

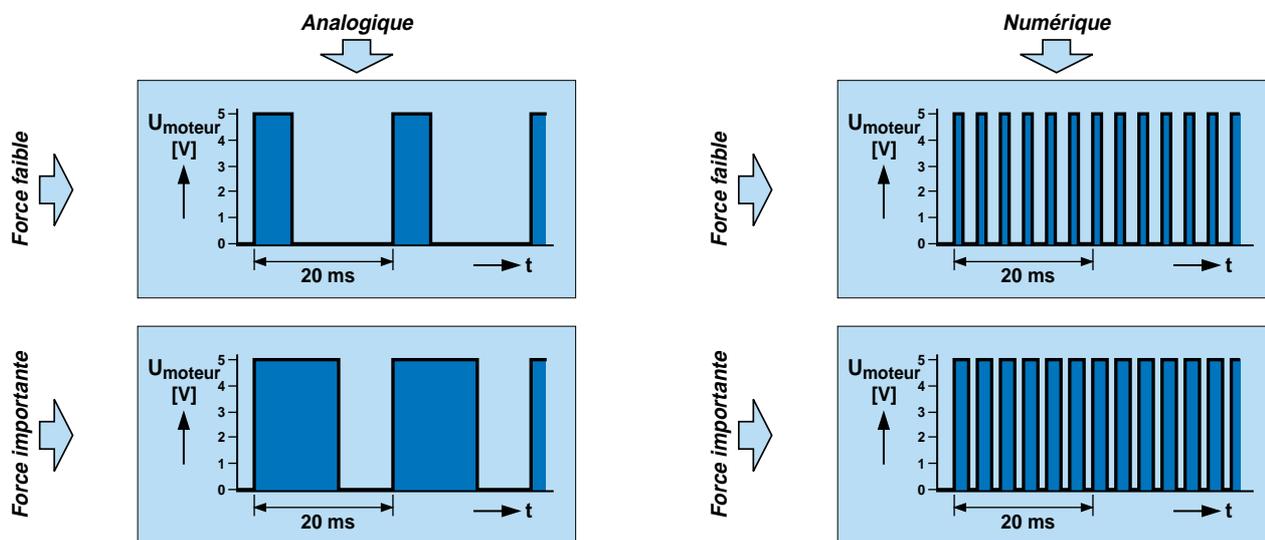


Figure 2. La tension de moteur pour le maintien d'une certaine position (analogique à gauche, numérique à droite).

n'est qu'en cas de changement du signal de commande que l'axe de sortie prendra une autre position.

## Fonctionnement

Un signal de commande prend la forme d'impulsions rectangulaires d'une amplitude de 5 V. La durée de période des impulsions est constante; la largeur des différentes impulsions du train détermine la position. Le neutre (point 0) d'une servo correspond à une largeur d'impulsion de  $\pm 1,5$  ms. En fonction de la rotation souhaitée, les impulsions devront soit s'allonger, soit devenir plus courtes. La durée de période de 20 ms ne change pas elle. L'axe de sortie est couplé à un potentiomètre. La position de son curseur correspond toujours à la position de l'axe de sortie de la servo. Partant, chaque position du potentiomètre correspond à une largeur d'impulsion donnée. Si la servo doit bouger, elle le fera jusqu'à ce que le curseur du potentiomètre ait atteint une

position qui corresponde à la largeur de l'impulsion reçue. L'électronique interne assure le maintien de cette position jusqu'à ce que la largeur d'impulsion du signal de commande change à nouveau.

## Numérique

Au cours des dernières années, les servos ont beaucoup évolué : dimensions, vitesse et puissance ne cessent d'être améliorées. La variante la plus récente, la « servo numérique » constitue un nouveau pas en avant. La servo numérique est en principe une servo standard. La seule différence est qu'un microprocesseur analyse le signal de commande et pilote le moteur. Sa structure est similaire à celle de son homologue « analogique ».

La mise en oeuvre d'un microprocesseur présente d'indéniables avantages. Comme nous le disions, sur une servo analogique, le moteur reçoit quelques impulsions pour contrer une force qui tendrait

à en modifier la position. Si le changement est faible il ne se passera pas grand chose. En effet, à petit changement petites impulsions ! La puissance effective envoyée au moteur est souvent trop faible pour le faire bouger. Ce comportement se traduit par une sorte d'effet d'hystérésis. En deçà d'une limite donnée, la réaction de la servo n'est pas linéaire par rapport aux changements de position induits. Dans le cas de la servo numérique la programmation du microcontrôleur tient compte des caractéristiques du moteur pour le pilotage. La largeur des impulsions envoyées vers le moteur sera ainsi plus large que ne le serait celle observée sur une servo standard. Le programme du processeur pouvant calculer avec précision la quantité et la longueur optimales des impulsions en cas d'action parasite dans une position donnée, une servo numérique est plus rapide et plus précise. Le meilleur comportement obtenu n'est pas uniquement dû à un programme intelligent. La fréquence de commutation des impulsions de commande

du moteur a été sensiblement accrue elle aussi, avec comme conséquence, une meilleure précision mais aussi la possibilité d'appliquer une « force » plus importante, le moteur se trouvant alimenté plus souvent. Un surplus de puissance tombe à pic non seulement pour maintenir une position mais aussi pour ne changer. L'accélération est plus rapide, une nouvelle position étant atteinte plus vite.

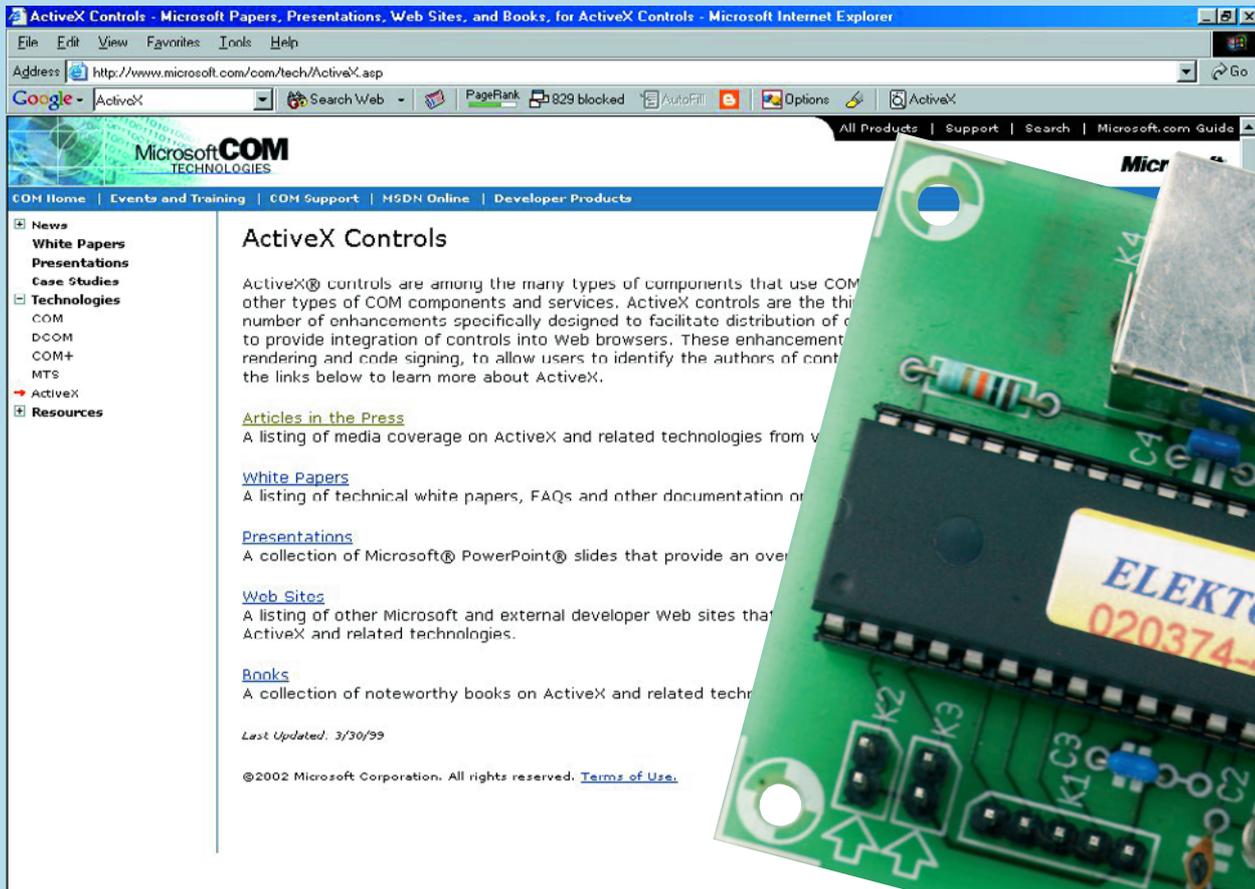
## Des inconvénients ?

Une activation plus fréquente du moteur se traduit bien entendu par une augmentation de la consommation d'énergie. C'est dans les applications de modélisme qu'il faudra en tenir compte. Les progrès de la technologie des accumulateurs font qu'il ne s'agit plus là d'un problème majeur.

(040041-1)

# ALLEZ ! ACTIVONS !

## ActiveX pour la centrale USB



La centrale d'acquisition USB décrite dans le numéro de novembre 2003 peut se targuer d'avoir été l'un des montages publiés dans Elektor ces dernières années ayant généré le plus de courrier et de Courriels (E-mail).

Le programme accompagnant ce montage connaissait des limites qu'il aurait été bon de faire sauter. L'auteur, sollicité de toutes parts, en était conscient, ce qui l'a amené à nous proposer un outil permettant à tout utilisateur de la centrale d'acquisition USB de créer son propre programme pour contrôler un périphérique de son choix.

Le module de contrôle *ActiveX* présenté ici permet de récupérer les grandeurs analogiques et les sorties numériques.

Ce type de contrôle permet de programmer l'interface à partir de n'importe quel langage de programmation évolué tel que Delphi, Visual-Basic ou C++ de Borland.

Comme Delphi est l'un des langages de programmation les plus utilisés, nous allons voir comment procéder en utilisant ce langage comme référence.

## Contrôle ActiveX, connais pas !

Un contrôle ActiveX est un composant logiciel qui s'incorpore à une application-hôte gérant les contrôles ActiveX, comme C++Builder, Delphi, Visual dBASE, Visual Basic, Internet Explorer ou Netscape Navigator et en étend les fonctionnalités.

Ainsi, Delphi est fourni avec plusieurs contrôles ActiveX : graphes, tableurs ou graphiques. Vous pouvez ajouter ces contrôles à la palette des composants de l'EDI (Environnement de Développement Intégré) et les utiliser comme tout composant VCL (*Visual Component Library*) standard en les déposant dans des fichiers et en définissant leurs propriétés avec l'inspecteur d'objets.

