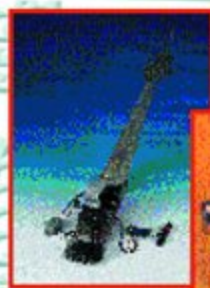


ELEKTOR

+ PC-PLUS:

- Bluetooth-transfert de données sans fil
- essayez-vous au port imprimante
- compilateur BASIC/Pascal pour PIC

Joe Jensen
(avec Embedded)
parle



«Tour du Rhin» mod 2

antenne-
cadre active**mono-carte « 537-Lite »**

visitez
le site
Elektor
nouveau



l'avenir de l'embarqué

*basé sur un exposé de, et une interview avec,
Joe Jensen, Director Embedded Intel Architecture*

Après un exposé public de près d'une heure sur les objectifs et la stratégie d'Intel en ce qui concerne les systèmes embarqués ou enfouis (*embedded*), Mr Joe Jensen d'Intel a bien voulu nous accorder une interview exclusive qui prit la forme d'un exposé plus spécifiquement centré sur les vues d'Intel quant à la globalisation d'Internet, monologue suivi de réponses substantiées aux quelques questions posées par la rédaction de ce magazine.



Commençons par prendre le temps de relever les points les plus importants de l'exposé de Joe Jensen. L'explosion du Web, car c'est bien de cela qu'il s'agit (cf. le graphique de la **figure 1**), est un phénomène irréversible. Si voici 2 ans, le département *Embedded* d'Intel ne faisait que 10% de son chiffre d'affaire dans ce domaine, Joe Jensen pronostique un bon 50% dans les 2 ans à venir. Le moto d'Intel pour l'an 2002 est : « *1 billion connected people, 1 trillion connected dollars* ». Nous savons donc à quoi nous en tenir.

Parallèlement, on constate un basculement du matériel vers le logiciel. Si voici quelques années le rapport de temps requis par chacune des étapes Matériel-Logiciel d'un développement était encore, respectivement, de 9 contre 1, il a basculé aujourd'hui dans la proposition inverse de 1 contre 9. Le leitmotiv des fabricants est de nos jours, le raccourcissement des temps de développement. S'il fallait à un concept, voici moins d'un lustre, 9 mois (voire plus) de validation auxquels s'ajoutaient 9 mois de mise en production, on cherche, aujourd'hui, à abaisser ces délais à 2 fois 3 mois seulement. Les fabricants doivent, pour survivre, concentrer leur attention à réduire les délais de fabrication, l'objectif étant d'arriver aux 6 mois, sachant que la durée de vie de la plupart des appareils entrant en question pour recevoir de l'électronique embarquée n'est plus guère, actuellement, que de 5 ans. Cela a une implication importante. En effet, vu le peu de temps disponible, les fabricants ne veulent plus faire de matériel, ils préfèrent se concentrer sur le logiciel, c'est-à-dire l'adaptation de leur savoir

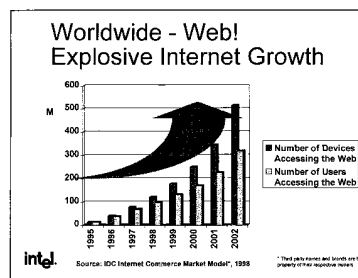


Figure 1. L'évolution du nombre de systèmes connectés au Web est quasi-exponentielle.

faire, de leur application, sur une plateforme matérielle existante. Les applications de l'embarqué sont innombrables, allant des feux de circulation aux magnétrons en passant par les téléphones, les TV setbox, les cartes-mère de systèmes industriels. On peut ainsi affirmer aujourd'hui que l'embarqué n'est pas un marché, mais un groupe d'applications. L'un des domaines majeurs de la mise en oeuvre de puissance embarquée est le domestique. Verrons-nous bientôt des réfrigérateurs à lecteur de code à barre intégré qui enregistreront la sortie d'un produit et en assureront un réapprovisionnement automatique. Lorsque la machine à laver détectera la fin du paquet de lessive elle se mettra en contact avec l'épicier du coin pour commander une recharge d'Ariel ou autre Omo. On en arrive même à parler d'écosystème Internet, cet éco n'ayant rien à voir

avec l'écologie, mais avec l'économie. Passons maintenant à l'interview. Son sujet principal était, d'après Joe Jensen, l'annonce de 3 nouveautés, importantes aux yeux d'Intel :

l'IACP

(Intel® Applied Computing Platform)

Intel est heureux d'annoncer la participation de pas moins de 10 nouveaux partenaires, fabricants au nombre desquels on compte Nokia, Alcatel, Siemens, pour n'en citer que quelques-uns, qui optent d'utiliser les noyaux proposés par Intel et se contentent de créer le firmware.

Avec la globalisation d'Internet, les frontières entre entreprises s'évaporent. Comme le dit Intel « *There is One Enterprise and it's Global. There is One Network and it's the Internet* ». Cette globalisation d'Internet met, pratiquement, du même coup, hors-jeu les nombreux Intranet établis dans les limites d'une société. Les sociétés ne disposant pas d'un Intranet peuvent se hausser au niveau des autres en mettant à profit les étonnantes capacités proposées par l'Internet. Même les personnels d'Intel communiquent avec leur quartier général par Internet, voire des techniques plus modernes dont nous n'avons pas encore la moindre idée en Europe, le réseau Intranet de leur société étant quasiment intégré dans la toile d'araignée mondiale.

On connaît, en électronique appliquée, outre le fameux « *The number of transistors on an integrated circuit doubles every 18 months* », une seconde loi, celle de Metcalf qui dit : « *The value of a network increases by the square*

of the number of its nodes ».

Quelles sont les conséquences de cette globalisation pour l'embarqué ?

Il faut penser de plus en plus en termes de connectivité au Net, de hautes performances, de systèmes de traitement rapides et surtout d'appareils conçus pour des applications dédiées.

Intel se tourne plus spécifiquement vers le PC industriel, une application à but bien défini régnant au coeur d'un central économique, connecté aux PC et serveurs d'une société, en vue de fournir en temps quasi-réel toutes les informations requises par les preneurs de décision (management). Ce PC industriel doit saisir et transmettre en temps réel toutes les données capitales concernant les liaisons, le rendement, les situations de stocks etc... Un tel PC industriel peut fort bien prendre la forme d'un SBC (*Single Board Computer*) doté d'une interface industrie standard, PC-104, CompactPCI ou autre...

Les fabricants ne veulent plus développer de matériel en raison du temps mis à valider un concept (6 mois à plus d'un an), ni non plus, en raison des *time to market* de plus en plus courts requis pour rester dans la course, perdre leur temps (*time is money*), au développement et à la validation d'un produit.

La seconde annonce concerne :

L'Entry-level Communications Appliance Reference Design

Il s'agit là d'un concept comportant les composants essentiels pour l'obtention des performances et fonctionnalités requises pour les appareils de communication de base (*entry level*). Le but de ce développement est de permettre à un client potentiel de se concentrer sur le marché en disposant dès le départ d'un module complet modifiable aisément pour répondre à ses besoins sans qu'il n'ait à modifier le coeur du microprocesseur ni à adapter le concept du chipset, tout cela en vue de permettre une réduction du temps de développement (*time-to-market*).

Le concept de référence utilise les composants suivants :

- Un processeur Celeron™ 300A d'Intel®,
- Le Chipset 440BX d'Intel®, ce qui sous-entend 2xEIDE, 2 ports USB, un BIOS de 2 Moctets Flash,
- 2 circuits 82559, composants d'interface Ethernet 10/100,
- et de la mémoire Flash (d'Intel® bien entendu), et tout cela sur une platine de 8,50" x 5,12" sur laquelle il est également prévu de la place pour 32 Moctets de mémoire en barrette SODIMM et pour l'alimentation.

Notons l'apparition des premières unités centrales CompactPCI à Pentium III® (Ziatech). Elles doivent équiper les commutateurs télécoms d'ATM aux USA.

La troisième annonce concerne :

l'Intel® AC SFL, (Intel® Applied Computing System Firmware Library V1.0)

Il s'agit d'une collection de primitives logicielles à utiliser comme fondation lors du développement de firmware de pointe embarqué. Elle a été optimisée pour être utilisée avec les microprocesseurs et chipsets d'Intel et fournit un set de fonctions pour une initialisation au niveau du composant tant en ce qui concerne le processeur que la mémoire et le chipset. Il ne s'agit pas d'un BIOS et ne comporte pas de services pour des systèmes reposant sur un programme travaillant par interruptions.

Un point important : SFL est gratuit pour le client. On pourra le télécharger du site Internet d'Intel voire le demander sur CD-ROM par carte d'enregistrement.

Cette bibliothèque de firmware est prévue pour travailler avec les produits d'Intel suivants : le Pentium II®, le Celeron™, la mémoire SDRAM et son chipset, le 440BX AGPset.

Les projets d'Intel au niveau du matériel ?

Il n'est pas exclu que l'on ait apparition, un jour ou l'autre, d'une version embarquée du Xeon.

Intel travaille à une carte-mère universelle qui pourrait recevoir tous les processeurs (ceux d'Intel bien entendu) allant du Celeron 300 MHz au Pentium III 700, voire 800 MHz le jour où il sortira, ce qui ne devrait plus être très loin puisque Intel vient tout juste d'annoncer son Pentium III® fabriqué en technologie 0,18 µm et travaillant jusqu'à 733 MHz.

Notons au passage qu'Intel vient d'introduire une nouvelle famille de cartes-mères baptisée les Intel® Desktop Boards, la première d'entre elles étant le SU810 Desktop Board, rien de moins. Il suffirait de définir les différentes tensions requises par le noyau (*core*) et de paramétrer le reste, par la mise en place ici et là, de l'un ou l'autre cavalier.

Investing in Innovation

Pour illustrer ce dont il parlait, et symboliser ce qui nous attend dans les prochaines années, Joe Jensen nous parla de BlackBerry™, un petit appareil portable d'une dizaine de centimètres de long sur 6 de large et 2 d'épaisseur (**figure 2**), une sorte de pager utilisant un réseau radio sans fil spécialement conçu à cet effet pour rester en communication permanente avec l'ensemble des autres systèmes connectés à ce réseau spécifique.



Figure 2. BlackBerry™ est-il le pré-curseur de ce qui nous attend tous dans les années qui viennent ? L'avenir nous le dira.

Il ne fonctionne (mal)heureusement (?) par encore en Europe.

Cet appareil est un système de communication bidirectionnel par radio permettant d'effectuer les opérations les plus classiques (actuellement) envoyer un E-mail, lire son courrier électronique; il est fait par RIM (*Research In Motion*), une société américaine.

C'est d'après Joe Jensen, l'exemple-type d'une application d'une architecture embarquée ne comportant guère plus qu'un microprocesseur 386™ d'Intel®, 2 Moctets de mémoire Flash, un bloc HF pour l'émission et la réception. Il permet de recevoir son courrier électronique et, par le biais de son minuscule clavier, d'y répondre, laborieusement peut-être, mais d'y répondre quasi-instantanément dès lors que l'on se trouve dans une zone couverte par le système radio requis, (parties les plus peuplées des USA et les villes les plus importantes du Canada).

Comme le montrent les quelques points abordés l'avenir est à nos portes et c'est à nous de voir comment nous le modèlerons.

Terminons par une parole célèbre de Robert Noyce : « *Don't be encumbered by history, go out and do something wonderful* » (ne restez pas esclave du passé, sortez de la grisaille et créez quelque chose de merveilleux). (000033)

Adresses Internet intéressantes :

<http://developer.intel.com/>,

le Developer's Site

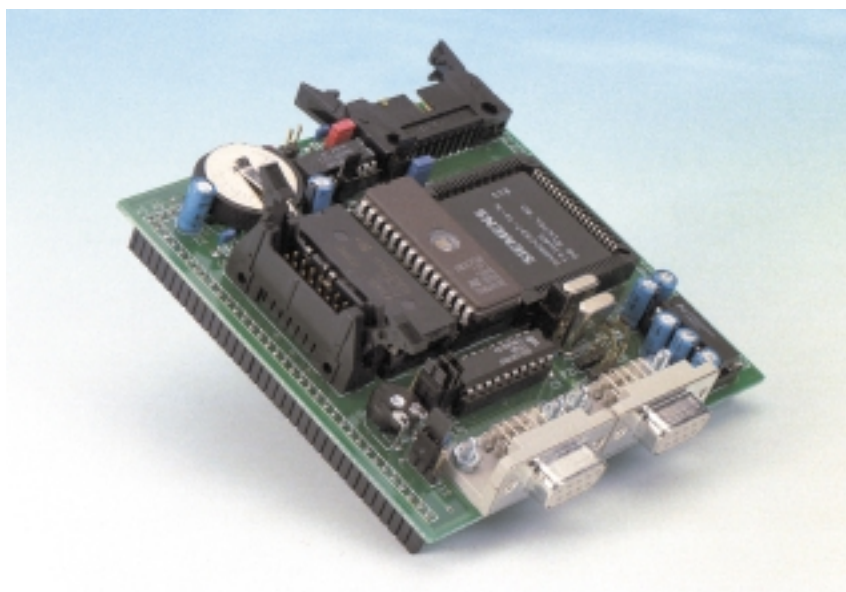
<http://developer.intel.com/platforms/applied/comm/index.htm>, la page d'accueil de la CACP (*Communications Applied Computing Platforms*).

Texte : Guy Raedersdorf

mono-carte « 537-Lite » (I)

module faible coût à microcontrôleur puissant et fiabilité améliorée

Siemens appelle tendrement son 80C537, un dérivé du 8051, son éléphant à 8 bits. Après vous avoir proposé, voici quelque temps déjà, un système d'évaluation et d'expérimentation sous la forme de la mono-carte à 80C537 (juin 1997), Elektor s'est mis en tête de vous proposer un module-noyau faible coût reprenant ce même microcontrôleur; il pourra, être utilisé, très flexiblement pour de nombreuses applications petites et moyennes en tant que système évolutif; il peut, en outre, se targuer d'une très grande fiabilité de fonctionnement.



Un coup d'oeil, aussi furtif qu'il soit, à la fiche de caractéristiques du microcontrôleur 80C537, aura vite fait de nous convaincre que nous nous trouvons en présence d'un composant aux talents multiples : 1 convertisseur analogique/numérique à 8 (voire 10) bits doté de pas moins de 12 entrées, 3 temporisateurs/compteurs dotés d'une possibilité de comparaison/capture,

une unité de coprocesseur mathématique, un chien de garde embarqué (*watchdog*), 7 ports d'entrées/sorties numériques bidirectionnels, 2 interfaces sérielles *full duplex*, 14 sources d'interruption, un mode d'économie d'énergie et une ribambelle d'autres caractéristiques intéressantes. Si vous voulez en savoir plus au sujet de ce contrôleur ô combien intéressant, nous

Caractéristiques techniques

- ♦ microcontrôleur 80C537, fréquence de fonctionnement commutable entre 12 et 16 MHz).
- ♦ Fiabilité accrue par la présence du surveillant de microcontrôleur MAX807.
- ♦ 32 Koctets d'EPROM, 32 Koctets de RAM, avec possibilité d'extension externe vers 64 Koctets pour chaque type de mémoire.
- ♦ mémoire RAM sauvegardée par pile.
- ♦ Présence de 4 adresses CS utilisables librement (dont l'une pour LCD si nécessaire)
- ♦ Interface sérielle 0 : aux normes RS-232
- ♦ Interface sérielle 1 : au choix aux normes RS-232 ou TTL.
- ♦ Interface permettant une connexion directe d'un affichage LCD disposant de 1 x 16 à 4 x 20 caractères, réglage de luminosité.
- ♦ Se laisse programmer dans tous les langages que connaît le 8051 (fichier INTEL-Hex) ou par mots-clé (token) Basic52
- ♦ Alimentation : 5 V, 75 mA (à 12 MHz), 80 mA (à 16 MHz), valeurs hors affichage LCD.

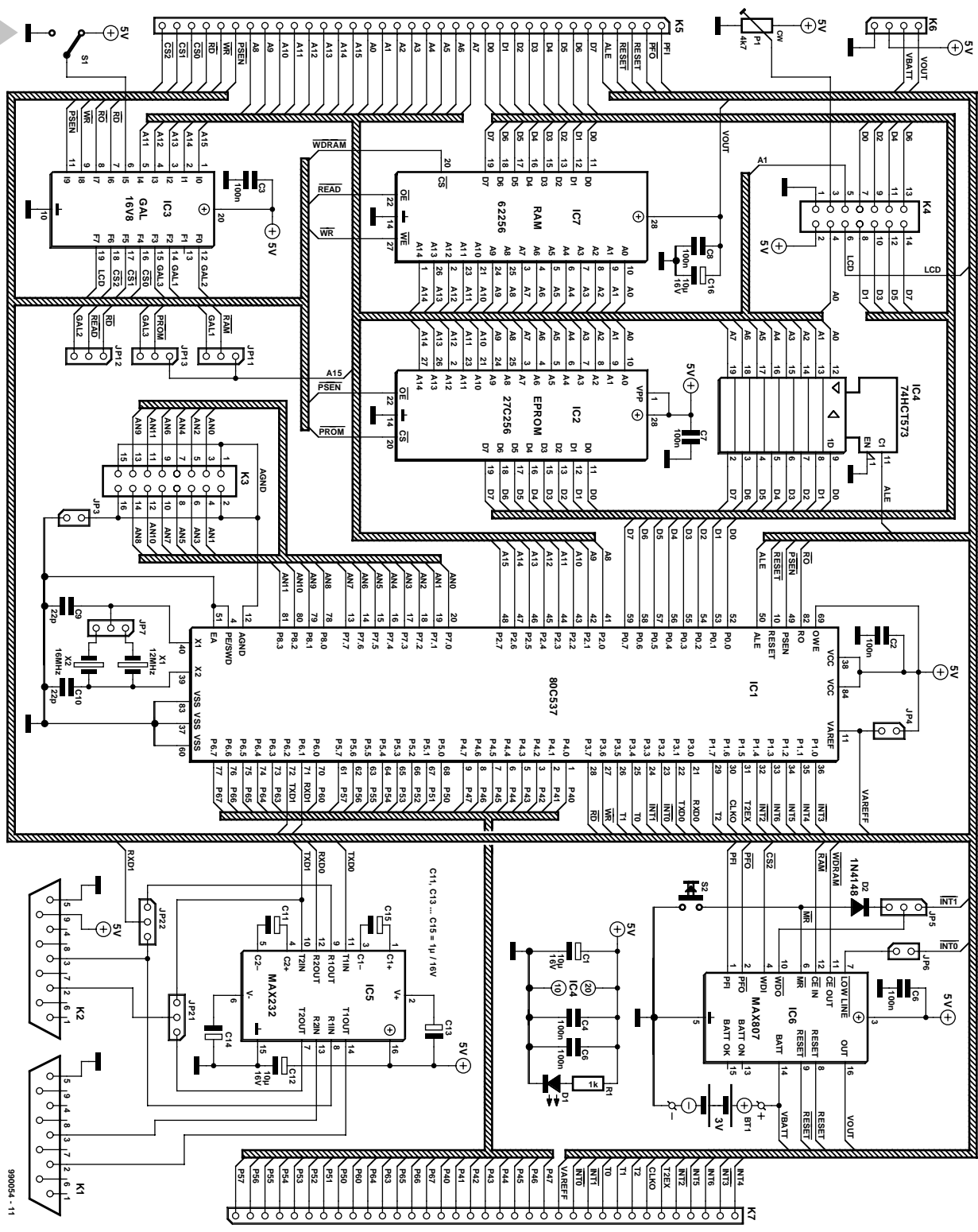


Figure 1. Le SAB80C537 est un excellent microcontrôleur méritant de trôner au coeur d'une carte extrêmement puissante.

vous renvoyons à la bibliographie donnée à la fin du second article.

LE MATÉRIEL

La mono-carte « 537-Lite » constitue, avec le SAB80C537 qui règne en son centre, un module à microcontrôleur à faible coût utilisable à toutes les sauces. Il pourra être enfiché, comme un circuit intégré monstrueux, au coeur même de l'application dont il doit assurer le pilotage. L'encadré énumérant les caractéristiques techniques permet de se faire une petite idée de la

puissance du module. Toutes les lignes importantes du microcontrôleur ont été reportées sur des embases, voire dérivées vers des borniers où l'utilisateur pourra s'en servir au gré de ses besoins pour d'éventuelles extensions. Le matériel constituant la carte et dont on trouve le schéma en **figure 1**, le laisse subdiviser en blocs fonctionnels dont les plus importants sont les suivants : **L'électronique du microcontrôleur** qui est en fait l'application de base d'un

microcontrôleur de la famille 8051 à la petite différence près qu'il est possible, par le biais du cavalier JP7, d'opter pour l'une des 2 fréquences de cadencement disponibles, 12 ou 16 MHz. **Les ports d'E/S numériques** sont directement accessibles par le biais de l'embase K7. Les entrées ne comportent pas d'électronique de protection, les sorties n'ont pas d'électronique de commande (*driver*). **Les ports d'entrées analogiques**

Figure 2. Cette platine relativement compacte pourra être montée comme un gros circuit intégré directement sur le montage d'application.

Le 80C537 possède 12 entrées analogiques multiplexées (0 à 5 V, résolution de 8 (voire de 10) bits) qui sont regroupées sur l'embase K3. Il faudra penser à ajouter, si nécessaire, l'électronique de protection et de filtrage externe requise. Les cavaliers JP3 et JP4 permettent de définir le niveau de référence des broches de la tension analogique, de la masse et de la tension de référence positive.

Les interfaces sérielles

Le 80C537 dispose de 2 interfaces sérielles, SS0 et SS1, dont les lignes ont été prolongées jusqu'aux embases Sub-D K1 et K2. SS0 travaille à des niveaux V24 fixes (niveaux RS-232 fournis par l'adaptateur de niveau MAX232); il est possible, au niveau de SS1, de paramétrer, par la mise en place de JP21 et JP22, la norme de fonctionnement, V24 ou TTL, de cette seconde interface sérielle.

Les bancs de mémoire

La connexion des circuits de mémoire (32 Koctets d'EPROM, IC2 et 32 Koctets de RAM, IC7) respecte l'approche classique du 8051, à savoir par le biais des ports 0 et 2 du contrôleur et du verrou d'adresses 74573, IC4.

La logique de décodage

L'électronique de codage et de décodage des signaux de commande prend la forme d'une GAL 16V8, IC3, ce qui permet au montage de gagner en compacité.

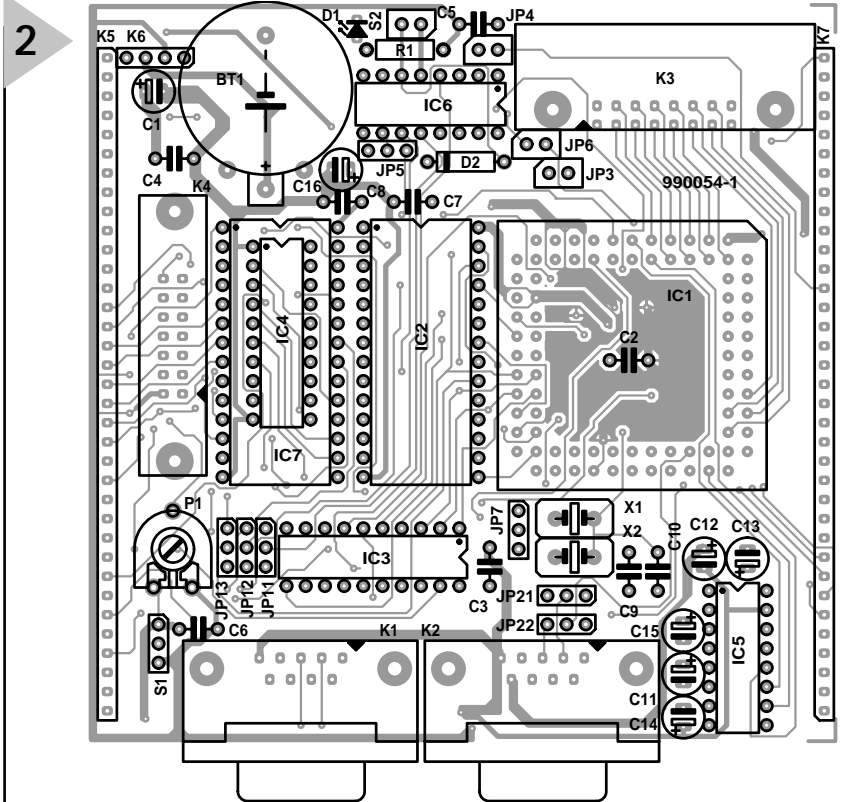
L'affichage LCD

Il est prévu de pouvoir connecter directement à l'embase K4 un affichage à cristaux liquides (LCD) alphanumérique si tant est qu'il dispose d'un contrôleur HD44780 de Hitachi ou tout autre type compatible, ce qui est le cas de près de 80% des affichages. Il est donc possible, de connecter et de piloter un affichage comportant entre 1 ligne de 16 caractères et 4 lignes de 20 caractères. L'ajustable P1 sert à régler la luminosité de l'affichage LCD.

Circuit de surveillance de microcontrôleur, MAX807

La sécurité de fonctionnement remarquable de cette mono-carte est due à l'utilisation d'un circuit intégré spécialisé, le MAX807. Ce composant remplit, dans un système à microcontrôleur, 7 fonctions de surveillance, de signalisation et d'alarme :

- ◆ Génération automatique d'un signal POR (Power On Reset = de remise à zéro à la mise sous tension)
- ◆ Remise à zéro manuelle par action sur une touche
- ◆ Fonction de chien de garde (watchdog)
- ◆ Protection de ligne de validation de circuit (CS = Chip Select) en cas de



- disparition de la tension système
 - ◆ Suivi de la tension système
 - ◆ Suivi de la tension d'alimentation
 - ◆ Connexion possible d'une pile de sauvegarde avec gestion de son énergie.
- Les pages « Infocarte » publiées ailleurs dans ce magazine vous proposent des informations additionnelles concernant ce composant.

RÉALISATION ET MISE EN FONCTION AISÉES

Il faudra, lors de l'implantation des composants sur la platine dont on retrouve le dessin des pistes et la séri-

graphie de l'implantation des composants en **figure 2**, respecter les règles de base classiques : commencer par les composants les plus petits, pour passer aux plus encombrants ensuite, utiliser des supports de qualité pour tous les circuits intégrés, et procéder à une vérification critique de la platine une fois les soudures terminées. La pile de sauvegarde ne sera mise en place qu'après que l'on se soit assuré du bon fonctionnement de la carte.

IC4 et IC7 appellent une remarque vu que ces 2 composants sont, pour des raisons d'économie de place, montés

Liste des composants

Résistances :

- R1 = 1 kΩ
- P1 = 4kΩ7

Condensateurs :

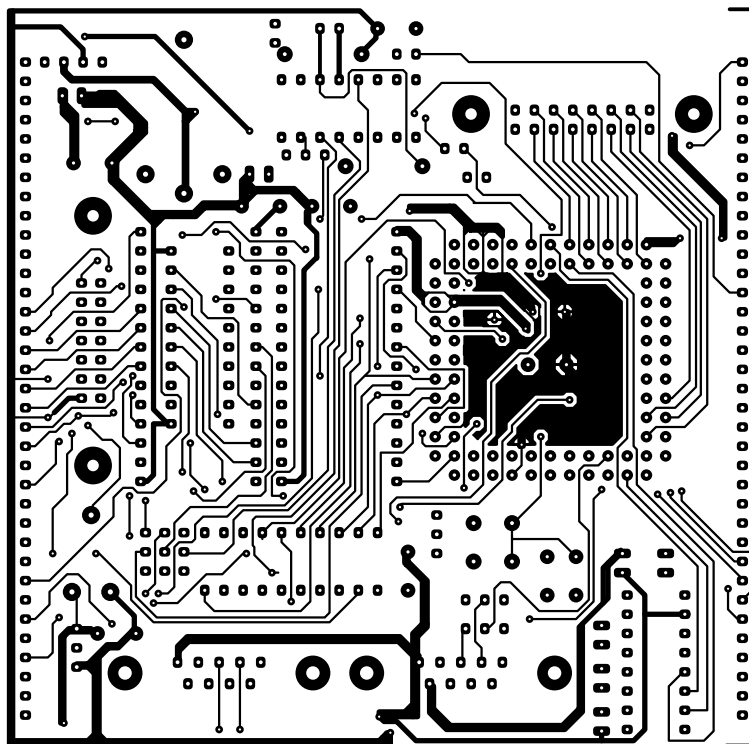
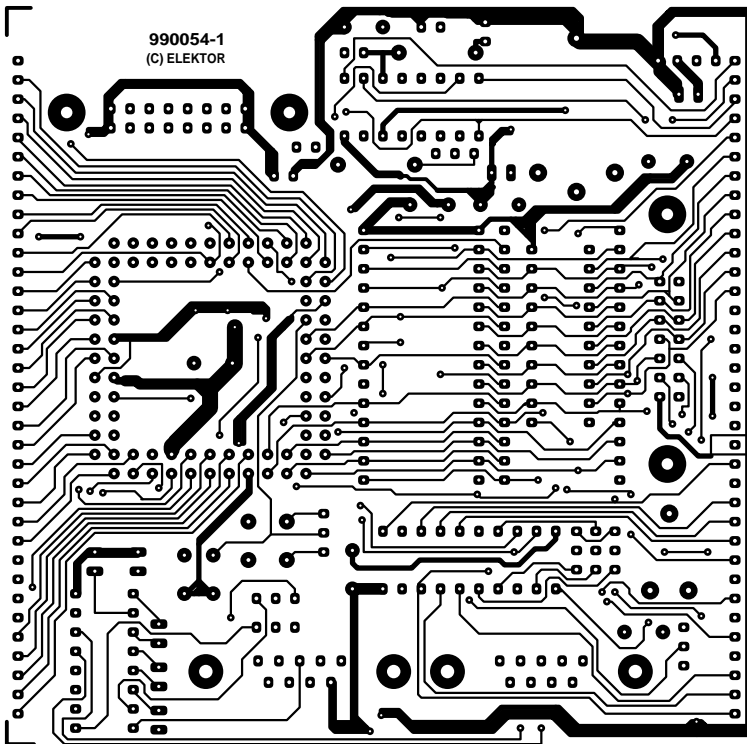
- C1, C12, C16 = 10 μF/16 V vertical
- C2 à C8 = 100 nF céramique
- C9, C10 = 22 pF céramique
- C11, C13 à C15 = 1 μF/16 V vertical

Semi-conducteurs :

- D1 = LED 3 mm rouge haut rendement
- D2 = 1N4148
- IC1 = SAB 80C537-16-N (Siemens)
- IC2 = 27C256 (programmée EPS 996510-1)
- IC3 = GAL 16V8 (programmée EPS 996515-1)
- IC4 = 74HCT573
- IC5 = MAX232 (Maxim)
- IC6 = MAX807 NCPE (Maxim)
- IC7 = RAM 62256 (70 ns)

Divers :

- K1, K2 = embase sub-D en équerre à 9 contacts encartable
- K3 = embase droite à 2 rangées de 8 contacts avec éjecteurs
- K4 = embase droite à 2 rangées de 7 contacts avec éjecteurs
- K5, K7 = embase autosécable SIL mâle à 1 rangée de 35 contacts
- K6 = embase autosécable SIL mâle à 4 contacts
- JP3, JP4, JP6 = embase mâle autosécable à 2 contacts + cavalier
- JP5, JP7, JP11 à JP13, JP21, JP22 = embase mâle autosécable à 3 contacts + cavalier
- X1 = quartz 12 MHz
- X2 = quartz 16 MHz
- S1 = embase autosécable à 3 contacts destinée à la connexion de l'inverseur unipolaire
- S2 = bouton-poussoir unipolaire
- BT1 = pile au lithium 3 V de 20 mm de diamètre + porte-pile



en « gigogne ». Ces 2 circuits intégrés étant montés sur support, il faudra surélever la RAM IC7 par la mise en place d'une rangée de contacts hauts voire la superposition d'un second support à 24 broches dont on aura supprimé la barrette transversale.

Après un dernier contrôle, une fois les composants mis en place, on optera, après avoir implanté l'EPROM moniteur, IC2, pour un paramétrage, par le biais des cavaliers, forçant la mono-carte en mode de fonctionnement n°2, *Program Download*.

▶ Mettre les cavaliers PP11, JP12 et JP13

en position 2/3

- ▶ Placer JP21 et JP22 en position 1/2
- ▶ Implanter les cavaliers JP3 et JP4
- ▶ Ne pas implanter ni JP5 ni JP6
- ▶ Le cavalier JP7 est mis en position 1/2 (fréquence d'horloge de 12 MHz)
- ▶ L'inverseur S1 est mis en position *Load RAM* (vers la masse).

Il est temps d'appliquer la tension d'alimentation. Il est recommandé, lors des essais, de ne pas brancher l'alimentation directement sur l'embase K6, mais d'alimenter la carte par le biais de l'embase sub-D K2 (masse à la broche 5 et +5 V à la

broche 9). L'alimentation doit être de 5 V exactement; attention à une inversion de sa polarité aux conséquences dramatiques. La consommation de courant ne doit pas dépasser 80 mA.

Après avoir coupé l'alimentation on connecte l'interface SS0 (K1) de la carte à un port COM libre du PC. On pourra fabriquer soi-même le câble requis en respectant le brochage suivant :

K1 de la mono-carte	PC (AT)
Sub-D 9	Sub-D 9
TxD 2	RxD 2
RxD 3	TxD 3
GND 5	GND 5

Il faudra, si l'on envisage de connecter la mono-carte à un PC de type XT, intervertir les contacts RxD et TxD et se rappeler que la masse se trouve à la broche 7 de l'embase Sub-D à 25 contacts du PC (XT).

K1 de la mono-carte	PC (XT)
Sub-D 9	Sub-D 25
TxD 2	RxD 3
RxD 3	TxD 2
GND 5	GND 7

On peut remettre la mono-carte sous tension. On place ensuite la disquette **EPS976008-1** dans le lecteur A et on lance le programme *term.exe* dans une fenêtre DOS. On fera, au premier lancement, *term 1* pour indiquer que l'on utilise le port COM1, ou encore *term 2* dans le cas du port COM2. Après apparition du message de copyright on verra s'afficher un menu à 4 points se terminant par un caractère #.

Une action sur la touche de RAZ de la mono-carte se traduit par l'apparition de l'entête

TFH-80C537er-Monitor

La liaison entre le PC et la mono-carte étant établie, on peut partant télécharger le premier programme. Il faut, avant de procéder au téléchargement d'un nouveau programme, effectuer une RAZ de la mono-carte. Cette RAZ fait repasser au début d'un programme déjà chargé, son lancement se faisant par une instruction *g0*.

On appuie sur la touche « F2 » (*Input from file*) pour le téléchargement d'un programme, entre le nom *p1.hex* comme nom de fichier et lance la commande par un « RETURN ».

On voit alors défiler quelques colonnes de chiffres sur l'écran; il s'agit du fichier INTEL-Hex du programme *p1.hex*. L'apparition d'un unique # marque la fin du téléchargement du programme.

Après une instruction *g0* suivie d'un « RETURN » le programme démarre à l'adresse 0. S1 se trouve alors en position *Load RAM*.

(990054-1)

Texte : Rolf Gerstendorf
Technique : Hans Bonekamp

cours de programmation BASIC Stamp (5)

5^e partie : Remote Rover

Nous allons, ce mois-ci, voir comment faire répondre le véhicule BoE-Bot aux instructions qui lui sont données par le biais d'une télécommande infrarouge.

par Al Williams

20

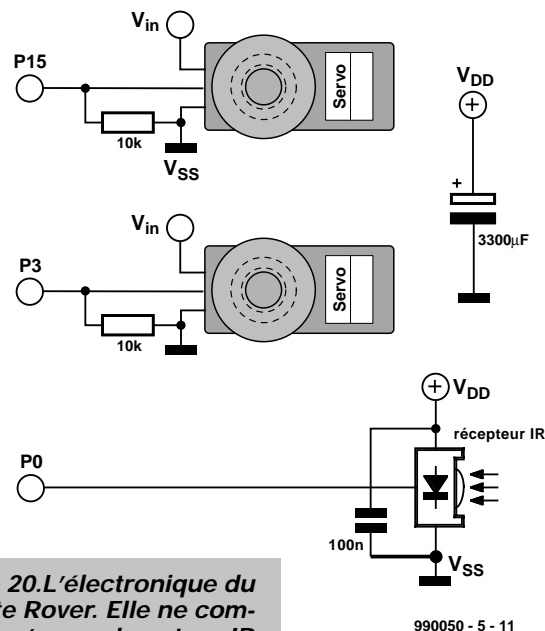


Figure 20. L'électronique du Remote Rover. Elle ne comporte qu'un seul capteur IR (Panasonic type 4602).

Savez-vous combien de télécommandes vous possédez. Si vous me ressemblez, leur nombre est ahurissant. La quasi-totalité des appareils grand public semblent ne pas pouvoir se passer de leur télécommande propre. Si vous avez le malheur de démolir ou d'égarer la télécommande fournie d'origine avec un appareil, il va vous falloir en acheter une nouvelle, puis une autre, aucune d'entre elles n'étant en fait en mesure de commander infailliblement l'appareil en question. Nous avons vu, le mois dernier, comment doter votre BoE-Bot d'un dispositif de détection d'obstacle par infrarouge (IR). Il existe une pléthore de composants IR pour la simple et bonne raison que les télécommandes utilisent pour la plupart l'IR pour l'envoi de commandes vers l'appareil concerné.

Aussi, avons-nous pensé, pourquoi ne pourrions nous pas « apprendre » à notre BoE-Bot à capter des instructions émises par une télécommande ? Si vous étiez en mesure de communiquer à distance avec votre robot vous pourriez lui ordonner de faire ce que vous voulez sans avoir à quitter votre fauteuil ! Ce faisant, vous découvrirez également comment le Basic Stamp mesure les impulsions et traite les tableaux (*arrays*).

DÉTAILS

L'électronique requise est, comme le montre le schéma de la **figure 20**, très simple. Il s'agit, à peu de choses près, du même circuit que celui utilisé dans l'article du mois dernier, à ceci près qu'il ne requiert qu'un seul capteur IR et ne comporte pas de LED. Les LED se

trouvent dans la télécommande. Il devrait être possible, après quelques essais, d'utiliser pratiquement n'importe quelle télécommande. É'ai, quant à moi, opté pour une télécommande Sony pour la simple raison que son protocole SIRCS (*Serial Infra Red Control Signal*; baptisé occasionnellement Control S) est parfaitement documenté sur Internet. Si vous ne possédez pas de télécommande Sony, achetez une télécommande universelle bon marché et programmez-la pour la commande d'un téléviseur Sony. Il existe un certain nombre de protocoles courants utilisés par les télécommandes (jetez un coup d'oeil à l'adresse : www.hut.fi/Misc/Electronics/docs/ir/ircodes.html pour vous faire une idée). La plupart d'entre elles utilisent une forme ou une autre de MLI (Modulation en Largeur

d'Impulsion = PWM pour *Pulse Width Modulation* en anglais). Dans le cas du protocole Sony, la télécommande émet un bit de début (*start*) souvent connu sous la dénomination d'impulsion AGC, d'une longueur relativement importante (supérieure à 2 ms). Cela permet au récepteur de se synchroniser et d'ajuster sa CAG (Commande Automatique de Gain, d'où le AGC mentionné plus haut); cet ajustement se fait à l'intérieur du module du détecteur IR.

Après le bit de début, la télécommande émet une série d'impulsions. Une impulsion de 600 μ s représente un 0, une impulsion de 1 200 μ s représentant elle un 1. 2 impulsions successives sont séparées par un intervalle de 600 μ s. Vu l'influence que différents facteurs peuvent exercer sur cette chronologie, il ne faut pas la prendre au pied de la lettre.

MESURES D'IMPULSIONS

Le Stamp est en mesure de lire des largeurs d'impulsion (par le biais de son instruction PULSIN). PULSIN requiert 3 arguments. Le premier spécifie la broche à utiliser pour la mesure de l'impulsion. Le Basic Stamp fait de ladite broche, si elle ne l'est pas déjà, une entrée. L'argument suivant indique à l'instruction s'il lui faut détecter une transition bas/haut (1) ou haut/bas (0). Le dernier argument est une variable de la taille d'un mot (*word-sized*) qui contient la durée de l'impulsion (si tant est qu'elle en ait une). Le Basic Stamp utilise une base de temps de 2 μ s; dans ces conditions si la variable contient le nombre 100, l'impulsion aura une largeur de 200 μ s. L'instruction « s'éteint » au bout de 131 ms. En l'absence d'impulsion au cours de cette durée l'instruction met la variable à 0. La taille d'impulsion maximale que l'on puisse mesurer est en effet de 65 353 x 2 μ s, d'où les 131 ms.

L'instruction PULSIN ne prend d'impulsion en compte que si elle détecte le flanc prévu. Supposons que nous voulions déterminer la durée d'une action sur un bouton-poussoir. Le bouton applique un 0 à la broche d'entrée tant qu'il est activé. Si on lance l'exécution de PULSIN après le « passage à zéro » du bouton on n'aura jamais mesure de quelque impulsion que ce soit. Il faut exécuter PULSIN avant l'arrivée du flanc de l'impulsion. Pour peu que l'on y réfléchisse, cela semble évident puisque cela évite que PULSIN ne fournisse des résultats erronés. L'instruction mesure à chaque fois une impulsion complète.

On pourra utiliser une variable d'un octet de large (*byte-wide*) si les impulsions à mesurer ne dépassent jamais 510 μ s. S'il devait alors se faire que la largeur d'impulsion dépasse cette valeur, le résultat fourni sera faux sans qu'il n'y ait d'avertissement. L'instruc-

tion utilise toujours, en interne, un temporisateur (*timer*) à 16 bits, de sorte que le choix d'une variable d'un octet ne change en rien la durée d'écoulement de la période du temporisateur (*timeout*). Il va nous falloir, avec le capteur IR, mesurer des impulsions d'une largeur supérieure à 2 ms, ce qui explique que tous les programmes listés dans cet article utilisent une variable d'un mot de large.

Si l'on suppose que le capteur IR se trouve connecté à la broche 0, il est extrêmement facile de mesurer la largeur d'une impulsion IR :

```
I RREAD:
  PULSIN 0, 0, W1
  IF W1=0 THEN I RREAD
    ' pas d'impul si on
  DEBUG ?W1
  GOTO I RREAD
```

On pourra essayer ce programme simple pour vérifier que l'on détecte des largeurs différentes en fonction des touches de la télécommande que l'on actionne. Comme on ne lit pas encore chacun des bits, les résultats obtenus n'ont pas encore de signification. La télécommande Sony émet 12 bits de données (13 au total si l'on tient compte du bit de début). Les 7 premiers bits représentent le code de fonction (le bit de poids faible en premier), les 5 bits suivants constitue le code d'identification (ID) d'un appareil. Vu que seules les touches de fonction nous intéressent, nous pouvons ignorer le code d'identification.

On pourrait, en principe, lire le mot de la façon suivante :

```
I RREAD:
  B0=0 ' Lecture de l'octet
  B1=1 ' masquage de bit
  PULSIN 0, 0, W5
  IF W5<1200 then I RREAD
    ' ce n'est pas un bit de début
  FOR B2 = 1 to 7
    PULSIN 0, 0, W5
  IF W5<400 THEN READZERO
  B0=B0+B1
    ' positionner 1 bit
  READZERO:
  B1=B1*2 ' supprimer masquage
  NEXT
```

Joli morceau de code, n'est-ce pas, mais il ne marche pas. Le principe est, quant à lui, sain. La première PULSIN lit le bit de début et rejette tout bit ne possédant pas la longueur correcte. Le code entre ensuite dans une boucle, lisant successivement tous les bits et plaçant dans B0 le bit correspondant lorsque la longueur de l'impulsion est supérieure à 800 μ s, longueur largement supérieure aux 600 μ s d'un 0; vu qu'un 1 a une durée nominale de 1 200 μ s, aucun d'entre eux ne devrait avoir une durée aussi courte que 800 μ s.

Le seul problème avec ce code est que l'intervalle entre les bits n'est que de 600 μ s. Cela ne laisse pas tellement de temps au Stamp pour récupérer après lecture de la dernière impulsion. Le Basic Stamp requiert un minimum de 470 μ s pour exécuter une instruction IF (la durée moyenne d'exécution d'une instruction est de quelque 330 μ s – certaines plus, d'autres moins).

Même les instructions les plus rapides nécessitent plus de 100 μ s. La présence d'instructions entre les différentes exécutions de la routine PULSIN, le Basic Stamp rate l'un ou l'autre bit.

L'une des options ouvertes consisterait à passer à un Basic Stamp IISX, beaucoup plus rapide qu'un Basic Stamp standard. Il est cependant possible, avec une programmation astucieuse, de faire en sorte qu'un Stamp II lise les impulsions IR à cette vitesse.

LA SOLUTION

Il nous faut, pour résoudre ce problème, réduire les instructions intercalées entre les instructions PULSIN. En fait, il faudrait, pour garantir un fonctionnement correct, éliminer les instructions intercalées entre les instructions PULSIN. Comment faire ? Se résoudre à stocker les résultats bruts fournis par PULSIN pour les traiter ultérieurement lorsque l'on dispose de plus de temps de traitement.

Le problème se pose de savoir où l'on stocke ces durées brutes ? On pourrait, bien entendu, envisager d'utiliser des variables, mais il nous faut un minimum de 8 variables-mot (une pour le bit de début, 7 autres pour les bits de données). Cela pourrait se traduire par du code on ne peut plus laid. Vous aurez peut-être pensé, si vous avez l'habitude de travailler avec d'autres langages de programmation, à stocker les comptages dans un tableau. Bonne idée, d'autant plus que le Basic Stamp sait travailler avec des tableaux.

TABLEAUX

Un tableau est une technique de regroupement des variables similaires en utilisant un nom de variable commun, un *nombre-index* servant à différencier une variable de l'autre. Supposons que nous voulions travailler avec des nombres impairs. On pourrait écrire :

```
oddnums var byte(5)
oddnums(0) = 1
oddnums(1) = 3
oddnums(2) = 5
oddnums(3) = 7
oddnums(4) = 9
```

La variable oddnums(2) réfère au troisième nombre impair (on se souviendra que l'index démarre à 0). Si l'on réserve 5 éléments, l'index utilisera les chiffres 0 à 4. Si l'on utilise un chiffre

Listage 8. Le programme du Remote Rover.

```
' Remote Rover par Al Williams
irsense con 0
irinput var in0
irthreshold con 450
irstartlow con 1100
irstarthi con 1300

value var byte ' resultat

raw var word(7)
start var word
dummy var word

right_servo con 3 ' moteur de servo droit
left_servo con 15 ' moteur de servo gauche
delay var byte ' durée de cycle du moteur
center con 750
speed var word
i var byte

delay=10
speed=100

top:
  gosub read_ir
  if value=1 then forward
  if value=3 then left
  if value=5 then right
  if value=7 then back
  goto top

forward:
  for i=1 to delay*2
    pul sout left_servo, center-speed
    pul sout right_servo, center+speed
    pause 20
  next
  goto top

back:
  for i=1 to delay
    pul sout left_servo, center+speed
    pul sout right_servo, center-speed
    pause 20
  next
  goto top

left:
  for i=1 to delay
    pul sout left_servo, center-speed
    pul sout right_servo, center-speed
    pause 20
  next
  goto top

right:
  for i=1 to delay
    pul sout left_servo, center+speed
    pul sout right_servo, center+speed
    pause 20
  next
  goto top

read_ir:
  ' Le problème est ici l'existence d'un trou de
  ' quelque 500 us entre les bits de sorte que le
  ' Stamp risque de rater quelque bits à moins que
  ' l'on ne lise la totalité en une seule passe.
  ' Il n'est, partant, pas possible de procéder à
  ' cette lecture à l'intérieur d'une boucle ni
  ' même de tester le bit de début tant que l'on
  ' n'en a pas terminé.
  if irinput=0 then noir ' Déjà au milieu d'une
  impulsion, autant la sauter
  pul sin irsense, 0, start
  pul sin irsense, 0, raw(0)
  pul sin irsense, 0, raw(1)
  pul sin irsense, 0, raw(2)
  pul sin irsense, 0, raw(3)
  pul sin irsense, 0, raw(4)
  pul sin irsense, 0, raw(5)
  pul sin irsense, 0, raw(6)
  ' Pourquoi ne pas les faire sortir
  pul sin irsense, 0, dummy
  pul sin irsense, 0, dummy
  pul sin irsense, 0, dummy
  pul sin irsense, 0, dummy
  ' vérifier bon bit de début
  if (start<irstartlow) or (start>irstarthi)
  then noir
  value=0
  for dummy=6 to 0
    value=value*2
    if raw(dummy)<irthreshold then ir0
    value=value+1

ir0:
  next

  return
noir:
  value=-1
  return
```

différent de ces 5, on écrit dans de la mémoire utilisée, le cas échéant, par une autre partie de son programme. Cette technique est pratique car elle permet de travailler avec des tableaux imbriqués dans des boucles. On pourrait, pour écrire tous les nombres impairs, écrire, par exemple, le code suivant :

```
l var byte
for l = 0 to 4
  Debug ?oddnms(l)
next
```

Il nous faut bien entendu respecter les limites du Basic Stamp au niveau de la mémoire. Les tableaux ne nous

donnent pas de mémoire supplémentaire, ils nous permettent tout simplement de mieux utiliser la mémoire disponible.

LECTURE D'UN SIGNAL IR

Une lecture avec succès du flux de données IR requiert une série de 13 instructions PULSIN (ou encore 8 si on décide d'ignorer les bits additionnels). Il nous suffit de stocker 8 de ces comptages. Voici à quoi votre code devrait ressembler :

```
irsense con 0
irstartlow con 1100 ' largeur
minimale du bit de début
```

```
irstarthi con 1300 ' largeur
maximale du bit de début
raw var word(7)
dummy var word
start var word
```

```
read_ir:
  pul sin irsense, 0, start ' lecture
  du bit de début potentiel
  pul sin irsense, 0, raw(0)
  pul sin irsense, 0, raw(1)
  pul sin irsense, 0, raw(2)
  pul sin irsense, 0, raw(3)
  pul sin irsense, 0, raw(4)
  pul sin irsense, 0, raw(5)
  pul sin irsense, 0, raw(6)
  pul sin irsense, 0, dummy
  ' ID composant - ignorer
```

```

pul sin irsense, 0, dummy
'ID composant - ignorer
pul sin irsense, 0, dummy
'ID composant - ignorer
pul sin irsense, 0, dummy
'ID composant - ignorer
pul sin irsense, 0, dummy
'ID composant - ignorer

```

Il ne vous reste plus qu'à traiter les données brutes. Il se peut que le premier bit pris en compte par le code ne soit pas le bit de début, un simple test nous évite une lecture d'un paquet de données en plein milieu :

```

if start<irstartlow or
start>irstarthi then noir

```

Il n'est malheureusement pas possible de pratiquer ce test juste après la lecture du bit de début, il nous faut lire l'ensemble du paquet et décider ensuite s'il est ou non valide.

La tâche suivante est de convertir la donnée brute en un nombre binaire. À ce niveau, la vitesse n'est pas critique :

```

value var byte
value=0
for dummy=6 to 0
  value=value * 2
  if raw(dummy)<irthreshold
then ir0
  value = value +1
ir0:
next
return

```

Ce code ne fait rien de plus qu'examiner chaque valeur (dans l'ordre inverse). Si la valeur brute dépasse le seuil, le code incrémente (ajoute 1) le résultat. En tout état de cause, le code multiplie la valeur par 2 (un décalage à gauche) à chaque fois que l'on se trouve dans la boucle. Le Stamp connaît un opérateur décalage à gauche (*shift left*) ce qui permet de remplacer l'instruction de multiplication par :

```
value = value << 1
```

Il est possible maintenant de lire facilement les touches de la télécommande. Les touches étant répétitives vous ne manquerez pas de vouloir en tenir compte dans votre programme.

REMOTE ROVER

Armé de ces routines pour le capteur IR, il reste à doter votre robot d'une télécommande. Tout ce qu'il vous faut savoir c'est la correspondance entre les touches de la télécommande et les valeurs émises, ce qui se laisse facilement déduire en faisant appel à la routine `ir_read` et en utilisant `debug` pour l'impression la variable représentant la valeur.

Dans le cas de la télécommande Sony,

Internet

<http://www.parallaxinc.com> – BASIC Stamp Manual Version 1.9, BASIC Stamp DOS et Windows Editor, exemples de programmes. Sources ation exemples. Sources de distribution internationale.

<http://www.stampsinclass.com> — documentation BoE-Bot, Robotics curriculum, BoE-Bot formats de dessin *.dxf and *.dwg, groupe de discussion pour l'utilisation du BASIC Stamp dans l'éducation.

chucks@turbonet.com — créateur du BoE-Bot et auteur de cette série d'articles. Assistance technique.

kgracey@parallaxinc.com — co-auteur de cet article. Assistance technique et questions concernant le programme d'éducation.

<http://www.selectronic.fr> — distributeur du BASIC Stamp de Parallax en France

la touche « 1 » retournait un 0, la touche « 2 » un 1, et ainsi de suite. L'auteur a choisi le « 2 » comme commande de marche avant, le « 8 » comme commande de déplacement vers l'arrière, les touches « 4 » et « 6 » correspondant à une rotation vers la gauche et la droite respectivement. Si l'on voit les positions de ces touches, ce choix est presque naturel.

Il n'est pas très difficile de procéder au test d'une touche donnée et d'envoyer les instructions correctes aux servos (à l'image des autres commandes de déplacement décrites dans les articles précédents). Le **listage 8** vous donne le code complet du Remote Rover.

L'auteur rencontra cependant une limitation. Après s'être amusé quelque temps avec le code du Remote Rover, il imagina obtenir un mouvement vers l'avant du robot lors d'une action sur la touche « 2 », mouvement devant se poursuivre jusqu'à l'entrée d'une nouvelle commande ou une action sur la touche « 5 ». Les choses tournèrent au problème.

Il est très facile de positionner un indicateur (*flag*) pour signaler un mouvement vers l'avant. Le problème auquel on se trouve cependant confronté est qu'il faut, lorsque l'on essaie de lire le signal IR émis par la télécommande, attendre que chaque instruction PULSIN se soit écoulée complètement avant que le programme ne retourne dans la boucle principale. Avec 13 instructions auxquelles il faut 131 ms pour s'écouler complètement cela se traduit par près de 2 s de temps mort entre 2 instructions de mouvement successives. D'où un mouvement relativement saccadé du robot. Ce phénomène est moins brutal avec l'approche originale, vu que les codes répétitifs émis par la télécommande terminent les instructions PULSIN sans que celles-ci n'aient à s'écouler totalement.

Il est bien évidemment possible d'améliorer les choses quelque peu en ne prenant pas en compte les codes d'identification (ID) des composants. Cette solution diminue le nombre de périodes devant s'écouler, mais présente l'inconvénient d'augmenter le nombre de fois que votre robot ratera le bit de début et aura partant à se resynchroniser avec la télécommande. Tous

comptes faits, le code resta ce qu'il était. Autre moyen d'accroître l'efficacité : tester le capteur avant d'interroger le bit de début. Si le capteur répond par un 0, c'est que l'on se trouve au milieu d'un paquet (de données) et que partant il n'existe aucune possibilité de lire, alors pourquoi se faire du soucis ? Le code donné dans le listage 8 procède à ce test.

ÉVOLUTIONS FUTURES

Il existe mille et une modifications simples à faire au programme du listage 8. On pourrait, par exemple, imaginer de programmer les boutons de volume et de canal en vue de modifier les variables de vitesse et de délai. Essayez-vous y.

Une fois que vous savez lire les codes de la télécommande vous pourriez doter votre robot de toutes sortes de fonctions plus sophistiquées. Il serait, par exemple, très simple de faire en sorte que certaines touches spécifiées produisent une série prédéfinie de pas. Avec un rien d'énergie additionnelle on pourrait imaginer d'utiliser la télécommande pour programmer une séquence, la stocker en EEPROM, pour la rappeler ultérieurement (à l'image d'une macro). On pourra, une fois saisi le protocole de la télécommande, envoyer au robot une série de commandes destinées au pilotage de votre téléviseur, mettre le robot dans une autre pièce et lui faire émettre les instructions prévues une fois qu'il se trouve en face du téléviseur. On pourrait même envisager d'utiliser ces techniques pour permettre à 2 robots de communiquer l'un avec l'autre sur des faibles distances.

Bien que les impulsions « crachées » par la télécommande IR soient très rapides, le Basic Stamp peut fort bien les traiter à condition que le programme ait été écrit correctement.

L'instruction PULSIN permettait une mesure simple et précise des impulsions. Bien qu'ils ne soient pas indispensables, les tableaux simplifient énormément la tâche. Avec un rien d'ingéniosité, il n'y a pratiquement pas de limite à ce que le Basic Stamp peut faire.

(990050-5)

Texte : J. Buiting
Technique : L. Lemmens

audio-DAC 2000

3^e partie : quelques aspects pratiques

Ceux d'entre nos lecteurs qui ont lu avec attention les 2 premiers articles consacrés au « sujet » connaissent tout du fonctionnement de notre nouveau convertisseur N/A haut de gamme. Les seuls aspects sur lesquels il nous faut encore attirer l'attention concernent la réalisation pratique. Ce sera là le thème majeur de ce troisième et dernier article.



Il nous faut, cependant, ajouter une remarque à l'introduction ci-dessus, vu qu'il reste un sous-ensemble du montage dont nous n'avons pas encore parlé, à savoir l'alimentation.

Nous avons signalé, dans le second article, que les circuits intégrés des convertisseurs N/A étaient alimentés en + et -5 V, tensions dérivées, par le biais des régulateurs IC13 et IC14, de la tension principale de ± 12 V destinée à l'alimentation de la partie analogique du circuit. Comme il est important de réduire au strict minimum les lignes d'alimentation vers les convertisseurs, lesdits régulateurs ont trouvé directement place sur la « platine des CNA ». Nous avons besoin, pour l'alimentation de la partie du récepteur et du reste de l'électronique présente sur la platine des CNA, d'une tension asymétrique de +5 V et d'une tension symétrique de ± 12 V. Sa génération se fait par l'intermédiaire de régulateurs de tension classiques (IC15 à IC17) qui, associés aux composants connexes qu'ils requièrent, trouvent place sur un cir-

cuit imprimé distinct. On retrouve, en **figure 5**, le schéma de l'alimentation; on voit immédiatement que l'alimentation numérique +5 V est totalement séparée de la tension symétrique de ± 12 V requise par l'électronique analogique. Les masses des 2 alimentations sont interconnectées, sur la platine des CNA, entre le filtre numérique et les CNA (au niveau de JP3 donc). Comme on le voit, l'alimentation ne se résume pas uniquement à quelques régulateurs intégrés mais requiert également la présence de quelques ponts de redressement et l'un ou l'autre condensateur. On trouve, coincées entre le pont de redressement et les condensateurs de lissage, les résistances d'arrêt, R55, R56 et R8, destinées à limiter le courant de charge et partant à éliminer encore mieux d'éventuels parasites. Les borniers K11 et K13 se voient connectés aux transformateurs de l'alimentation, un ± 15 V/4 VA et un 8 V/4 VA respectivement. Nous vous laissons le choix du modèle, la liste des composants donnant un

certain nombre de type de transformateurs utilisables. Nous avons développé une platine spécifiquement destinée à ce montage vous permettant l'utilisation différents types de transformateurs, circuit imprimé sur lequel viennent également se monter les 2 filtres secteur, les porte-fusibles et les borniers de connexion. Nous avons consacré un article distinct à ce projet, article publié ailleurs dans ce magazine.

LES PLATINES

Comme vous l'ont appris les articles précédents, l'audio-DAC 2000 se subdivise en 3 sous-ensembles : le récepteur, la conversion N/A et l'alimentation. Le récepteur est associé à une mini-platine d'indication sur laquelle se trouve une paire d'afficheurs 7 segments à LED servant à visualiser la fréquence d'échantillonnage. Le dessin de circuit imprimé (double face) représenté en **figure 6** se compose ainsi en fait de 4 platines distinctes que l'on pourra aisément séparer de par la présence de lignes de fracture fraisées, opération à

laquelle on procédera avant de passer à l'étape de la mise en place des composants et de leur soudure. Cette séparation se fera en douceur. Si vous avez des doutes quant aux risques que sous-entend cette opération, rien ne vous interdit de saisir une scie à lame et dentition fines.

Il faudra, mais pourquoi le répétons-nous donc à chaque article ?, lors de la réalisation, respecter la liste des composants et la sérigraphie de leur implantation. Attention à l'orientation des circuits intégrés et à la polarité des condensateurs électrochimiques, toute erreur à ce niveau peut avoir des conséquences catastrophiques.

On soudera de préférence tous les interrupteurs DIP directement sur la platine, exception faite de S2 si tant est que l'on envisage, ultérieurement, d'utiliser une commande par processeur. On utilisera alors un support à 8 broches de bonne qualité (contacts tulipe) dans lesquels on soudera les conducteurs requis.

Toutes les connexions ayant trait à l'alimentation prennent la forme de borniers encartables, ce qui facilite le câblage voire, le cas échéant, un éventuel dépannage. Le +5 V du numérique est appliqué à la platine des CNA, la platine du récepteur recevant son alimentation par le biais de la liaison entre K5 et K3. La diode D4 présente sur la platine des CNA protège, dans une certaine mesure, l'électronique contre une surtension au niveau de l'alimentation.

En ce qui concerne la connexion de la mini-platine de l'affichage, le câble en nappe à 10 conducteurs y est fixé par le biais d'un connecteur encartable. L'autre extrémité du câble plat est dotée d'un connecteur à 2 rangées de 5 contacts. Attention à la longueur du câble en nappe. Les 2 afficheurs sont soudés directement sur la platine.

Du côté de la platine du récepteur, IC1, le CS8414, est lui aussi soudé directement sur la platine. Attention aux décharges électrostatiques qui risqueraient d'endommager ce composant. IC4 et IC5 pourront prendre place sur un support. Il est recommandé, pour une meilleure protection contre les parasites, de souder l'oscillateur à quartz, IC3, lui aussi sur la platine (il se trouvera ainsi plus près du plan de masse).

Il est recommandé, lorsque l'on passe au montage des composants sur la platine des CNA, de commencer par la soudure du DF1704 (figure 7), composant miniature s'il en est. Ce circuit intégré CMS est un 28 broches en boîtier SSOP, ce qui implique que les broches se trouvent à 0,65 mm seulement l'une de l'autre ! Cette opération exige un soin extrême, une panne très fine, voire l'utilisation d'une loupe pour vérifier la qualité des soudures et

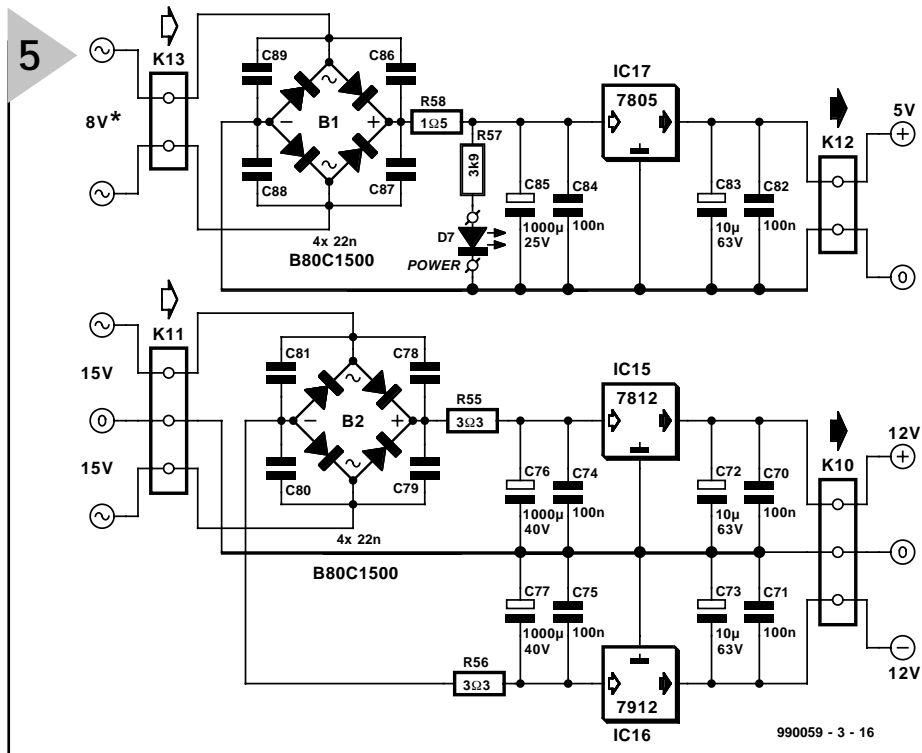


Figure 5. Schéma de l'alimentation. Au niveau de la platine, l'alimentation numérique de +5 V et l'alimentation ±12 V analogique sont parfaitement séparées l'une de l'autre.

l'absence de court-circuit entre broches. On pourra utiliser de la tresse à dessouder pour enlever un excédent de soudure en veillant à éviter toute surchauffe du circuit intégré, ce que l'on obtient en lui laissant le temps de refroidir de temps à autre.

Une fois que l'on est sûr que IC6 est soudé correctement, on pourra passer au montage des 2 CNA, IC7 et IC8. Le PCM1704 étant proposé sous la forme d'un boîtier CMS 20 broches standard (SOIC), sa mise en place est relativement aisée. Les amplificateurs opérationnels IC9 à IC12 pourront, de préférence, être montés sur supports (de bonne qualité).

Les condensateurs faisant partie du filtre analogique, C27 à C38, sont des EMZ à 1%; de forme carrée radiale, leurs broches se trouvent diamétralement opposées. De par leur pas de 7,1 mm, il n'y a pas d'empêchement physique à les remplacer par des condensateurs MKT, mais il faudra se rappeler qu'une tolérance trop importante se traduit par une courbe de filtrage et un comportement en phase moins bons. Les condensateurs EMZ sont dotés d'un repère (une ligne) qui indique laquelle de ses broches est reliée à la couche diélectrique extérieure. Veillez, lors du montage, à ce que soit bien cette broche qui soit reliée à la masse ou à la sortie d'un amplificateur opérationnel. Ceci augmente l'insensibilité du filtre analogique aux parasites. Il en va de même en ce qui concerne les condensateurs axiaux C25 et C26; on fera en sorte que l'anneau soit du côté des sorties des amplifica-

teurs opérationnels IC9 et IC11 respectivement. Les relais eux aussi sont soudés directement sur la platine. N'oubliez pas non plus de monter les ponts de câblage JP2 et JP3; on optera, comme il s'agit de liaisons fixes, pour un conducteur de section plus importante. Terminons par un détail qui a son importance. Il est recommandé, pour une meilleure diaphonie (séparation des canaux) aux fréquences élevées, de blinder la section gauche du filtre analogique par rapport à sa section droite. On disposera, pour ce faire, une languette de tôle de 86 mm de long et de 13 mm (au minimum) de haut entre Re2 et IC12. Cette tôle ira ainsi du bord de la platine jusqu'au niveau de l'interrupteur DIP S4; la sérigraphie marque l'endroit de placement sous la forme d'une ligne pointillée. On enlèvera un peu de vernis sur la platine à l'aide d'un outil acéré au niveau des extrémités de la plaquette de manière à assurer un bon contact électrique entre elle et la surface de cuivre qui fait office de masse présente sur le dessus de la platine. Nous avons constaté, lors de nos mesures, une amélioration de la diaphonie de quelque 12 dB à 20 kHz due à la mise en place de cette languette de blindage.

« MISE EN BOÎTE »

Une fois terminée la réalisation, dans les règles de l'art, des différentes platines et la vérification de l'absence d'erreurs de placement et de soudure faite,

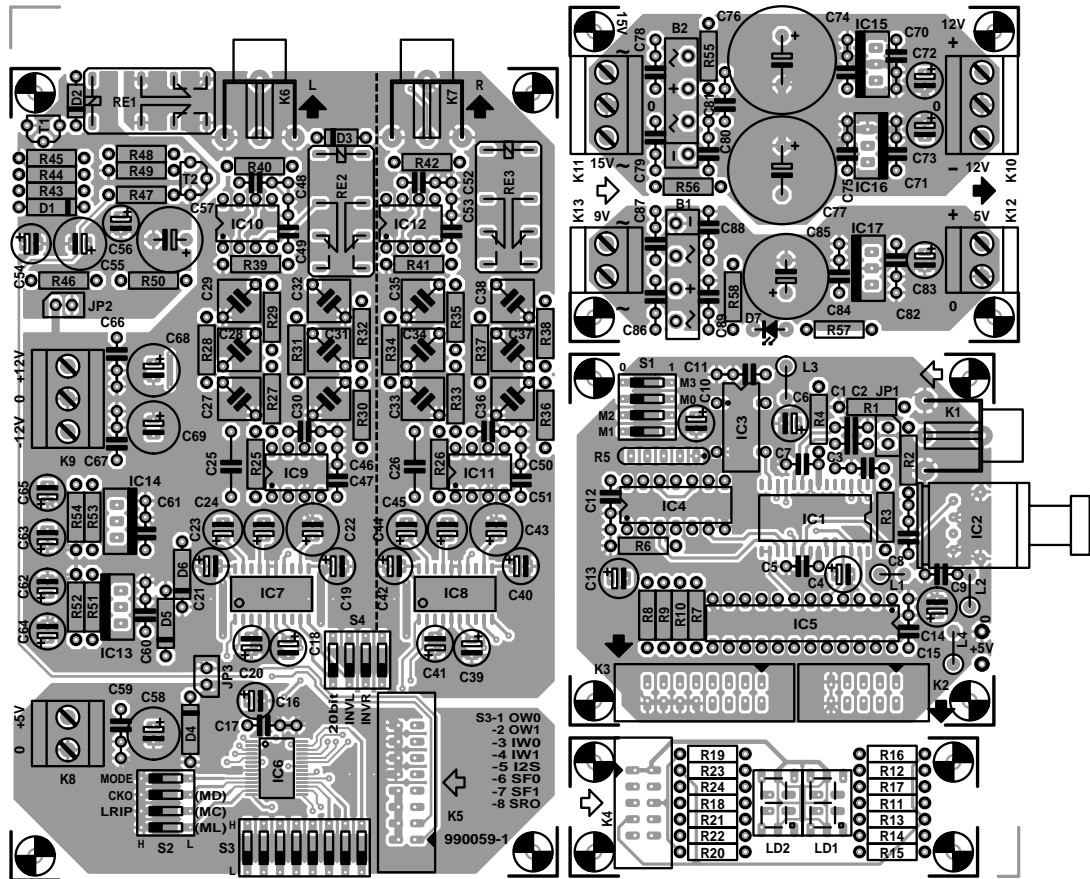


Figure 6. Des lignes de fracture permettent la séparation des 4 platines double face à trous métallisés que requiert cette réalisation. La sérigraphie de l'implantation des composants a été reproduite ici à 85% de sa taille réelle.