Vous pouvez également déployer sur le Web un contrôle ActiveX, en le référencant dans des documents HTML et en le visualisant dans un navigateur Web gérant ActiveX.

Ce qui suit est une présentation générale de la façon de créer un contrôle ActiveX dans l'environnement Delphi. Vous n'y trouverez pas les détails complets de l'écriture des contrôles ActiveX. Pour cela, consultez la documentation *Microsoft Developer's Network* (MSDN). Le site web de Microsoft sur ActiveX donne également des informations sur ce sujet.

Vous pouvez également déployer sur le Web un contrôle ActiveX, en le référencant dans des documents HTML et en le visualisant dans un navigateur Web gérant ActiveX.

## Le contrôle ActiveX picus\_brxProj1.ocx :

Pour le réaliser l'auteur s'est inspiré du programme BASIC précédent, la compilation ayant été faite sous Delphi 4. Il est évident qu'il n'est pas question dans le cadre restreint de cet article, de nous lancer dans un cours sur la création d'un contrôle ActiveX. Delphi et sa fonction Aide (*Help*) vous en apprendront bien plus.

7 autres fichiers sont nécessaires pour l'installation complète de cet ActiveX.

Tous les fichiers nécessaires sont inclus dans le répertoire `fich_ox` (cf. recopie d'écran de la **figure 1**).

## Installation

On va utiliser une étape de base, en effet cette approche est préférable à une installation automatique qui ne nous apprend rien.

- 1) Installer le fichier `picus_brxProj1.ocx` dans le répertoire `c:\windows\system` pour Windows 98 et Me, ou dans `c:\windows\system32` dans Windows 2000 et XP, ou `c:\WINNT\system32` pour Windows 2000 pro. Cela permettra de recenser le contrôle ActiveX grâce à `regsvr32.exe` du système d'exploitation.
- 2) Tous les autres fichiers du répertoire `fich_ox` dans le répertoire `c:\Program Files\Borland\Delphi4\Imports` pour Delphi 4.

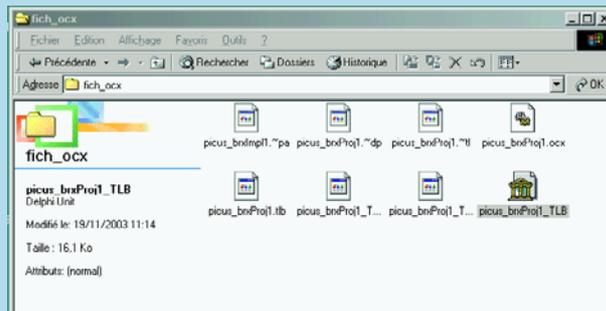


Figure 1. Le répertoire `fich_ox` comprend tous les fichiers du projet.

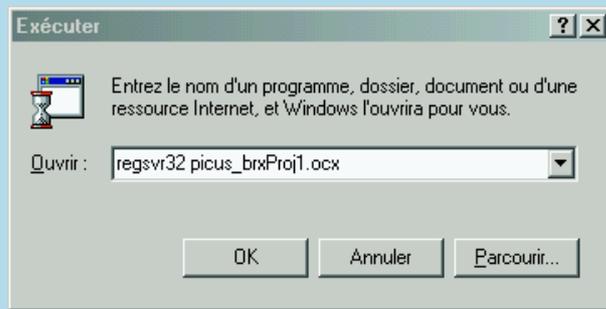


Figure 2. Le module d'exécution que tout le monde a déjà utilisé une fois dans sa vie.



Figure 3. Un message qui fait plaisir.

- 3) En cliquant sur *Démarrer* puis *Exécuter* `regsvr32 picus_brxProj1.ocx` (cf. recopie d'écran de la **figure 2**). Si cela réussit on a le message de succès ci contre (**figure 3**).
- 4) Sous Delphi 4, choisir, sous *Composant*, l'option de menu *Importer un contrôle ActiveX*, à l'aide de l'ascenseur sélectionner `picus_brxProj1.ocx`, puis installer, sauvegarder (**figure 4**).  
Si tout se passe bien sous le menu ActiveX on doit voir apparaître le contrôle ActiveX.

Vous voilà prêt à exploiter votre centrale à plein.

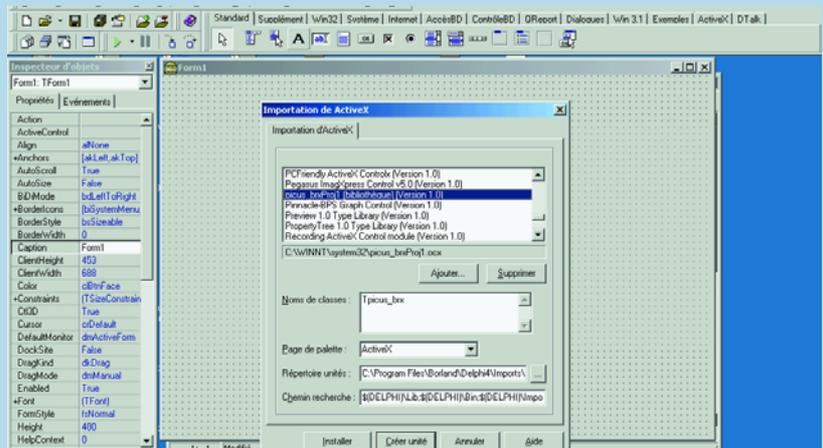


Figure 4. Les différents modules ont vite fait d'encombrer l'écran.

Figure 5. Schéma d'un petit circuit de capture de température à base de LM(3)35, un fameux capteur de l'écurie National Semiconductor.

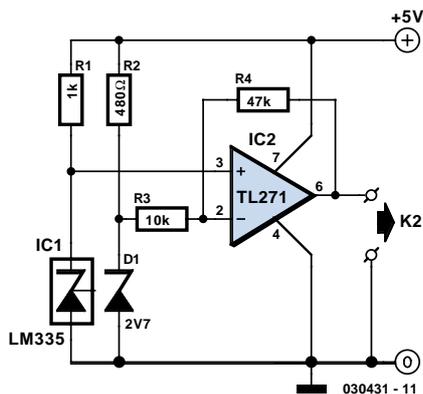


Figure 6. Le bouton USB Centrale a été placé dans la Form1.

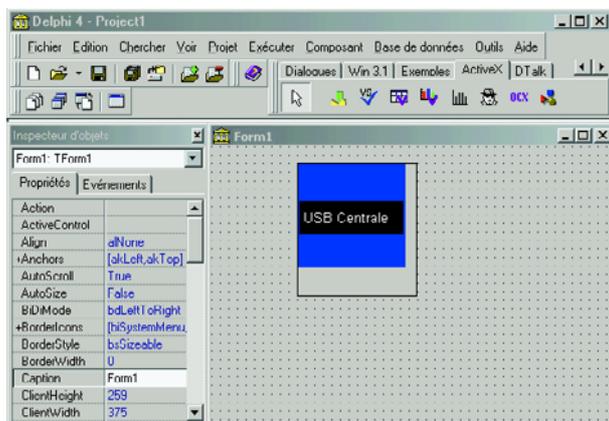


Figure 7. Les boutons TEMP et un timer ont été mis en place dans Form1.

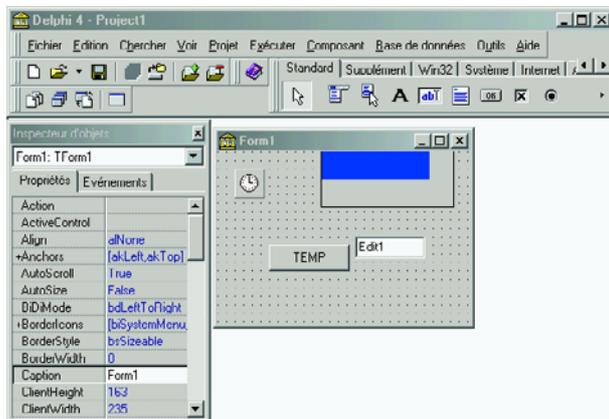
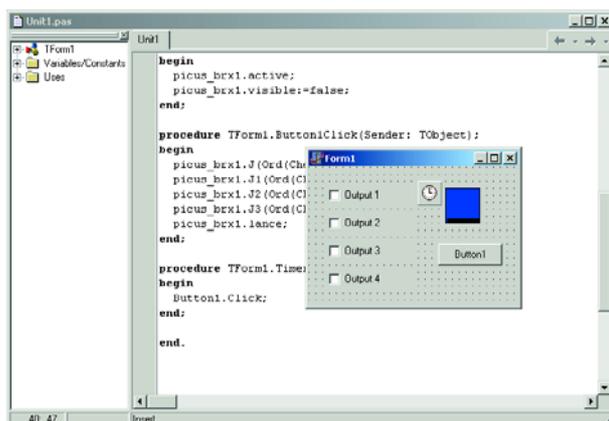


Figure 8. La fonction de l'ocx permet de contrôler la centrale.



## Mise en oeuvre de notre contrôle ActiveX à l'aide d'un exemple

Rien de tel qu'un exemple pour saisir le fonctionnement. Nous allons faire tout simple : tenter l'acquisition de température en s'aidant d'un classique capteur de température LM335 de Dallas Semiconductor. Celui ci est donc tout simple, puisque le signal proportionnel à la température est amplifié par un circuit non inverseur faisant appel à un amplificateur opérationnel du type TL271. Le schéma de l'électronique est représenté en figure 5. La tension d'alimentation positive nécessaire,  $V_{DD}$ , est fournie par une sortie numérique de la centrale USB, le point K2 étant connecté à la première entrée analogique.

## Le programme

1) Ouvrir Delphi puis après avoir lancé une nouvelle application aller dans le menu ActiveX,

Cliquer sur le composant picus\_brx, et le placer sur la fiche form1 (figure 6)..

2) Placer un bouton (à récupérer dans le menu Standards des composants Delphi), puis un edit.

Ensuite placer un timer (à récupérer dans le menu Standards des composants Delphi) cf. les recopies d'écran (figure 7).

3) Cliquer deux fois sur bouton et ajouter le code suivant :

```

procedure TForm1.Button1Click(Sender: TObject);
var Val_T:real;
begin
  Val_T:=(picus_brx1.AN0*5/255); {conversion tension température}
  edit1.text:=floattostr(val_T);
end;

```

et cliquer deux fois sur timer :

```

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  TEMP.Click;
end;

```

lancer l'application et vous avez la tension variant en fonction de la température de votre bureau.