Liste des composants

Résistances :

R1 = 75 Ω
 R2 = 220 Ω
 R3 = 470 Ω
 R4, R50 = 4Ω7
 R5 = réseau SIL de 4 · 10 kΩ
 R6 = 10 kΩ
 R7 à R10 = 22 Ω
 R11 à R24 = 820 Ω
 R25, R26 = 2kΩ49/1%
 R27, R30, R33, R36 = 3kΩ57/1%
 R28, R34 = 4kΩ12/1%
 R29, R35 = 3kΩ92/1%
 R31, R37 = 3kΩ65/1%
 R32, R38 = 3kΩ32/1%
 R39, R41, R45, R49 = 1 MΩ
 R40, R42 = 100 Ω
 R43, R44, R47, R48 = 150 kΩ
 R46 = 10 Ω
 R51, R53 = 249 Ω/ 1%
 R52, R54 = 750 Ω/1%
 R55, R56 = 3Ω3
 R57 = 3kΩ9
 R58 = 1Ω5

Condensateurs :

C1, C2 = 10 nF céramique
 C3 = 68 nF
 C4, C6, C10, C16, C62 à C65, C72, C73, C83 = 10 μF/63 V radial
 C5, C7 = 47 nF céramique
 C8, C9, C11, C12, C15, C17, C46 à C53, C59 à C61, C66, C67, C70, C71, C74, C75, C82, C84 = 100 nF céramique
 C13, C14, C23, C24, C44, C45, C54 = 47 μF/25 V radial

C18 à C21, C39 à C42 = 4μF7/63 V radial
 C22, C43, C58, C68, C69 = 100 μF/25 V radial
 C25, C26 = 47 pF¹ axial (EMZ)
 C27, C33 = 2nF2/1%¹ (EMZ)
 C28, C34 = 4nF7/1%¹ (EMZ)
 C29, C35 = 330 pF/1%¹ (EMZ)
 C30, C36 = 1 nF/1%¹ (EMZ)
 C31, C37 = 1nF5/1%¹ (EMZ)
 C32, C38 = 270 pF/1%¹ (EMZ)
 C55 = 220 μF/25 V radial
 C56 = 1 μF/63 V radial
 C57 = 470 μF/25 V radial
 C76, C77 = 1 000 μF/40 V radial
 C78 à C81, C86 à C89 = 22 nF cér.
 C85 = 1 000 μF/25 V radial

¹ polystyrène/polypropylène

Selfs :

L1 à L4 = 47 μH

Semi-conducteurs :

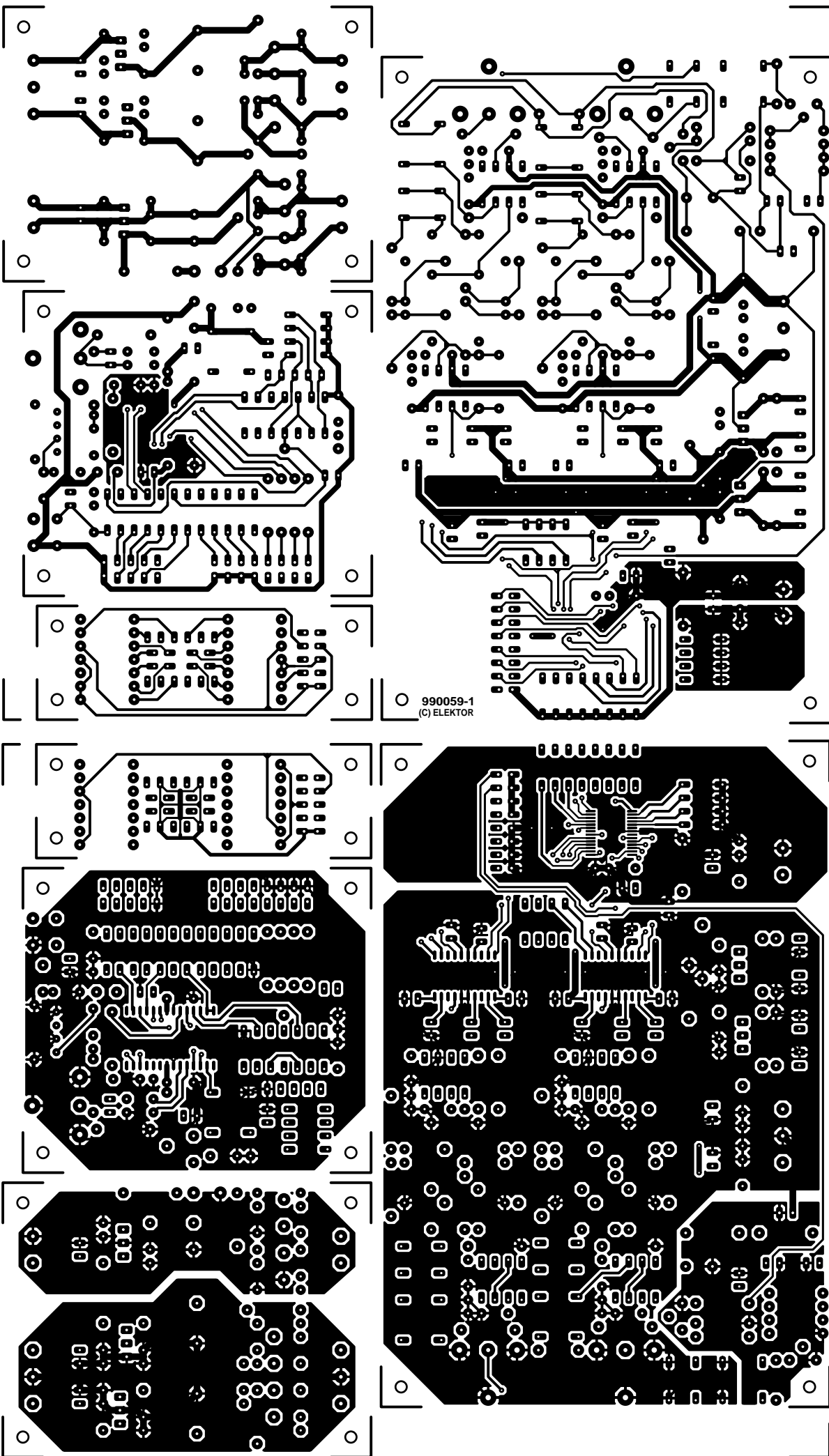
D1 = 1N4001
 D2, D3 = 1N4148
 D4 à D6 = diode zener 5V6/1W3
 D7 = LED à haut rendement rouge
 LD1, LD2 = HDN10750 (Siemens)
 T1, T2 = BC517
 IC1 = CS8414-CS (Crystal Semiconductor)
 IC2 = TORX173 (Toshiba)
 IC3 = 6,144 MHz SG531P (Seiko Epson)
 IC4 = 74HCT32
 IC5 = GAL22V10B-25LP (programmée, **EPS996530-1**)
 IC6 = DF1704E (Burr-Brown)
 IC7, IC8 = PCM1704U (Burr-Brown)

IC9 à IC12 = OPA627AP (Burr-Brown)
 IC13 = LM317 (TO220)
 IC14 = LM337 (TO220)
 IC15 = 7812
 IC16 = 7912
 IC17 = 7805

Divers :

JP1 = embase autosécable mâle à 2 contacts + cavalier
 JP2, JP3 = pont de câblage *
 K1, K6, K7 = embase Cinch encartable (Monacor T-709G par exemple)
 K2 = embase encartable mâle à 2 rangées de 5 contacts
 K4 = embase encartable mâle à 2 rangées de 5 contacts (pour câble plat)
 K3, K5 = embase mâle encartable à 2 rangées de 8 contacts
 K8, K12, K13 = bornier encartable à 2 contacts au pas de 5 mm
 K9 à K11 = bornier encartable à 3 contacts au pas de 5 mm
 S1, S2, S4 = quadruple interrupteur DIP
 S3 = octuple interrupteur DIP
 B1, B2 = B80C1500 vertical
 Re1 à Re3 = relais V23042-A2003-B101, 12 V/600 Ω (Siemens)
 transfo d'alimentation : 2x15 V/4 VA (tel que, par exemple, Block FLD4/15, Hahn BVUI 3020165, Monacor FTR-415), et 2x8 (ou 9) V/4 VA (tel que, par exemple, Block FLD4/8, Hahn BVUI 3020161, Monacor FTR-49 - cf. platine du transformateur ailleurs dans ce numéro)

* cf. texte



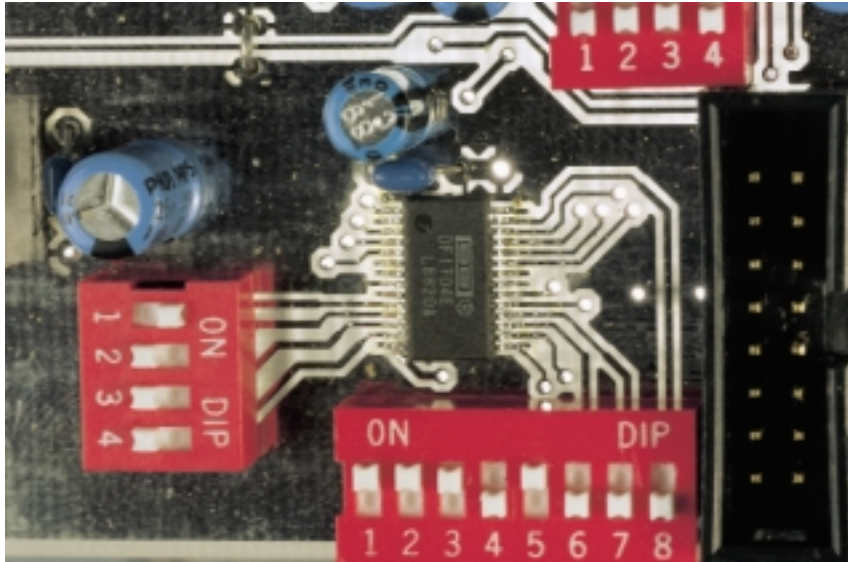


Figure 7. Le montage du DF1704, IC6, requiert une main très ferme et un matériel adéquat.

on pourra en envisager l'interconnexion pour réaliser notre audio-DAC 2000 qu'il faudra bien entendu habiller. La solution optimale consiste à utiliser un coffret en métal qui aura de préférence les dimensions et l'aspect du reste des éléments de la chaîne audio où il doit trouver place. Nous avons utilisé, pour notre prototype, un coffret du type UC251/SW de Monacor. Ses dimensions, 435 mm de large, 44 mm de haut seulement et 230 mm de profondeur sont classiques dans le monde de l'audio. Il s'agit, comme l'illustre la **figure 8**, d'un coffret relativement compact qui offre cependant suffisamment de place pour le placement des différentes platines. La disposition adoptée ici est l'une des plus favorables que l'on puisse envisager, aussi conseillons-nous à ceux qui envisageraient de réaliser ce montage d'adopter la disposition suivante : la platine des CNA dans un coin, celle du récepteur juste à côté, la platine de l'alimentation juste devant et le transformateur d'alimentation dans le coin opposé. Nous avons utilisé une platine pour transformateur spéciale, décrite, nous l'avons dit plus haut, dans un article distinct présenté ailleurs dans ce même numéro.

Les seuls éléments montés sur la face avant sont l'interrupteur secteur et l'affichage visualisant la fréquence d'échantillonnage. On peut également y positionner la LED de tension d'alimentation D7, mais cela n'a rien d'impératif vu que les afficheurs LD1 et LD2 servent, implicitement, de dispositif signalant la présence de la tension d'alimentation (marche/arrêt).

La figure 8 montre clairement le câblage des différents éléments qu'il va falloir réaliser. Énumérons, pour éviter tout oubli, les interconnexions à effectuer :

- ◆ il part, de K2 de la platine du récepteur, un câble plat à 10 conducteurs qui arrive sur l'embase K4 située sur le dos de la platine d'affichage.
- ◆ un câble en nappe à 16 conducteurs va de K5 de la platine des CNA à l'embase K3 de la platine du récepteur. Cette liaison fait également office de lignes d'alimentation de la platine du récepteur. Vérifiez bien la position de la broche 1 lors de la connexion !
- ◆ 2 conducteurs partent de K12 de la platine d'alimentation (5 V) pour arriver à K8 de la platine des CNA.
- ◆ 3 conducteurs vont du bornier K10 de la platine d'alimentation (± 12 V) à l'embase K9 de la platine des CNA.

EN GUISE DE CONCLUSION

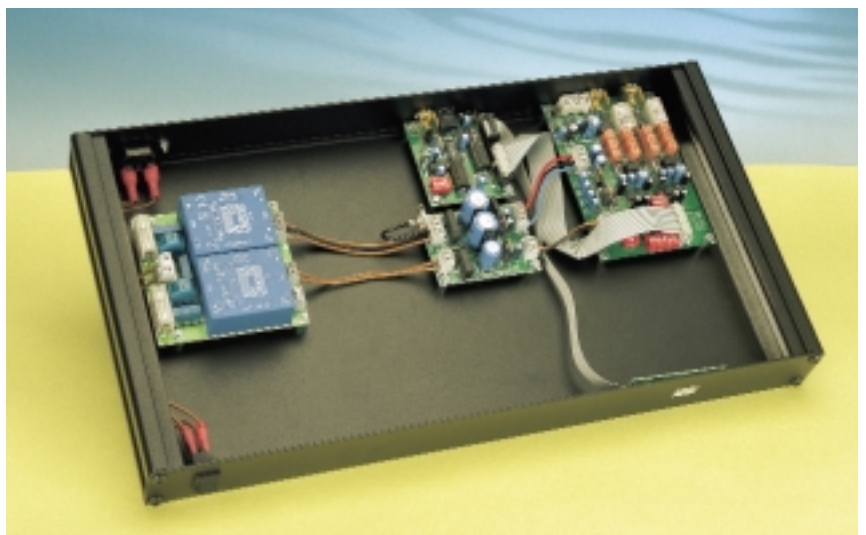
Juger à l'écoute de la qualité d'un

convertisseur A/N n'est pas une sinécure. Il ne faut pas s'attendre à découvrir des différences dramatiques comme cela peut être le cas lors d'une écoute comparative d'enceintes. Les améliorations par rapport aux audio-DAC ayant eu précédemment l'honneur des colonnes d'Elektor sont à chercher au niveau de nuances très subtiles. Nous avons trouvé que notre dernier-né, l'audio-DAC 2000, fournissait un son sensiblement meilleur que toutes les versions existantes dont nous disposons encore, le plus impressionnant étant sa netteté et la finesse de l'image stéréo. Nous vous renvoyons, pour un jugement plus « mesuré », aux résultats de mesure donnés en encadré ci-après. Permettez-nous quelques remarques d'ordre général : la largeur de la bande passante aux fréquences d'échantillonnage de 32, 44,1 et 48 kHz, est exactement égale à la moitié de la fréquence d'échantillonnage vu qu'aux dites fréquences la bande passante du filtre analogique est plus grande que celle du filtre numérique (dont la raideur de pente est très importante). Aux fréquences de 88,2 et 96 kHz, la bande passante dépend du filtre analogique. La distorsion harmonique totale + bruit (THD + N pour *Total Harmonic Distorsion + Noise*) à une fréquence d'échantillonnage de 96 kHz a été mesurée à une bande passante de 22 kHz, vu qu'à une fréquence d'échantillonnage plus faible le filtre de sortie analogique présente une bande passante de 26 kHz. Il est plus facile, dans ces conditions, d'effectuer une comparaison entre les 3 mesures.

(990059-3)

Texte: Sief van Rooij
Technique: Ton Giesberts

Figure 8. Un coup d'oeil au coeur de notre prototype. On constate que le coffret que nous avons utilisé permet une disposition facile des différents composants de cette réalisation.



Caractéristiques techniques :

- ▶ 1 entrée coaxiale + 1 entrée optique
- ▶ travaille à toutes les fréquences d'échantillonnage comprises entre 32 et 96 kHz
- ▶ visualisation de la fréquence d'échantillonnage par 2 chiffres
- ▶ octuple suréchantillonnage (8x)
- ▶ filtre numérique 24 bits
- ▶ convertisseurs N/A 24 bits
- ▶ désaccentuation numérique
- ▶ filtre de sortie analogique du 3ème ordre commutable
- ▶ tensions d'alimentation séparées pour les parties numérique et analogique

Spécifications

tension d'entrée nominale de l'entrée coaxiale : $0,5 V_{CC}/75 \Omega$
 tension de sortie nominale : $2,1 V_{eff}$
 Plage de fréquences (-3 dB) : $0 \text{ à } f_s/2$ ($f_s = 32/44,1/48 \text{ kHz}$)
 $0 \text{ à } 42 \text{ kHz}$ ($f_s = 88,2/96 \text{ kHz}$)
 amplitude 20 kHz : $-0,94 \text{ dB}$ ($f_s = 44,1/48 \text{ kHz}$)
 $-0,66 \text{ dB}$ ($f_s = 88,2/96 \text{ kHz}$)
 bande passante des filtres analogiques : 26 kHz
 (Butterworth pour $f_s = 32/44,1/48 \text{ kHz}$)
 42 kHz (Bessel pour $f_s = 88,2/96 \text{ kHz}$)
 Impédance de sortie : 100Ω
 Rapport signal/bruit : $> 114 \text{ dBA}$
 THD+N (1 kHz, B = 80 kHz) : $0,0016 \%$ ($44,1 \text{ kHz}/16 \text{ bits}$)
 $0,001 \%$ ($48 \text{ kHz}/24 \text{ bits}$)
 $0,0008 \%$ ($96 \text{ kHz}/24 \text{ bit}$, B = 22 kHz)
 Distorsion d'intermodulation (IMD) (60 Hz/7 kHz, 0 dB) : $0,0035 \%$
 Dérive en linéarité : $< 0,5 \text{ dB}/-90 \text{ dB}$ (d'après fiche de caractéristiques)
 $0,2 \text{ dB}/-110 \text{ dB}$ (mesurée)
 Diaphonie (1 kHz) : $> 115 \text{ dB}$
 Plage de dynamique : $> 100 \text{ dB}$

Les mesures ont été effectuées avec le paramétrage suivant des interrupteurs DIP.

S1	S2	S3	S4
-1 off	-1 on	-1 on	-1 off
-2 of	-2 off	-2 on	-2 off
-3 off	-3 off	-3 on	-3 off
-4 on	-4 off	-4 off	-4 N.C.
		-5 on	
		-6 off	
		-7 off	
		-8 off	

Courbes de mesure

Nous vous proposons, pour que vous ayez le maximum d'informations, en plus d'un bulletin chiffré, quelques courbes de mesure de l'audio-DAC 2000 commentées.

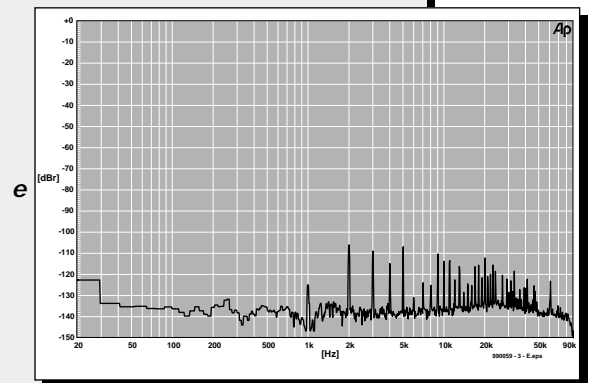
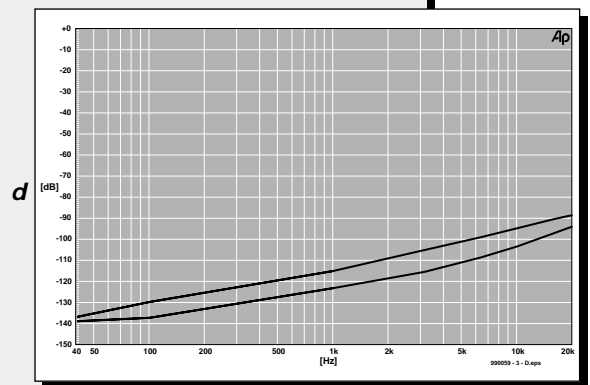
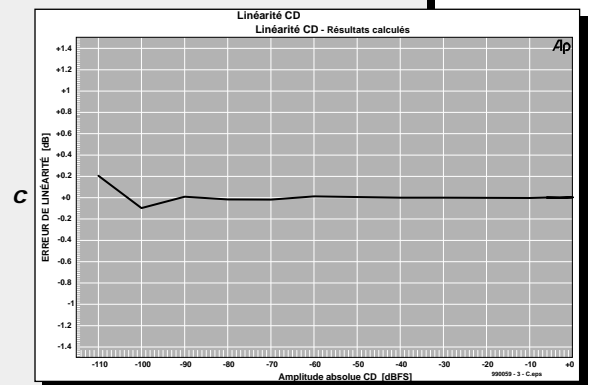
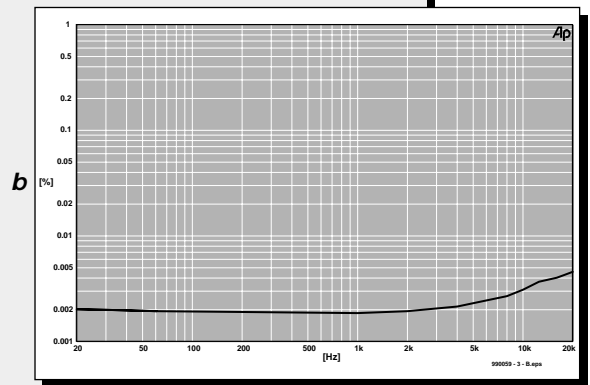
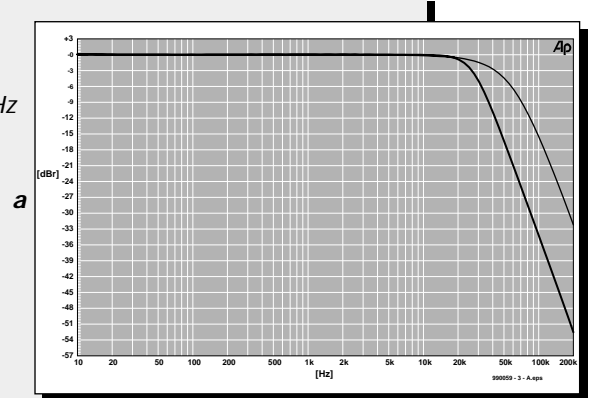
La courbe **a** montre la réponse en fréquence des filtres analogiques de sortie. Leur mesure se fait par l'application d'un courant au niveau des convertisseurs courant/tension, ce qui implique également une prise en compte du filtrage effectué par les condensateurs C25 et C26.

La courbe **b** montre le THD+N à pleine modulation. Cette mesure a été faite dans une situation pratique, par le biais d'un CD de test (16 bits/44,1 kHz). La croissance de la distorsion au-delà de 3 kHz est étonnamment faible, ce facteur restant inférieur à 0,005 % jusqu'à plus de 20 kHz. Aux fréquences audio élevées, la vitesse des CNA joue bien entendu un rôle certain.

La courbe **c** illustre la linéarité des CNA. Pas de remarque à faire. Le balayage en amplitude a été fait à l'aide de signaux de test de 400 Hz à activation (dither); en l'absence d'activation, il ne serait pas possible d'effectuer des mesures 16 bits jusqu'à -110 dB.

La courbe **d** montre la diaphonie entre les canaux à partir de 40 Hz. En-deçà de cette valeur on ne mesure que du bruit de fond. La diaphonie est excellente, elle est toujours, dans les 2 cas, supérieure à 88 dB et ce même à 20 kHz. Ces valeurs ont été relevées blindage entre les canaux en place (cf. texte).

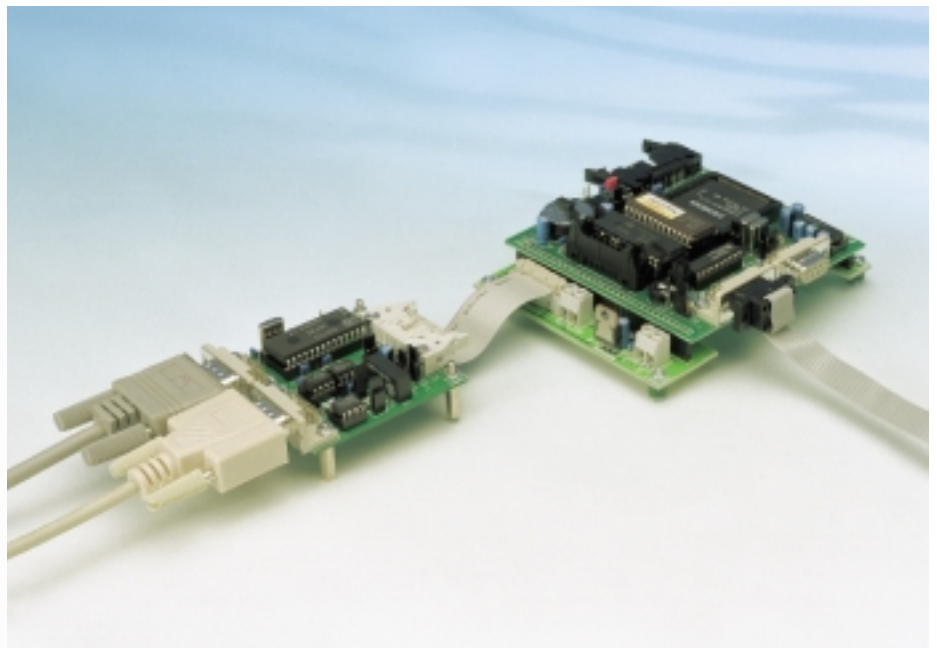
La courbe **e** montre, pour finir, le spectre de fréquences à 1 kHz à pleine modulation et à une fréquence d'échantillonnage de 48 kHz/24 bits. Comme on le voit, toutes les harmoniques se trouvent bien en-deçà de -100 dB.



pratique du bus CAN

en BASIC 537

Celui qui, pour la première fois, est aux prises avec un bus CAN désire avant tout assister à une transmission réussie. N'empêche, il faut au préalable que deux systèmes à microcontrôleur soient connectés



correctement au contrôleur CAN, qu'un bus d'échange de données ait été câblé et que deux logiciels aient pu entrer en relation. Quand enfin les données circulent sur le bus, la suite est un jeu d'enfant. Cet article n'a d'autre ambition que de faciliter autant que faire se peut les premiers pas de l'utilisateur.

L'interface pour bus CAN présentée dans le magazine *Elektor* de novembre 99, on peut sans difficulté la faire travailler en BASIC 537, une version améliorée et spécialement adaptée au 82C537 du célèbre BASIC MCS51 d'Intel, familier à la plupart de nos lecteurs. Ce langage et son interpréteur ont été conçus à l'origine pour le microcontrôleur 8052AH-BASIC, lequel n'est d'ailleurs plus fabriqué actuellement, mais ils poursuivent néanmoins leur carrière grâce à une version en ROM externe destinée à d'autres contrôleurs de la famille 80xx. À ce sujet, vous pouvez utilement vous reporter aux articles sur l'automate-

gigogne 80C32-BASIC parus dans *Elektor* de février et mars 1998. Les nouvelles variantes du BASIC pour les microcontrôleurs 80C535 et 80C537, nous y reviendrons très prochainement, d'ailleurs. Une platine unique à l'usage du microcontrôleur 80C537 a déjà été présentée dans le magazine de juin 97 sous l'appellation de mono-carte. Mais plus spécifiquement pour les applications pratiques en relation avec l'interface pour le bus CAN, nous vous proposons ici une mono-carte « 537 Lite », encore plus compacte et moins chère que la précédente, telle que vous l'avez aperçue en photographie dans l'en-tête du présent article.

MATÉRIEL

Pour assurer une liaison aisée entre la mono-carte « 537 Lite » équipée de l'EPROM porteuse du BASIC 537 et la platine d'interface CAN, nous avons dessiné un adaptateur sur circuit imprimé. Vous en trouverez le schéma à la figure 1, tandis que la morphologie des pistes et l'organisation de l'implantation des composants se trouvent à la figure 2. La photo vous le fait découvrir, la platine du 537 se branche directement sur l'adaptateur, résolvant ainsi la question du câblage. L'autre liaison, celle avec la carte d'interface CAN, sera réalisée simplement à l'aide d'un câble en nappe, comme la photo en début d'article le laissait présager.

Reste à trouver une solution expéditive pour l'alimenter. À cet effet, un régulateur de +5 V, flanqué d'une diode de protection contre l'inversion de polarité à l'entrée de la platine d'adaptation, assure en même temps l'alimentation des deux autres cartes. Il n'y a qu'à connecter à K2 un module secteur bon marché –un courant de 300 mA sera plus que suffisant– dont la tension de sortie continue non stabilisée s'élève entre 9 et 12 V. Si l'on dispose déjà d'un 5 V stabilisé dans les environs, on peut laisser de côté IC1, D1, C1 et C2, il suffit alors de relier cette source aux bornes correspondantes de K1 sur l'adaptateur.

Par raison d'économie, la platine de l'adaptateur est nettement moins spacieuse (encore) que celle du 537 à laquelle elle se branche. Si l'on divise en deux la platine de l'adaptateur suivant la ligne représentée en sérigraphie, puis qu'on fixe les deux sections à bonne distance sur un châssis, une tôle d'aluminium par exemple, il sera bien facile d'y appliquer la mono-carte « 537 Lite ». C'est à dessein que le signal d'interruption INT2 est relié par fil entre la broche 12 de K3 et la broche 32 de K6. Pour vous en convaincre, observez la photo de la carte à 537 avec les deux sections disjointes de l'adaptateur. Chacune de ces parties est pourvue en conséquence de points de connexion et d'ailleurs on voit bien sur la sérigraphie de la figure 2 un pont de câblage dessiné à cet endroit.

LA COMMANDE EN BASIC

Il nous suffit, pour commander la carte d'interface CAN, d'un programme dont l'unique tâche consiste à dialoguer avec une pile de registres du contrôleur CAN, situés dans un domaine qui commence à l'adresse F000_H. Pour accéder aux adresses dans les domaines de RAM externe et des périphériques, le BASIC 537 fait appel à l'opérateur XBY.

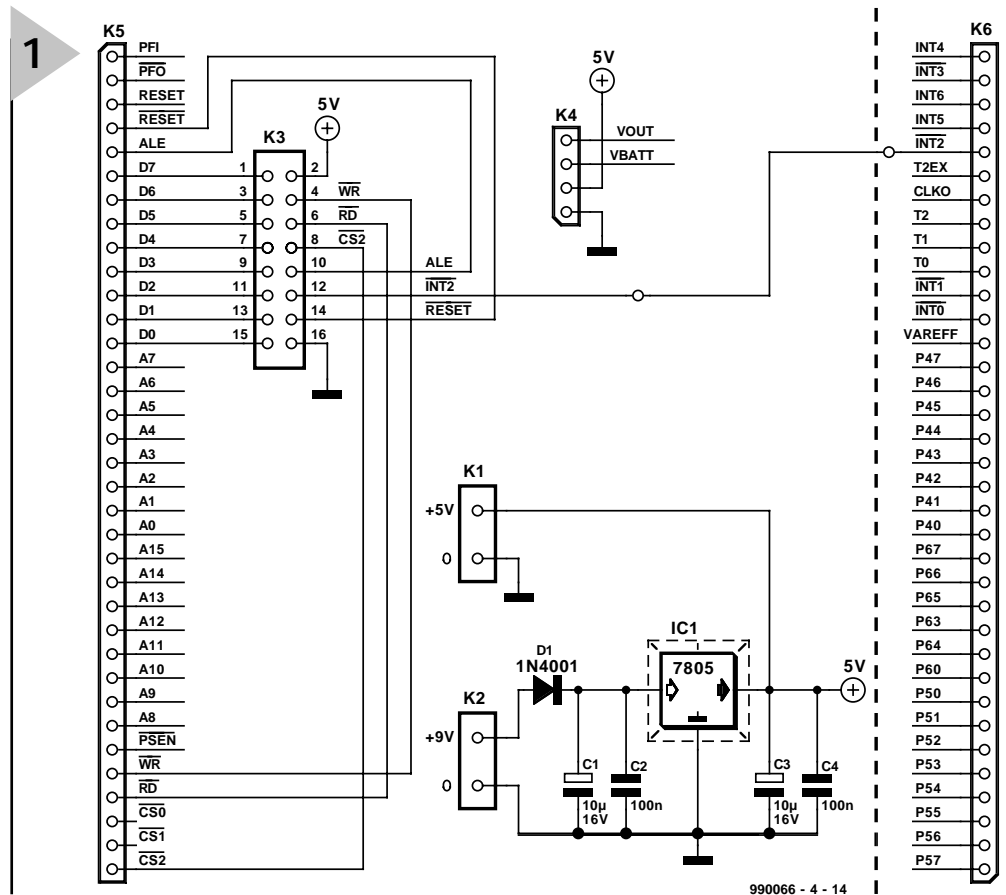


Figure 1. Le schéma de la platine d'adaptation destinée à faciliter la liaison entre la mono-carte 537 Lite et l'interface de bus CAN.

En guise d'entraînement, démarrons avec un système simplifié, une transmission de données entre deux ensembles équipés du 80C537. L'environnement indispensable y sera recréé par le logiciel. Le débit est fixé à 20 kbit/s. Les messages sont transmis sans bit RTR, il n'y aura donc pas de demande de réponse. Les deux ensembles doivent remplir les tâches suivantes.

L'ensemble 1 expédie sans cesse des messages portant l'identité 300 et dans lesquels on a inséré huit octets. Les données proviennent des huit premiers canaux du convertisseur A/N. Il y a donc en permanence des mesures effectuées sur huit entrées analogiques et le ou les autres ensembles connectés au bus peuvent recevoir et utiliser ces données.

L'ensemble 2 réceptionne toutes les données qui circulent sur le bus et les répercute vers le PC, à travers l'interface RS232. On dispose ainsi d'un superviseur simple du CAN, apte à observer le trafic dans son ensemble. La figure 3 nous expose cet assemblage sous forme de bloc-diagramme. Les liaisons entre ensembles ne réclament pas de câble spécial. Lors des premières expériences sur le banc de laboratoire, là où les distances sont toujours inférieures au mètre, un bout de câble à

deux conducteurs suffit à relier les bornes 4 et 8 de la prise CAN aux homologues de l'autre ensemble. Sur d'aussi courtes distances, on ne remarque d'ailleurs aucune différence si les terminaisons sont bouclées ou non sur leur impédance caractéristique.

LE LOGICIEL D'ÉMISSION ET LE TEST

Le listage 1 reprend le programme d'émission destiné au contrôleur 1. Le système à 80537 interpellé le contrôleur CAN SJA 1000 à l'adresse 0F000_H. C'est la ligne 95 qui fixe l'adresse de départ (BA=0F000_H). Sur un autre système, il n'y aurait donc qu'à changer la valeur attribuée à BA. La séquence d'initialisation, nous l'avons décrite dans l'article sur le matériel CAN. Aux lignes 110 et 200 s'opère la vérification de la programmation des registres. Le logiciel patiente ici jusqu'à ce qu'il soit certain que tous les bits de ces registres se trouvent dans l'état souhaité. S'il ne détecte pas l'autre contrôleur sur le bus ou s'il n'obtient pas de réponse comme prévu, le programme reste coincé ici. En cas de réussite, on peut lire les rapports suivants :

Reset OK

Init OK

Il ne reste plus alors qu'à réaliser l'initialisation jusqu'à la ligne 200. Une première vérification consiste à observer le signal rectangulaire présent au point

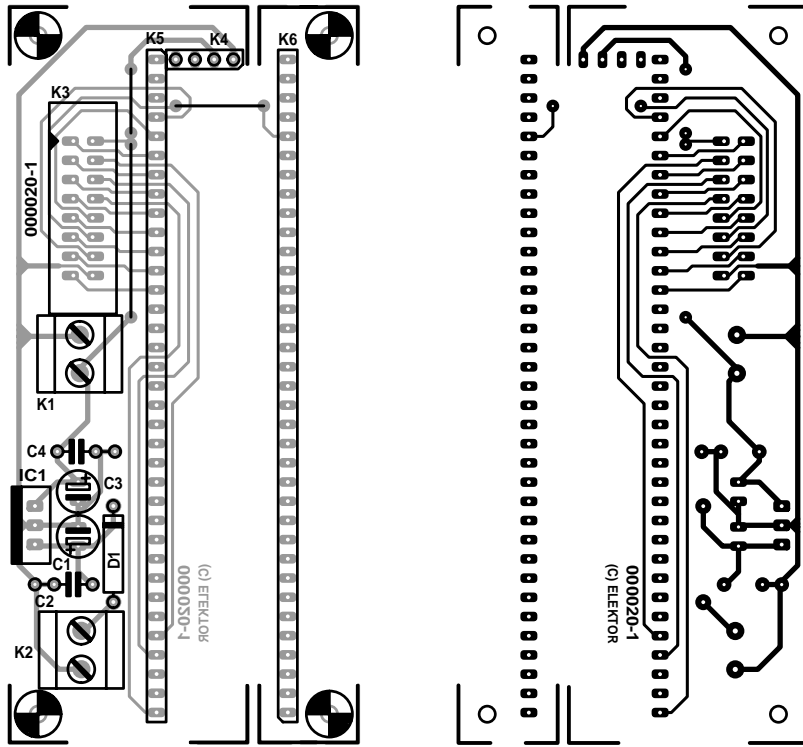


Figure 2. Le tracé des pistes et l'implantation des composants de l'adaptateur.