## Utilisation des sorties

Une fonction de l'ocx permet de contrôler les sorties de la centrale, nous allons l'examiner à l'aide d'un exemple.

Mettre en forme la fiche Delphi en y ajoutant les différents composants suivants :

- 4 composants edit à récupérer dans la palette standard
- 1 timer à récupérer dans la palette système
- 1 bouton à récupérer dans la palette standard

et bien sur notre fameux ocx picus\_brx1 dans la palette active x (cf. recopie de la figure 8).

double cliquer sur la fiche et ajouter le code suivant :

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  picus_brx1.active;
  picus_brx1.visible:=false;
end;

```

double cliquer sur le composant button et ajouter le code suivant :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
picus_brx1.J(Ord(Checkbox1.Checked));
picus_brx1.J1(Ord(Checkbox2.Checked));
picus_brx1.J2(Ord(Checkbox3.Checked));
picus_brx1.J3(Ord(Checkbox4.Checked));
picus_brx1.lance;
end;
```

double cliquer sur le composant Timer et ajouter le code suivant :

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
button1.Click;
end;
```

mettre des 0 ou des 1 dans les edit et regarder l'action sur les sorties.

## Conclusion

Mais là ne s'arrêtent pas les applications possibles de la combinaison Centrale USB + ActiveX. Nous laissons à votre féconde imagination la tâche d'en trouver d'autres. N'hésitez pas à nous en faire part, l'interaction lectorat/rédaction est importante si un magazine veut rester tel que le souhaitent ses lecteurs.

(030431-1)

## Liens Internet :

### ActiveX Controls

[www.microsoft.com/com/tech/ActiveX.asp](http://www.microsoft.com/com/tech/ActiveX.asp)

### JEDI Visual Component Library

<http://homepages.borland.com/jedi/jvcl/>

### VCLComponents.com

[www.vclcomponents.com/](http://www.vclcomponents.com/)

### Site de l'auteur (en construction) :

<http://dorelec.web.oo.fr>

# VB SOUS WORD

## Créer ses programmes VISUAL BASIC sous WORD

Vous avez bien lu : un PC équipé de « Word » suffit pour créer un programme en langage Visual Basic, et s'initier à la programmation. C'est à la portée de tous mais beaucoup de gens l'ignorent.

Cet article est destiné à permettre à ceux qui n'ont pas Visual Basic, et qui souhaitent s'initier à la programmation sous Windows, de débiter facilement, tout en se faisant plaisir. L'astuce consiste à utiliser « Visual Basic pour Applications », présent aussi, notamment, sous « Excel ».

### Etape 1

Ouvrir Word. Toutes les images et les programmes présentés ici ont été créés sous Word 2000, mais les autres versions fonctionnent aussi, même Word 97 !

### Etape 2

Cliquer sur le menu « Affichage », puis sur « Barre d'outils ». Cocher « Visual Basic » (figure 1).

### Etape 3

La barre d'outils « Visual Basic » apparaît (figures 2 et 3).

### Etape 4

Vous créez ainsi un bouton, appelé « CommandButton », qui apparaît sous la forme ci-contre (figure 4). Pour voir ses caractéristiques, déplacer la souris au dessus du bouton, cliquer sur le bouton droit, et appuyer sur « Propriétés ». Une nouvelle fenêtre apparaît, dans laquelle sont décrites toutes les propriétés de l'objet. Les deux plus importantes propriétés sont « name » et « caption » (figure 5).

Donner un NOM au bouton. Dans notre exemple, ce sera « BoutonTestMessage ».

Puis remplacer le texte affiché sur le bouton par « Veuillez cliquer sur ce bouton! ». On peut aussi modifier la taille, la police, la couleur du fond, placer une image en arrière plan,...

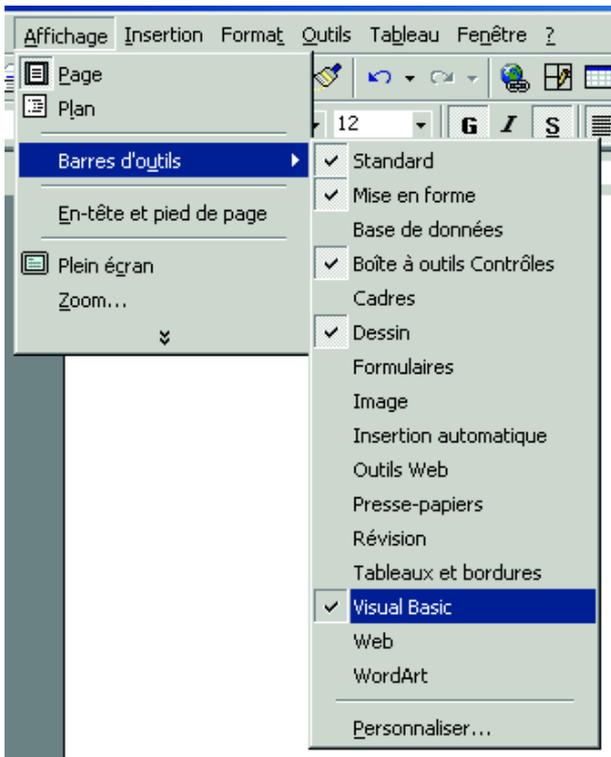
*Remarque sur les fenêtres :* Sous Visual Basic, les fenêtres sont nombreuses et rangées d'une façon très simple : rien n'est rangé ! Cette charge est laissée à l'utilisateur. À vous de ne pas vous laisser embrouiller par toutes les fenêtres.

### Etape 5

Effectuer un double clic sur le bouton. L'éditeur Visual Basic apparaît. La procédure « BoutonTestMessage\_Click » est créée si elle n'existait pas et le curseur est placé ICI, à l'endroit où vous allez écrire votre programme ! (figure 6)

Cette fonction est exécutée chaque fois que l'on « Click » sur le bouton « BoutonTestMessage ».

En termes plus évolués, l'événement « Click » est associé à l'objet « BoutonTestMessage », de type « CommandButton », (on dit aussi appartenant à la classe « CommandButton »). Le code correspondant est situé dans cette fonction « privée », c'est-à-dire propre à l'objet.



1



Cliquer sur la boîte à outils « Contrôles »

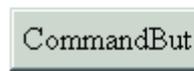
Activer le mode « Création »

2

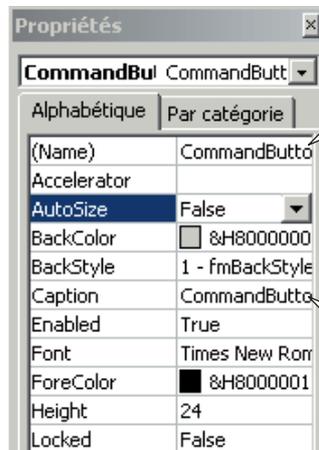


Cliquer sur l'icône « Bouton de commande »

3



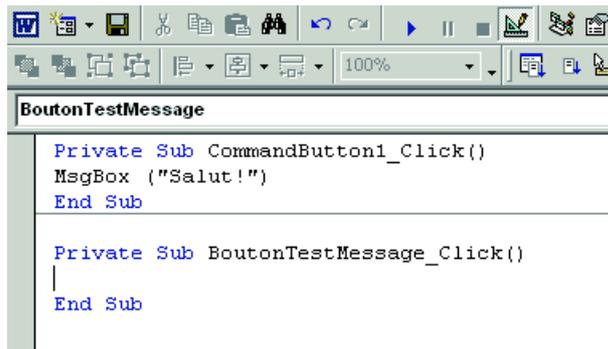
4



Le NOM ou « name » du bouton. Choisir un nom COHERENT et SIGNIFICATIF, qui indique à la fois le type d'objet et son rôle. Par exemple : « BoutonAllumer ».

L'inscription figurant sur le bouton ou « caption ». Attention à ne pas confondre avec le nom !

5



6



7

## Etape 6

Entrer au clavier la ligne : `MsgBox ("Vous avez cliqué ici !")`

## Etape 7

Retourner dans « Word », par exemple en cliquant sur « ALT TAB », et passer en mode « Exécution », en désélectionnant les deux icônes.

Vous pouvez aussi enlever la barre d'outils « Visual Basic », si vous souhaitez un document « propre ». Mais il vous faudra la faire réapparaître, pour modifier le programme (figure 7).

## Etape 8

Enregistrer au format « Word », et tester le fonctionnement.

## Conclusion

Il suffit de quelques secondes pour créer un programme sous « Word ». Vous avez maintenant une idée de ce qu'il est possible de faire. On de même aussi programmer en « Visual Basic » depuis « Excel » : créer des fenêtres complètes avec des boutons, des zones de saisie, des images et enregistrer des données, les afficher sous forme de graphiques, personnaliser le menu de « Word » avec des commandes « perso », lancer d'autres programmes... Il est aussi possible, mais ceci mériterait d'autres articles, d'utiliser les API de Windows, pour programmer le port parallèle, par exemple. Des applications complexes ont été écrites ainsi, en cachant le menu par défaut. L'utilisateur a alors l'impression d'être dans un programme spécial alors qu'il travaille en réalité sous « Excel » ou « Word » !

# Mon programme ne fonctionne pas !

Trois solutions possibles et quelques conseils...

### 1 ) Niveau de sécurité :

Si un bouton par exemple ne fonctionne pas, vérifiez que vous êtes au niveau de sécurité « moyen ». Ce niveau vous laisse le choix, lorsque des macros ou du code « Visual Basic » sont trouvés dans un programme, de les exécuter ou non. « Word » vous propose systématiquement la question de l'ouverture du fichier. En niveau « haut », ce choix ne vous est pas laissé. Il n'est pas possible d'exécuter des programmes écrits en « Visual Basic ».

Modifiez votre niveau de sécurité sous Word :

- cliquer dans le menu « outils » puis sur « macros » et sur « sécurité ».
- Choisir le niveau de sécurité « moyen ».
- Fermer le fichier, sortir de « Word ».
- Ouvrir à nouveau le document.

Lors de l'ouverture, « Word » vous demande si vous souhaitez activer les macros. Dans le cas présent, cliquer sur « oui ».

### 2 ) Problème d'exécution du programme :

Votre programme peut aussi contenir une erreur, qu'il faut localiser.

Le « débogueur », ou « indicateur d'erreur » est un outil qui arrête automatiquement votre programme et pointe sur l'endroit où elle se situe ! La recopie d'écran ci dessous montre la fenêtre qui s'affiche dans ce cas.

L'éditeur « VisualBasic » s'ouvre alors automatiquement.

### 3 ) Si cela ne marche toujours pas :

Il se peut que le problème ne vienne pas de vous. Essayez alors cette astuce : Enregistrez votre programme et fermez « Word ». Ouvrez alors à nouveau votre document. Cela suffit souvent à faire fonctionner miraculeusement les programmes...

Mais si rien ne marche, il vous reste la Toile, et les forums de discussion.

### Liens Internet

Pour en savoir plus sur Visual Basic

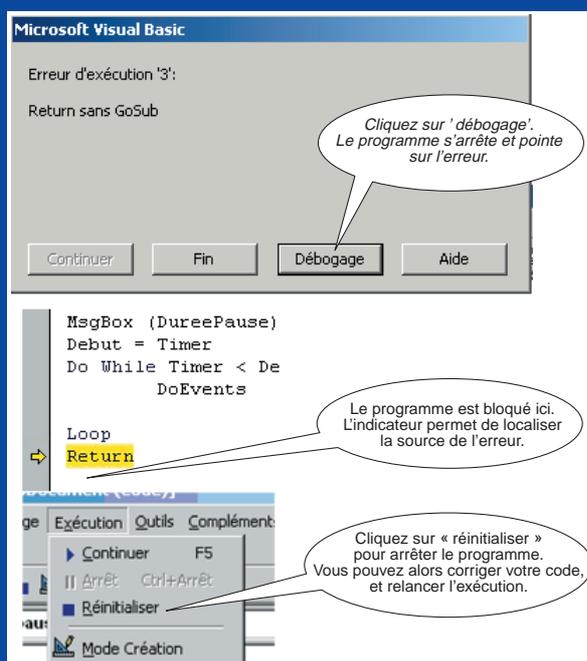
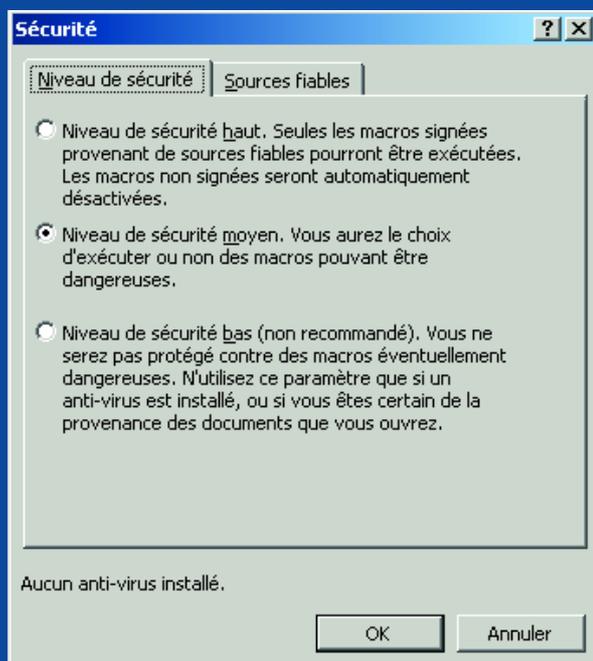
<http://msdn.microsoft.com/vbasic/>

Visual Studio

<http://msdn.microsoft.com/vstudio/>

Source de codes en VB

<http://www.vbcode.com/>





**ICS502**

**Multiplicateur d'horloge**

**Fabricant :**

Integrated Circuit Systems

<http://www.icst.com/pdf/ics502.pdf>

**Caractéristiques :**

- Boîtier SOIC à 8 broches ou puce (die)
- Disponible en boîtier sans plomb
- Horloge PLL bon marché
- Pas d'erreur de multiplication
- Facile à cascader avec la série ICS5xx
- Fréquence de quartz d'entrée 5 à 27 MHz
- Fréquence d'horloge d'entrée 2 à 50 MHz
- Fréquence d'horloge de sortie jusqu'à 160 MHz
- Gigue (*jitter*) faible
- Compatible avec toutes les CPU courantes
- Rapport cyclique de 45/55 jusqu'à 160 MHz
- Tension de service de 3,0 à 5,5 V
- Courant de sortie de 25 mA à niveau TTL

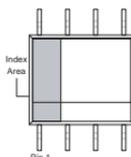
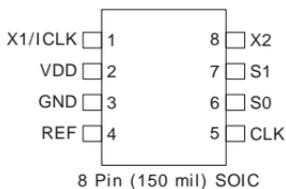
**Description :**

Le ICS502 constitue une option bon marché pour la génération d'un signal d'horloge haute fréquence de haute qualité. On pourra utiliser comme élément de base un quartz de fréquence faible ou un signal d'horloge basse-fréquence. Cette approche permet de se passer, dans la plupart des systèmes, d'oscillateurs à quartz « rapides » chers. Le ICS502 fait appel à un circuit de PLL pour rehausser la fréquence d'un quartz standard bon marché jusqu'à 160 MHz.

La ROM interne mémorise 6 facteurs différents pouvant servir de multiplicateurs pour la fréquence d'entrée. Il est possible ainsi de choisir nombre de fréquences d'horloge courantes. La sélection des facteurs se fait par le biais d'une paire d'entrées logiques. Ayant été développé pour la génération d'horloge, il est logique que le ICS502 soit caractérisé par une gigue (*jitter*) faible en sortie, il n'a cependant pas été prévu de définition de l'asymétrie du signal entre l'entrée et la sortie (*input to output skew*). Si l'application concernée requiert une valeur garantie de ce facteur il est recommandé de lui préférer le ICS570B.

**Formes de boîtier et brochage :**

Broche	Dénomination	Fonction
1	X1/ICLK	Connexion du quartz ou entrée d'horloge
2	VDD	Tension d'alimentation +3,3 ou +5 V
3	GND	Masse
4	REF	Sortie tamponnée de l'oscillateur à quartz
5	CLK	Sortie de l'horloge multipliée
6	S0	Entrée de sélection du rapport d'horloge (VDD ou GND)
7	S1	Entrée de sélection du rapport d'horloge (VDD, GND ou Ouvert)
8	X2	Connexion du quartz, ouvert si entrée d'horloge



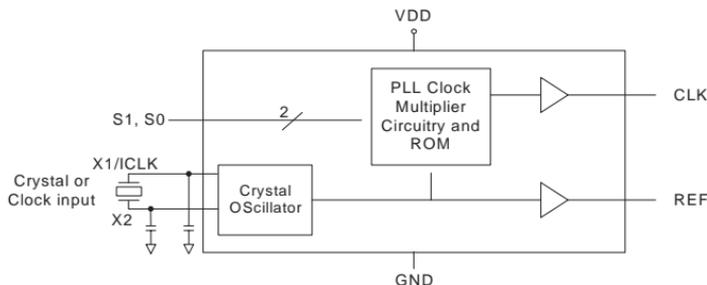


**Décodage de l'horloge :**

SI	S0	CLK
0	0	× 2
0	1	× 5
ouvert	0	× 3
ouvert	1	× 3,33
1	0	× 4
1	1	× 2,5

**Valeurs maximales :**

Paramètre	Valeur
Tension d'alimentation $V_{CC}$	7 V
Tension sur toutes Entrées/Sorties	-0,5...( $V_{DD}+0,3$ V)
Plage des températures de service	-40...+85 °C



**Application typique :**

Attila, bloqueur de pub, Elektor juin 2004

**Composants externes :**

*Condensateur de découplage*

Comme c'est le cas avec tout circuit intégré « mixed-signal » puissant, le ICS502 se doit lui aussi d'être parfaitement découplé de la tension d'alimentation de manière à réduire au maximum le bruit et permettre un fonctionnement optimal du circuit intégré. Il faudra prendre un condensateur de découplage de 100 nF entre les broches  $V_{DD}$  et GND et ceci le plus près possible des pattes de manière à limiter au mieux les influences inductives parasites des pistes. Le ICS502 ne requiert pas de filtrage externe de la tension d'alimentation.

*Résistance de terminaison*

La ligne CLK peut être dotée d'une résistance de terminaison de 33  $\Omega$ . La capacité totale représentée

par la puce est de 12 pF. Il faudra, s'il oscille à sa fondamentale, connecter un quartz en résonance parallèle.

*Condensateurs de charge du quartz :*

Les condensateurs de faible valeur pris respectivement entre les connexions X1 et X2 et la masse ont pour fonction d'assurer une charge correcte du quartz et une réduction au minimum de l'influence des capacités parasites de la platine. Comme, en principe, ceci ne peut être obtenu que par une augmentation de la charge, il est important, dès le départ, de minimiser au maximum les capacités parasites (pistes courtes, pas de métallisation entre le quartz et le circuit intégré). La valeur (exprimée en pF) se calcule par la formule  $(C_L - 12 \text{ pF}) * 2$ , formule dans laquelle  $C_L$  représente la capacité de charge du condensateur.

## Caractéristiques électriques :

$V_{DD} = 5,0 \text{ V} \pm 5\%$ , Température ambiante de 0 à 70 °C, sauf mention contraire

Paramètre	Symbole	Conditions	Min.	Typ.	Max.	Unité
<b>En continu</b>						
Tension d'alimentation	$V_{DD}$		3		5,5	V
Tension d'entrée Haut	$V_{IH}$	$I_{CLK}$ , broche 1	$(V_{DD}/2) + 1$			V
Tension d'entrée Bas	$V_{IL}$	$I_{CLK}$ , broche 1			$(V_{DD}/2) - 1$	V
Tension d'entrée Haut	$V_{IH}$	OE, broche 7	2			V
Tension d'entrée Bas	$V_{IL}$	OE, broche 7			0,8	V
Tension d'entrée Haut	$V_{IH}$	S0,S1	$V_{DD}-0,5$			V
Tension d'entrée Milieu	$V_{IM}$	SI		$V_{DD}/2$		V
Tension d'entrée Bas	$V_{IL}$	S0,S1			0,5	V
Tension de sortie Haut	$V_{OH}$	$I_{OH} = -25 \text{ mA}$	2,4			V
Tension de sortie Bas	$V_{OL}$	$I_{OL} = 25 \text{ mA}$			0,4	V
Consommation de courant	IDD	quartz 20 MHz, facteur 5, hors-charge		20		mA
Courant de court-circuit		sortie CLK		$\pm 70$		mA
Résistance pull-up intégrée		broche 7		270		k $\Omega$
Capacité d'entrée		SI, S0, OE		4		pF

## Caractéristiques électriques :

$V_{DD} = 5,0 \text{ V} \pm 5\%$ , Température ambiante de 0 à 70 °C, sauf mention contraire

Paramètre	Symbole	Conditions	Min.	Typ.	Max.	Unité
<b>En alternatif</b>						
Fréquence d'entrée	$F_{IN}$	Entrée quartz	5		27	MHz
Fréquence d'entrée	$F_{IN}$	Entrée horloge	2		50	MHz
Fréquence de sortie	$F_{OUT}$	$V_{DD} = 4,5 \text{ à } 5,5 \text{ V}$ , 0 à 70 °C	14		160	MHz
Fréquence de sortie	$F_{OUT}$	$V_{DD} = 3,0 \text{ à } 3,6 \text{ V}$ , -40 à 70 °C	14		160	MHz
Temps montée horloge sortie	$t_{OR}$	0,8 à 2,0 V		l		ns
Temps chute horloge sortie	$t_{OR}$	2,0 à 8,0 V		l		ns
Rapport cyclique horloge sortie	$t_{OD}$	1,5 V, jusqu'à 160 MHz	45	49...51	55	%
Bande passante PLL			10			kHz
Durée validation sortie		OE Haut jusqu'à activation sortie		50		ns
Durée désactivation sortie		OE Bas jusqu'à désactivation sortie		50		ns
Gigue absolue période d'horloge	$t_{ja}$			±70		ps
Gigue I Sigma période d'horloge	$t_{js}$			25		ps

# Orgue lumineux de poche

Minuscule beauté

Burkhard Kainka

Faire un tour  
en disco peut  
avoir son attrait,  
mais l'être humain  
n'est pas fait pour se  
trouver confronté

journallement à une telle  
violence acoustique. Un mini-orgue  
lumineux permet d'ajouter une teinte disco lorsque  
l'on écoute de la musique chez soi. *LED it be...*

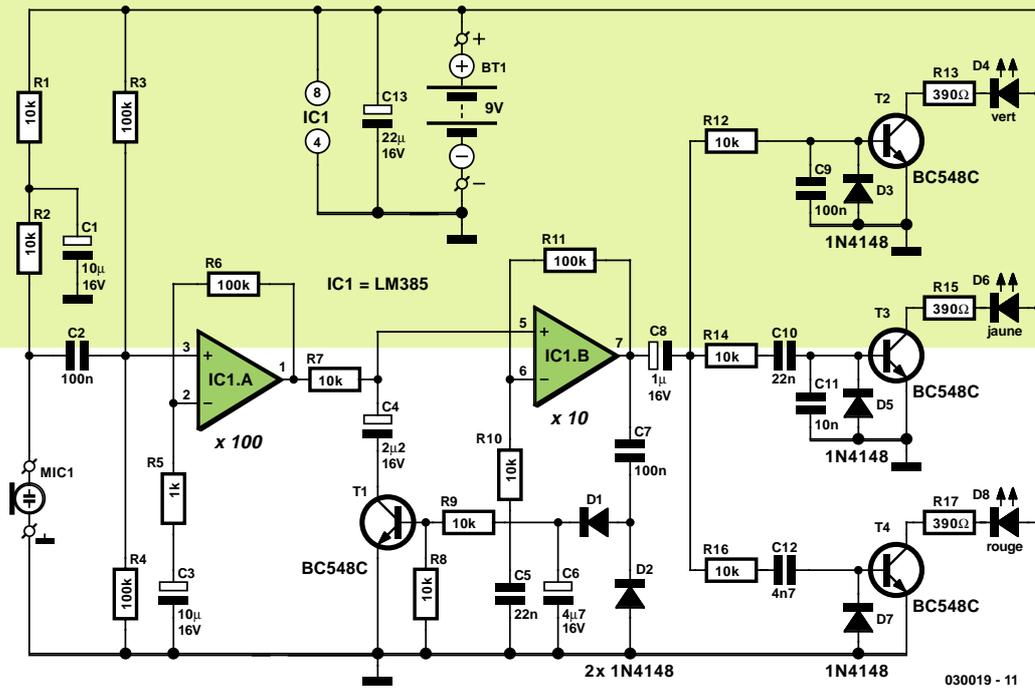


Figure 1. L'électronique de l'orgue lumineuse constitué d'un amplificateur piloté doté en aval d'un filtre 3 voies.

Un orgue lumineuse transforme des sons en impressions lumineuses. On utilise, le plus souvent, 3 canaux pour différentes plages de fréquence. Le canal des graves reproduit le rythme de la musique par un clignotement plus ou moins synchrone, les 2 autres canaux rendant les plages de fréquence plus élevées.

Avec ses 3 LED, la version de poche d'orgue lumineuse présentée ici brille moins violemment que son homologue en grandeur nature, mais son fonctionnement est identique. Il n'est cependant pas besoin de prévoir une liaison sonore étant fournie par un micro. Autre spécificité : la sensibilité s'adapte au niveau sonore du moment. Tout ceci permet de se passer de tout organe de commande. L'alimentation de l'orgue fait appel à une pile compacte 9 V. La consommation de courant moyenne n'est que de quelque 20 mA. Il est recommandé, pour en avoir pour son argent quant à la durée de vie de la pile, d'opter pour des LED de très forte luminosité.

## Amplifier

Il est nécessaire, pour que le signal du micro à électret ait un quelconque effet, de l'amplifier très fortement. Sur le schéma de la **figure 1**, c'est la paire