- Liste des composants**
- Condensateurs :
 C1, C3 = 10 μ F/16 V (vertical)
 C2, C4 = 100 nF (céramique)
- Semi-conducteurs :
 D1 = 1N4001
 IC1 = 7805
- Divers :
 K1, K2 = bornier encartable à 2 contacts au pas de 5 mm
 K3 = embase HE-10 à 2 rangées de 8 contacts
 K4 = embase autosécable à 1 rangée de 4 contacts
 K5, K6 = embase autosécable droite à 1 rangée de 35 contacts

de mesure de la platine du contrôleur. Alors qu'avant l'initialisation on apercevait une fréquence de 8 MHz, on ne doit plus y trouver, à présent, que 2 MHz. Si c'est effectivement le cas, on peut déjà se dire que le contrôleur est convenablement piloté. En avant pour le chargement du programme complet et le démarrage. À cet instant, l'électronicien confirmé aura naturellement l'envie de prendre l'oscilloscope et d'examiner les données. Avant que ne soit obtenue la liaison

avec un autre ensemble, on détecte déjà un signal sur les conducteurs de données. À la suite d'une mise à zéro matérielle et sans initialisation, on peut mesurer sur chacun des fils un niveau de repos voisin de 2,5 V. Une fois le programme d'émission démarré, on reconnaît les données sous la forme de signaux rectangulaires d'une amplitude de 1 V. Les états les plus brefs durent 50 μ s, d'où la conclusion que le débit de 20 kbit/s est respecté. Pourtant, on s'aperçoit que le transfert répétitif

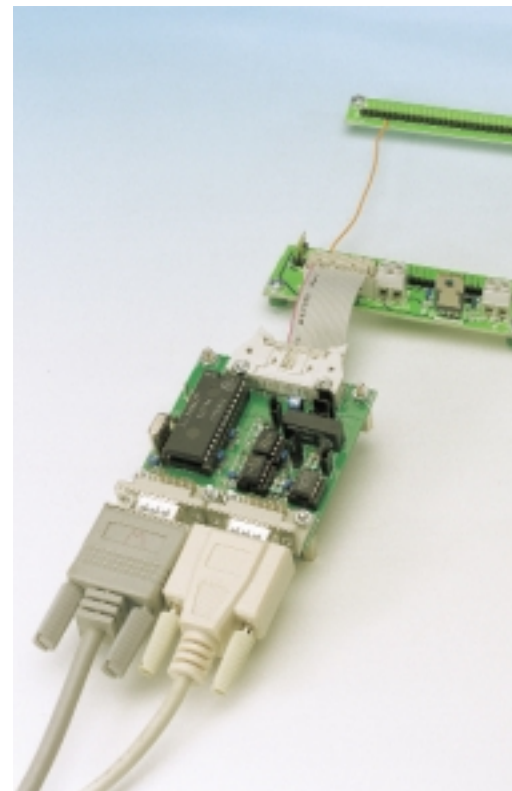
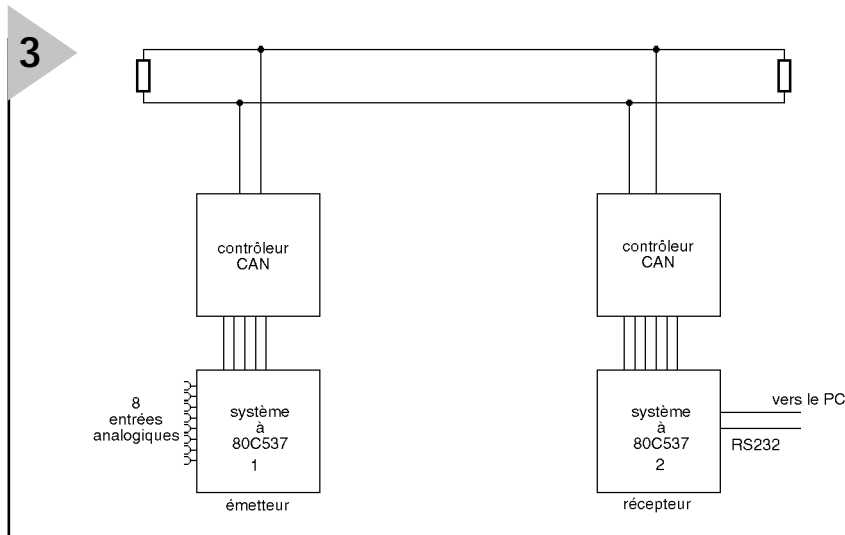


Figure 3. Diagramme fonctionnel de la liaison par bus CAN entre deux systèmes à 80C537 programmés en BASIC.



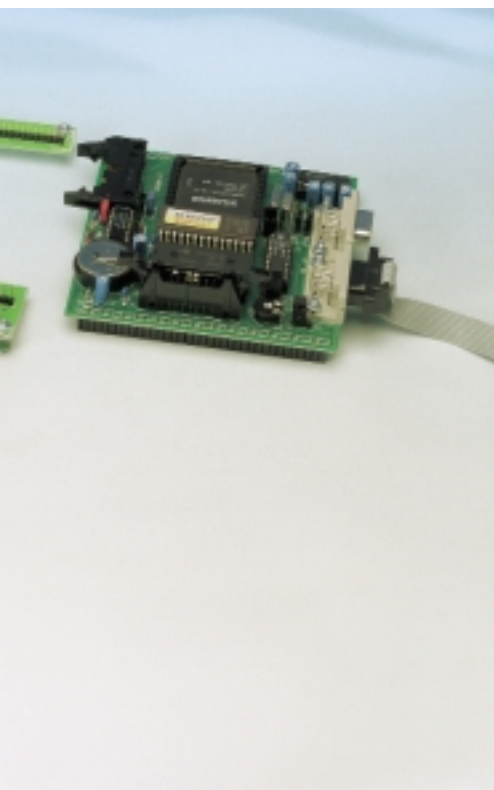
de données est interrompu par de brèves pauses de 2 ms, au lieu d'assister au passage de courts paquets de données. Pas de panique, c'est le comportement normal du contrôleur aussi longtemps qu'il n'a pas salué de collègue sur le bus. Mais voilà, il ne suffit pas de raccorder au câble un second contrôleur, encore faut-il l'initialiser. Le signal d'attente de la station émettrice à la recherche d'un partenaire se poursuit même lorsque le programme en BASIC est interrompu.

LE LOGICIEL DE RÉCEPTION

Le moment est donc venu de nous intéresser au logiciel de réception, pour un système composé de deux ensembles, tel qu'il est consigné dans le listage 2. Nous pouvons y relever

que l'initialisation ne diffère pas de celle d'un ensemble émetteur. Aussitôt qu'elle est terminée, la mention « Init OK » apparaît à l'écran et l'ensemble émetteur entame son fonctionnement normal. Dès maintenant, nous pouvons observer de petits groupes de données, d'une longueur d'à peu près 5 ms, déambuler sur le bus. Enfin, le bus CAN opère de la manière prévue, des paquets de données s'écoulent, peu importe que quelqu'un les lise ou pas.

Entrons dans le logiciel de réception proprement dit, à la ligne 500, où il espère être averti de l'arrivée d'un message, par l'intermédiaire du bit 0 du registre d'état du contrôleur. Dès qu'un bloc de données est arrivé, le logiciel peut obtenir la lecture d'un



total de 10 octets de la part du contrôleur. Les deux premiers renferment l'identité, laquelle est, à la ligne 570, reconstruite à partir de ces deux octets et affichée. Comme on s'y attendait, c'est bien l'identité 300, celle que nous avons insérée dans le logiciel d'émission.

Les données utiles sont extraites par une boucle et affichées lors de l'exécution de la ligne 610. On y trouve en permanence les valeurs de mesure des huit entrées analogiques du premier ensemble. La figure 4 vous montre, dans une fenêtre du terminal, les données reçues.

ENFIN À TROIS EN BUS

À vrai dire, tout ce que nous avons fait jusqu'à présent, nous aurions très bien pu le réaliser par simple interface

Programme d'émission CAN1.BAS (Listage 1)

```

90   REM Init CAN Controller
95   BA=0F000H
100  XBY(BA+00H)=01H : REM Reset Mode
110  IF (XBY(BA+00H).AND.1)<>1 THEN GOTO 110
111  PRINT "Reset OK"
120  XBY(BA+1FH)=43H : REM CDR, 2 MHz
130  XBY(BA+04H)=0 : REM ACR
140  XBY(BA+05H)=0FFH : REM AMR, Acceptance Mask, all
150  XBY(BA+06H)=53H : REM BTRO, 20 Kbit/s*
160  XBY(BA+07H)=2FH : REM BTR1
170  XBY(BA+08H)=1AH : REM OCR;
180  XBY(BA+01H)=0EH : REM CMR, end sleep mode
190  XBY(BA+00H)=0 : REM CR, end reset mode
200  IF (XBY(BA+00H).AND.1)>0 THEN GOTO 200
201  PRINT "init ok"
500  REM ***** Main Loop *****
501  REM Send 8 Bytes of AD-Data in message 300
510  FOR N=0 TO 7
520  XBY(BA+0CH+N)=AD(N) : REM fill TB1..TB8
530  NEXT N
540  ID=300 : REM Message Identifier
550  DFL=8 : REM 8 Bytes
560  GOSUB 1000 : REM Send Message
570  FOR T=1 TO 1000 : NEXT T
580  GOTO 500
1000 REM ***** Send CAN Telegram *****
1010 IF (XBY(BA+02H).AND.4)=0 THEN GOTO 1010 : REM SR
1020 XBY(BA+0AH)=INT(ID/8) : REM IDT1
1030 XBY(BA+0BH)=(ID-8*INT(ID/8))*32+DFL : REM IDT2
1040 XBY(BA+01H)=0DH : REM CMR, start transmission
1050 RETURN

```

Programme de réception CAN2.BAS (Listage 2)

```

90   REM Init CAN Controller
95   BA=0F000H
100  XBY(BA+00H)=01H : REM Reset Mode
110  IF (XBY(BA+00H).AND.1)<>1 THEN GOTO 110
111  PRINT "Reset OK"
120  XBY(BA+1FH)=43H : REM CDR, 2 MHz
130  XBY(BA+04H)=0 : REM ACR
140  XBY(BA+05H)=0FFH : REM AMR, Acceptance Mask, all
150  XBY(BA+06H)=53H : REM BTRO, 20 Kbit/s*
160  XBY(BA+07H)=2FH : REM BTR1
170  XBY(BA+08H)=1AH : REM OCR;
180  XBY(BA+01H)=0EH : REM CMR, end sleep mode
190  XBY(BA+00H)=0 : REM CR, end reset mode
200  IF (XBY(BA+00H).AND.1)>0 THEN GOTO 200
201  PRINT "Init OK"
500  REM ***** Receiver Main Loop *****
510  SR=XBY(BA+02H) : REM Status Register
520  REM Error Detection and Clear Data Overrun
530  IF (SR.AND.2)=2 THEN XBY(BA+01H)=8 : Goto 510
540  REM Get Receive Status
550  IF (SR.AND.1)=0 THEN goto 510
560  REM Read received message
570  ID=XBY(BA+14H)*8+INT(XBY(BA+15H)/32) : PRINT ID
580  DFL=XBY(BA+15H).AND.15 : REM Data Length
590  RTR=(XBY(0FE15H).AND.16)/16 : REM RTR not used
600  FOR N=0 TO 7
610  PRINT N , XBY(BA+16H+N)
620  NEXT N
630  XBY(BA+01H)=0CH : REM Release Receive Buffer
640  GOTO 510

```

Programme d'émission/réception CAN3.BAS sans initialisation (Listage 3)

```

500 REM ***** Main Loop *****
505 REM ***** Receiver *****
510 SR=XBY(BA+02H) : REM Status Register
520 REM Error Detection and Clear Data Overrun
530 IF (SR.AND.2)=2 THEN XBY(BA+01H)=8 : GOTO 510
550 IF (SR.AND.1)=0 THEN GOTO 510
560 REM Read received message
570 ID=XBY(BA+14H)*8+INT(XBY(BA+15H)/32) : Print ID
580 DFL=XBY(BA+15H).AND.15 : REM Data Length
590 RTR=(XBY(0FE15H).AND.16)/16 : REM RTR not used
600 IF ID<>300 THEN GOTO 660
610 PORT=0
620 IF XBY(BA+16H+0)>100 THEN PORT=PORT+1
630 IF XBY(BA+16H+1)>100 THEN PORT=PORT+2
640 IF XBY(BA+16H+2)>100 THEN PORT=PORT+4

```

```

650 WRSFR OE8H,PORT : REM Port 4 Output
660 XBY(BA+01H)=0CH : REM Release Receive Buffer
800 REM ***** Send AD-Data *****
810 FOR N=0 TO 7
820 XBY(BA+0CH+N)=AD(N) : REM fill TB1..TB8
830 NEXT N
840 ID=500 : REM Message Identifier
850 DFL=8 : REM 8 Bytes
860 GOSUB 1000 : REM Send Message
870 FOR T=1 TO 1000 : NEXT T
880 GOTO 500
1000 REM ***** Send CAN Telegram *****
1010 IF (XBY(BA+02H).AND.4)=0 THEN GOTO 1010 : REM SR
1020 XBY(BA+0AH)=INT(ID/8) : REM IDT1
1030 XBY(BA+0BH)=(ID-8*INT(ID/8))*32+DFL : REM IDT2
1040 XBY(BA+01H)=ODH : REM CMR, Start Transmission
1050 RETURN

```

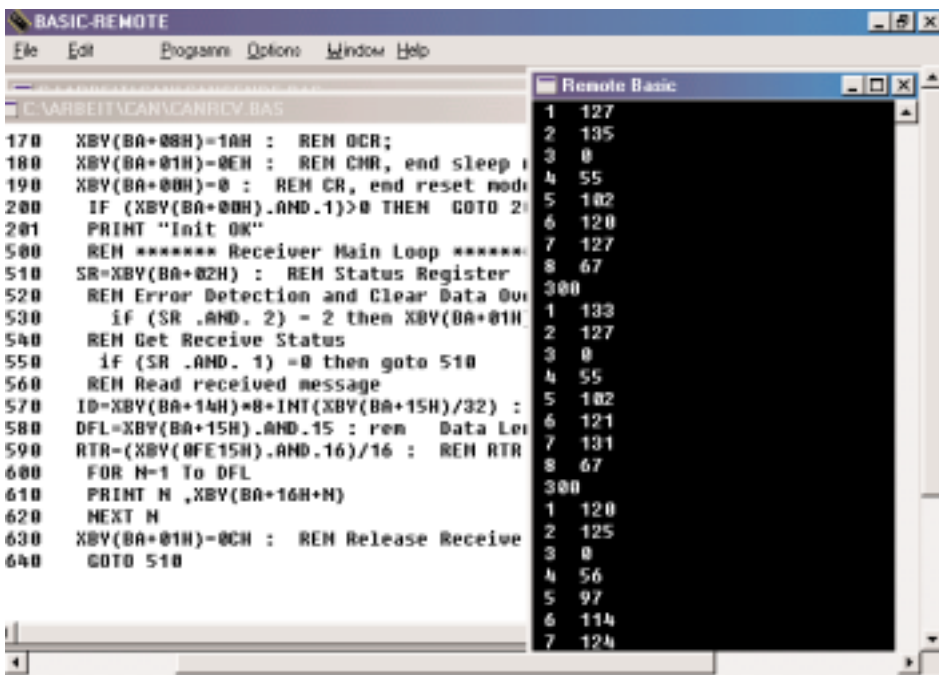


Figure 4. Dans la fenêtre du terminal sous BASIC 537, les données réceptionnées.

RS 232. Là où le bus CAN déploie son savoir-faire, c'est quand il parvient à réunir plus de deux participants. L'occasion pour nous de faire intervenir un troisième larron. Exactement ce que se propose de faire le logiciel CAN3.BAS, repris, sans la séquence d'initialisation, dans le listage 3.

Il réceptionne tous les messages, mais ne traite que ceux porteurs de l'identité 300. Les trois premières valeurs de mesure transmises y sont comparées à des limites définies et ont pour effet de modifier, en cas de dépassement, les niveaux de trois broches du port P4. Après avoir traité les informations, il répond encore par un autre message, identifié 500, qui entraîne une nouvelle séance de mesure des canaux A/N et la transmission des résultats. Quand le troisième système est connecté au bus, le deuxième fournit également au terminal les informations véhiculées sous l'identité 500, comme on le voit sur la figure 5.

990066-4

Texte: Ernst Krempelsauer
Technique: Hans Bonekamp

Référence.

Les trois listages dont nous avons discuté au cours de cet article sont disponibles en téléchargement sur le site Elektor (www.elektor.presse.fr).

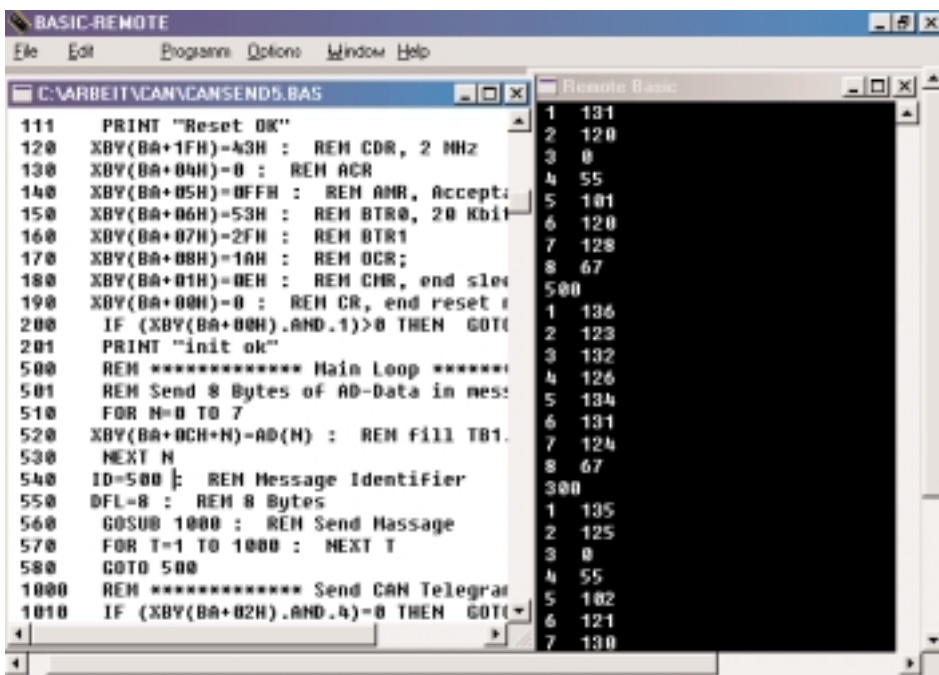


Figure 5. Après réception, les messages identifiés 300 et 500.

électronique en ligne

faites votre radio à cristal

retour aux origines

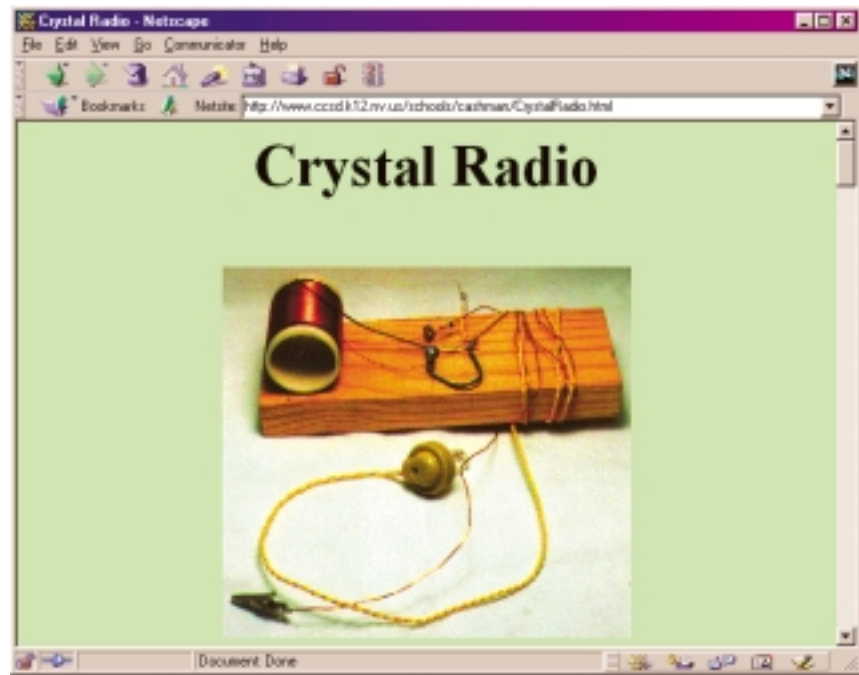
Tous ceux qui ont un minimum de connaissances en électronique savent de quoi il retourne lorsque l'on parle de radio à cristal : le premier modèle de récepteur et le plus simple qui soit, ne comportant rien de plus qu'une diode (le cristal), une antenne, une bobine et un écouteur. On trouve, sur Internet, de nombreux exemples de récepteurs à cristal, certains d'entre eux accompagnés de descriptions exhaustives et de photos fort aguichantes.

Il est tout à fait normal, dans notre société « haute technologie » hypermoderne, de trouver, dans de nombreux appareils, une électronique complexe. Et pourtant, à tout prendre, l'électronique n'existe en fait que depuis peu. Nombre d'amateurs d'électronique se souviendront sans doute, avec un rien de nostalgie, leurs premières expériences d'électronique, la réalisation d'un récepteur simple dont le seul élément actif était une diode.

Les nombreux sites traitant du sujet « radio à cristal » nous donnent à penser qu'il ne s'agit pas uniquement de nostalgie.

Un tour sur Internet nous permet de découvrir différentes réalisations de radios à cristal simples. L'origine en est souvent des établissements scolaires, mais nombre d'amateurs proposent également une information fort intéressante. Quelques adresses :

Building a basic AM broadcast crystal set
(www.midnightscience.com/project.html)



Crystal Radio

(www.ccsd.k12.nv.us/schools/cashman/CrystalRadio.html)

Crystal radio project

(www.datasync.com/~ignatz/electro/crystal/crystal.htm)

La complexité des projets est variable elle aussi. Le vrai récepteur à cristal ne comporte, en tant qu'élément actif, rien de plus qu'une diode au germanium, mais nous avons également trouvé des réalisations complexes utilisant des transistors voire des circuits intégrés. En voici 3 exemples :

High power crystal set

(<http://hibp.ecse.rpi.edu/~john/xtal.html>)

a "convertible" crystal radio

(www.glynn.k12.ga.us/~opool/XTAL/radio.htm)

short wave radio project

(www.datasync.com/~ignatz/electro/receive/receive.htm)

Nous trouvons, chez **Radio Winkles**, à l'adresse :

(www.antiqueradio.org/econmain.htm)

l'exemple d'une description claire de la réalisation d'un récepteur O.C. à cristal baptisé « EconOceanic ». Il permet la réception des O.M. et de la plupart des O.C. comprises entre 1,7 et 17 MHz.

Si vous voulez en savoir plus au sujet des différents concepts pouvant être

mis en oeuvre pour la réalisation de récepteurs à cristal, nous vous renvoyons à la page

« **Crystal set design** ». Cela commence par un simple détecteur à diode et se termine avec un projet à double réseau. L'adresse ? :

(www.webex.net/~skywaves/xtalset102/xtalset102.htm)

On trouve également un certain nombre de clubs d'amateurs de radio à cristal sur Internet. Rendez-vous chez **Xtal Set Society**

(www.midnightscience.com/) et

au **Crystal Set radio club**

(<http://clubs.yahoo.com/clubs/theystalsetradioclub/>),

ces 2 clubs ne s'intéressant qu'à la réalisation et à l'expérimentation de récepteurs radio simples.

Al Klase, un électronicien professionnel nous propose sur son site quelques montages faits main fort attrayants. Appréciez les schémas et les photos !

(www.webex.net/~skywaves/HP002/HP-002.htm)

On trouve également chez **Scott's Crystal Radio's**, à l'adresse :

(<http://members.aol.com/scottswim/>)

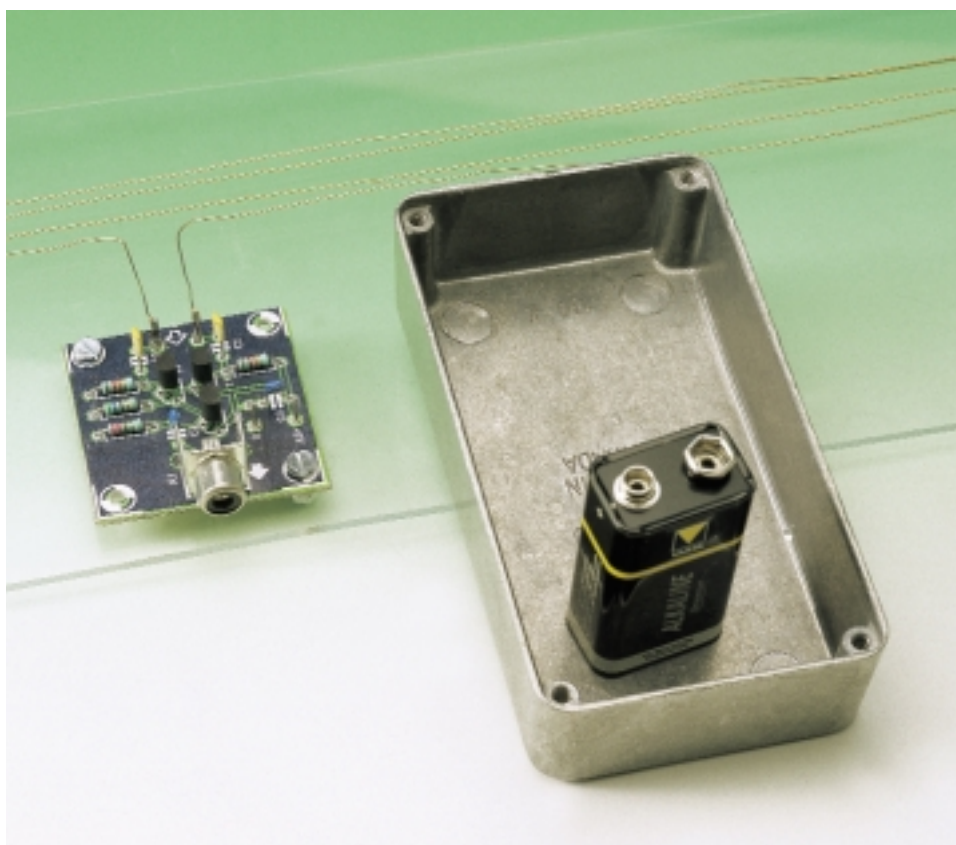
une belle collection de postes à cristal anciens pour ne pas dire antiques.

(005005)

antenne-cadre

pour récepteur mondial

En réception ondes courtes, rien de tel qu'une antenne filaire longue, mais tout un chacun n'a pas nécessairement la place pour en installer une. Alors, l'antenne active décrite ici offre une solution de rechange remarquable. Elle est simple à construire, fournit d'excellentes prestations et en plus, elle est très bon marché.



Celui qui décide d'utiliser son récepteur portatif à ondes courtes pour écouter les émissions internationales ailleurs qu'en vacances s'apercevra vite des limites de l'antenne télescopique. Le rendement d'un fouet court est très restreint et d'autre part, à la maison, les parasites sont légion. S'intéresse-t-on quelque peu à la réception à longue distance, il faut rechercher une voie plus sérieuse.

Mais laquelle ? Il y a tellement de livres qui regorgent de techniques de captage et l'on dénombre une telle diversité d'antennes. Pour s'y retrouver dans cette forêt, balisons les sentiers au départ des qualités que l'on attend d'une telle antenne.

Tout d'abord, une antenne à ondes courtes ou pour les émissions internationales se doit de posséder une largeur de bande suffisante, question de

couvrir avec un rendement convenable la totalité de la gamme, soit de 3 MHz à 30 MHz, environ. C'est alors indiscutablement la bonne vieille antenne à long fil de cuivre qui représente encore la formule la plus simple. Il existe bien de nombreuses alternatives adéquates, mais leur mise en œuvre est singulièrement plus compliquée. Ce qui les caractérise toutes, c'est qu'elles sont assez volumineuses et donc particulièrement destinées à une installation extérieure. Que faire si la place manque ? Se contenter du brin télescopique ? Peut-être existe-t-il d'autres procédés abordables.

PEUT-ON L'ENCADRER ?

À l'intérieur, un choix intéressant, c'est l'antenne-cadre. Certains, à l'évocation du ce type de capteur, penseront

immédiatement qu'il s'agit là d'un objet qui n'a plus sa place qu'au musée. Or, si l'antenne-cadre a rendu bien des services aux pionniers de la TSF, elle n'a rien perdu de ses attraits. Au contraire, sa forme et ses qualités la destinent tout spécialement à l'usage domestique. Quelques clous suffiront à la fixer, à un chambranle de fenêtre par exemple ; elle ne prend que fort peu de place et demeure presque invisible. Elle présente en outre l'avantage estimable de n'être sensible qu'à la composante magnétique du champ de l'émetteur, ce qui permet de frapper d'ostracisme tout le fatras des perturbations électriques. Voilà pourquoi on parlait déjà aux temps héroïques (1928) de cadre antiparasite.

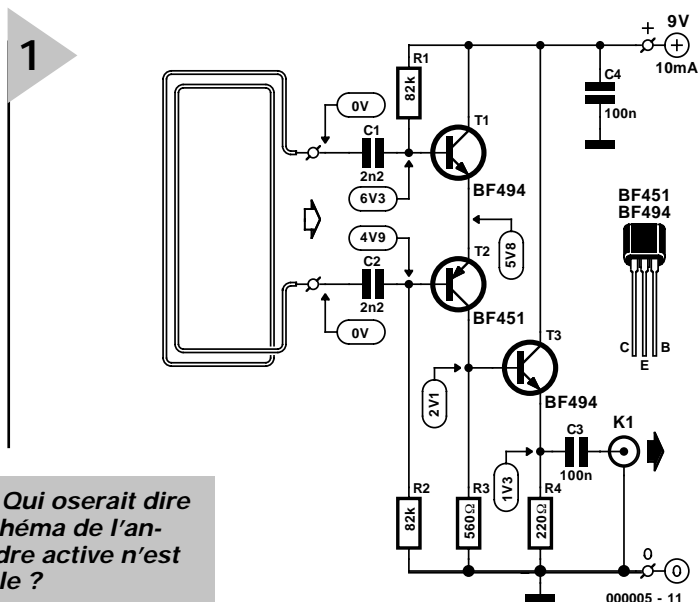


Figure 1. Qui oserait dire que le schéma de l'antenne cadre active n'est pas simple ?

ACTIVE

Du point de vue de la surface effective, l'antenne présentée ici s'apparente à un modèle filaire de longueur moyenne, équivalente à 10 ou 15 m de fil. Cependant, du fait qu'elle est enroulée et placée à l'intérieur, ses caractéristiques sont différentes et son rendement légèrement moindre.

Ce sont précisément les raisons qui conduisent à la rendre « active », en d'autres termes, à la doter d'un pré-amplificateur embarqué. Mais c'est aussi l'occasion rêvée d'accorder son impédance au standard du câble coaxial à 75 Ω.

Pareil amplificateur n'a nul besoin d'être horriblement compliqué, comme en témoigne le schéma de la **figure 1**. On découvre ainsi que l'antenne cadre est reliée aux entrées d'un amplificateur différentiel à composants discrets, basé sur le célèbre transistor à radiofréquence BF494 et son complémentaire PNP, le BF451. Cet étage différentiel apporte à peu près un gain de 10 et jouit d'une bande passante d'un bon 30 MHz, si bien que tout le spectre des ondes courtes s'en trouve couvert.

Le transistor T3 travaille en tampon adaptateur d'impédance. Le signal d'antenne amplifié sort via C3 sur le connecteur K1 et peut immédiatement attaquer un câble coaxial à destination du récepteur.

LA CONSTRUCTION

La partie active de l'antenne cadre peut se monter sur une platine, dont la **figure 2** détaille la disposition des pistes et des composants. Elle n'est pas disponible chez Publitronic Service, il faudra donc la graver personnellement, mais vu le petit nombre de composants, elle ne devrait poser de piège à personne.

Le bobinage, lui aussi, reste la simplicité même. À l'entour d'un cadre d'un mètre sur 1,5 m, en gros, garni d'un clou à chaque angle, on bobine deux à quatre spires de fil de câblage isolé. Le

diamètre du conducteur n'est pas spécialement critique. Le support du fil, le cadre proprement dit, ne peut évidemment pas être métallique, sous peine d'échec garanti. Notre prototype mesurait 82 × 133 cm et donnait des résultats probants avec trois spires. L'amplificateur, nous l'avons logé dans une petite boîte fixée à la base du cadre. Comme le montage ne consomme qu'une dizaine de milliampères, une pile de 9 V suffira amplement. Bien sûr, il est loisible à chacun d'y substituer une alimentation secteur stabilisée, à condition, on le devine, qu'elle soit convenablement déparasitée.

RÉSULTATS

Pour une antenne, un essai pratique en dit beaucoup plus long qu'une kyrielle de chiffres. Aux fins de tests, nous l'avons donc branchée au récepteur d'ondes courtes publié en janvier 1999. Grâce à la possibilité offerte de commuter instantanément de l'une à

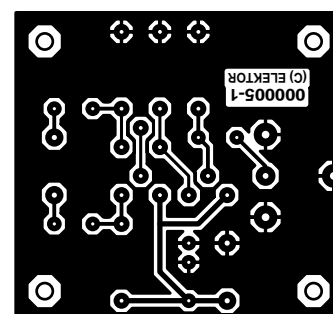
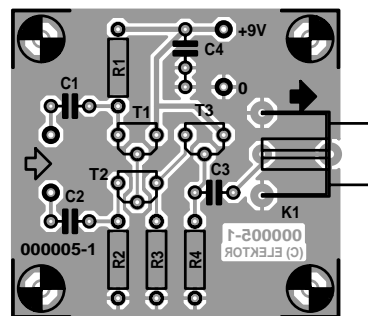
l'autre, nous avons pu comparer ses prestations avec une antenne filaire conventionnelle. Sur l'ensemble du domaine, les deux antennes ont fourni des résultats comparables, mesures effectuées au S-mètre (indicateur de force de réception). En comparaison, l'antenne fouet nous a semblé « tirer trop court », généralement au tiers des déflexions du S-mètre dans les autres configurations.

(000005)

Rédacteur : J. van Rooij
À la technique : K. Walraven

2

Figure 2. Et qui prétendra qu'un petit montage comme celui-ci se montrerait piègeur lors de la réalisation ?



Liste des composants

Résistances :
R1, R2 = 82 kΩ
R3 = 560 Ω
R4 = 220 Ω

Condensateurs :
C1, C2 = 2nF2 cér. au pas de 5 mm
C3 = 100 nF cér. au pas de 5 mm
C4 = 100 nF cér. au pas de 7,5 mm

Semi-conducteurs :
T1, T3 = BF494
T2 = BF451 (BF450)

Divers :
K1 = embase tulipe ou BNC
10 à 15 mètres de fil de montage

horloge

« Tour du Rhin » Mod 2

avec une platine à l'esthétique indéniable

L'horloge « Tour du Rhin » décrite le millénaire dernier, en 1998 plus exactement, était une reproduction miniaturisée pilotée par DCF de la fameuse horloge monumentale se trouvant sur les bords du Rhin à Dusseldorf. L'auteur, aiguillonné par les nombreuses réactions positives suscitées par son projet, a décidé de se réattacher au programme en vue de le doter de nouvelles fonctionnalités. Nous avons, quant à nous, redessiné la platine en lui donnant une esthétique très proche de la tour dans sa version originale.



Réaliser son propre projet donne encore plus de plaisir lorsque le résultat obtenu est exclusif. Ceux d'entre nos lecteurs qui se sont attaqués, voici plus de 2 ans, à l'horloge « Tour du Rhin » (modèle 1) l'auront sans aucun doute constaté eux-mêmes. Toute personne se trouvant « confrontée » pour la première fois à une horloge de ce genre ne manquera pas de se demander comment elle marche.