d'amplificateurs qui s'en charge; elle introduit un gain de l'ordre de 1 000x. L'entrée du premier amplificateur opérationnel, IC1.A, est reliée au micro à capsule électret par le biais d'un condensateur, C2. La capsule se trouve au 9 V par le biais des résistances R1 et R2. Épaulée par C1, R1 découple la tension d'alimentation du micro sensible, R2 constituant la résistance de travail pour la capsule électret, aux bornes de laquelle la tension de signal du micro apparaît sous la forme de petites variations de la tension (continue). Le condensateur C2 fait en sorte que seule cette ondulation (la composante alternative de la tension) arrive à l'entrée de l'ampli op. Le diviseur de tension constitué par R3 et R4 fournit la tension continue à l'entrée de l'ampli op. Ces résistances ayant la même valeur, la tension continue à l'entrée de IC1.A est égale à la moitié de la tension d'alimentation soit, avec une pile de 9 V, 4,5 V. Cette même tension continue se trouve aussi à la sortie du premier ampli op et, à l'entrée du second, IC1.B, par le biais de R7, la sortie de IC1.B prenant la même valeur. En d'autres termes, les 2 amplis op sont forcés, par le biais de R3 et R4, en continu, à la moitié de la tension d'alimentation. Les 2 amplis op ne sont pas inverseurs et ont un gain unitaire (1x) pour les tensions continues.

Pour les tensions alternatives, et partant le signal du micro, le gain est beaucoup plus élevé. Au niveau du premier ampli op, il est déterminé par le rapport des résistances R6/R5, par celui de R11/R10 pour le second ampli op. Cela nous donne, pour IC1.A, un gain de 100, le gain introduit par IC1.B étant de 10x.

On a, sur le second ampli op, un réglage de gain automatique qui entre en fonction aux niveaux sonores plus important et évite toute surmodulation. La tension de signal à la sortie de IC1.B, redressée par les diodes D1 et D2, lissée par C6, attaque le transistor NPN T1 qui, en combinaison avec R7, constitue un diviseur de tension pour la tension de signal en aval du premier ampli op. Si la tension de signal augmente, la tension redressée aux bornes de C6 augmente, le courant de base du transistor, arrivant via R9, augmente, celui-ci voit alors son impédance diminuer, réduisant ainsi la tension alternative BF en aval de R7.

On pourra s'étonner de l'utilisation ici d'un transistor NPN sans courant continu de collecteur. On aurait pu s'attendre à trouver à cet endroit un transistor à effet de champ (FET) faisant office de résistance pilotée, approche courante. Ce que l'on sait moins est qu'il est également possible d'utiliser un transistor ordinaire bon marché, un

# Liste des composants

## Résistances :

R1,R2,R7 à R10,R12,R14,R16 = 10 kΩ  
R3,R4,R6,R11 = 100 kΩ  
R5 = 1 kΩ  
R13,R15,R17 = 390 Ω

## Condensateurs :

C1,C3 = 10 μF/16 V radial  
C2,C7,C9 = 100 nF  
C4 = 2 μF/16 V radial  
C5,C10 = 22 nF  
C6 = 4 μF/16 V radial

C8 = 1 μF/16 V radial  
C11 = 10 nF  
C12 = 4 nF  
C13 = 22 μF/16 V radial

## Semi-conducteurs :

D1 à D3,D5,D7 = 1N4148  
D4 = LED verte (cf. texte)  
D6 = LED jaune (cf. texte)  
D8 = LED rouge (cf. texte)  
IC1 = LM385N (sur support)

T1 à T4 = BC548C

## Divers :

BT1 = pile compacte 9 V avec connecteur à pression  
MIC1 = capsule micro à électret, telle que, par exemple, 302147 - 8A (Conrad)  
Platine EPS 030019-1 (dessin des pistes et au téléchargement sous [www.elektor.fr](http://www.elektor.fr))

BC548C ici). En fait, un transistor NPN constitue lui aussi une résistance variable se laissant piloter sur une large plage. Il faut cependant, si l'on veut limiter la distorsion du réglage de volume, que la tension de signal appliquée au collecteur ne dépasse pas quelques millivolts. Ce n'est pas le cas ici, le réglage de la tension de sortie se faisant à de l'ordre de 1 VCC. Avec un gain de 10 du second étage, on a de l'ordre de 100 mV<sub>CC</sub> sur le collecteur. Ceci se traduit par des distorsions inacceptables pour d'autres applications. Cela ne pose pas de problème dans le cas de cet orgue lumineux, le signal de sortie n'étant pas à écouter mais à visualiser sous la forme de lumière. Il faudra, si l'on veut utiliser ce circuit de réglage à des fins audio, n'attaquer le transistor qu'à des tensions de signal faible et le placer plus en amont de la chaîne d'amplification.

## Filtres

Le signal de sortie du second ampli op attaque directement, mais par le biais de filtres rustiques, les étages de commande de LED de l'orgue lumineux, dotés chacun d'un transistor NPN, T2 à T4. Pour éviter que la base des transistors ne puisse se charger négativement, chacun d'entre eux est doté d'une diode, D3, D5 et D7, montée en tête-bêche (anti-parallèle). De ce fait, l'étage n'est passant que lors des demi-ondes positives. En dépit de cela, on a quand même, en raison des fréquences relativement élevées, une impression de luminosité moyenne. Les filtres sont des réseaux RC classiques se laissant dimensionner grossièrement pour leur fréquence de coupure ( $f_c = 1/(2 \pi R C)$ ). Avec ses valeurs de 10 kΩ et 100 nF, le filtre passe-bas R12/C9 possède une fréquence de coupure de 160 Hz. Les sons de fréquence inférieure à cette valeur pilotent le canal de la LED verte. Le canal central

comporte un filtre passe-haut/passe-bas combiné (R14/C10 et C11). Le dernier canal est piloté par un filtre passe-haut, R16/C12. On pourra, par modification prudente des valeurs de condensateurs, adapter à son goût le comportement du montage.

## La platine

Bien que ce montage soit simple, la platine représenté en **figure 2** en simplifie la réalisation. Les 2 amplis op se trouvent dans le même circuit intégré, un LM385, que l'on placera de préférence sur support. Attention à la polarité. De par l'absence de pont de câblage, il suffit de se concentrer sur la polarité des condensateurs, des diodes et des LED. On pourra utiliser n'importe quel type de capsule micro à

électret à 2 broches du commerce.

Le courant de LED maximal atteint, à modulation maximale des transistors, quelque 18 mA, de sorte que l'on pourra utiliser des LED standard, haute luminosité ou très forte luminosité. Si l'on utilise ce dernier type de LED, l'effet d'orgue lumineux sera plus marqué; pour une utilisation « en chambre » on pourra se contenter de LED standard. Si l'on veut utiliser des LED faible courant, il faudra augmenter la valeur des résistances de limitation R13, R17 et R17 à 3kΩ3 voir 3kΩ9. Ceci permettra, comme le courant tombe à 2 mA par LED, à l'orgue de rester fonctionnel sensiblement plus longtemps sur la même pile.

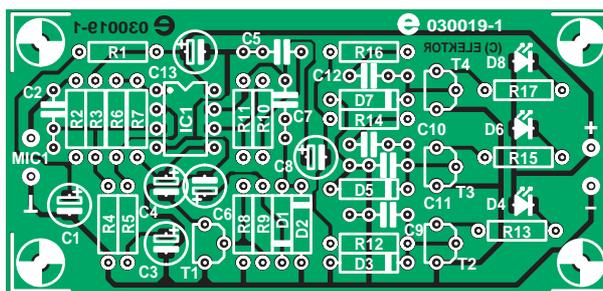
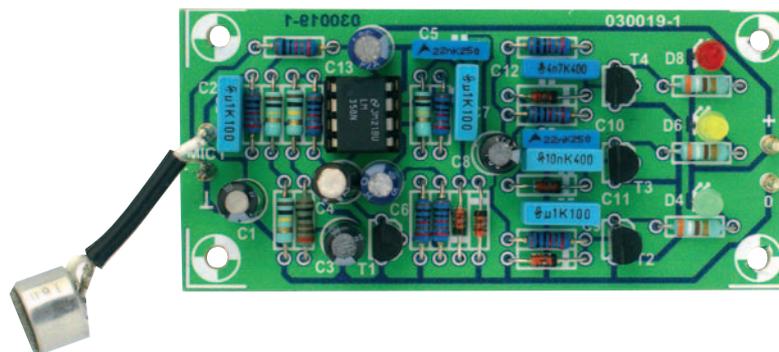


Figure 2. Une platine simple face sans pont de câblage.



(030019-1)

### Gradateur ou pas gradateur, that's the question

La prise d'un gradateur entre le réseau secteur et une lampe permet-elle d'économiser de l'énergie ou vaut-il mieux utiliser une ampoule de wattage plus faible ?

(Question posée à plusieurs reprises)

*La prise d'un gradateur entre une ampoule et le réseau secteur diminue la quantité de lumière et aussi la consommation d'énergie. Le remplacement de l'ampoule d'origine par une ampoule de wattage plus faible sans utiliser de gradateur permet des économies d'énergie encore plus importantes. Cela est dû à la température du filament. Une ampoule ordinaire produit, outre de la lumière, également de la chaleur, ces 2 phénomènes requérant de l'énergie. N'oublions pas qu'une ampoule ne convertit en lumière que 10% environ de l'énergie qu'elle dérive du réseau secteur, le reste étant « gaspillé » en chaleur. Plus la température du filament augmente, plus les 2 types d'énergies consommées augmentent, mais la production de lumière augmente plus que la production de chaleur (cf. la loi dite « du déplacement de Wien »). Ainsi, le rendement d'une ampoule augmente avec l'augmentation de la température du filament. Un filament à forte température a une durée de vie plus courte. Il a donc fallu trouver un compromis entre ces 2 facteurs de rendement et de durée de vie. La combinaison ampoule de wattage plus important + gradateur se traduit par un abaissement de la température du filament, et partant du rapport lumière/chaleur et donc du rendement. L'utilisation d'un gradateur a une autre conséquence : celui d'augmenter la « chaleur » de la lumière fournie, plus de rouge et moins de bleu. À noter que ceci ne vaut que pour les ampoules à incandescence classiques et non pas pour les ampoules faible consommation qui ne*

*possèdent pas de filament mais utilisent le principe de la décharge gazeuse et produisent moins de chaleur et ont de ce fait besoin de moins d'énergie pour produire la même quantité de lumière.*

### Remplaçant pour le LM16A21

Impossible de trouver le LM 16A21 de Sharp pour le testeur de transistor intelligent. Après de nombreux mois de recherches, j'ai enfin trouvé un affichage LCD compatible avec le LM16A21 de Sharp utilisé par Elektor dans le testeur de transistor intelligent.

Chez Go-tronic, le LCD 1602A (réf. 03335) fait exactement l'affaire, le brochage étant cependant exactement inversé par rapport à celui de l'affichage Sharp. Mais il doit également exister des équivalents chez Selectronic pour cet affichage 2 x 16 caractères.

Louis Caruelle

**Projet c+** Salut la rédaction, J'ai lu ma revue aujourd'hui avec du retard mais cet article m'a tout de même interpellé. J'ai donc testé avec succès le montage proposé !-. Seule petite amélioration j'ai utilisé une impulsion dont la forme répond à la loi de poisson (d'avril évidemment) !!! @+ Philippe

*Nous apprécions le fait que certains de nos lecteurs aient apprécié ce petit clin d'oeil au mois d'avril, une bien vieille tradition...*

### Silence sur la fréquence

Ma carte à MSC1210 (Elektor 09/03) fonctionne sans pro-

blème avec le programme fourni (les valeurs de température changent lors d'un échauffement) mais le téléchargement d'un nouveau programme ne marche pas, de même d'ailleurs que la démonstration Ti-MS120-ADC. Le Ti-Downloader se rebiffe lors d'un transfert et émet le message suivant :

'No Response to initial CR'

Uwe Kall

*Je serais enclin à penser à un câble zéro modem mal câblé, ou à un positionnement erroné des cavaliers. Vous trouverez dans l'article « carte à MSC1210 » du numéro de décembre 2003 en page 340, un exemple de câblage de zéro modem (null modem) ou sur mon « vieux » site Internet à l'adresse [www.mikrocontroller.info/kabelsalat/](http://www.mikrocontroller.info/kabelsalat/) (en allemand). Pour le téléchargement il vous faut doter les 2 embases à cavalier alignées, JP1 et JP2, d'un cavalier, seule l'embase JP3 doit rester ouverte, ce dernier cavalier n'étant utilisé qu'à de très rares occasions.*

Jürgen Wickenhäuser

### Un bon petit vieux oscille

J'ai réalisé, tout récemment, The Compact, un amplificateur décrit en 1997. Lorsque je tourne le potentiomètre de réglage de la polarisation (*bias*) en butée à gauche, je retrouve toutes les valeurs indiquées sur le schéma. Si au contraire je tourne le potentiomètre vers la droite aux 3/4 de sa course les 2 amplificateurs (gauche et droite) se mettent à osciller entre 1 et 4 MHz. J'ai la quasi-circonférence que je n'ai pas créé de boucle de terre dans mon appareil et cela ne doit partant pas être la raison de mon problème. Avez-vous déjà eu des échos de problèmes de ce genre, ou aurais-je

quand même fait une erreur quelque part ?

Dick Flanderij

*Pour autant que nous nous en rappelions, il s'agit en effet d'une réalisation de fin 1996 ! Nous pouvons à raison qualifier le Compact de vrai « evergreen » et aimons y revenir à l'occasion. Si nous devons en croire le nombre de platines vendues, cet amplificateur a été énormément construit, mais il est exact que nous avons eu à l'occasion une observation du même genre que la vôtre. La solution est simple : placez un condensateur de 100 pF entre la base du transistor T16 et la masse. Ceci présente cependant l'inconvénient d'avoir une (légère) influence sur la caractéristique de transfert de l'amplificateur.*

**Sons** Il était possible, du temps du Commodore C64, de programmer soi-même le circuit intégré générateur de son. J'avais même à l'époque écrit un programme qui me permettait de faire passer un test d'audition à toute la famille. En cette ère du PC, il est quasiment impossible de faire quoi que ce soit avec une carte-son. Du moins c'est ma première impression. J'ai ouvert différents ouvrages de programmation mais je ne m'en sors pas. Ne pourriez-vous pas, amis d'Elektor, m'aider pour faire en sorte que mon PC produise mes sons ?

Klaus Bartges

*Notre propre expérience sur le sujet est limitée. Quelques recherches sur Internet nous ont fournies, entre autres, les adresses suivantes :*

[www.vbexplorer.com/VBExplorer/sound1.asp](http://www.vbexplorer.com/VBExplorer/sound1.asp)  
<http://users.bigpond.net/au/gradley/vbatutor/VBALesson6.htm>  
[www.officecomputertraining.com/](http://www.officecomputertraining.com/)

[vbutorial/tupages/page41.asp](http://vbutorial/tupages/page41.asp)  
<http://experts.about.com/q/1048/3376005.htm>

À noter au passage que dans l'ouvrage Publitronec « Je programme les interfaces de mon PC sous Windows » malheureusement épuisé, comportait quelques chapitres expliquant comment travailler avec la carte-son sous Delphi. Il y était même décrit un test auditif automatisé !

**Climat-Logger** Dans la liste des composants du Climat-Logger (Elektor de janvier 2004) il est mentionné, au niveau des condensateurs C1 à C3, cf. texte; dans le schéma aussi il est cf. texte, mais dans le texte on ne mentionne que C1, encore que sa valeur ne soit indiquée nulle part. Ma question est partant : quelles sont les valeurs des condensateurs C1 à C3 ? J'ai en outre remarqué que vous avez interverti les piles BT1 et BT2. BT1 est la paire de piles AA, BT2 la pile au lithium. C'est d'ailleurs ce que confirme la photo.

**H.R. Jannink**

La taille de C1 à C3 n'apparaît en effet nulle part dans le texte. Le but de la manoeuvre est de faire en sorte que la combinaison C1 à C3 ait la valeur la plus proche possible de 370 pF. L'auteur l'a obtenue à par la mise en parallèle de 2 condensateurs, l'un de 220 pF l'autre de 120 pF. Nous n'avions pas remarqué l'interversion des piles, mais vous avez parfaitement raison.

**EDiTS Pro** J'ai acquis la carte de contrôleur d'EDiTS Pro (juin 1999). Il supporte le format Motorola II. Comme les décodeurs de locomotive d'Elektor étaient trop

minuscules pour que je puisse les réaliser, j'ai acheté un décodeur de locomotive chez Conrad (type LD-W-1). Conrad indique que ce décodeur supporte le format Motorola II. Je pensais partant qu'il me serait possible d'utiliser les décodeurs de loco de Conrad avec EDiTS Pro. Cependant, il s'avère qu'ils ne sont pas interchangeables : avec EDiTS Pro il est possible de définir un maximum de 80 adresses alors que le décodeur de Conrad est capable de 255 adresses ! Ai-je mal compris quelque chose ou s'avère-t-il en pratique que la mention « supporte Motorola II » n'est pas suffisante. Existe-t-il peut-être plusieurs versions de ce fameux format ?

**W.G. Wouterse**

L'auteur Steffen van de Vries répond : « Officiellement, (selon Märklin) le format Motorola II ne supporte que 80 adresses. Sachant cependant que le format permet en principe jusqu'à 255 adresses, certains décodeurs de loco supportent cette possibilité (la dernière version du super-décodeur de locomotive d'EDiTS Pro et le LD-W-1). La version 1.2 du logiciel d'EDiTS Pro supporte elle aussi 255 adresses. »

**Mini-carte-son USB** Une petite question concernant la mini-carte-son USB (décembre 2002) : il est difficile voire impossible d'obtenir les mini-selfs L1 et L2 (selfs CMS blm31a601a) par paire et elles coûtent trop cher pour qu'on en achète 10 ! D'où ma tentative de les remplacer par des selfs plus courantes, mais ni le site de Farnell, ni celui de Murata ne donnent de valeur. Ma question : quelle est la valeur en  $\mu\text{H}$  ou  $\text{mH}$  de ces selfs ?

**Flor Coppens**

Murata indique pour ce type de selfs une impédance de 600  $\Omega$  à 100 MHz, la résistance ohmique étant elle de 0,9  $\Omega$ . Si nous nous basons sur ces valeurs, nous en dérivons une valeur de l'ordre de 1 mH. Cette valeur n'a d'ailleurs rien de critique, mais il est important d'utiliser des selfs aussi compactes que possible pour qu'elles soient efficaces.

Vous pouvez aussi envisager de faire vous-même vos 2 selfs. Bobinez par exemple 2 ou 3 couches de fil de cuivre émaillé fin l'une sur l'autre, diamètre intérieur de 1 mm, longueur 2 mm. Leur fonction est principalement l'élimination de parasites, le montage fonctionnera sans aucun doute avec elles aussi.