L'horloge intégrée dans la Tour du Rhin prend la forme d'une série de points lumineux disposés pour constituer une ligne verticale. Nous avons, pour visualiser l'heure, remplacé les 39 ampoules d'origine par autant de LED jaunes. On retrouve, sur le croquis de la **figure 1**, les contours de la Tour du Rhin. Les 9 LED inférieures représentent les unités de secondes (0 à 9). Les 5 LED placées au-dessus de celles

des secondes visualisent les dizaines de secondes. Le codage des minutes (unités et dizaines) et des heures (unités) reprend la même formule. Il reste, pour finir, 2 LED servant à indiquer chacune une dizaine d'heures. Un examen du croquis de la figure 1 et le principe devrait être évident. À 23:59:59 toutes les LED sont allumées, une seconde plus tard, à 00:00:00, toutes les LED sont éteintes. Un nouveau cycle de 24 heures recommence.

On constate, par rapport à la première version de cette horloge, les adjonctions suivantes : outre les 4 luminaires donnant les 4 points cardinaux (N, E, S et O), nous avons également monté la balise lumineuse au sommet de la tour de sorte que notre modèle réduit reproduit le plus fidèlement possible l'original.

Nous avons, outre ces modifications cosmétiques, doté le programme de nouvelles fonctions. Il n'est plus indispensable maintenant, bien que cela soit plus simple et plus précis, de faire appel à un récepteur DCF. Cette horloge peut donner l'heure exacte sans réception de signal horaire, mais cela ne manquera pas, à moyen terme, de se traduire par une certaine dérive par rapport à l'heure réelle.

Il nous a fallu, pour pouvoir nous passer du récepteur DCF, ajouter quelques touches qui serviront à la mise à l'heure de l'horloge. On se souviendra que le signal horaire donne également le jour de la semaine (que l'on pourra, maintenant, entrer par le biais des touches). L'horloge disposant de cette information additionnelle, on pourra également s'en servir en tant que réveil (qui tiendra compte du jour de la semaine, ce que ne font pas les radio-réveils courants). Le montage comporte, bien entendu, un résonateur servant à produire le signal du réveil.

Il nous a fallu, pour implémenter toutes ces fonctions, utiliser un processeur disposant d'une capacité de mémoire plus importante. C'est là le prix à payer pour les toutes ces fonctions additionnelles.

COMMENT LES OCTETS ALLUMENT-ILS LES AMPOULES ?

La figure 2 donne le schéma de notre horloge à LED. Une grande partie du schéma est identique à celui de la première version décrite en mai 1998. Le type de microcontrôleur utilisé est resté le même, ni la circuiterie de commande des LED ni l'alimentation n'ont changé. Ce qu'il y a de neuf est le circuit de commande du résonateur piézo. On pourra utiliser ici, avec le dimensionnement choisi, n'importe quel résonateur travaillant à une tension continue de 5 V et ne consommant pas plus de 50 mA. Les 3 boutons-poussoirs attaquent des entrées du processeur non utilisées sur la pre-

mière version de ce projet. Les LED supplémentaires sont venues s'intercaler dans la matrice aux emplacements non utilisés précédemment.

À nouveau il est possible, par le biais de JP1, d'implanter une LED additionnelle servant à signaler la présence de la tension d'alimentation. R4 et R5 permettent de s'essayer à d'autres dispositifs d'affichage. Normalement, ces 3 résistances ne sont pas nécessaires, de sorte que l'on pourra, dans la majorité des cas, ne pas implanter R4, R5 ni R7 (la résistance prise en amont de JP1).

Est-il bien nécessaire de préciser que bien que le matériel soit très proche de celui constituant la première version de cette horloge, le logiciel programmé dans le microcontrôleur a beaucoup évolué lui. Il vous faudra donc, pour disposer des nouvelles fonctions, utiliser un nouveau microcontrôleur programmé. Rien n'interdit non plus de décider de n'utiliser que le nouveau circuit imprimé, sachant qu'il ne sera pas nécessaire, dans ce cas-là, d'implanter les nouvelles LED.

LA RÉALISATION

Nous vous proposons, en figure 3, le dessin des pistes. On constate au premier coup d'œil que nous sommes loin d'un « lay-out » classique. Si la forme de la platine est spéciale, ses dimensions le sont aussi. Ceci explique que nous ayons dû réduire sa taille à 55% de la taille réelle. Il vous faudra donc, si vous voulez réaliser votre propre platine, procéder à un agrandissement à 182% de ce dessin. Il vous est possible, si vous avez accès à Internet, de chercher, sur le site d'Elektor (<http://www.elektor.presse.fr>) le dessin des pistes à l'échelle 1 (sous la forme d'un document .pdf).

L'affaire est plus simple si vous achetez la platine toute faite, vu qu'elle possède déjà la forme de la Tour du Rhin. Si vous gravez votre propre platine il vous faudra lui donner la forme requise à l'aide d'une scie à lame fine. On pourra envisager de doter, avant la mise en place des composants, le côté « composants » de la platine d'une fine (et jolie) couche de peinture en évitant de boucher les orifices devant recevoir les composants.

Le pied de la tour comporte une paire de platines séparables du reste. La platine inférieure est celle du clavier; juste au-dessus nous avons celle du processeur chargé du pilotage de l'ensemble. À vous de choisir si vous voulez effectuer cette double (voire simple) séparation ou garder la tour d'une seule pièce, ce choix ayant bien évidemment des conséquences sur la réalisation finale du montage.

Il va falloir, vu que les 3 platines sont isolées électriquement l'une de l'autre, procéder à leur interconnexion.

On commencera par la mise en place

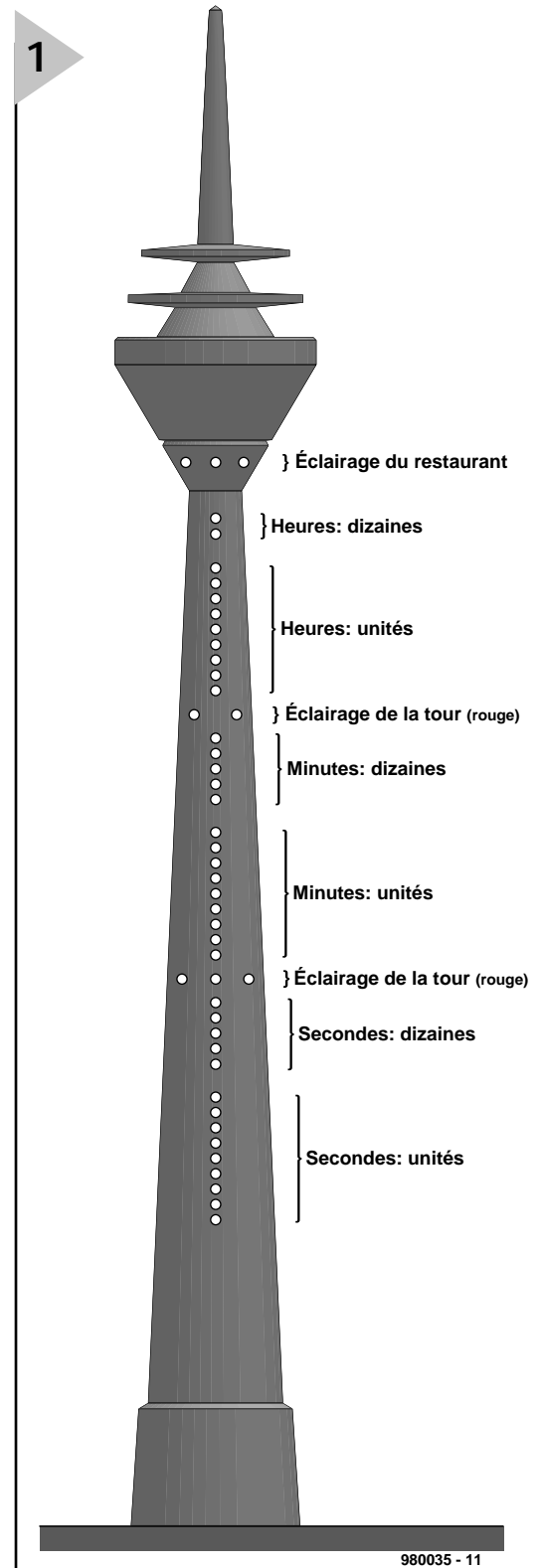
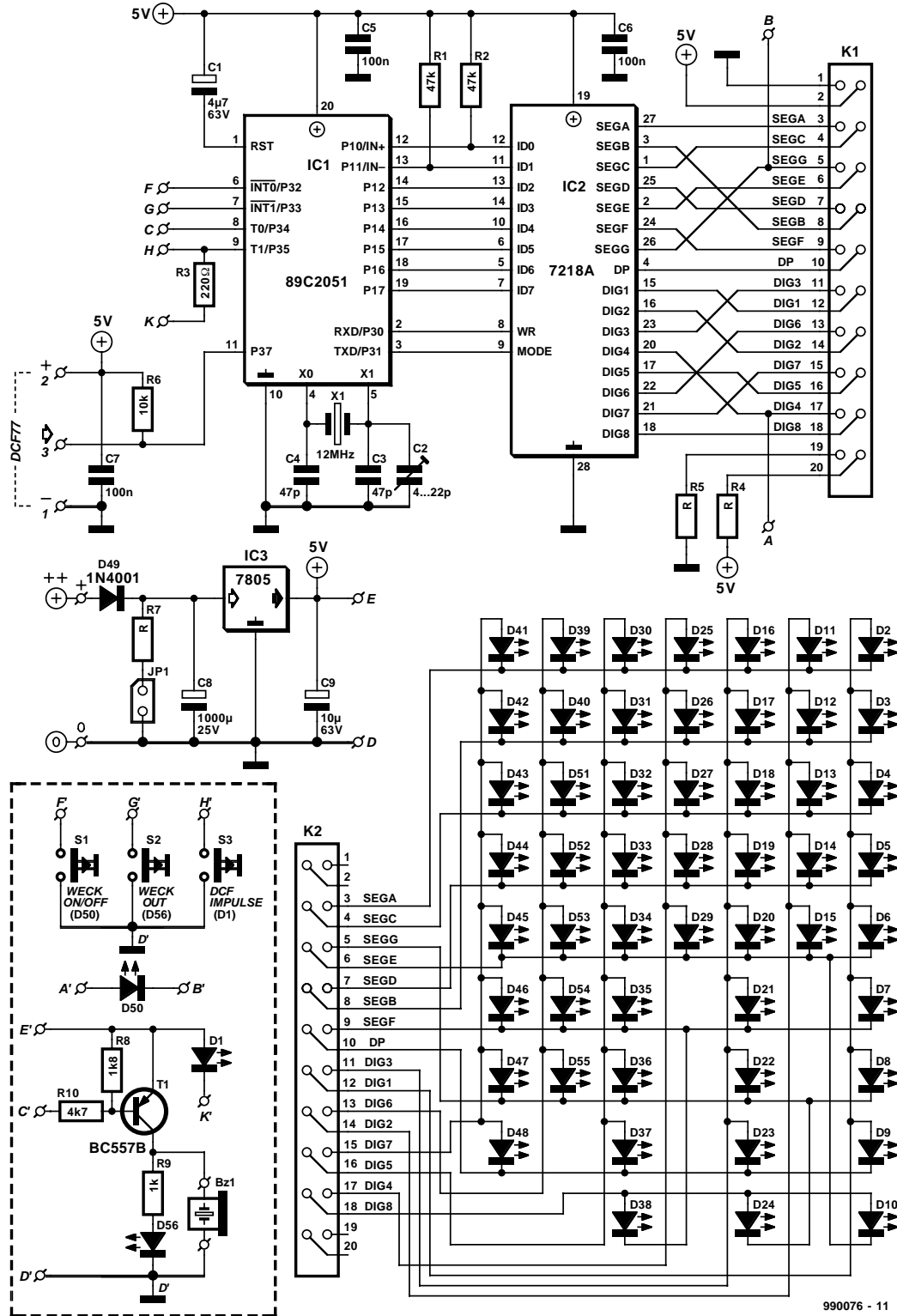


Figure 1. On voit sur ce croquis le principe sur lequel repose l'affichage de l'heure de la Tour du Rhin de Düsseldorf.

des ponts de câblage et des picots requis. On prévoira, de préférence, pour IC1 et IC2, des supports pour C.I. Attention, lors de l'implantation des nombreuses LED que comporte ce montage, à bien en respecter la polarité. Une erreur est vite faite.



990076 - 11

Figure 2. L'électronique. Il n'est pas difficile, à condition d'avoir lu le premier article, de repérer les similitudes entre la version 2 de l'horloge de la « Tour du Rhin » et sa version originale d'il y a 2 ans.

Une fois que l'on en a terminé avec l'implantation des composants il faudra, interconnecter les platines à l'aide de quelques fils de liaison et d'un morceau de câble plat et cela que l'on ait choisi de séparer les platines ou de les garder d'une pièce. On reliera le point A au point A', B à B' et cela jusqu'à K à K'.

Il faut ensuite relier les embases K1 et K2 à l'aide d'un morceau de câble en nappe à 20 conducteurs. Le récepteur DCF optionnel sera branché aux picots PC1 à PC3. La CPU étant une

source indiscutable de parasites qui pourraient influencer de façon néfaste le fonctionnement de l'horloge DCF, il faudra veiller à disposer le récepteur à quelque 20 à 30 cm de la CPU. On utilisera un petit morceau de câble blindé pour effectuer la liaison entre le récepteur et les picots en question.

L'alimentation fait appel à un adaptateur secteur pouvant fournir un courant de l'ordre de 250 mA à une ten-

sion continue comprise entre 8 et 12 V. Une fois les connexions des différentes platines effectuées, on pourra enficher l'adaptateur dans une prise secteur. Si l'on utilise un module DCF et que la réception du signal horaire est correcte, on devrait voir la LED D50 clignoter au rythme des secondes. Ceci est également vrai pour certaines des balises. En

3

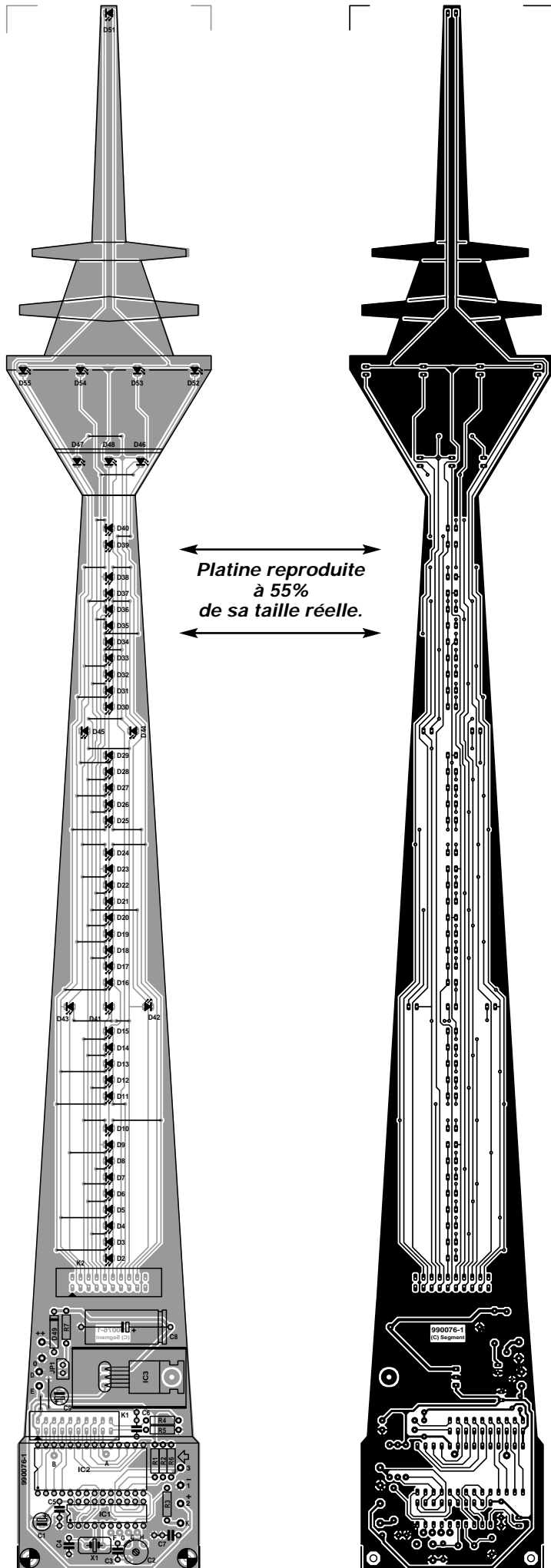
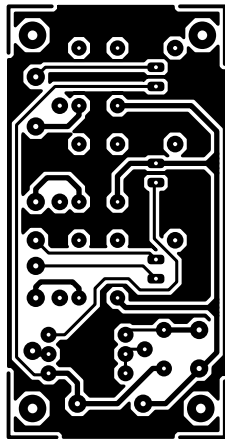
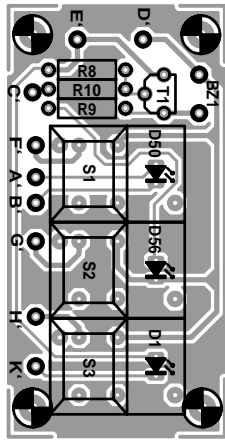


Figure 3. Ce n'est peut-être pas la platine la plus grande que nous vous avons jamais proposée, mais sans doute celle pouvant se targuer d'être la plus esthétique.

cas de réception correcte (confirmée par le clignotement régulier de D50), l'horloge devrait, au bout de 2 à 3 mn, afficher l'heure exacte.

Il faudra, en l'absence de module DCF, procéder à une mise à l'heure manuelle et démarrer le mode « sans DCF ». Pour ce faire, il faut appuyer sur la touche S3 à l'instant de l'application de la tension d'alimentation. On verra alors l'éclairage du restaurant s'allumer, toutes les autres LED restant éteintes. Après avoir relâché S3 on pourra s'attaquer à la mise à l'heure. S1 sert à entrer les heures, chaque action sur cette touche se traduisant par une incrémentation du compteur des heures. On passe ensuite à S2 pour entrer les minutes. Attention, le passage par 60 du compteur des minutes se traduit par une incrémentation du compteur des heures.

S3 sert à entrer le jour de la semaine. Cette touche remplit une double fonction. Une action rapide (entre 0,1 et 1 s) entre le jour de la semaine. Chaque

nouvelle action d'une durée comprise entre ces 2 valeurs incrémente le compteur de jour. Lors de l'entrée du jour de la semaine, la première LED des secondes représente le lundi, la seconde le mardi et ainsi de suite. Une fois le dimanche passé (7 LED allumées) on repasse, logiquement, au lundi (1 LED allumée).

Une fois l'horloge mise à l'heure, on pourra la démarrer par une action sur S3. Une action prolongée (d'une durée supérieure à 1 s) démarre l'horloge à une minute ronde. On actionnera partant S3 à lors de la 59ème seconde d'une minute de façon à ce que l'horloge démarre exactement à la minute ronde suivante.

Au bout de quelques remises à l'heure vous verrez que ce processus est étonnamment logique. Les utilisateurs d'un module DCF n'ont pas à se casser la tête pour la remise à l'heure de leur horloge vu que celle-ci se fait automatiquement !

ENTRÉE DE L'HEURE DE RÉVEIL

On pourra, une fois que l'horloge est remise à l'heure et qu'elle marche, entrer l'heure de réveil. Tout au long de cette opération, l'horloge reste à l'heure. Le module DCF est cependant, lui, mis hors-circuit pendant ce réglage. Ceci n'a pas de conséquence sachant que l'horloge dispose d'un quartz qui lui fournit un signal d'horloge relativement précis (nombre d'horloges DCF alimentées par piles, économisent de l'énergie en ne procédant, chaque jour, qu'à quelques synchronisations seulement sur le signal DCF. Ceci suffit à garantir une précision de l'horloge plus qu'acceptable).

Il faudra, pour entrer l'heure de réveil, commencer par appuyer sur S3. Toutes les LED s'éteignent alors voire donnent la dernière heure de réveil (avec le jour de la semaine) choisie précédemment. Chacune des 3 touches remplit alors une double fonction en fonction de la durée de l'action : action brève (moins de 0,6 s) ou action longue (durée supérieure à 0,6 s). L'heure de réveil se définit sous la forme d'un nombre d'heure et d'un nombre de minutes.

Voici les fonctions attribuées aux touches :

Action brève sur S1 : incrémentation des heures

Action brève sur S2 : incrémentation des minutes

Action brève sur S3 : incrémentation du jour de la semaine

Action longue sur S1 : réveil on/off

Action longue sur S2 : programmation du jour de la semaine

Action longue sur S3 : retour à l'affichage de l'heure

Chaque action sur S1, se traduit par une incrémentation de l'heure de réveil. S2 sert, de façon similaire, à entrer les minutes de l'heure de réveil, la touche S3 servant quant à elle, à entrer le jour de réveil. Après une première action sur S3, on verra clignoter la LED correspondant au lundi. Il faudra, si le réveil ne doit se déclencher, par exemple, que le lundi, appuyer sur S2 aussi longtemps que nécessaire pour que la LED de jour de la semaine qui clignote reste allumée fixement. Si l'on peut programmer plusieurs jours dans la semaine on incrémentera le compteur de jours de la semaine par le biais de S3 et on activera chacun de ces jours par le biais de S2. Lors d'un passage du dimanche au lundi, tous les jours de la semaine programmés sont effacés; il faudra partant procéder à une nouvelle sélection des jours.

La dernière position prise par le curseur reste mémorisée après le passage en mode « horloge ». Il est possible ainsi, ultérieurement, d'ajouter d'autres jours au programme de réveil. Pour terminer, une action prolongée sur S1 active la fonction de réveil. La LED intégrée dans cette touche s'allume. Il reste à actionner S3 suffisamment longtemps pour repasser en mode « horloge ». Cette touche servira également à faire taire le réveil s'il lui prenait l'envie de se manifester (ce qui est d'ailleurs une réaction normale de la part d'un réveil).

La sortie de réveil est en fait une sortie logique attaquant un transistor de commutation auquel est connecté un résonateur CC. On peut également envisager de lui faire piloter un générateur de signaux acoustiques, un relais, voire tout autre dispositif électronique. Il peut s'avérer nécessaire, dans certains cas, de prendre un tampon à cette sortie. Comme vous le constatez, nous avons choisi de vous

Liste des composants

Résistances :

R1,R2 = 47 kΩ
R3 = 220 Ω
R4,R5,R7 = cf. texte
R6 = 10 kΩ
R8 = 1kΩ8
R9 = 1 kΩ
R10 = 4kΩ7

Condensateurs :

C1 = 4μ7F/63 V radial
C2 = ajustable 4 à 22 pF
C3,C4 = 47 pF
C5,C6,C7 = 100 nF
C8 = 1 000 μF/25 V
C9 = 10 μF/25 V radial

Semi-conducteurs :

D1,D46,D47,D48 = LED à haut rendement verte
D2 à D40,D56 = LED à haut rendement jaune
D41 à D45,D50 à D55 = LED à haut rendement rouge
D49 = 1N4001
T1 = BC557B
IC1 = AT89C2051-12PC (programmé EPS996519-1)
IC2 = ICM7218A IJI (Intersil = Harris)
IC3 = 7805

Divers :

JP1 = embase mâle autosécable à 3 contacts + cavalier
K1,K2 = embase HE-10 à 2 rangées de 10 contacts
morceau de câble à 20 conducteurs doté de 2 connecteurs DIP à 20 broches
S1,S2,S3 = touche Digitast avec LED
X1 = quartz 12 MHz
Bz1 = résonateur à tension continue 5 ou 6 V
radiateur pour IC3 (15 K/W, ICK35 par exemple)
Optionnel : module DCF (Conrad 64 11 38-55)

permettre de donner libre-cours à votre imagination.

(990076-1)

texte : Hans Steeman
technique : Karel Walraven

L'horloge de la « Tour du Rhin » sur Internet

Les amateurs de la « Tour du Rhin » et de l'horloge qu'elle abrite peuvent trouver sur Internet nombre d'informations intéressantes la concernant. On trouve, sur le site :

<http://www.duesseldorf.de/tourist/download/index.html>,

un économiseur d'écran (version pour Windows) basé sur la « Tour du Rhin ». Il existe également un site consacré au thème « horloges » à l'adresse :

<http://www.hsp.de/~wiegels/programm/uhren.htm>.

Ce site met bien entendu à disposition l'un ou l'autre programme d'émulation de la « Tour du Rhin ».

Vous pourrez, si vous voulez en savoir plus au sujet de la « Rheinturm » elle-même, faire un tour au site « Hohe Túrme » sis à l'adresse :

<http://www.hsp.de/~wiegels/tuerme/hoch.htm>

où vous apprendrez tout des caractéristiques de cette tour et de bien d'autres.

les décharges statiques

une suite de faits « choquants »

S'il est un phénomène que les composants électroniques « maîtrisent » mal, c'est bien celui des décharges statiques. Il vaut partant mieux faire attention et en tenir compte. Nous vous proposons ci-après quelques trucs & astuces pratiques, et verrons pourquoi il vaut mieux que... les éléments ne reniflent pas trop de la trompe du côté de circuits intégrés.



990064-51

Les charges statiques sont un phénomène encore mal connu, tout comme cela fut le cas des premiers trains (à vapeur) il y a plus d'un siècle. Nombre de scientifiques d'alors étaient persuadés que le corps humain se désintégrerait de lui-même à une vitesse supérieure à 100 km/h. Nous savons depuis lors que cela n'est pas le cas. Le facteur important n'est pas la vitesse mais sa variation (l'accélération). Rouler à 100 km/h dans une voiture ne pose plus, de nos jours, le moindre problème, mais si l'on percute un mur à cette même vitesse, la vitesse passe, en quelques fractions de seconde, de 100 à 0 km/h et c'est bien là le problème auquel on se trouve alors confronté. Il en va de même pour l'électricité. Il vous est sans doute déjà arrivé de vous charger d'électricité statique en caressant un chat ou en marchant sur une surface plastique (moquette ou lino). La charge se fait progressivement et partant est imperceptible. Ce n'est que lorsque l'on touche une poignée de

porte ou une personne au potentiel différent que l'on se rend compte de l'existence d'une charge statique qui ne manque pas, littéralement, de faire des étincelles. La variation de charge se traduit par un choc, plus surprenant que douloureux d'ailleurs. Ce que nous ressentons n'est pas la différence de potentiel mais le minuscule courant de décharge qui se met alors à passer. Plus l'intensité de ce courant est élevée, plus la douleur ressentie est forte. Notons qu'il est possible de réduire le courant de décharge en employant une résistance de 10 ou de 100 k Ω et de s'en servir comme « paratonnerre ». Avec ce petit truc on ne ressent plus rien, bien que la différence de potentiel disparaisse (encore que cette égalisation se fasse plus lentement).

COMPOSANTS FRAGILES

Les composants électroniques sont encore plus sensibles aux (dé)charges statiques que ne le sont les êtres humains. Certains d'entre eux sont fra-

giles au point d'être irrémédiablement endommagés lors d'une égalisation de potentiels.

Nous savons depuis que le niveau de la tension n'a pas d'importance que l'on applique 1 000 volts ou plus à un composant, si tant est que ce même niveau de tension règne dans son environnement immédiat et que partant il n'y ait pas de différence de potentiel. En cas de différence de tension, cela peut avoir pour conséquence, au cœur du composant, un endommagement de l'isolation (en cas de dépassement de la tension de disruption) voire la circulation d'un courant ayant pour effet d'annuler les différences de charge. Ce courant peut lui aussi détruire le composant.

Une fois que l'on sait tout cela on peut se doter d'un code de conduite pour une manipulation en toute sécurité des composants électroniques. Le principe primordial est donc, avant d'entrer en contact avec un composant, de toujours faire en sorte qu'il n'y ait pas de

différence de potentiel entre soi et le composant.

Un exemple : vous venez d'acheter de nouveaux modules de mémoire pour votre PC et vous voulez les y monter. La pochette dans laquelle se trouvent les modules peut avoir un potentiel quelconque. L'intérieur de cette pochette est faite de matériau conducteur de sorte qu'il ne peut pas y avoir de différence de potentiel entre les modules eux-mêmes et les modules et la pochette : les circuits intégrés sont parfaitement à l'abri. Mais, en ce qui vous concerne, vous pouvez présenter un potentiel quelconque par rapport à la pochette. Il va donc falloir faire en sorte que la différence de tension entre vous-même et la pochette disparaisse. Si vous n'avez pas crainte d'un (petit) choc, il suffit tout simplement de prendre la pochette (pas son contenu) en main. Ceci fait, vous pouvez sortir les modules de leur pochette sans leur faire courir le moindre risque. N'ayez crainte alors de toucher les modules, au contraire, car vous annulez alors toute différence de potentiel ayant pu se créer depuis. Répétons-le, il est préférable de prendre alors les modules en main plutôt que par le biais d'une pince (plate) isolée.

La mise des modules dans le PC constitue l'étape suivante. De son côté, le PC peut présenter un potentiel quelconque par rapport au vôtre. Touchez, pour cette raison, de l'autre main, pas celle dans laquelle vous tenez les modules de mémoire, le coffret du PC à un endroit conducteur, une vis non peinte par exemple. Ce contact peut se traduire par un petit choc mais ce faisant, vous venez d'annuler toute différence de tension entre les modules et le PC. Maintenez le contact avec le PC d'une main et enfichez les modules de l'autre.

MISE À LA TERRE ?

Et qu'en est-il de la terre ? Ne vaudrait-il pas mieux de commencer par mettre les modules en contact avec la terre ? La terre ne présente-t-elle pas un potentiel nul ? Il n'y a pas le moindre risque n'est-ce pas ! Examinons la situation :

De même que vous pouvez parfaitement vous sentir à l'aide au sommet de la Tour Eiffel (tant que vous ne sautez ou n'en tombez pas), un composant n'a que faire de se trouver à un potentiel stable nul (celui de la terre) ou de n'importe quelle autre valeur (élevée). Si le potentiel de la terre ne présente ne fait pas la moindre importance, il n'en est pas moins très pratique. Tant que le PC se trouve au départ au potentiel de la terre et que cela est également le cas des modules, on est assuré, sans avoir à le mesurer, d'une absence de différence de potentiel et, partant, de danger latent.

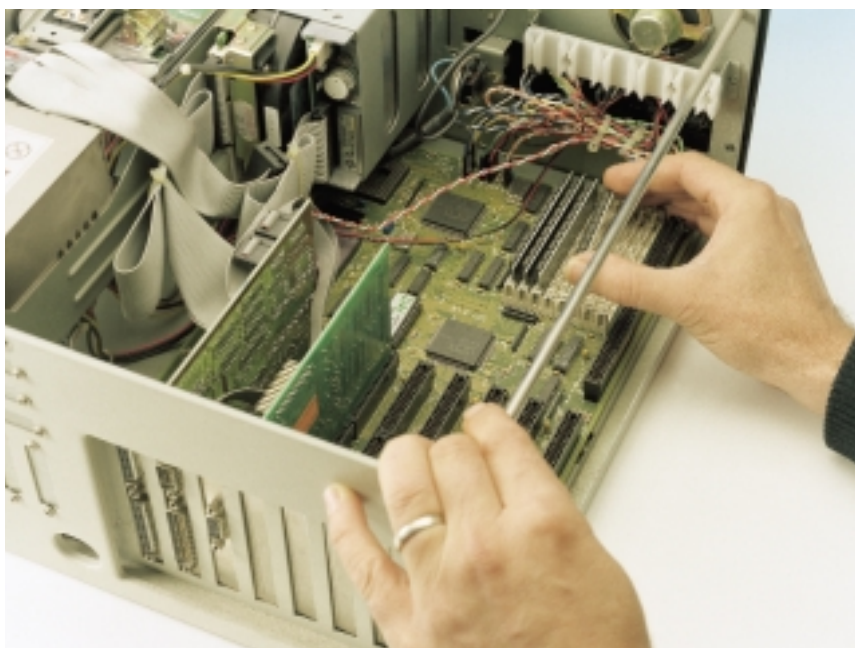


LA DIFFÉRENCE ENTRE UN ÊTRE HUMAIN ET UN ÉLÉPHANT

Que peut supporter un circuit intégré ? Prenons, comme exemple, les familles logiques modernes. Les fabricants se sont rendu compte que leurs clients étaient moins rigoureux qu'ils ne le pensaient. En dépit des avertissements aux couleurs les plus chatoyantes, nombre de composants étaient endommagés vu qu'on les empoignait sans la moindre précaution. Ceci explique qu'ils aient doté leurs produits de protections leur permettant de supporter sans dommage la décharge « moyenne » d'un corps humain. Sachez, à titre info, qu'un circuit logique de la famille HC(T) peut supporter des tensions jusqu'à 2 000 volts (!). On peut admettre que le corps

humain possède une capacité de l'ordre de 100 à 150 pF par rapport à son environnement. Cette capacité se décharge lors du contact avec le composant. Si un éléphant, dont la capacité est notablement plus élevée vu son corps plus massif, le prenait « en trompe » ce circuit intégré ne manquerait pas de trépasser illico-presto. De même, si votre potentiel dépasse les 2 000 V, le circuit intégré serait endommagé. La pratique nous a appris que l'on peut fort bien travailler dans les conditions actuelles. Les courants entrant en jeu lors de la décharge méritent d'être pris en compte vu que l'on parle de quelques centaines de mA sur une durée de quelques millisecondes (μs).

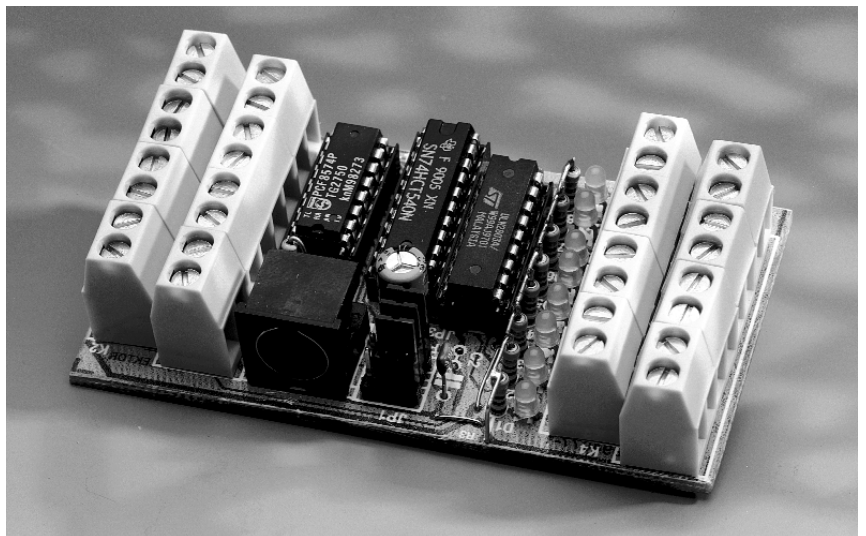
(990064)



port d'E/S 8 bits

commander par l'interface I²C

Le lecteur assidu d'Elektor y découvre, à intervalle plus ou moins régulier, la description de l'une ou l'autre interface d'Entrées/Sorties (E/S). Chacun de ces montages possède cependant ses spécificités qui le rendent plus ou moins unique. Le présent port d'E/S 8 bits est doté de 8 sorties tamponnées, possède 8 entrées à niveau TTL et communique avec le PC par le biais de la fameuse interface I²C. Il est en outre remarquable par sa compacité.



Le choix de l'interface I²C présente des avantages indéniables. La communication se fait par le biais d'une interface compacte se résumant à 2 fils. Il s'agit d'une interface fiable, relativement rapide et parfaitement documentée. Il existe en outre, pour ladite interface, de nombreux composants d'E/S au prix très abordable. Comme on le constate, de nombreux avantages, mais connaît-elle aussi ses inconvénients cette interface ? Elle n'en a, en fait qu'un seul. Le système de commande, qu'il s'agisse d'un ordinateur ou d'un microcontrôleur, se doit de posséder une interface I²C.