**Ah ces schémas**

**Elektor !** Chère Rédaction, quels logiciels ou programmes de CAD utilisez-vous pour « croquer » vos schémas et dessiner vos platines ? Je suis curieux car votre style de dessin semble remonter à la nuit des temps et en 1990 il n'existait pas encore, pour autant que je le sache, de programme de dessin.

**Serdar Ulukonakci (E-mail)**

Cher Serdar, cette question nous est posée en moyenne au moins une fois par semaine, ce qui explique que nous ayons pris la décision de rendre publique un secret bien gardé. Nous avons commencé par dessiner nos schémas sur PC dès 1985, avec OrCAD, les platines étant faites à l'aide de Ultiboard. Aujourd'hui nous utilisons des versions de McCad ([www.mcdac.com](http://www.mcdac.com)) et Ultiboard ([www.ultiboard.com](http://www.ultiboard.com)) fortement modifiées il est vrai et dotées de nos propres bibliothèques de symboles de circuits et de formes de boîtiers.

## MISES AU POINT

**MIDI-Lights, Elektor mars 2001, page 70 (000179)**

Quelques lecteurs signalent que la lampe du projecteur scintille au cours du réglage, à luminosité constante ce phénomène n'apparaît pas. Ceci est dû à une impulsion de synchronisation trop courte. Le remède est d'augmenter quelque peu C1 et C2 et de faire passer leur valeur de 2nF2 à 3nF3.

**Codeur pour simulateur de vol, Elektor février 2004, page 28 (030066-I)**

Dans la liste des composants, le numéro de nomenclature des mini-manches P1 et P2 correspond à la version 100 k $\Omega$ . Le schéma utilise lui des manches de 10 k $\Omega$  (modèle CTS 25A103A60TB). Les versions 10 et 100 k $\Omega$  fonctionnent tout aussi bien dans le montage.

## Règles du jeu

- Publication de la correspondance de lecteurs à la discrétion du Rédacteur en chef
  - Les points de vue et opinions exprimées par les correspondants ne sont pas nécessairement ceux du Rédacteur en chef ou de l'Éditeur.
  - La correspondance pourra, le cas échéant, être traduite ou éditée en longueur, clarté et style.
  - En cas de réponse à COURRIER, veuillez S.V.P. indiquer le numéro concerné.
  - Veuillez S.V.P. adresser votre correspondance : [redaction@elektor.fr](mailto:redaction@elektor.fr) ou
- Rédacteur en chef Elektor  
 Chez W.W.S. 38,  
 Rue Croix des Petits Champs  
 75001 Paris - France

[vbutorial/tupages/page41.asp](http://vbutorial/tupages/page41.asp)  
<http://experts.about.com/q/1048/3376005.htm>

À noter au passage que dans l'ouvrage Publitronec « Je programme les interfaces de mon PC sous Windows » malheureusement épuisé, comportait quelques chapitres expliquant comment travailler avec la carte-son sous Delphi. Il y était même décrit un test auditif automatisé !

**Climat-Logger** Dans la liste des composants du Climat-Logger (Elektor de janvier 2004) il est mentionné, au niveau des condensateurs C1 à C3, cf. texte; dans le schéma aussi il est cf. texte, mais dans le texte on ne mentionne que C1, encore que sa valeur ne soit indiquée nulle part. Ma question est partant : quelles sont les valeurs des condensateurs C1 à C3 ? J'ai en outre remarqué que vous avez interverti les piles BT1 et BT2. BT1 est la paire de piles AA, BT2 la pile au lithium. C'est d'ailleurs ce que confirme la photo.

**H.R. Jannink**

La taille de C1 à C3 n'apparaît en effet nulle part dans le texte. Le but de la manoeuvre est de faire en sorte que la combinaison C1 à C3 ait la valeur la plus proche possible de 370 pF. L'auteur l'a obtenue à par la mise en parallèle de 2 condensateurs, l'un de 220 pF l'autre de 120 pF. Nous n'avions pas remarqué l'interversion des piles, mais vous avez parfaitement raison.

**EDiTS Pro** J'ai acquis la carte de contrôleur d'EDiTS Pro (juin 1999). Il supporte le format Motorola II. Comme les décodeurs de locomotive d'Elektor étaient trop

minuscules pour que je puisse les réaliser, j'ai acheté un décodeur de locomotive chez Conrad (type LD-W-1). Conrad indique que ce décodeur supporte le format Motorola II. Je pensais partant qu'il me serait possible d'utiliser les décodeurs de loco de Conrad avec EDiTS Pro. Cependant, il s'avère qu'ils ne sont pas interchangeables : avec EDiTS Pro il est possible de définir un maximum de 80 adresses alors que le décodeur de Conrad est capable de 255 adresses ! Ai-je mal compris quelque chose ou s'avère-t-il en pratique que la mention « supporte Motorola II » n'est pas suffisante. Existe-t-il peut-être plusieurs versions de ce fameux format ?

**W.G. Wouterse**

L'auteur Steffen van de Vries répond : « Officiellement, (selon Märklin) le format Motorola II ne supporte que 80 adresses. Sachant cependant que le format permet en principe jusqu'à 255 adresses, certains décodeurs de loco supportent cette possibilité (la dernière version du super-décodeur de locomotive d'EDiTS Pro et le LD-W-1). La version 1.2 du logiciel d'EDiTS Pro supporte elle aussi 255 adresses. »

**Mini-carte-son USB** Une petite question concernant la mini-carte-son USB (décembre 2002) : il est difficile voire impossible d'obtenir les mini-selfs L1 et L2 (selfs CMS blm31a601a) par paire et elles coûtent trop cher pour qu'on en achète 10 ! D'où ma tentative de les remplacer par des selfs plus courantes, mais ni le site de Farnell, ni celui de Murata ne donnent de valeur. Ma question : quelle est la valeur en  $\mu\text{H}$  ou  $\text{mH}$  de ces selfs ?

**Flor Coppens**

Murata indique pour ce type de selfs une impédance de  $600 \Omega$  à 100 MHz, la résistance ohmique étant elle de  $0,9 \Omega$ . Si nous nous basons sur ces valeurs, nous en dérivons une valeur de l'ordre de 1 mH. Cette valeur n'a d'ailleurs rien de critique, mais il est important d'utiliser des selfs aussi compactes que possible pour qu'elles soient efficaces.

Vous pouvez aussi envisager de faire vous-même vos 2 selfs. Bobinez par exemple 2 ou 3 couches de fil de cuivre émaillé fin l'une sur l'autre, diamètre intérieur de 1 mm, longueur 2 mm. Leur fonction est principalement l'élimination de parasites, le montage fonctionnera sans aucun doute avec elles aussi.

**Ah ces schémas**

**Elektor !** Chère Rédaction, quels logiciels ou programmes de CAD utilisez-vous pour « croquer » vos schémas et dessiner vos platines ? Je suis curieux car votre style de dessin semble remonter à la nuit des temps et en 1990 il n'existait pas encore, pour autant que je le sache, de programme de dessin.

**Serdar Ulukonakci (E-mail)**

Cher Serdar, cette question nous est posée en moyenne au moins une fois par semaine, ce qui explique que nous ayons pris la décision de rendre publique un secret bien gardé. Nous avons commencé par dessiner nos schémas sur PC dès 1985, avec OrCAD, les platines étant faites à l'aide de Ultiboard. Aujourd'hui nous utilisons des versions de McCad ([www.mcdac.com](http://www.mcdac.com)) et Ultiboard ([www.ultiboard.com](http://www.ultiboard.com)) fortement modifiées il est vrai et dotées de nos propres bibliothèques de symboles de circuits et de formes de boîtiers.

## MISES AU POINT

**MIDI-Lights, Elektor mars 2001, page 70 (000179)**

Quelques lecteurs signalent que la lampe du projecteur scintille au cours du réglage, à luminosité constante ce phénomène n'apparaît pas. Ceci est dû à une impulsion de synchronisation trop courte. Le remède est d'augmenter quelque peu C1 et C2 et de faire passer leur valeur de  $2\text{nF}2$  à  $3\text{nF}3$ .

**Codeur pour simulateur de vol, Elektor février 2004, page 28 (030066-I)**

Dans la liste des composants, le numéro de nomenclature des mini-manches P1 et P2 correspond à la version 100 k $\Omega$ . Le schéma utilise lui des manches de 10 k $\Omega$  (modèle CTS 25A103A60TB). Les versions 10 et 100 k $\Omega$  fonctionnent tout aussi bien dans le montage.

## Règles du jeu

- Publication de la correspondance de lecteurs à la discrétion du Rédacteur en chef
  - Les points de vue et opinions exprimées par les correspondants ne sont pas nécessairement ceux du Rédacteur en chef ou de l'Éditeur.
  - La correspondance pourra, le cas échéant, être traduite ou éditée en longueur, clarté et style.
  - En cas de réponse à COURRIER, veuillez S.V.P. indiquer le numéro concerné.
  - Veuillez S.V.P. adresser votre correspondance : [redaction@elektor.fr](mailto:redaction@elektor.fr) ou
- Rédacteur en chef Elektor  
 Chez W.W.S. 38,  
 Rue Croix des Petits Champs  
 75001 Paris - France

# Proteus VSM

Essayez-vous à la simulation de  $\mu\text{C}$

David Daamen

Dessiner un schéma, faire une platine et tester.

De nos jours, tout cela se fait sur un PC !

Les programmes de simulation actuels sont même en mesure actuellement de simuler des microcontrôleurs.

Proteus est un set complet offrant des possibilités professionnelles à un prix abordable.

**Proteus Virtual System Modelling (VSM)** est un logiciel permettant de simuler sur PC des circuits numériques, analogiques, voire une combinaison des deux. La première étape consiste bien évidemment à la saisie du schéma. Ici c'est le programme **ISIS Schematic Capture** qui s'en charge. De la même source, **ARES**, un programme de dessin de circuits imprimés qui s'accouple parfaitement à ISIS. À eux deux, ils constituent une combinaison classique qui fait de la conception d'un circuit et de sa « mise sur cuivre » pratiquement un jeu d'enfant.

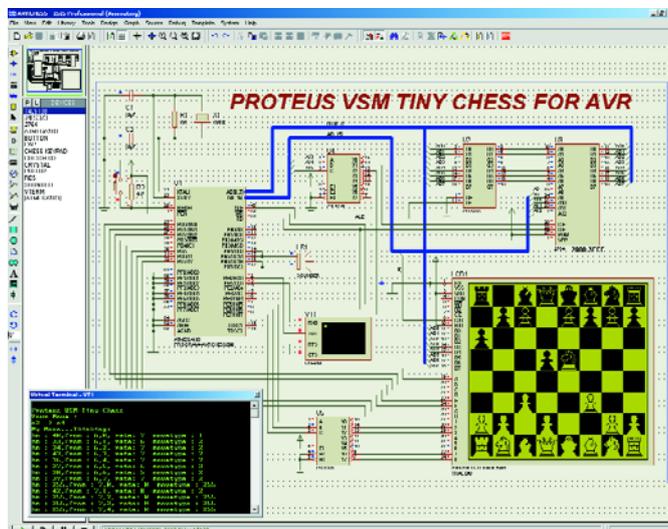
Il comprend, cela va de soi, des fonctions telles que vérification des connexions (ERCC = *Electrical Rule & Connectivity Check*), placement automatique des composants (*Auto Placer*) et des pistes (*Auto Router*). Rien de neuf jusqu'à présent direz-vous, il existe bien d'autres programmes présentant ces fonctionnalités.

Dans le cas de Proteus, c'est très précisément la partie

simulation qui présente des options extrêmement intéressantes. Il est en effet rare de pouvoir incorporer des microcontrôleurs dans le circuit. Il est vrai que d'autres programmes permettent également de le faire, mais Proteus VSM est également en mesure de simuler le code à l'intérieur du contrôleur. En principe cela non plus n'est pas unique, mais il est possible, en cours de simulation, de travailler de façon interactive avec le circuit : il est possible de commander les interrupteurs et les potentiomètres tout comme on le ferait dans la réalité, le résultat étant visualisé en temps réel sur des LED ou un affichage LCD par exemple. La simulation est même en mesure de faire écouter, via la carte-son, des signaux périodiques dont la fréquence se situe à l'intérieur du domaine audio. ET tout cela, comme nous le disions, en temps réel. On pourra ainsi, sur un système à Pentium II 300 MHz, simuler un système à 8051 standard tournant à 12 MHz.

## Le simulateur

**ProSPICE** est le noyau du simulateur. Il s'agit d'une



combinaison entre un simulateur analogique et un simulateur numérique dit à commande par événement (*event-driven*), ce qui permet de simuler sans problème, un mélange d'électronique analogique et numérique (*mixed-mode*). L'avantage de SPICE est qu'il est possible, sans arrière-pensée, d'ajouter les modèles de composants proposés par les fabricants. À noter que le logiciel en comporte près de 6 000 en standard.

Le simulateur fonctionne également fort bien avec ces modèles animés : des animations graphiques dont la forme peut changer pour, par exemple, donner le sens de déplacement du courant ou signaler l'allumage d'une petite ampoule.

Il est également possible de définir soi-même des modèles (animés), en principe sans même avoir à programmer. Le logiciel est accompagné d'un SDK (*Software Developers Kit*) documenté permettant la création de modèles en tant que fichiers .DLL.

## Mesurer

Qui dit tests virtuels dit du même coup mesures virtuelles. Et ce non pas uniquement

avec un simple voltmètre ou ampèremètre, mais aussi à l'aide d'un oscilloscope, d'un générateur de fonctions, d'un générateur de patrons, d'un compteur et d'un terminal sériel. Leur mise en oeuvre est celle de leurs homologues « réels ». Petit extra pratique : une possibilité de visualisation en temps réel du niveau logique de broches quelconques à l'aide d'un petit trait de couleur.

Des mesures plus exhaustives, sous forme graphique par exemple, ou la mesure de caractéristiques de fréquence, analyse de distorsion ou de bruit sont disponibles dans le set optionnel « **Advanced Simulation Option** ».

## Co-simulation de microcontrôleurs

Comme nous le disions, VSM est en mesure d'établir un lien entre la simulation du logiciel du microprocesseur et la simulation de l'ensemble de l'électronique qui l'environne. Ceci signifie que lorsque le code écrit vers un port du processeur, les niveaux du circuit changent en conséquence. Inversement, en cas de changement de niveau sur le circuit, le programme simulé en

prend note.

Les modèles de VSM ne supportent pas uniquement des ports d'E/S simples, mais aussi, selon le processeur utilisé, des interruptions, des temporisateurs, des USART, etc. Il est même possible de mettre plusieurs processeurs dans un même circuit : il suffit de les placer et de les connecter !

Lors de l'écriture de ces lignes, Proteus VSM est disponible pour les processeurs des types BASIC Stamp, PIC, AVR, HC11 et 8051.

### Débogage

Comme il est fort peu probable que l'on réussisse à faire fonctionner un circuit à base de microprocesseur du premier coup, l'élimination des erreurs, le débogage est un processus extrêmement important. Sur Proteus VSM c'est un point auquel on énormément travaillé. La fonction pas à pas est très utile. À l'image du débogueur de logiciel, il est également possible d'examiner, ligne après ligne, le code dans le processeur. On voit alors instantanément les

effets du code sur l'ensemble du circuit, y compris sur toute l'électronique connectée au contrôleur. Il est même possible, en fonction de la famille de processeur et des outils de programmation utilisés, de faire du pas à pas dans le code écrit dans un langage de programmation de haut niveau comme C.

Nous ne pouvons que recommander Proteus aux concepteurs de réalisations combinant l'électronique numérique à l'électronique analogique.

(040043-1)

### Prix de Proteus VSM :

449 €, (ex-TVA, par famille de processeur)

### Pour de plus amples informations :

Antratek  
E-mail: [info@antratek.nl](mailto:info@antratek.nl)  
Internet: [www.antratek.nl](http://www.antratek.nl)

Publicité

## Répertoire des annonceurs

BETA LAYOUT	<a href="http://www.pcb-pool.com">www.pcb-pool.com</a>	57
ESPACE COMPOSANT ELECTRONIQUE	<a href="http://www.ibcfrance.fr">www.ibcfrance.fr</a>	23
EUROCIRCUITS	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	77
FARNELL IN ONE	<a href="http://www.farnellinone.be">www.farnellinone.be</a>	17
GÉNIBRÈDE	<a href="http://www.genibrede.com">www.genibrede.com</a>	57
GSE	<a href="http://www.gsenet.com">www.gsenet.com</a>	57
HFC AUDIOVISUEL		63
HI TECH TOOLS	<a href="http://www.hitechtools.com">www.hitechtools.com</a>	31
IMPRELEC		36
LEXTRONIC	<a href="http://www.lextronic.fr">www.lextronic.fr</a>	84
LKC ELECTRONIQUE		63
MULTIPOWER	<a href="http://www.multipower.fr">www.multipower.fr</a>	31
NATIONAL INSTRUMENTS	<a href="http://www.ni.com/france">www.ni.com/france</a>	31
OPTIM INFO	<a href="http://www.optiminfo.com">www.optiminfo.com</a>	57
PROGRAMMATION	<a href="http://www.programmation.fr">www.programmation.fr</a>	22
SELECTRONIC	<a href="http://www.selectronic.fr">www.selectronic.fr</a>	2

### Réservation d'espaces publicitaires

Réservez dès aujourd'hui votre espace publicitaire dans le magazine Elektor du mois de juillet/août 2004!  
Date limite de réservation : **le 7 juin 2004.**

Pour toute information concernant la publicité aussi bien dans notre magazine que sur notre site internet [www.elektor.fr](http://www.elektor.fr) contactez :

SL Régie – Sophie Lallonder  
39, Rue Lamarck, 75018 Paris  
Tel : 06 98 29 27 04  
E-mail : [sophie.lallonder@wanadoo.fr](mailto:sophie.lallonder@wanadoo.fr)

**ThePCBShop.com**

**ELEKTOR CIRCUITS IMPRIMÉS ET ENCORE PLUS**

# www.elektor.fr

**Plus qu'une simple vitrine**

...

**c'est aussi un atelier d'applications !**

- Accès
- Plus qu'une simple vitrine ... C'est aussi un atelier d'applications !
- Carte de programmation et d'expérimentation en kit montée prêt à l'emploi !
- Nouveau livre : Programmation de microcontrôleurs
- Elektor Item Tracer : nouvelle version !
- Nouveau CD-ROM : CD Elektor 2003
- Nouveau livre : Réception des satellites de Télécom mondiale
- Offrez un an d'abonnement !
- Le GB590 pour le Gameboy est à nouveau disponible !
- Carte d'expérimentation à MSPC1230 complètement montée pour 99 C
- Trois numéros pour 12,50 C sans engagement !