Nous avons décrit, par le passé, un certain nombre d'interfaces pour PC parfaitement adaptées à cette tâche, la dernière l'ayant été dans le numéro de mars de cette année. Il s'agit en l'occurrence d'une interface I²C venant se brancher sur le port imprimante. Ladite interface est supportée par des DDL auxquels on pourra faire appel en environnement Windows. Il nous arrive également, de temps à autre, de nous intéresser à des systèmes à microcontrôleur disposant d'une interface de ce type.

Il n'y a plus la moindre raison de ne

pas vous attaquer au présent montage. Il est possible, pour des applications « lourdes », de connecter un maximum de 16 modules en parallèle ce qui se traduit, pour l'utilisateur, par la mise à disposition de pas moins de 128 lignes d'E/S bidirectionnelles.

3 CIRCUITS INTÉGRÉS EN TOUT ET POUR TOUT

La figure 1 vous donne le schéma du montage. Le coeur de ce projet est un PCF8574(A), un circuit intégré de l'écurie Philips que nous avons utilisé dans plusieurs montages décrits dans Elektor (cf. l'encadré).

L'électronique située sur la gauche de ce composant constitue l'interface I²C proprement dite. La communication avec le monde extérieur peut se faire de 2 façons, soit par une embase DIN miniature soit par le biais d'un double bornier encartable à vis. On trouve, au niveau de l'embase mini-DIN, les 2 signaux de commande SCL (*Serial CLock* = horloge sérielle) et SDA (*Serial DA* = données sérielles) ainsi que la tension d'alimentation et le potentiel de la masse. Le dernier contact de cette embase est utilisé par la ligne d'interruption optionnelle

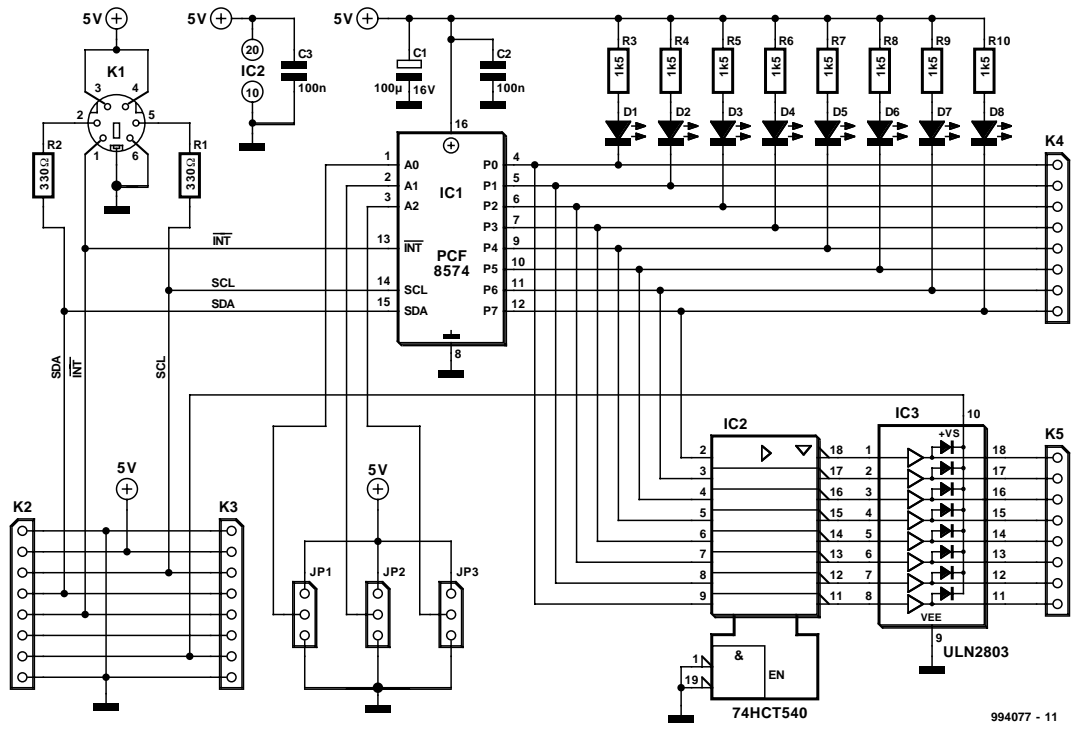


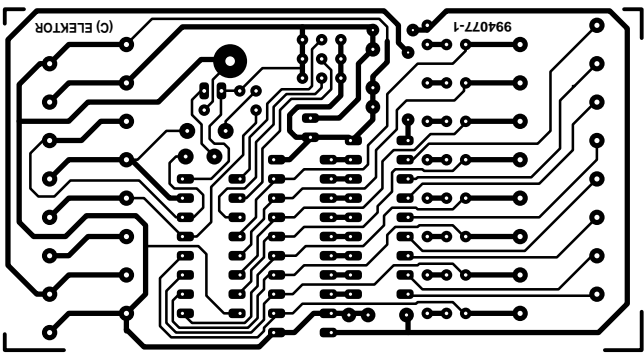
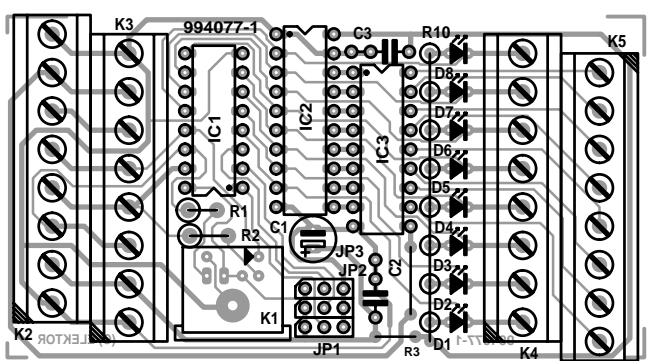
Figure 1. L'électronique du port d'E/S à 8 bits piloté par le biais d'une interface I²C.

(INT = *INT*errupt) du bus I²C. C'est au cas où l'on utiliserait une combinaison de plusieurs de ces interfaces que le second mode de connexion proposé prend tout son intérêt. Le brochage des borniers K2 et K3 est tel qu'il est possible d'interconnecter aisément les interfaces par le biais de faibles longueurs de conducteurs. À l'image de l'embase min-DIN, les borniers mettent à disposition, outre les signaux I²C, également la tension d'alimentation et le potentiel de la masse. On y trouve aussi, nous y reviendrons plus tard, la ligne de cathode commune des diodes

de protection (diodes dites de roue libre) que comporte IC3. Les cavaliers JP1 à JP3 permettent de (re)définir l'adresse de base du PCF8574(A). Si l'on utilise, pour IC1, un PCF8574 (sans le suffixe (A)) l'adresse de base sera \$0100xxx (40_{HEX}). Si au contraire on a opté pour un PCF8574(A), cette adresse de base sera \$0111xxx (70_{HEX}). Les 3 x de l'adresse de base sont définis-

sables par le biais de cavaliers. La combinaison de 2 adresses de base et de 8 combinaisons de bits se traduisent par un total de 16 adresses différentes, ce qui explique que l'on puisse connecter, à un bus I²C, un maximum de 16 modules. Ces 3 bits sont en fait suivis d'un dernier bit qui sert lui à déterminer la direction des données (lecture ou écriture). Passons maintenant à la partie droite du schéma. IC1 dispose de 8 sorties bidirectionnelles (P0 à P7). Chacune de ces lignes d'E/S est dotée d'une résistance de forçage au niveau

Figure 2. Dessin des pistes et sérigraphie de la platine développée à l'intention de cette réalisation.



- Liste des composants**
- Résistances :
 R1, R2 = 330 Ω
 R3 à R10 = 1kΩ5
- Condensateurs :
 C1 = 100 μF/16 V radial
 C2, C3 = 100 nF
- Semi-conducteurs :
 D1 à D8 = LED à haut rendement
 IC1 = PCF8574P ou PCF8574AP
 IC2 = 74HCT540
 IC3 = ULN2803
- Divers :
 JP1 à JP3 = embase autosécable mâle à 1 rangée de 3 contacts
 K1 = embase mini DIN à 6 contacts encartable
 K2 à K5 = bornier encartable à 8 contacts au pas de 5 mm

Le PCF8574 sous la loupe

Le PCF8574(A) est un circuit intégré CMOS pouvant être utilisé pour doter des microcontrôleurs de fonctions d'E/S additionnelles au travers d'un bus I²C. Ce composant intègre un port quasi-bidirectionnel et une interface I²C. Le terme de quasi-bidirectionnel désigne un port combinant entrée et sortie. Ce port peut servir d'entrée à condition d'avoir été mis, auparavant, au niveau haut. Le logiciel doit, partant appliquer un « 1 » logique à la dite sortie. Une fois que la sortie se trouve au niveau haut elle peut être utilisée pour la prise en compte de niveaux logiques présents sur la ligne de port. On peut ainsi savoir, si l'on procède, à l'aide d'une instruction de lecture, à la lecture de la sortie, à quel niveau se trouve l'entrée. Il faut, pour faire passer au niveau bas (« 0 ») un port défini en entrée, drainer un courant externe de 100 µA au moins vers la masse.

Les ports de sortie ont une puissance suffisante pour attaquer directement une LED. On se référera au croquis de la figure 3 pour voir comment procéder. Un port ne peut fournir que 100 µA au maximum et ne peut commuter à la masse qu'un courant de 25 mA au plus.

Ce composant est doté d'une possibilité de générer une interruption. Lorsqu'un port est défini comme étant une entrée, chaque changement de niveau (flanc montant ou descendant) générera une interruption. L'interruption est levée dès que disparaît l'état qui l'a généré (le port a retrouvé sa valeur d'origine) ou après une opération de lecture ou d'écriture du port par le biais de l'interface I²C.

La figure 3 donne la structure interne du PCF8574(A). Une remise à zéro matérielle (Reset) fait passer au niveau haut les sorties de tous ports. Elles se trouvent ainsi dans l'état requis pour être également utilisées en entrée.

Les figures 4a et 4b montrent les signaux placés sur le bus I²C pour obtenir le transfert de données, dans un sens ou dans l'autre vers et depuis le microcontrôleur.

Figure 3. Structure interne du PCF8574(A). Ce circuit intégré a été développé spécifiquement pour la réalisation d'E/S numériques par le biais d'une interface I²C.

3

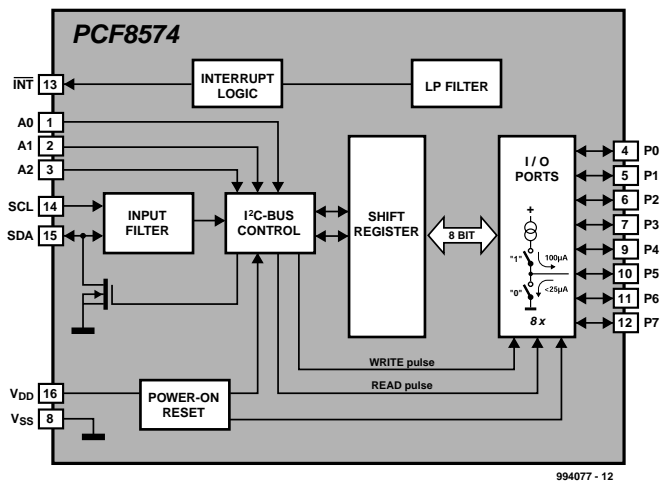
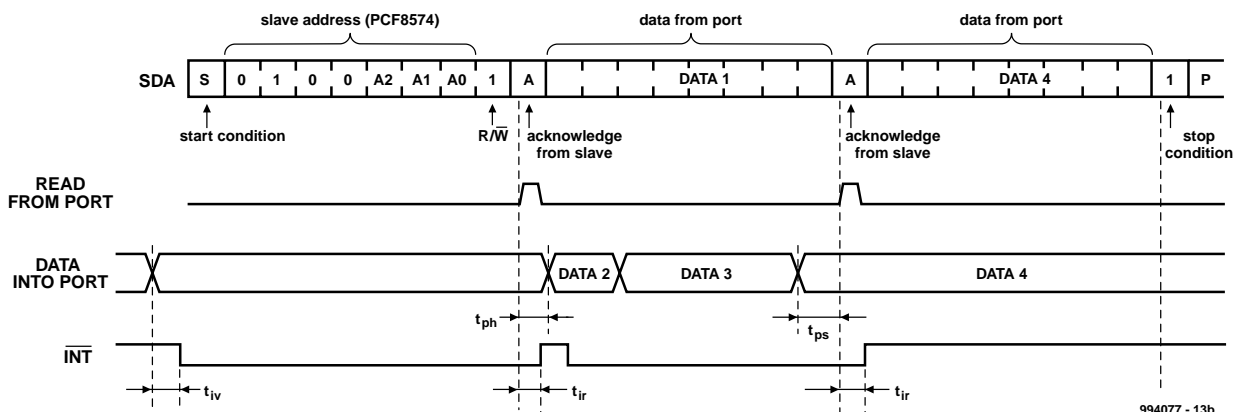
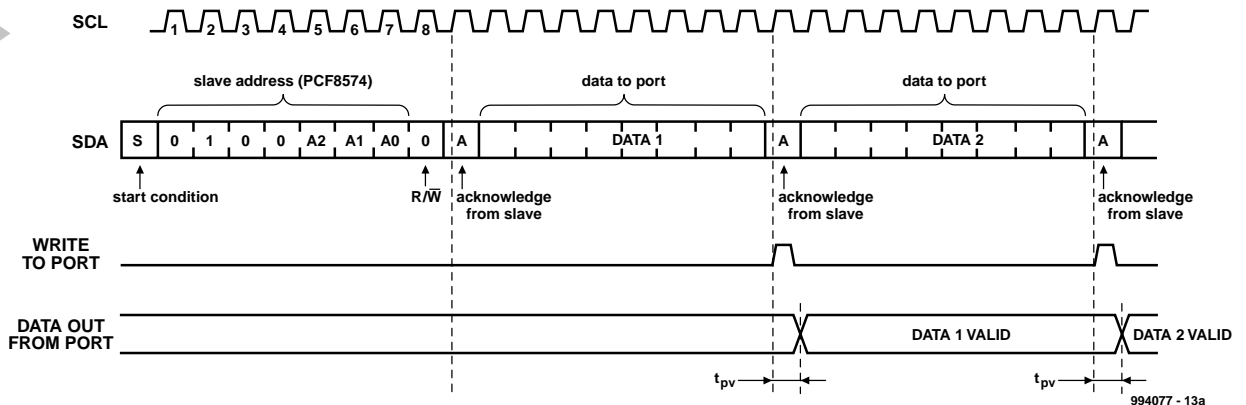


Figure 4. Chronodiagramme des opérations de lecture et d'écriture en direction du PCF8574(A).

4



haut (*pull up*) et d'une LED prise en série. Si l'une des broches d'E/S est définie comme étant une sortie, la LED correspondante visualise l'inverse du niveau de sortie : ainsi, si la sortie se trouve au niveau bas la LED correspondante est allumée; à l'inverse, si elle est haute, la LED sera éteinte. Il n'est pas nécessaire, de par la présence de la résistance de forçage au niveau haut, de mettre une entrée non utilisée à un potentiel quelconque.

Une broche d'E/S peut également servir de sortie et c'est là très précisément la raison de la présence de IC2 et de IC3. IC2 est ce que l'on appelle un circuit de commande de bus (*bus-driver*); il intègre 8 tampons inverseurs qui travaillent à niveau TTL. Ils servent à l'attaque du circuit de commande de puissance (*power driver*), IC3, un ULN2803. Ce circuit intègre 8 étages de commande à transistor dotés d'une sortie à collecteur ouvert. Notons que toutes les sorties sont dotées d'une diode de roue libre dont les cathodes sont reliées à la ligne commune allant à la broche 10 de IC3. Ces circuits de commande spéciaux de la famille ULN conviennent parfaitement à la commande (mise en et hors-circuit) de charges ohmiques et inductives faibles (50 V/50 mA au maximum). Cela signifie qu'il est possible d'activer directement des ampoules de faible

puissance et des relais. Il ne faudra pas oublier de connecter la masse de l'alimentation servant aux charges aux bornes de masse de K2 ou K3 (contacts 1 et 8). La borne 1 du connecteur est identifiée, tant sur le schéma que sur la sérigraphie de la platine, par une marque biseautée. La mise en oeuvre de diode de roue libre est indispensable si l'on doit commuter des charges inductives. Il faudra, dans ce cas-là, relier les cathodes des diodes de roue libre intégrées dans IC3 à la tension d'alimentation utilisée. Il faudra alors également amener ladite tension d'alimentation à la broche 7 de K2 ou K3.

LA RÉALISATION : PLUS DE CONNECTEURS QUE DE CIRCUITS INTÉGRÉS

La réalisation de ce montage sans la moindre chausse-trappe ne devrait pas poser de problème. Nous avons développé à son intention une platine simple face dont la **figure 2** donne la sérigraphie de l'implantation des composants et le dessin des pistes. L'électronique se trouve au centre, entourée des connecteurs, mini-DIN et borniers, servant à la mettre en contact avec le monde extérieur. Les borniers à vis garantissent un contact électrique très fiable. De par leur positionnement côté

à côte, ces 2 borniers devront être montés « dos à dos ». Une solution plus élégante consiste à implanter 2 borniers de hauteurs différentes qui pourront alors être orientés dans la même direction sans interférer l'un avec l'autre; ils sont malheureusement difficiles à dénicher.

Une fois que l'on en a terminé avec les soudures et que l'on a vérifié son montage tant au niveau des soudures qu'à celui des composants et de leur positionnement, on pourra le mettre à contribution. Il reste à définir, par le biais des cavaliers, l'adresse convenable, à connecter la carte à une interface convenable pour disposer d'une carte d'E/S numérique opérationnelle. De par ses dimensions, la platine pourra être implantée dans un coffret standard qui pourra être cliqué sur un rail DIN.

Est-il nécessaire de préciser que l'interface a besoin, pour fonctionner, d'un rien de logiciel. En règle générale, l'interface I²C utilisée devrait être pilotée à l'aide d'un programme pilote (*driver*) permettant l'émission de données vers une adresse donnée ainsi que la lecture de données disponibles à une certaine adresse. Cette fonction est « nécessaire et suffisante » pour obtenir, avec la présente carte, le fonctionnement souhaité.

(994077)

AIDES À LA RÉALISATION

Elektor ne fait pas la vente de composants. Ceux-ci sont normalement à trouver chez un revendeur de composants. Il nous a cependant semblé nécessaire, suite à de nombreuses lettres, de résumer sur cette demi-page les informations cruciales pour la lecture et la compréhension des articles publiés dans Elektor. Nous utilisons, pour l'indication des valeurs de composants, les préfixes (classiques) suivants :

E (exa) = 10 ¹⁸	a (atto) = 10 ⁻¹⁸
P (peta) = 10 ¹⁵	f (femto) = 10 ⁻¹⁵
T (tera) = 10 ¹²	p (pico) = 10 ⁻¹²
G (giga) = 10 ⁹	n (nano) = 10 ⁻⁹
M (mega) = 10 ⁶	μ (micro) = 10 ⁻⁶
k (kilo) = 10 ³	m (milli) = 10 ⁻³
h (hecto) = 10 ²	c (centi) = 10 ⁻²
da (deca) = 10 ¹	d (deci) = 10 ⁻¹

Dans certains schémas et dans la liste des composants nous préférons utiliser, contrairement aux recommandations IEC et BS, le préfixe + symbole comme caractère délimiteur en remplacement de la virgule. 2 exemples :

3kΩ9 = 3,9 kΩ 4μF7 = 4,7 μF

Sauf mention contraire, la tolérance des résistances est ±5% et leur wattage 1/3 à 1/2 watt. La tension de service des condensateurs est de ≥ 50 V.

Lors de la mise en place des composants on commencera en règle générale par l'implantation des composants passifs de la taille la plus faible, c'est-à-dire les ponts de câblage, les résistances et les petits condensateurs; on passera ensuite aux supports pour circuits intégrés, aux relais, aux condensateurs de forte capacité tels que les électrolytiques et aux connecteurs et autres embases. Les semi-conducteurs vulnérable et les circuits intégrés fragiles seront montés en dernier.

Le soudage. On utilisera un fer à souder d'une puissance de 15 à 30 W doté d'une pointe fine et de la soudure à âme de résine (60/40). On enfiche les connexions du composant concerné dans les orifices prévus à cette intention, on les replie légèrement, on les coupe à la bonne longueur et on procède à leur soudure; on attend de 1 à 2 secondes jusqu'à ce que l'alliage étain/plomb devienne liquide et vienne souder relier la connexion au métal de l'orifice. On peut alors enlever le fer à souder. Attention à éviter de surchauffer le composant en particulier les circuits intégrés et les semi-conducteurs. S'il faut désouder un composant on utilisera de préférence un fer à dessouder à pompe aspirante ou un appareil spécialement prévu à cet effet.

Le dépannage. Si le circuit ne fonctionne pas correctement, il faudra comparer soigneusement les composants mis en place sur la platine avec la sérigraphie de l'im-

plantation des composants et vérifier leurs caractéristiques à l'aide de la liste des composants. Tous les composants se trouvent-ils à leur place (celle prévue sur la sérigraphie)? Les polarités des composants en ayant une a-t-elle bien été respectée. N'avez-vous pas fait d'erreur dans le branchement des lignes d'alimentation? Toutes les soudures faites sont-elles « saines »? N'avez-vous pas oublié de pont de câblage? Si le schéma de la réalisation en cause comporte des valeurs de mesure, les éléments mesurés sur le circuit imprimé correspondent-ils à ces valeurs – on peut accepter une dérive de ±10% des dites valeurs.

La valeur d'une résistance est indiquée à l'aide d'un code de couleurs qui définit comme suit :



couleur	1 ^{er} chiffre	2 ^{ème} chiffre	facteur multiplicateur	tolérance
noir	--	0	--	--
marron	1	1	x10 ¹	±1%
rouge	2	2	x10 ²	±2%
orange	3	3	x10 ³	--
jaune	4	4	x10 ⁴	--
vert	5	5	x10 ⁵	±0,5%
bleu	6	6	x10 ⁶	--
violet	7	7	--	--
gris	8	8	--	--
blanc	9	9	--	--
or	--	--	x10 ⁻¹	±5%
argent	--	--	x10 ⁻²	±10%
rien	--	--	--	±20%

Exemples :

marron-rouge-marron-or = 120 Ω, ±5%
jaune-violet-orange-marron = 47 kΩ, ±1%

Il arrive que nous ayons à publier des corrections éventuelles concernant l'une ou l'autre réalisation, ce que nous faisons dans les plus brefs délais dans l'un des magazines publiés ultérieurement. On notera que la rubrique « le coin du lecteur » contient de temps à autre des commentaires ou/et des informations additionnelles concernant des montages publiés dans un numéro précédent.

platine « spéciale-transfos »

pour l'audio-DAC 2000 (et autres réalisations)

La mini-platine universelle décrite dans le présent article met à disposition la place requise par une paire de petits transformateurs d'alimentation, 2 filtres secteur et une paire de porte-fusibles.

Elle pourra constituer une aide pratique indéfectible lors de la réalisation de nombre de montages.

Les lecteurs fidèles d'Elektor ne seront guère étonnés de nous entendre affirmer que de moins en moins de projets décrits dans ce magazine comportent un transformateur monté directement sur la platine. Pourquoi donc ? Pour des aspects de sécurité tout d'abord. Travailler avec la tension du secteur demande la prise de certaines précautions et un montage non soigné peut se traduire par la présence, à des endroits où on ne l'attend pas, de la tension du secteur. Ceci explique que nous utilisons, autant que faire se peut, des adaptateurs secteurs autonomes.

Même si la tension requise ne peut pas être fournie par un bloc secteur standard, nous préférons, si possible, ne pas monter le transformateur sur la platine du montage concerné. Nous avons 2 raisons pour ce choix. La première est un coût sensiblement plus important de la platine, vu l'augmentation de la taille de cette dernière que cette approche implique, d'autant plus que nombre d'entre nos lecteurs préfèrent résoudre eux-mêmes le problème du transformateur. La seconde raison est qu'il est toujours souhaitable de placer un transformateur d'alimentation le plus loin possible de la partie du

montage chargée du traitement du signal, ceci en vue d'éviter tout problème de bruit ou d'interférences.

2 TRANSFORMATEURS

On se trouve alors confronté au problème de fixation du transformateur. Il se veut malheureusement que dans la plupart des cas les transformateurs proposés n'existent qu'en version « moulé » et partant sans la moindre possibilité de les fixer.

C'est en vue de trouver une solution élégante à ce problème de fixation que nous avons conçu une platine « spéciale-transfos ». Elle a, comme première utilisation, l'audio-DAC 2000 (auquel ce numéro consacre un nouvel article), mais nous sommes certains que cette platine trouvera preneur pour nombre d'autres applications.

À l'image de nombre de réalisations modernes, l'audio-DAC 2000 requiert 2 tensions d'alimentation, une tension asymétrique de +5 V et une tension symétrique de ± 12 V. Les tensions à fournir par le transformateur sont de ce fait de 8 (voire 9) et de ± 15 V respectivement. En vue de rendre la platine la plus universelle possible nous

avons choisi, en tant que source de la première tension, un transformateur de 2 x 8 (voire 9) volts dont un seul des enroulements secondaires est utilisé. Le second enroulement pourra servir à autre chose.

Le schéma de la figure 1 montre que nous avons profité de l'occasion pour monter sur la platine, outre les 2 transformateurs, également une paire de filtres secteur, L1/C2 et L2/C3, ainsi qu'un condensateur de déparasitage additionnel, C1. Vu d'autre part que tous les transformateurs ne sont pas protégés contre les courts-circuits, nous avons

prévu en outre une paire de porte-fusibles (avec fusible).

La figure 2 vous propose le dessin des pistes et la sérigraphie de l'implantation des composants. La platine a été dessinée pour des transformateurs moulés plats de 4 VA fournis par de nombreuses sociétés, telles que Block, Hahn, Monacor et autres Myrra. Le premier de ces fabricants utilise un brochage différent mais la platine est prévue pour les 2 variantes. La liste des composants donne les modèles convenant à l'audio-DAC 2000.

Signalons, à l'intention de ceux qui voudraient utiliser cette platine pour une autre application, que Block et Monacor disposent d'une version 6 VA, Hahn propose des versions 3, 10 et 16 VA. Ces types diffèrent au niveau de leur hauteur, mais leur largeur reste constante ainsi que le brochage des contacts. Il existe bien entendu d'autres marques, Myrra entre autres, qui doivent avoir des transformateurs pouvant prendre place sur cette platine.

Quelques aspects pratiques en guise de conclusion. La valeur de fusible de 32 mA proposée concerne les transformateurs 4 VA non protégés contre les courts-circuits de Hahn et Monacor. Il faudra, en cas d'utilisation de transformateurs de puissance plus importante, adapter en conséquence ladite valeur. Attention, lors du montage de la pla-

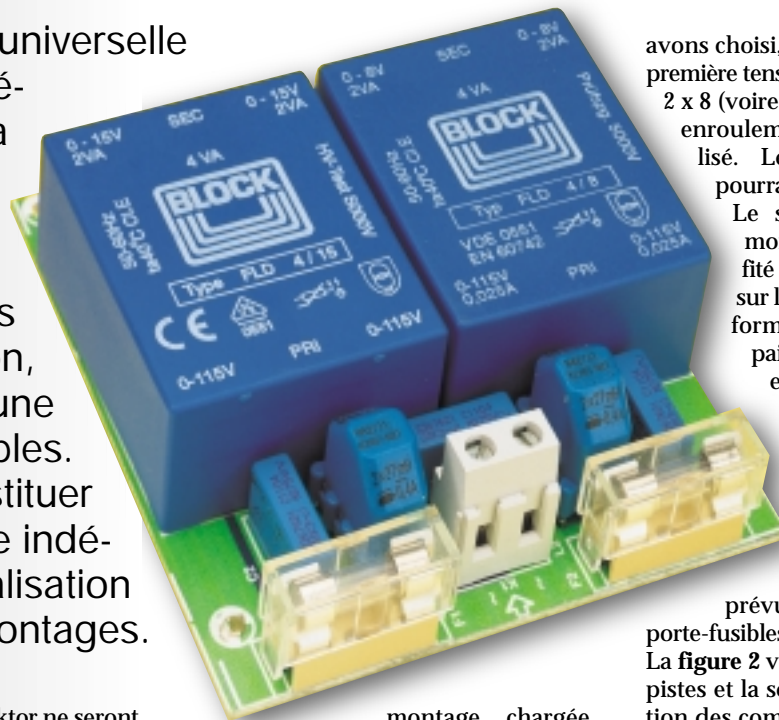


Figure 1. On trouve sur la platine, outre les 2 transformateurs, également une paire de filtres secteur et 2 porte-fusibles (avec leur fusible s'entend).

tine, au respect des règles de sécurité. Les porte-fusibles, qui véhiculent la tension du secteur se trouvant sur le bord de la platine, il faudra disposer cette dernière à 6 mm au minimum de toute partie métallique d'un coffret (pour une protection en classe 2). Il faut en outre monter la platine sur des entretoises de 10 mm de haut au minimum pour garantir une isolation suffisante entre les pistes véhiculant la tension du secteur et la masse.

Une dernière remarque pratique concerne la tension de transfo requise pour l'alimentation 5 V de l'audio-DAC 2000. Si l'on utilise non pas un 2 x 8 mais un 2 x 9 V, le régulateur 7805, IC17, voit sa dissipation augmenter sensiblement, aussi faudra-t-il envisager de le doter d'un petit radiateur.

(000001)

Texte : Sjeff van Rooij
Technique : Ton Giesberts

Figure 2. Le dessin de la platine est celui requis par des transformateurs moulés de 4 VA, mais il est également possible d'y monter d'autres types de transformateurs.

Liste des composants

(pour l'audio-DAC 2000)

Condensateurs :

C1 à C3 = 100 nF/275 VAC X2, au pas de 15 mm

Sels :

L1, L2 = 2 x 27 mH/0,4 A tel que, par exemple, Siemens B82721-K2401-N21

Divers :

K1 = bornier encartable à 2 contacts au pas de 7,5 mm

K2 = bornier encartable à 3 contacts au pas de 5 mm

K3, K4 = bornier encartable à 2 contacts au pas de 5 mm

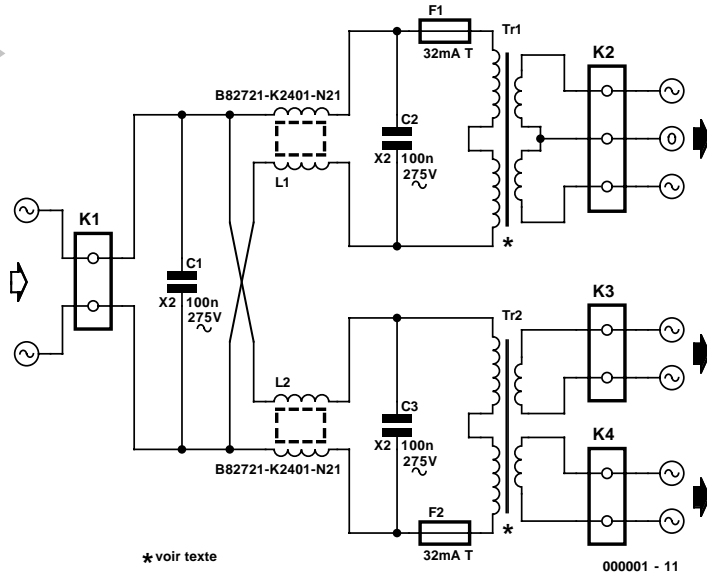
F1, F2 = fusible 32 mA T, avec porte-fusible encartable

Tr1 = transformateur moulé 2 x 15 V/4 VA, tel que, par exemple, Block FLD4/15, Hahn BV UI 302 0165, Monacor FTR-415 *

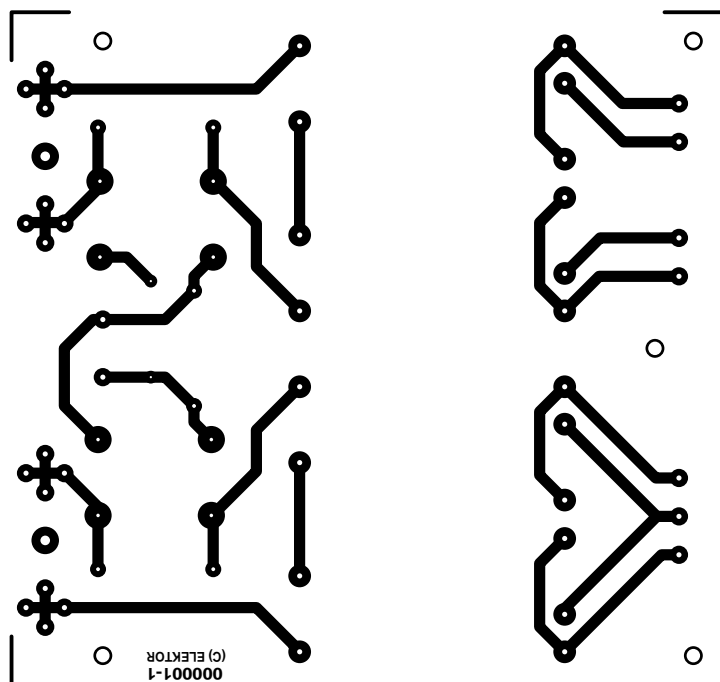
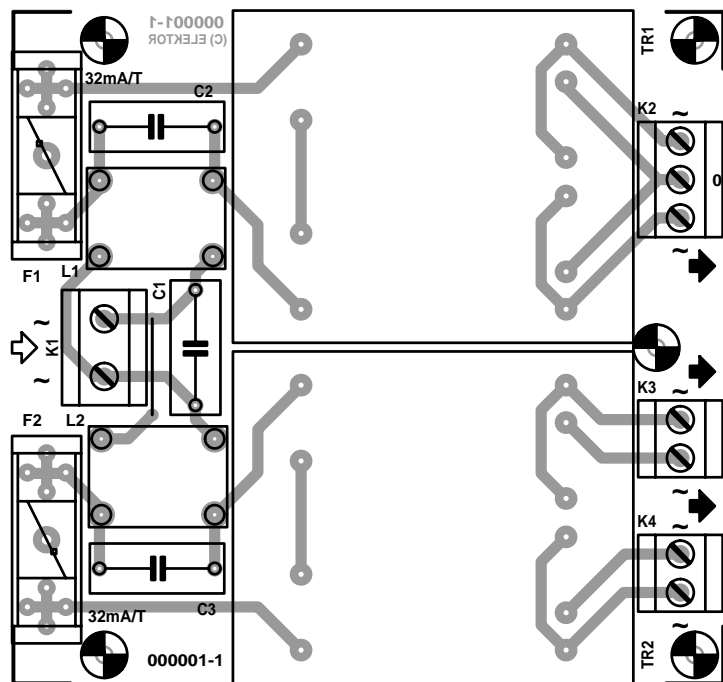
Tr2 = transformateur moulé 2 x 8 (ou 9)V/4 VA, tel que, par exemple, Block FLD4/8 ou FLD4/9, Hahn BV UI 302 0161, Monacor FTR-49

*) Les séries de Hahn et Monacor ne sont pas protégées contre les courts-circuits, ceux de Block le sont elles.

1



2



DF1704

Circuits intégrés
Fonction spéciale, Audio


INFOCARTE 1/2000

Commandes de mode matériel

Dénomination	Numéro de broche	Description
RESV	13	Réservé et partant non utilisée
LRIP	12	Polarité LRCIN LRIP = H: LRCIN = H = Canal Gauche, LRCIN = L = Canal Droit LRIP = L: LRCIN = L = Canal Gauche, LRCIN = H = Canal Droit
CKO	11	CLKO Fréquence de sortie CKO = H: Fréquence CLKO = XT1/2 CKO = L: Fréquence CLKO = XT1
MUTE	15	Commande de silencieux progressif : H = Mute Off, L = Mute On
I ² S	3	Commande de format des données d'entrée I ² S IW1 IWO FORMAT D'ENTRÉE L L L 16 bits, Standard, MSB en premier, justification à droite L L H 20 bits, Standard, MSB en premier, justification à droite L H L 24 bits, Standard, MSB en premier, justification à droite L H H 24 bits, MSB en premier, justification à gauche H L L 16-bits, I ² S H L H 24-bits, I ² S
IWO	4	
IW1	5	
SRO	27	Progressivité du filtre : H = lente, L = raide
OW0	19	Commande de longueur de mot en entrée OW1 OW0 FORMAT DE SORTIE L L 16 bits, MSB en premier L H 18 bits, MSB en premier H L 20 bits, MSB en premier H H 24 bits, MSB en premier
OW1	20	
SF0	17	Sélection du taux d'échantillonnage pour la désaccentuation numérique
SF1	18	
		SF1 SF0 TAUX D'ÉCHANTILLONNAGE L L 44,1 kHz L H Réservé, non utilisé H L 48 kHz H H 32 kHz
DEM	16	Désaccentuation numérique : H = On, L = Off

(Dans les tableaux, H = High, Haut, L = Low, Bas).

DF1704

Circuits intégrés
Fonction spéciale, Audio


INFOCARTE 1/2000

DF1704

Filtre d'interpolation numérique Soundplus™
Stéréo, 24 bits, 96 kHz à octuple suréchantillonnage

Fabricant

Burr-Brown, P.O.
Box 11400, Tuc-
son, AZ 85734,
U.S.A.


Tél. (520) 746-1111, Fax (520) 889-1510.

Internet: <http://www.burr-brown.com/>

Caractéristiques techniques :

- ◆ FILTRE NUMÉRIQUE DESTINÉ A ETRE ASSOCIÉ AU PCM1704, UN AUDIO DAC 24 BITS
- ◆ FILTRE A HAUTES-PERFORMANCES
Atténuation de la bande d'arrêt : -115 dB
Ondulation dans la bande passante : ±0,000 05 dB
- ◆ INTERFACE AUDIO :
Formats d'entrée des données : Standard, justifié à gauche et I²S
Longueur du mot d'entrée : 16, 20, ou 24 bits
Longueur du mot de sortie : 16, 18, 20, ou 24 bits
Fréquence d'échantillonnage : 32 à 96 kHz
- ◆ HORLOGE-SYSTEME :
256 f_S, 384 f_S, 512 f_S, 768 f_S
- ◆ OSCILLATEUR A QUARTZ EMBARQUÉ
- ◆ FONCTIONS PROGRAMMABLES :
Modes de commande par matériel et par logiciel
Réponse de filtrage raide ou à évolution lente
Silencieux progressif
Désaccentuation numérique
Atténuation numérique Gauche/Droite indépendante
- ◆ NE REQUIERT QU'UNE TENSION D'ALIMENTATION DE +5V
- ◆ BOITIER MINIATURE SSOP A 28 BROCHES

Exemple d'application

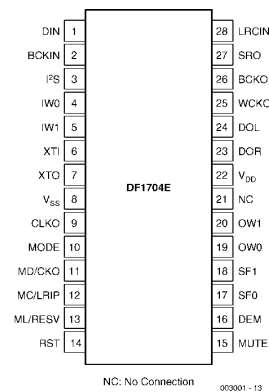
Audio-DAC 2000, Elektor n° 257, 258 et 259,
novembre 1999 à janvier 2000.

Description

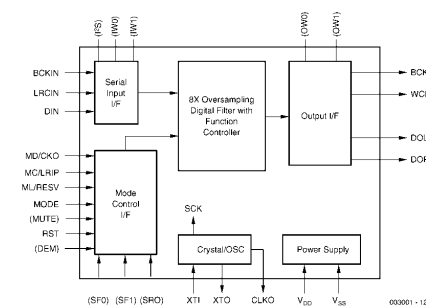
Le DF1704 est un filtre d'interpolation numérique stéréo hautes-performances avec octuple suréchantillonnage conçu pour les applications grand public et

professionnelles haut de gamme. Le DS1704 travaille à 24 bits, 96 kHz et dispose de fonctions programmables par l'utilisateur, dont la réponse de filtrage, la désaccentuation, l'atténuation et les formats d'entrée et de sortie.

Le DF1704 est le compagnon idéal du PCM1704 de Burr-Brown, un convertisseur numérique/analogique à 24 bits. Cette combinaison permet la réalisation de systèmes et sous-ensembles audio de très haut niveau.



Brochage.



Structure interne du DF1704.

DF1704

Circuits intégrés
Fonction spéciale, Audio

INFOCARTE 1/2000

Fonctions programmables

Le DF1704 possède un certain nombre de fonctions programmables, dont la plupart sont accessibles soit par matériel (*Hard*) soit par logiciel (*Soft*).

Le tableau ci-dessus résume les fonctions programmables par l'utilisateur dans chacun des modes de fonctionnement.

FONCTION	LOGICIEL (MODE = H)	MATÉRIEL (MODE = L)	RAZ par défaut (Mode Logiciel)
Sélection du format des données d'entrée	0	0	Format Standard
Sélection de la longueur du mot d'entrée	0	0	16 bits
Sélection de la longueur du mot de sortie	0	0	16 bits
Sélection de la polarité de LRCIN	0	0	Gauche/Droite = H/L
Désaccentuation numérique	0	0	OFF
Silencieux logiciel	0	0	OFF
Atténuation numérique	0	X	0dB, G/D Indépendants
Taux d'échantillonnage de la fonction de désaccentuation	0	0	44,1 kHz
Sélection de l'évolution du filtre	0	0	Sélection filtrage raide
Sélection de la fréquence de sortie	0	0	Identique B la sortie XTI

Légende : 0 = Programmable par l'utilisateur, X = Non disponible

Commande de mode par matériel

Si MODE = BAS, le DF1704 peut être configuré par le biais d'un certain nombre de broches programmables par l'utilisateur. On trouvera, ci-après un résumé succinct des fonctions de broches. Le tableau fournit des informations additionnelles quant au paramétrage des commandes de mode matériel.

Les broches I²S, IWO, et IW1 servent à la sélection du format des données audio d'entrée et de la longueur de mot.

Les broches OW0 et OW1 servent à la sélection de la longueur du mot de donnée de sortie.

La broche DEM sert à valider et inhiber la fonction de désaccentuation numérique. Seuls les taux d'échantillonnage de 32, 44,1 et 48 kHz peuvent subir une désaccentuation.

Les broches SF0 et SF1 servent à la sélection du taux d'échantillonnage de la fonction de désaccentuation.

La broche SRO sert à la sélection de la réponse du filtre qui peut être soit raide (*sharp*), soit progressive (*slow roll-off*).

La broche MUTE sert à la validation ou l'inhibition de la fonction de silencieux progressif (*soft mute*).

La broche CKO sert à la sélection de la fréquence présente à la broche LKO, soit XTI soit XTI/2.

La broche LRIP sert à la sélection de la polarité utilisée pour l'horloge gauche/droite d'entrée, LRCIN.

Pour terminer, la broche RESV n'est pas mise à contribution dans le DF1704 actuel, elle est réservée pour des fonctions potentielles ultérieures.

DF1704

Circuits intégrés
Fonction spéciale, Audio

INFOCARTE 1/2000

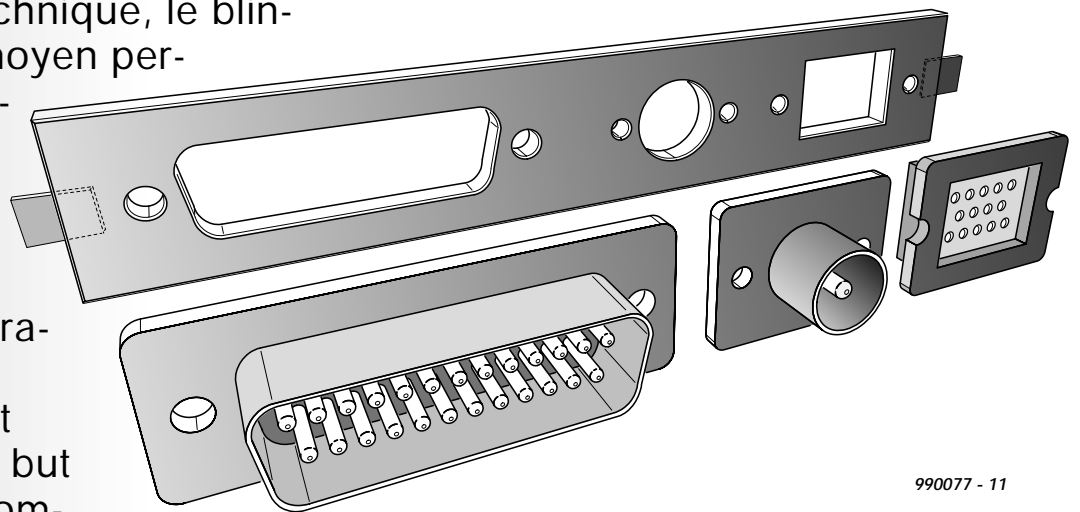
Brochage			
Broche	Nom	E/S	Description
1	DIN	IN	Entrée de données audio sérielles ⁽³⁾
2	BCKIN	IN	Entrée d'horloge de bit pour données audio sérielles ⁽³⁾
3	I ² S	IN	Sélection du format des données d'entrée ^(2, 4)
4	IWO	IN	Sélection de mot de donnée audio d'entrée ^(2, 4)
5	IW1	IN	Sélection de mot de donnée audio d'entrée ^(2, 4)
6	XTI	IN	Entrée oscillateur/entrée d'horloge externe
7	XTO	OUT	Sortie oscillateur
8	VSS	—	Masse numérique
9	CLKO	OUT	Sortie d'horloge-système tamponnée
10	MODE	IN	Sélection du mode de commande (H : logiciel, L : matériel) ⁽¹⁾
11	MD/CKO	IN	Sélection de la fréquence de l'horloge de sortie/de l'entrée des données de commande ^(1, 5)
12	MC/LRIP	IN	Sélection de la polarité de LRCK/horloge des données de commande ^(1, 5)
13	ML/RESV	IN	Verrou de données de commande/réserve ^(1, 5)
14	RST	IN	Lorsque cette broche se trouve au niveau BAS, le filtre est forcé en mode de RAZ. ⁽¹⁾
15	MUTE	IN	Commande du silencieux ^(1, 4)
16	DEM	IN	Commande de la désaccentuation ^(2, 4)
17	SFO	IN	Choix du taux d'échantillonnage de la désaccentuation ^(2, 4)
18	SF1	IN	Choix du taux d'échantillonnage de la désaccentuation ^(2, 4)
19	OW0	IN	Sélection du format et du mot de donnée audio de sortie ^(2, 4)
20	OW1	IN	Sélection du format et du mot de donnée audio de sortie ^(2, 4)
21	NC	—	Non connecté
22	V _{DD}	—	Alimentation numérique, +5V
23	DOR	OUT	Rch, Sortie de données audio sérielles
24	DOL	OUT	Lch, Sortie de données audio sérielles
25	WCKO	OUT	Horloge de mot pour la sortie audio sérielle
26	BCKO	OUT	Horloge de bit pour la sortie audio sérielle
27	SRO	IN	Sélection de la réponse du filtre ^(2, 4)
28	LRCIN	IN	Entrée d'horloge G/D (f _G) pour les données audio sérielles ⁽³⁾

NOTES: ⁽¹⁾ Broches 10 à 15; entrée B trigger de Schmitt avec résistance de forçage au niveau haut.
⁽²⁾ Broches 3 à 5, 16 B 20, 27; entrée à trigger de Schmitt avec résistance de forçage au niveau bas.
⁽³⁾ Broches 1, 2, 28; entrée à trigger de Schmitt
⁽⁴⁾ Broches 3 à 5, 15 à 20, 27; ces broches sont invalides lorsque la broche MODE (10) se trouve au niveau haut (H).
⁽⁵⁾ Broches 11 à 13; ces broches remplissent des fonctions différentes en fonction de l'état H ou L de la broche MODE (10).

le blindage = un impératif croissant

comment obtenir de bons résultats à l'aide de techniques simples

En électro-technique, le blindage est le moyen permettant d'empêcher un rayonnement électromagnétique gênant de parasiter d'autres appareils. Cet article a pour but de montrer comment il est possible, avec des moyens simples, de réaliser un blindage efficace autour de circuits électroniques.



990077 - 11

Figure 1. On pourra réaliser un transfert blindé à peu de frais à l'aide d'une languette métallique, de fiches et de connecteurs. On veillera à une bonne conduction électrique entre les différents composants.

Le blindage est aujourd'hui, ne serait-ce seulement qu'en raison des normes de CEM (Compatibilité ÉlectroMagnétique) récemment édictées, la pierre d'achoppement de tout électronicien, même amateur. En raison de l'évolution rapide du numérique dont les fréquences ne cessent d'augmenter, le parasitage électromagnétique devient un problème de plus en plus grave. Ceci explique que l'on -l'état c'est nous- pose des exigences de plus en plus strictes lorsqu'il s'agit du blindage, celui des montages électroniques en particulier. Comme vous le savez sans doute, il n'est plus permis de mettre sur le marché des produits électroniques et électriques ne possédant pas un blindage adéquat.

LA SOURCE

La circulation d'un courant alternatif dans un conducteur ou la présence

d'un potentiel alternatif entre 2 conducteurs se traduit par la naissance d'un rayonnement électromagnétique. Il est, partant, inévitable de constater l'existence de ce type de rayonnement sur tout circuit électronique.

Ce sont tout particulièrement les microprocesseurs qui travaillent à des fréquences d'horloge de plusieurs MHz et qui placent des signaux rectangulaires présentant de telles fréquences sur, par exemple, un bus système, qui contribuent très sensiblement à la production de rayonnements électromagnétiques. Sachant que ces fréquences de commutation élevées sont indispensables à l'exécution des instructions requises, et que l'on ne saurait se passer de bus pour la transmission des signaux de la source vers le destinataire, toute électronique numérique est, par définition, une

source de parasites. Une fois que l'on s'est convaincu de cette évidence, il ne reste qu'une solution : emballer le circuit (ou du moins la partie qui génère les signaux parasites) le plus efficacement possible dans un blindage capable d'arrêter le rayonnement. Il est possible, en pratique, d'arriver à, voire même de garantir, un bon résultat à l'aide de moyens simples.

METTONS LE PAQUET

Le problème de la CEM est aujourd'hui, dans l'industrie électro-technique, un phénomène dont il est tenu compte. Dès le début même de la conception on prend des mesures destinées à minimiser la génération de rayonnement. On réduit la longueur des pistes au strict nécessaire, on groupe les composants critiques et on développe le boîtier en respectant des normes parfaitement définies.

Lorsqu'il s'agit de fréquences inférieures à 10 MHz, c'est la circulation de courants qui est la source principale de problèmes. Il faut, en cas d'utilisation de conducteurs pour relier les différents sous-ensembles d'un système, utiliser alors du câble blindé efficace. Il est essentiel de relier le blindage du câble au potentiel de terre du montage. Il faudra doter le câble non blindé d'un filtre adéquat. On trouve dans le commerce des filtres spéciaux et des noyaux de ferrite conçus à cette intention. Un câble non blindé ou introduisant un filtrage insuffisant se comporte comme une antenne et ne fait qu'aggraver le problème. L'une des techniques simples permettant de doter un boîtier d'un point de transfert blindé consiste à faire appel à des connecteurs métalliques, des connecteurs de transfert et des fiches. On positionnera ces matériaux de connexion sur une languette (*strip*) conductrice (en aluminium par exemple), en veillant à ce que cette plaquette soit bien mise à la terre (cf. **figure 1**). On pourra, dans le cas d'un boîtier en plastique, placer une telle plaquette sur l'extérieur du boîtier.

UN BOÎTIER ADÉQUAT

Il n'y a, pour les signaux de fréquence supérieure à 10 MHz, qu'un seul remède : emballer la source de signal. Il suffit, dans le cas de signaux de 40 MHz ou plus, d'une fine couche métallique conductrice pour réaliser un blindage suffisant. On pourra utiliser un boîtier recouvert à l'intérieur d'une couche conductrice ou d'une fine épaisseur de métal. Un boîtier hermétique en aluminium convient parfaitement. Autre option, utiliser l'un ou

l'autre plastique conducteur voire emballer le circuit dans une housse conductrice spécialement prévue à cet effet.

Il est important, pour un résultat optimal, que tous les orifices percés dans le boîtier, ceux destinés au passage des câbles blindés par exemple, soient parfaitement étanchéifiés pour empêcher tout passage de rayonnement.

Il existe du matériau de transfert spécial prenant la forme de languettes élastiques dotées de matériau conducteur (cf. **figure 2**).

On utilisera, dans le cas d'un refroidissement forcé, l'un ou l'autre ventilateur prévu à cette intention et assurant un blindage suffisant.

Les fréquences très faibles, 50 Hz ou moins, exigent elles la prise de mesures sévères. Cela est principalement le cas pour des applications militaires. Il n'est pas nécessaire, normalement, d'en tenir compte.

BLINDAGE PAR TÔLE

On obtient le meilleur résultat si l'on place le blindage le plus près possible de la source. On disposera, si cela est possible, un enclôt métallique autour de la platine ou, sur la platine, des composants source de rayonnement (tant sur le dessous que le dessus de la platine). Il est important de penser, lors de la conception du blindage, à se laisser, pour plus tard, une possibilité de modification ou de dépannage. On trouve, dans le commerce des enclos à couvercle cliquant qui apportent une solution à ce problème. Un circuit très compact pourra y trouver place en entier, pour des projets plus grands on s'en servira pour le blindage d'une par-

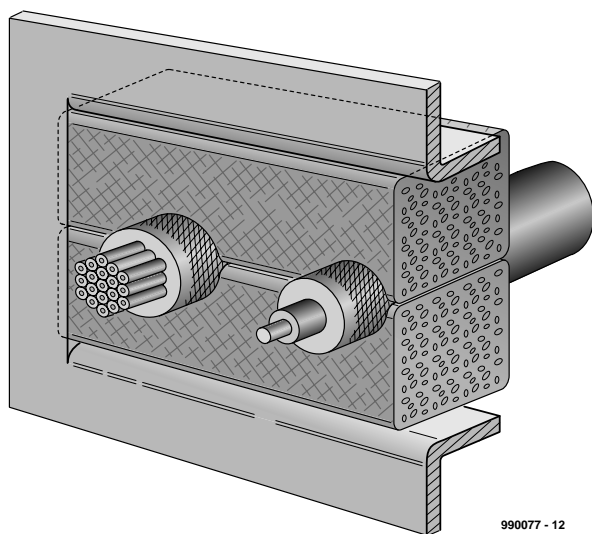
tie du montage, le microcontrôleur et sa mémoire RAM/ROM par exemple. Le blindage métallique pourra être fixé par le biais de picots placés aux endroits requis sur la platine et reliés à la masse. La **figure 3** donne un exemple de réalisation pratique.

Il est également possible, si nécessaire, d'isoler des parties rayonnantes du circuit à l'aide d'une fine épaisseur de tôle (cloisonner). Cette précaution est primordiale lorsque l'on risque une interférence néfaste entre plusieurs parties du circuit. Quelques exemples pratiques : cloisonnement entre la section HF d'un récepteur et son circuit de pilotage par microprocesseur.

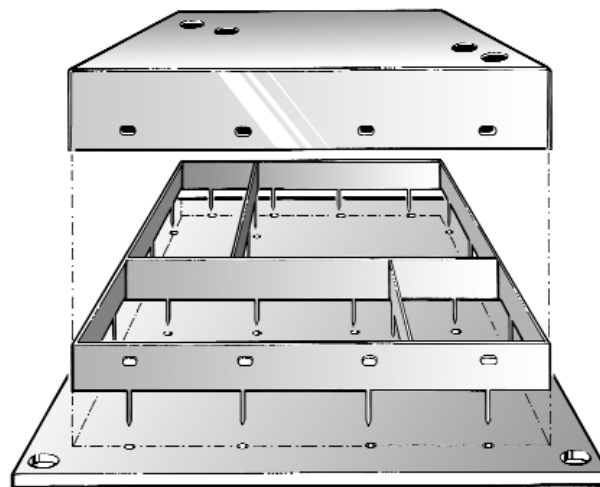
Les niveaux d'atténuation à obtenir dépendent des normes nationales et internationales. Des organismes tels que les CISPR, FCC et VDI ont traduit ces normes sous la forme de réglementations et de fiches de normes. Il est quasiment impossible pour un amateur de s'assurer que son montage répond à ces spécifications. Cependant, si l'on prend à coeur les recommandations données tout au long de cet article, on peut se persuader que l'on a fait son possible. Les professionnels démarreront toujours avec une construction optimale pour ensuite, dans un environnement conditionné, procéder à des mesures pour voir quelles mesures de blindage peuvent être supprimées tout en assurant, à un coût acceptable, le blindage requis par les normes.

(990077)

Cet article est basé sur une brochure de Holland Shielding Systems B.V. de Dordrecht (NL).



990077 - 12



990077-13

Figure 2. On pourra réaliser un passage des câbles à l'aide d'une languette conductrice élastique. Il est important d'assurer une liaison électrique entre le blindage du câble et le boîtier.

Figure 3. On pourra isoler certaines parties du montage par le biais de cloisons et d'enclôts à couvercle. On veillera à relier le blindage à la masse, par le biais, par exemple, de picots eux-mêmes reliés à la masse.

Cet article explique comment écrire des programmes en assembleur PIC 16C84 en utilisant un compilateur écrit pour le système Windows 95/98. Le compilateur PIC permet d'écrire des programmes dans un langage de haut niveau et fournit le code assembleur nécessaire.

Le code peut être modifié, assemblé ou simulé en utilisant le logiciel (libre) Microchip MPLAB.

par Roger Thomas

compilateur pour PIC 16C84

avec optimisation de code

Dispositifs principaux du compilateur

- pour le processeur PIC 16C84
- manuel complet (56 pages) sur disque
- trois exemples décrits sur disque
- syntaxe semblable à Visual Basic et Pascal
- compatible Windows 95/98, aucunes DLL requises
- génère un code assembleur pour le logiciel Microchip MPLAB (logiciel libre)
- variables : booléenne, octet, texte
- opérations arithmétiques : +, -, /, *, mod
- formats numériques : décimal, hexadécimal, binaire, caractère
- fonctions booléennes : =, >, <, >=, <=, <>
- opérateurs booléens : AND, OR, XOR
- commandes du compilateur : if...then...else, select/case, while...loop, table, read, write/read, EEPROM, procedure, directive, ASM directive, input, output, alias, pin-name, RTTC, prescaler, wait, picfuse,
- gestionnaire d'équations
- optimisateur de code
- messages d'erreur
- gestion des interruptions

Il est très avantageux d'utiliser un compilateur. L'écriture d'un programme PIC (anagramme dont la signification est sujette à discussion) à l'aide d'un vocabulaire proche de l'anglais est plus facile que sa programmation directe en langage assembleur. Le temps mis pour écrire et tester un logiciel est généralement beaucoup plus court avec un langage compilé, et en guise de preuve le compilateur PIC Windows lui-même a été écrit en utilisant un compilateur.

Le compilateur PIC n'est basé sur aucun langage de haut niveau mais comporte des éléments de Pascal et de Visual Basic. De plus, le compila-

teur est souple et s'adapte à différentes syntaxes de programmation. Il est écrit pour l'environnement Windows, ce qui facilite l'utilisation du logiciel. Comme tous les codes PIC utilisés sont définis à l'intérieur du compilateur, aucune initialisation ou fichier en-tête ne sont nécessaires.

Le compilateur produit un code assembleur directement depuis le programme source de haut niveau, permettant au programmeur de ne pas se soucier des dédales de la programmation assembleur. Il peut aussi avoir une fonction éducative pour les apprentis programmeurs assembleur qui verront comment des ordres de haut niveau,

faciles à comprendre, sont traduits en code assembleur PIC équivalent.

Les fichiers résultants en code assembleur contiennent à la fois le programme original de haut niveau (commenté) et le code assembleur PIC prêt pour l'assemblage (ou la simulation) par le logiciel Microchip MPLAB version 4.12 ou supérieure. Disposer simultanément des codes source et assembleur facilite la mise au point du programme. Programmer en langage compilé n'empêche pas de modifier ou de compléter le code assembleur lorsqu'on utilise le logiciel MPLAB. Celui-ci est disponible gratuitement sur le site de Microchip : www.microchip.com.

Le coût (en terme de programmation) de l'utilisation d'un code compilé peut être la réduction de la vitesse d'exécution, parce que le programme en code machine n'est peut-être pas aussi efficace qu'un programme écrit directement en langage assembleur. Ce n'est pas le cas avec ce compilateur PIC, qui produit dans la plupart des circonstances le code le plus rapide possible. Il y a très peu de surcharge de compilation dans le code assembleur en termes de variables supplémentaires requises ou de multiplication d'instructions assembleur. Le seul code supplémentaire ajouté concerne les commandes booléennes et arithmétiques. L'arithmétique fonctionne soit en 8 bits (non signé) ou en 16 bits (non signé). Le compilateur a besoin de plusieurs octets pour conserver les résultats arithmétiques, qui sont baptisés `_STACxx` dans le fichier assembleur résultant.

Le compilateur traduit le programme

source en deux passes. La première crée une liste de déclarations de procédures parce qu'il peut rencontrer un appel à une procédure avant d'avoir détecté la déclaration de celle-ci. Lors de la seconde passe, les appels à procédures sont affectés aux déclarations de procédures correspondantes.

La syntaxe du compilateur n'est pas sensible à la typographie (majuscules/minuscules), mais le logiciel MPLAB peut l'être, c'est l'option sélectionnée dans le fichier hexadécimal projet. Pour cette raison, tous les noms de procédures et de variables sont rédigés en majuscules dans le fichier de sortie assembleur.

Pour pouvoir utiliser le compilateur avec n'importe quel microcontrôleur PIC, le compilateur n'impose aucune restriction de code. La tâche est laissée à l'assembleur MPLAB qui vérifie la taille du programme et peut plus aisément produire une cartographie de l'utilisation de la mémoire et des fichiers de références croisées.

Utilisation du compilateur

Utiliser un éditeur de textes (comme Notepad, WordPad ou l'éditeur de MPLAB) pour créer le programme source de haut niveau ; sauvegarder le fichier texte avec une extension .psf (*PIC Source File*). S'assurer que le fichier sauvegardé est en mode texte seulement et ne contient aucun texte enchâssé renfermant de l'information. À l'inverse d'un programme assembleur qui nécessite un colonnage très strict (étiquettes, mnémoniques, opérateurs, commentaires), un programme de haut niveau utilise librement l'espace pour indenter le programme. Ces espacements n'ont aucune signification pour l'exécution du programme et sont ignorés par le compilateur. Mais leur utilisation facilite la lecture et la mise au point du programme.

Utilisation de l'interface

Le compilateur PIC Windows est très facile à utiliser et, mis à part le chargement et la sauvegarde des fichiers, il s'occupe de tout !

Comme le montre la **figure 1**, la barre des tâches comporte un certain nombre d'icônes.

load (chargement) – appuyez sur le bouton « load » et une boîte de dialogue répertoire affiche tous les fichiers source (extensions .psf) du répertoire, sélectionnez et chargez le fichier source adéquat. Par défaut, le compilateur va s'ouvrir dans le répertoire le

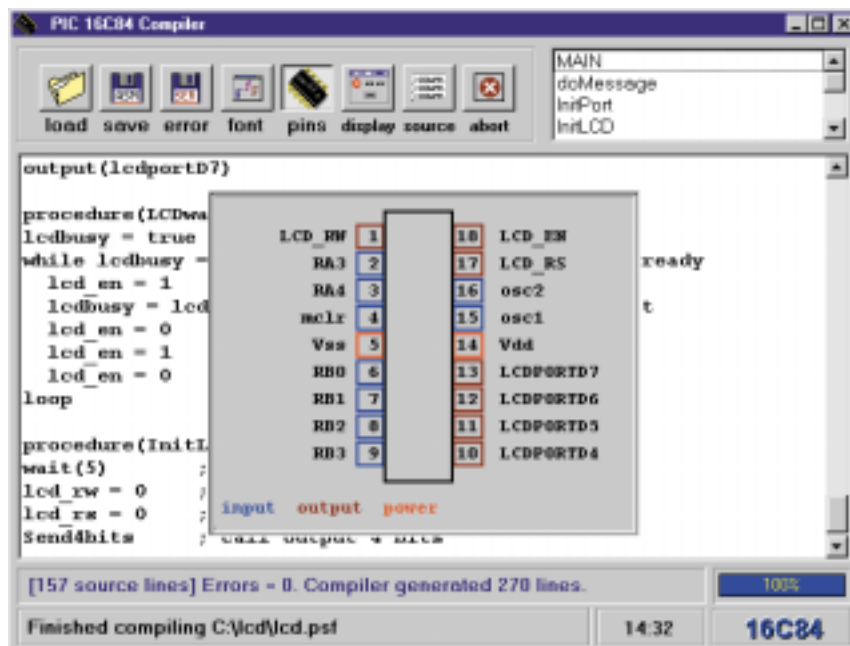


Figure 1. La fenêtre du compilateur. Notez que les fonction des broches du PIC sont visibles dans leur ensemble.

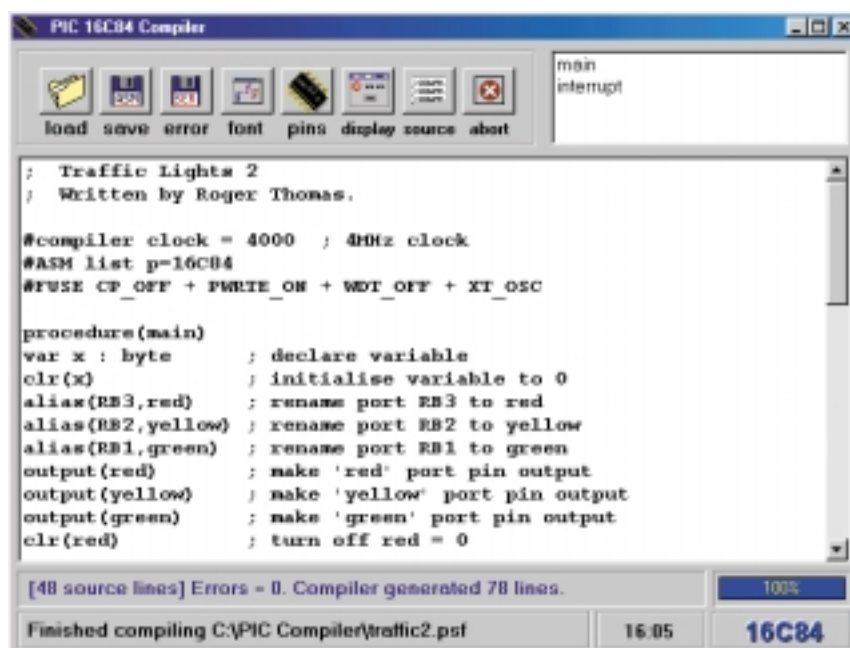
plus récemment utilisé. Lorsque le compilateur est utilisé pour la première fois, le répertoire choisi est celui hébergeant le compilateur.

save (sauvegarde) – après une compilation réussie, sauvegardez le fichier source assembleur (même nom de fichier mais avec une extension .asm) en appuyant sur ce bouton. Le fichier en question est sauvegardé dans le

même répertoire que le fichier source. Le fichier de sortie assembleur contient le code PIC supplémentaire requis puisque le compilateur va automatiquement y ajouter les routines de support.

error (erreur) – ce bouton enregistre le fichier d'erreurs dans le même répertoire que le fichier source. Le fichier a le même nom de fichier mais avec

Figure 2. Un autre exemple du compilateur PIC en action. Ici, un programme pilote de feux de signalisation, est en cours d'écriture. Notez les noms de procédure dans la fenêtre en haut à droite.



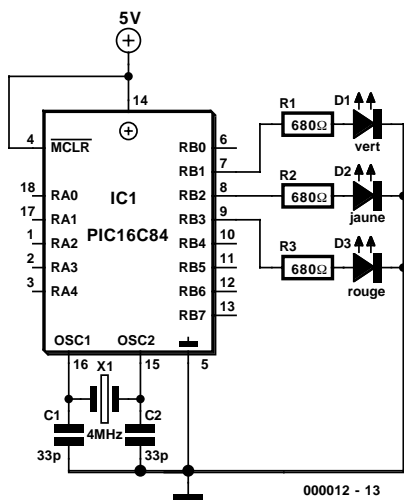


Figure 3. Le matériel de démonstration des feux tricolores.

une extension `.ser` (*source error* = erreur source). Il contient tous les messages d'erreur (avec le numéro de ligne dans le programme source) mais pas de code source ou assembleur.

font (police) – pour changer la police ou la taille de la police du texte affiché sur l'écran, appuyez sur le bouton « font ». Une boîte de dialogue police s'affiche, dans laquelle vous pouvez choisir la police et sa taille.

pins (broches) – affiche les noms des broches du PIC et les broches des ports d'entrée ou de sortie avec leur code couleur.

display (affichage) – sur option, le compilateur s'arrête et affiche la boîte de dialogue messages lorsqu'il y a des messages d'erreur de compilation insérés dans le fichier de sortie assembleur (par défaut, affichage du message d'erreur). Lors du premier passage, tout message d'erreur est affiché quelle que soit l'option choisie.

source (source) – permet de choisir si les lignes source doivent être insérées dans le fichier assembleur comme commentaires (par défaut, insertion du code source). Les lignes source ne contenant qu'un commentaire sont toujours insérées.

abort (arrêt) – arrête la compilation.

À côté des huit boutons se trouve une liste de tous les noms de procédure utilisés dans le programme. Pour trouver et afficher une procédure particulière, sélectionnez-la dans la liste (le nom sera brièvement surligné) en cliquant sur le bouton gauche de la sou-

ris. Le texte affiché se déplace et révèle la procédure correspondante (figure 2).

La procédure principale du programme PIC est appelée « main » (*main* = principale). L'exécution du programme débute avec cette procédure, qui est appelée à chaque démarrage du contrôleur PIC. Le programme d'application PIC boucle généralement en permanence une fois l'initialisation réalisée, en attente d'événements. Il est extrêmement rare qu'on utilise un programme PIC pour une seule action.

Optimisation du code compilé

Après la compilation sans erreur d'un segment du programme source, l'optimiseur contrôle le programme assembleur pour chercher le code redondant. Ce code redondant prend généralement la forme d'initialisations inutiles ou de validations des diverses balises d'état PIC.

Par exemple, la procédure de sortie d'une partie de la gestion de feux de trafic donnera ces résultats :

(Extraits du code source original)

```
green = 0 ; turn green off
yellow = 0 ; turn yellow off
red = 1; turn on red
```

```
; code not optimised
; green = 0 ; turn green off
MOVLWH' 00'
BTFSS _STATUS, _Z
BSF PORTB, GREEN
BTFSC _STATUS, _Z
BCF PORTB, GREEN
; yellow = 0 ; turn yellow off
MOVLWH' 00'
BTFSS _STATUS, _Z
BSF PORTB, YELLOW
BTFSC _STATUS, _Z
BCF PORTB, YELLOW
; red = 1 ; turn on red
MOVLWH' 01'
BTFSS _STATUS, _Z
BSF PORTB, RED
BTFSC _STATUS, _Z
BCF PORTB, RED
```

Dans le code non optimisé, le compilateur a calculé l'équation après le signe égal et affecté le bit selon le résultat : zéro ou un. L'optimiseur vérifie le code et trouve qu'il garde une valeur fixe puisque les résultats ne changent jamais ; il élimine donc les calculs intermédiaires :

```
; code optimised
; green = 0 ; turn green off
BCF PORTB, GREEN
```

```
; yellow = 0 ; turn yellow off
BCF PORTB, YELLOW
; red = 1 ; turn on red
BSF PORTB, RED
```

Interruptions PIC

Lorsque se produit une interruption, le compteur du programme est chargé avec l'adresse de l'emplacement 4, qui contient le code sauvegardant l'état du programme et appelle la procédure de gestion de l'interruption. Une fois l'interruption traitée (« interrupt »), le programme PIC exécute une instruction « retour » (*return from interrupt* = retour après interruption). Le programme source n'a pas besoin de rétablir l'autorisation d'interruption, l'ordre est déjà donné automatiquement par l'instruction « retour ». La routine de gestion de l'interruption doit être appelée « interrupt ».

L'utilisation des interruptions rend un programme PIC plus efficace, l'autre solution ayant été d'interroger en permanence les balises pour contrôler si un événement particulier ne s'est pas produit.

Il existe quatre sources d'interruptions que le gestionnaire d'interruptions du PIC 16C84 doit traiter :

1. interruption externe sur la broche RB0 ;
2. interruption lors de changement sur les broches RB4-RB7 ;
3. débordement du minuteur RTCC ;
4. affecté à la fin de l'écriture dans l'EEPROM.

L'interruption « fin de l'écriture dans l'EEPROM » est prise en compte par la fonction « écrire dans l'EEPROM » (*writeEEPROM*).

Pour valider les interruptions, utilisez la commande « `irq_enable = true` ». La source individuelle d'interruptions doit être sélectionnée avant de lancer cette commande.

Pour invalider toutes les interruptions, utilisez la commande « `irq_enable = false` ». Cette commande agit globalement par l'intermédiaire de la balise GIE (*Global Interrupt Enable* = valider globalement les interruptions), indépendamment de l'interruption individuelle utilisée.

La procédure de gestion des interruptions a besoin de quelques octets de stockage intermédiaire, et la routine de gestion de l'interruption doit avoir sa zone de stockage variable propre car elle ne peut partager la mémoire avec le reste du programme.

Comme l'interruption peut arriver n'importe quand, il se peut qu'avec l'arithmétique à 16 bits, elle se produise en plein milieu d'une procédure arithmé-

tique. L'attribution d'une variable de 16 bits à une autre variable de 16 bits nécessite de multiples instructions pour déplacer la valeur de l'octet inférieur, puis de l'octet supérieur. Si une interruption se produit au cours du processus, la variable peut conserver l'ancienne valeur de l'octet inférieur et accepter la nouvelle valeur de l'octet supérieur. C'est la raison pour laquelle le compilateur n'autorise pas le traitement des opérations arithmétiques à 16 bits par la procédure de gestion des interruptions.

Si une seule source d'interruptions a été validée, il n'est pas nécessaire de contrôler les balises de validation des interruptions individuelles. D'une manière générale, il est préférable de définir une procédure de gestion des interruption la plus petite et la plus rapide à exécuter possible, en utilisant des équations simples. Notez qu'aucune autre procédure ne peut être appelée depuis la procédure de gestion des interruptions.

Il est aussi préférable de faire une copie des variables utilisées par le gestionnaire d'interruptions et de travailler avec cette copie. Pointer sur une variable que le gestionnaire d'interruptions utilise directement peut aboutir à des résultats imprévisibles. Par exemple, si la variable « x » est modifiée par le gestionnaire d'interruptions, le programme ci-dessous peut ne pas donner les résultats attendus. La valeur de x peut avoir été modifiée entre la première et la deuxième commande de comparaison, aucune instruction ne sera donc exécutée.

```
var x : byte
procedure(main)
if x >=6 and x <= 10 then
; x = 4
begin
    statement
end
; l'interruption se produit maintenant
if x >= 0 and x <=5 then
; x changes to 6
begin
    statement
end
```

Si une variable doit être incrémentée ou décrétementée ou mise à zéro par le programme de gestion de l'interruption, utilisez les instructions inc(x) ou dec(x) ou clr(x) qui se compilent en une seule instruction assembleur.

Exemples de programmes

Pour démontrer les avantages de l'utili-

sation d'un compilateur et pour rendre plus claire la syntaxe du langage, le fichier de documentation contient quelques exemples de programmes qui illustrent les diverses commandes du compilateur. Ces programmes étant

présentés pour leur valeur didactique, ils ne présentent pas forcément la meilleure programmation. Notez que certaines des lignes de commentaires ont été supprimées et le fichier assembleur nettoyé pour la publication.

Listage 1. Feux de signalisation (1) Programme source

```
; Traffic Lights 1
; Written by Roger Thomas.

#compiler clock = 4000 ; 4MHz clock
#ASM list p=16C84
#FUSE CP_OFF + PWRTE_ON + WDT_OFF + XT_OSC

var x : word ; create 16 bit variable
var y : byte ; create 8 bit variable

procedure(main)
alias(RB3, red) ; rename port RB3 to red
alias(RB2, yellow) ; rename port RB2 to yellow
alias(RB1, green) ; rename port RB1 to green
output(red) ; make 'red' port pin output
output(yellow) ; make 'yellow' port pin output
output(green) ; make 'green' port pin output
clr(red) ; turn off red = 0
clr(yellow) ; turn off yellow = 0
clr(green) ; turn off green = 0

clr(x) ; initialise = 0
clr(y) ; initialise = 0

while true
inc(x) ; x = x + 1
if x = 1500 then
begin
inc(y) ; y = y + 1
clr(x) ; x = 0
end
else
begin
if (y >= 0) AND (y<= 49) then
begin
red = 1 ; turn on red
yellow = 0 ; turn off yellow
green = 0 ; turn off green
end

if (y >= 50) AND (y<= 75) then
begin
red = 1 ; turn on red
yellow = 1 ; turn on yellow
green = 0 ; turn off green
end

if (y >= 76) AND (y<= 110) then
begin
red = 0 ; turn off red
yellow = 0 ; turn off yellow
green = 1 ; turn on green
end

if (y >= 111) AND (y<= 130) then
begin
red = 0 ; turn off red
yellow = 1 ; turn on yellow
green = 0 ; turn off green
end

if y = 131 then
begin
clr(x)
clr(y)
end
end
loop
```

```

;
;                               16C84
;          RA2  1  | i  i  | 18  RA1
;          RA3  2  | i  i  | 17  RA0
;          RA4  3  | i  i  | 16  osc2
;          mclr 4  | i  i  | 15  osc1
;          Vss  5  | p  p  | 14  Vdd
;          RB0  6  | i  i  | 13  RB7
;          GREEN 7  | o  i  | 12  RB6
;          YELLOW 8 | o  i  | 11  RB5
;          RED   9  | o  i  | 10  RB4
;
_PCL EQU H' 02'
_STATUS EQU H' 03'
_C EQU H' 00'
_Z EQU H' 02'
_RPO EQU H' 05'
PORTB EQU H' 06'
_PCLATH EQU H' 0A'
_INTCON EQU H' 0B'
IRQ_ENABLE EQU H' 07'
_STACK0 EQU H' 0C'
_STACK1 EQU H' 0D'
_STACK2 EQU H' 0E'
_STACK3 EQU H' 0F'
_STACK4 EQU H' 10'
_STACK5 EQU H' 11'
_STACK6 EQU H' 12'
_STACK7 EQU H' 13'
_STACK8 EQU H' 14'
_STACK9 EQU H' 15'
X EQU H' 16'
XH EQU H' 17'
Y EQU H' 18'
RED EQU H' 03'
YELLOW EQU H' 02'
GREEN EQU H' 01'

ORG 0

goto MAIN

; Traffic Lights 1
; Written by Roger Thomas.

list p=16C84
_config H' 3FF9'

; var x : word ; create 16 bit variable
; var y : byte ; create 8 bit variable

MAIN
; alias(RB3, red) ; rename port RB3 to red
; alias(RB2, yellow) ; rename port RB2 to yellow
; alias(RB1, green) ; rename port RB1 to green
; output(red) ; make 'red' port pin output
BSF _STATUS, _RPO
BCF PORTB, RED
; output(yellow) ; make 'yellow' port pin output
BCF PORTB, YELLOW
; output(green) ; make 'green' port pin output
BCF PORTB, GREEN
; clr(red) ; turn off red = 0
BCF _STATUS, _RPO
BCF PORTB, RED
; clr(yellow) ; turn off yellow = 0
BCF PORTB, YELLOW
; clr(green) ; turn off green = 0
BCF PORTB, GREEN

; clr(x) ; initialise = 0
CLRF X
CLRF XH
; clr(y) ; initialise = 0
CLRF Y

; while true
_WHI LE0
; inc(x) ; x = x + 1
INCF X, F

```

```

BTFSC _STATUS, _Z
INCF XH, F
; if x = 1500 then
_IF1
MOVF X, W
MOVWF _STACK0
MOVF XH, W
MOVWF _STACK1
MOVLW H' FF'
MOVWF _STACK2
MOVLW H' DC'
SUBWF _STACK0, F
BTFSS _STATUS, _Z
CLRF _STACK2
MOVLW H' 05'
SUBWF _STACK1, F
BTFSS _STATUS, _Z
CLRF _STACK2
MOVF _STACK2, W
MOVWF _STACK0
MOVWF _STACK1
BTFSC _STATUS, _Z
GOTO _ELSE1
; begin
; inc(y) ; y = y + 1
INCF Y, F
; clr(x) ; x = 0
CLRF X
CLRF XH
; end
; else
GOTO _END1
_ELSE1
; begin
; if (y >= 0) AND (y <= 49) then
_IF2
MOVF Y, W
MOVWF _STACK0
MOVLW H' 00'
SUBWF _STACK0, W
CLRW
BTFSC _STATUS, _C
ADDLW H' FF'
MOVWF _STACK4
MOVF Y, W
SUBLW H' 31'
CLRW
BTFSC _STATUS, _C
ADDLW H' FF'
ANDWF _STACK4, W
BTFSC _STATUS, _Z
GOTO _ELSE2
; begin
; red = 1 ; turn on red
BSF PORTB, RED
; yellow = 0 ; turn off yellow
BCF PORTB, YELLOW
; green = 0 ; turn off green
BCF PORTB, GREEN
; end
; if (y >= 50) AND (y <= 75) then
_ELSE2
_IF3
MOVF Y, W
MOVWF _STACK0
MOVLW H' 32'
SUBWF _STACK0, W
CLRW
BTFSC _STATUS, _C
ADDLW H' FF'
MOVWF _STACK4
MOVF Y, W
SUBLW H' 4B'
CLRW
BTFSC _STATUS, _C
ADDLW H' FF'
ANDWF _STACK4, W
BTFSC _STATUS, _Z
GOTO _ELSE3
; begin

```

```

;      red = 1      ; turn on red
BSF  PORTB, RED
;      yellow = 1  ; turn on yellow
BSF  PORTB, YELLOW
;      green = 0   ; turn off green
BCF  PORTB, GREEN
;      end

;      if (y >= 76) AND (y<= 110) then
_ELSE3
_IF4
  MOVF Y, W
  MOVWF _STACK0
  MOVLW H' 4C'
  SUBWF _STACK0, W
  CLRW
  BTFSC _STATUS, _C
  ADDLW H' FF'
  MOVWF _STACK4
  MOVF Y, W
  SUBLW H' 6E'
  CLRW
  BTFSC _STATUS, _C
  ADDLW H' FF'
  ANDWF _STACK4, W
  BTFSC _STATUS, _Z
  GOTO _ELSE4
;      begin
;      red = 0      ; turn off red
BCF  PORTB, RED
;      yellow = 0  ; turn off yellow
BCF  PORTB, YELLOW
;      green = 1   ; turn on green
BSF  PORTB, GREEN
;      end

;      if (y >= 111) AND (y<= 130) then
_ELSE4
_IF5
  MOVF Y, W
  MOVWF _STACK0
  MOVLW H' 6F'
  SUBWF _STACK0, W
  CLRW
  BTFSC _STATUS, _C
  ADDLW H' FF'
  MOVWF _STACK4
  SUBLW H' 82'
  CLRW
  BTFSC _STATUS, _C
  ADDLW H' FF'
  ANDWF _STACK4, W
  BTFSC _STATUS, _Z
  GOTO _ELSE5
;      begin
;      red = 0      ; turn off red
BCF  PORTB, RED
;      yellow = 1  ; turn on yellow
BSF  PORTB, YELLOW
;      green = 0   ; turn off green
BCF  PORTB, GREEN
;      end

;      if y = 131 then
_ELSE5
_IF6
  MOVF Y, W
  SUBLW H' 83'
  MOVLW H' 00'
  BTFSC _STATUS, _Z
  ADDLW H' FF'
  ANDLW H' FF'
  BTFSC _STATUS, _Z
  GOTO _ELSE6
;      begin
;      clr(x)
  CLRF X
  CLRF XH
;      clr(y)
  CLRF Y
;      end
;      end
_ELSE6
_END1
;      loop
  GOTO _WHILE0
END

```

Listage 2. Feux de signalisation (1) Programme assembleur

Toutes les étiquettes variables engendrées par le compilateur sont précédées d'un signe « souligné » pour les différencier des variables utilisées dans le programme source.

La méthode la plus simple pour mettre en oeuvre une **séquence de feux de signalisation** serait d'utiliser la commande « wait » (attente) après avoir mis les diodes LED appropriées en ou hors service.

```

green = 0      ; turn green off
yellow = 0    ; turn yellow off
red = 1       ; turn on red
wait(3000)    ; wait for 3 seconds

```

Si votre intention est de construire le circuit présenté en **figure 3**, prenez conscience des limites du logiciel PIC. L'intensité de sortie maximum sur le port B est de 100 mA, chaque broche dispose d'une intensité maximum de sortie de 20 mA. Il faut incorporer la résistance de limitation d'intensité appropriée

(R) en série avec la diode LED (dans une plage de 470 Ω à 1 kΩ selon la LED). Ici, la suggestion est de 680 Ω.

Le programme exemple continuera de s'exécuter jusqu'à suppression de la tension appliquée au processeur PIC. Le code source du programme est présenté dans le **listage 1**. La variable « x » est incrémentée à chaque boucle du programme. Après l'atteinte d'un certain chiffre, celui-ci incrémente une variable « y ». Ceci est fait pour ralentir le programme – si la variable « x » était utilisée directement, les feux basculeraient trop vite. Les parenthèses séparant les conditions « y » ne sont pas destinées au compilateur, mais à la meilleure compréhension du programme. Le code assembleur résultant est présenté dans le **listage 2**.

Le fichier de documentation comporte d'autres exemples de programmes : *Traffic Lights (2)* (feux de signalisation 2) et *LCD Display Driver* (pilote

d'affichage de diode LCD). Le code source et les listes de code assembleur se trouvent aussi dans le fichier de documentation.

Description de la syntaxe et des commandes

Une description complète de toutes les commandes disponibles et de la syntaxe demandée par le compilateur existe dans les 56 pages du fichier de documentation. Ce fichier, sous la forme d'un document MS Word, se trouve dans la disquette numéro **EPS-996033-1** qui peut être commandée auprès des adresses habituelles. La disquette contient aussi les fichiers des codes source (.psf) des exemples et, bien sûr, le compilateur lui-même (Compiler84.EXE). Le fichier « readme.txt » (lisezmoi.txt) décrit l'installation très simple.

Texte : Jan Buiting

(000012)

Cette platine expérimentale a été conçue et réalisée pour servir d'aide au développement pour tous ceux d'entre nos lecteurs qui aimeraient s'essayer à la conception de l'une ou l'autre platine d'extension devant se connecter au port parallèle de l'un de leurs PC. L'auteur a mis le prototype de cet montage à contribution au cours de ses expériences avec, entre autres cobayes, des CNA (convertisseur numérique/analogique) et des affichages à cristaux liquides (LCD).

Technique et logiciel : Adrian Grace

expérimentateur LPT

porte ouverte sur le port parallèle du PC

La platine, dont on retrouve le schéma en **figure 1**, se branche, par le biais d'un morceau de câble en nappe à 25 conducteurs, sur l'embase sub D à 25 contacts qui constitue la sortie de l'interface parallèle d'un PC; elle comporte un connecteur de transition (cf. les informations données dans le paragraphe « La construction »). L'électronique se résume en fait à une triplette de sous-ensembles –un pour chacun des registres– à savoir les registres de données (*DATA*), d'état (*STATUS*) et de commande (*CONTROL*), requis par un port d'imprimante parallèle.

Les registres

Commençons par une courte description des registres. Le registre de données est un verrou de sortie à 8 bits servant normalement au transfert des données vers l'imprimante. Le registre d'état comporte 5 entrées à lecture seule; l'ordinateur s'en sert pour monitorer l'état de l'imprimante au niveau des messages d'erreur, etc. Le registre de commande dispose lui de 4 sorties; il sert normalement à mettre en oeuvre les fonctionnalités de l'imprimante.

Registre de données

Les 8 sorties du registre de données attaquent 8 LED (D5 à D12) au travers de résistances de limitation de courant (R5 à R12). Les LED utilisées sont du type 5 x 2 mm et partant se blotissent à merveille dans une grille classique au pas de 2,54 mm (1/10").

Registre d'état

Le registre d'état comporte 5 entrées que l'on simule par le biais d'un quin-

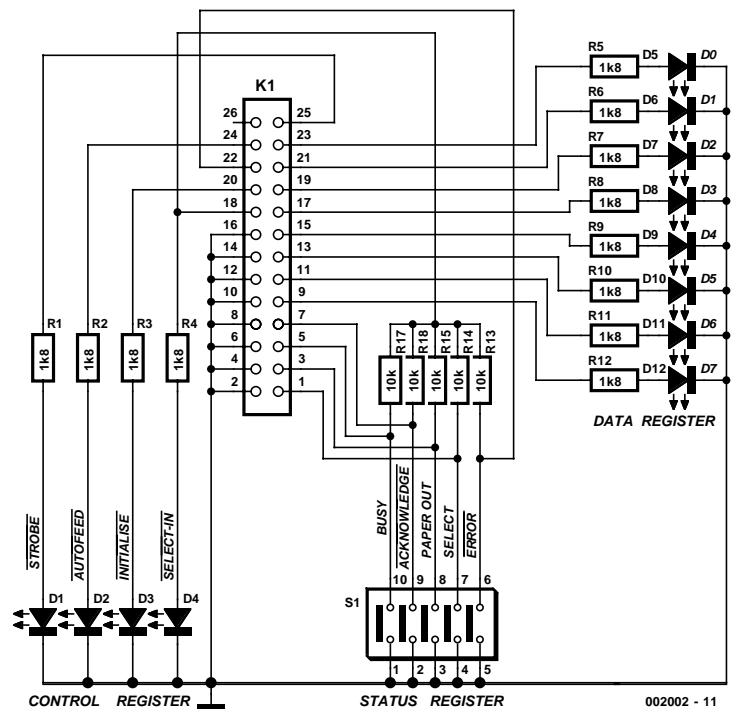


Figure 1. L'électronique de la platine d'expérimentation pour port LPT.

tuple interrupteur DIL. Les entrées remplissent les fonctions suivantes :

D7	D6	D5	D4	D3	D2	D1	D0
Busy	Acknowledge	Paper out	Select	Error	Non utilisé	Non utilisé	Non utilisé

Les contacts du quintuple interrupteur sont forcés au niveau haut (*pull-up*) par les biais des résistances R13 à R17; il est possible d'en lire l'état si tant est que la ligne **SELECT IN**, le bit D3, se trouve au niveau logique haut. Ceci permet une auto-alimentation du montage, ce qui signifie que l'on peut se passer d'alimentation externe.

Registre de commande

Le registre de commande possède une

largeur de 4 bits –le quartet de poids faible (*low nibble*). À l'image des sorties du registre de données, chacune des sorties du registre de commande attaque, par le biais d'une résistance de limitation de courant, une LED. On utilisera le même type de LED que précédemment, mais d'une couleur différente de préférence. Voici les fonctions

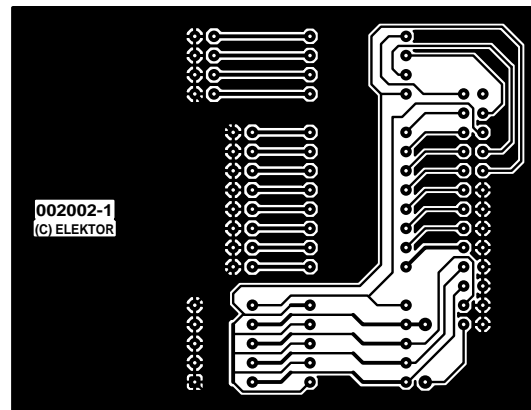
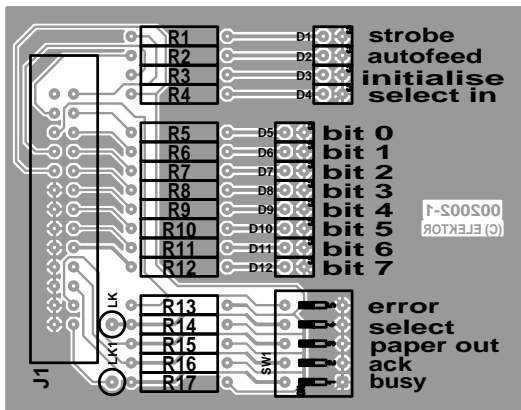


Figure 2. Dessin, recto-verso, de la platine, proposé par l'auteur.

de chacun des bits :

stice du connecteur de transition K1 de

D7	D6	D5	D4	D3	D2	D1	D0
Non utilisé	Non utilisé	Direction	IRQ	Select In	Initialise	Autofeed	Strobe

La LED D4 remplit une double fonction; lorsqu'elle est alimentée, elle fournit le courant requis pour la lecture des contacts du quintuple interrupteur connectés au registre d'état.

Le quartet de poids fort du registre de commande comporte un certain nombre de bits de commande qui ne sont pas adressés dans le présent concept.

La construction

Les fichiers de dessin (format Ultiboard) du circuit imprimé dessiné par l'auteur pour le présent projet ont été mis sur une disquette (ESS 006002-1) disponible auprès des adresses habituelles. L'implantation des composants sur la platine représenté en **figure 2** est à la portée de Monsieur-Tout-Le Monde. On plantera la totalité des composants, exception faite de J1. Attention à ne pas oublier le pont de câblage LK1. On pourra ensuite doter le câble en nappe du connecteur J2, après avoir enlevé le conducteur en excédent, il n'est pas possible en effet, sans forcer, de glisser 26 conducteurs dans un connecteur qui est prévu de n'en comporter que 25 ! Le conducteur coloré, qui identifie la broche 1, doit correspondre à la broche 1 de J2.

C'est maintenant que les choses deviennent intéressantes. Placez la platine devant vous de manière à ce que J1 soit le plus éloigné possible de vous. Disposez le câble plat sur le dessus du montage de sorte que l'extrémité du câble multi-brin se trouve au-dessus de J1. L'orientation de ce câble doit être telle que le conducteur repère soit sur la droite. Pas de PB pour le moment ? Glissez le câble en nappe dans l'inter-

manière à ce que la ligne non occupée se trouve sur la droite et que le câble soit en butée sur la gauche. Il est temps d'écraser J1 en veillant à ce que les contacts soient établis correctement. On pourra utiliser, pour protéger les broches de J1, quelques chutes de platine d'expérimentation à pastilles montées en gigogne. Il reste ensuite à retourner le câble et à fixer la barette anti-arrachement. On pourra ensuite monter J1 à l'endroit prévu sur la platine et l'y souder.

Le logiciel

Comme nous le disions plus haut, le logiciel requis par ce projet est disponible sur disquette. Il a été écrit en C et tourne sous DOS. Le programme lui-même (un fichier exécutable) se présente sous la forme d'un menu. Le programme commence par interroger le BIOS de l'ordinateur pour établir l'adresse du port imprimante avant de l'afficher à l'écran. Dans l'état actuel des choses, le programme ne s'intéresse qu'au premier port LPT1. Le menu principal offre 3 options, une pour chacun des registres et une voie de sortie. La première option concerne le registre d'état. Dans cette option, la ligne SELECT IN est automatiquement activée, permettant ainsi la lecture des positions des contacts de l'interrupteur. L'écran affiche alors la valeur hexadécimale des contacts jusqu'à une nouvelle action sur une touche du clavier. La seconde option est celle du registre de données. Lors de sa sélection, on voit apparaître un second sous-menu affichant la valeur actuelle du registre de données. Le sous-menu permet à l'utilisateur de décaler les bits vers la

Liste des composants

- R1 à R12 = 1kΩ8/0W25
- R13 à R17 = 10 kΩ/0W25
- D1 à D4 = LED 5 x 2 mm verte à haut rendement
- D5 à D12 = LED 5 x 2 mm rouge à haut rendement
- SW1 = quintuple interrupteur DIL
- J1 = connecteur de transition à 26 contacts à barette anti-arrachement
- J2 = connecteur Sub D mâle à 25 contacts
- morceau de câble plat à 26 conducteurs coupé à la longueur requise et ramené à 25 conducteurs à l'une des extrémités

gauche ou vers la droite ainsi qu'une incrémentation ou décrémentation des valeurs du registre.

La valeur du registre est envoyée vers la sortie du port parallèle de sorte que les LED visualisent le pattern de bits.

La troisième option concerne elle le registre de commande. Cette option ne fait rien de plus que d'illuminer de façon cyclique l'une des 4 LED associées au registre de commande, jusqu'à une nouvelle action sur une touche.

La disquette EPS006002-1 comporte le code source d'origine ainsi que le fichier objet intermédiaire et le fichier exécutable. Le code-source est bien commenté (en anglais) de sorte que vous pourrez expérimenter à loisir.

Nous vous recommandons instamment, si vous avez l'intention de poursuivre et de procéder à vos propres expériences en rapport avec le port parallèle, d'envisager l'achat d'une seconde carte (pour pas bien cher) pour port LPT. Cela vous évitera d'endommager le port parallèle principal qui est, bien souvent, intégré sur la carte-mère. Rien de plus catastrophique que d'endommager votre carte-mère, ce n'est pas là le but de nos expériences !

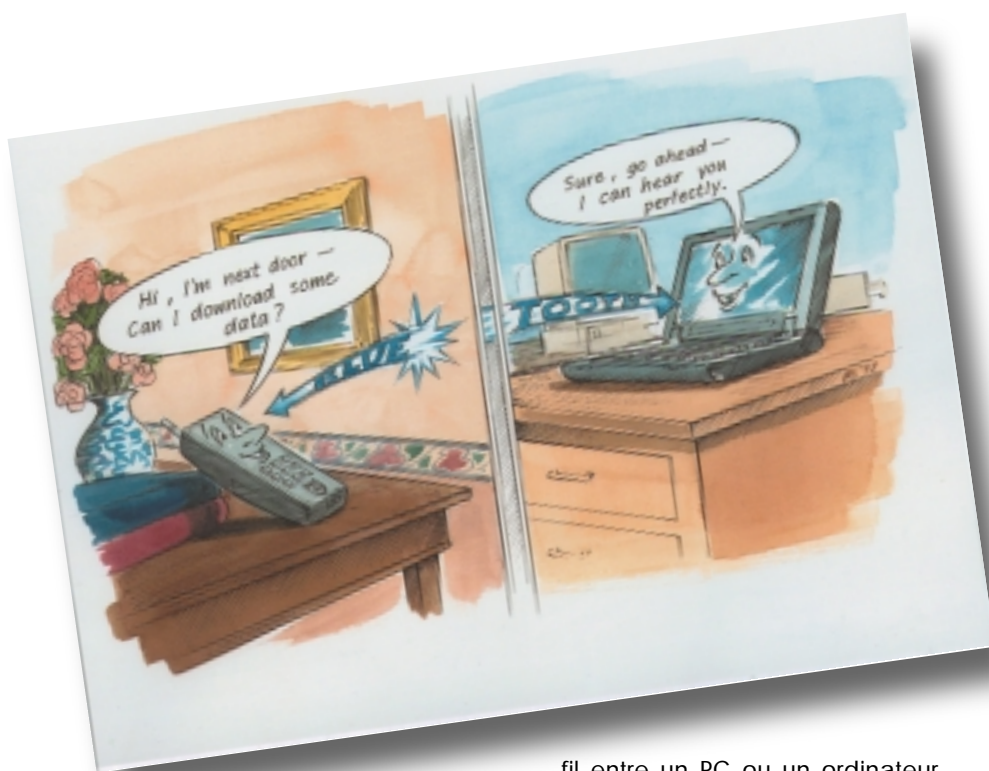
(002002-1)

Technique : Karel Walraven
Texte : Jan Buiting

L'échange de données entre PC et imprimantes, numériseurs, télécopieurs et modems, la communication entre ordinateur et téléphone, tous deux portatifs : le câblage embrouillé des appareils précédant l'installation laborieuse des pilotes est incontournable, même de nos jours. Il serait merveilleux que tous les appareils de traitement des données utilisés aujourd'hui puissent converser de façon autonome. Plus de câblage, plus d'installation. C'est enfin possible : avec la communication des données à courte distance, Bluetooth rend tout câblage superflu et ouvre la voie à de nouvelles applications.

par l'ingénieur diplômé Gregor Kleine

Bluetooth communication de données à 2,4 GHz sans câble d'ordinateur



Nokia Mobile Phones, Ericsson Mobile Communications, IBM, Intel et Toshiba, conscientes de ce besoin, ont fondé en mai 1998 le groupe à intérêt commun *Bluetooth Special Interest Group (BSIG)* [1,2] voué à la définition d'une liaison de communication universelle entre appareils de traitement des données. Il ne s'est agi au début que de définir un mode de communication de données entre un téléphone mobile et des périphériques évitant les inconvénients du câblage. Dans une phase ultérieure, on a étendu les applications à la liaison sans

fil entre un PC ou un ordinateur mobile et ses périphériques, jusqu'à la réalisation ad hoc de réseaux sans fil. Le groupe à intérêt commun Bluetooth (voir encadré) compte aujourd'hui plus de 1 000 entreprises. Tous les fabricants d'ordinateurs connus et toutes les entreprises de télécommunications sont bien entendu représentés, mais on y trouve aussi les fabricants de voitures et les entreprises d'électronique de divertissement. Le groupe BSIG a présenté en automne 1999 le cahier des charges temporaire 1.0 de Bluetooth. Chacun peut y avoir accès sur Internet [2] afin de permettre à la méthode de transmission Bluetooth de se développer

sans entraves. Le cahier des charges ne comporte pas moins de 1 500 pages. Le but principal du développement de Bluetooth consiste à réaliser des modules miniaturisés extrêmement peu coûteux incorporés à différents terminaux. Le prix basé sur la fabrication en grande série devrait se situer au-dessous de 5 dollars.

Fréquences

Les bandes **ISM** (*Industrial, Scientific & Medical*) remplissent les conditions requises : mondiales et gratuites. Ces bandes de fréquence destinées aux appareils de l'industrie, de la science et de la médecine ne requièrent pas de licence et sont gratuites, mais leur puissance d'émission est limitée. Le **tableau 1** en fournit la liste complète. L'utilisation des autres bandes **ISM**, 5,8 GHz, 24,250 GHz et 122,5 GHz n'est malheureusement pas encore technologiquement rentable. Comme la présence d'utilisateurs des bandes **ISM** qui ne détiennent pas de licence constitue une cause toujours possible de perturbations, et comme Bluetooth exige de surcroît un débit d'environ 1 Mbit/s, il ne reste plus que la bande 2,4 GHz (2 400 à 2 483 MHz). Mais comme sa largeur est de 83 MHz, elle offre assez de place de réserve lorsqu'un four à micro-ondes occupe par exemple une fréquence particulière. L'accès à cette bande de fréquence est mondial et seuls le Japon

(de 2,471 GHz à 2,497 GHz), la France et l'Espagne imposent une bande légèrement plus limitée.

Pour venir à bout des perturbations dont la plupart se produisent à une fréquence fixe, Bluetooth fait appel à un processus d'étalement du spectre (*spread spectrum*) : des changements de fréquence rapides (jusqu'à 1 600 fois par seconde) pseudo-aléatoires fournissent un assez grand nombre de fréquences dépourvues de perturbations pour assurer sans erreur le débit désiré (figure 1). On nomme cette méthode dite FHSS (*Frequency Hopping Spread Spectrum*), c'est-à-dire « étalement du spectre par bonds de fréquence ». Bluetooth fait appel à 79 fréquences dans la gamme des fréquences 2,402 GHz à 2,480 GHz séparées par un intervalle de 1 MHz. Conformément aux restrictions mentionnées plus haut, il n'existe que 23 fréquences de saut en France, en Espagne et au Japon.

Picoréseaux et réseaux de diffusion

Deux appareils équipés de Bluetooth entrent spontanément en contact l'un avec l'autre par bonds de fréquence. Pour que le reste de l'appareillage Bluetooth à proximité ne rende pas toute liaison impossible en se servant des mêmes fréquences, on dérive la suite des fréquences utilisées de l'adresse et de l'horloge de l'appareil qui prend l'initiative de la liaison. Cette appareil devient le maître du circuit de transmission tandis que tous les autres participants sont des esclaves qui se synchronisent sur le maître.

Un canal peut être utilisé par 8 radios Bluetooth au maximum. Ce réseau se nomme un *picoréseau* (le *Piconet* de la figure 2). Par *canal*, on veut dire que tous les participants de ce picoréseau se servent de la même suite de sauts, donc de la même suite de fréquences. Comme une adresse individuelle de 48 bits est attribuée de façon centralisée à chaque appareil Bluetooth, les bonds de fréquence ne peuvent pas avoir lieu dans deux canaux identiques. La constitution automatique de picoréseaux supplémentaires permet de se servir de plus de 8 appareils Bluetooth au même endroit, par exemple dans un bureau, sans risque d'interférence et ce avec un débit élevé.

Un émetteur-récepteur esclave intégré dans un picoréseau Bluetooth déterminé est interpellé par le maître lors du premier intervalle par voie multiplexée dans le temps (TDM = *Time Division*

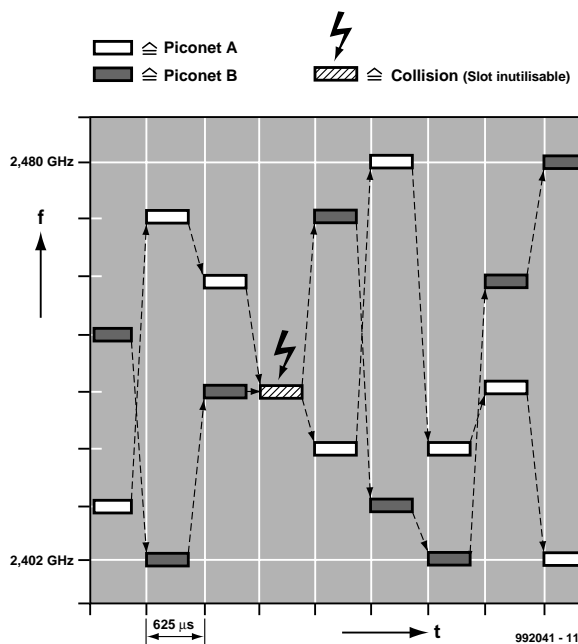


Figure 1. Bonds de fréquence.

Multiplexing) et peut répondre lors de l'intervalle suivant. L'esclave peut participer à un autre picoréseau au cours

des autres intervalles. Pour ce faire, il règle son récepteur sur la fréquence de saut momentanée du second pico-

Tableau 1. Gammes de fréquences ISM.		
Gamme de fréquences ¹	Largeur de bande	Co-utilisateurs/remarques
26,957 MHz à 27,283 MHz	0,326 MHz	Transmission CB, interphones de surveillance, ...
40,660 MHz à 40,700 MHz	0,040 MHz	Faible largeur de bande
433,050 MHz à 434,790 MHz	1,74 MHz	Radiophonie amateurs et professionnelle
868 MHz à 870 MHz	2,00 MHz	Faiblement occupée mais trop étroite
2,400 GHz à 2,483 GHz	83,00 MHz	Four à micro-ondes, majordome électrique

¹ Remarque : Autres bandes ISM à 5,8 GHz, 24,250 GHz et 122,5 GHz

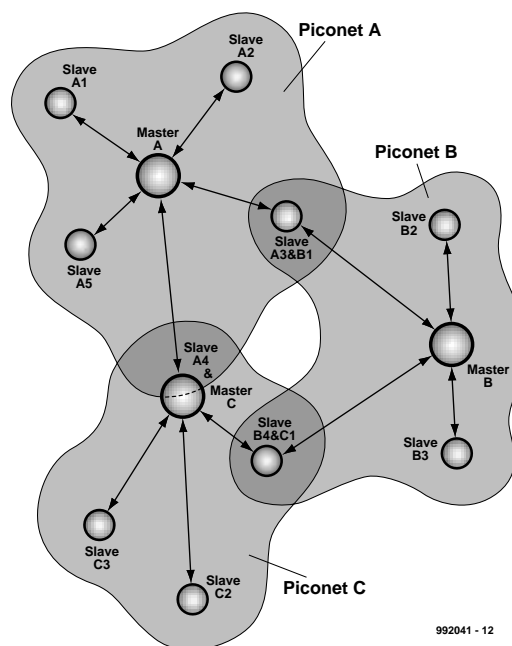


Figure 2. Picoréseau et réseau de diffusion.

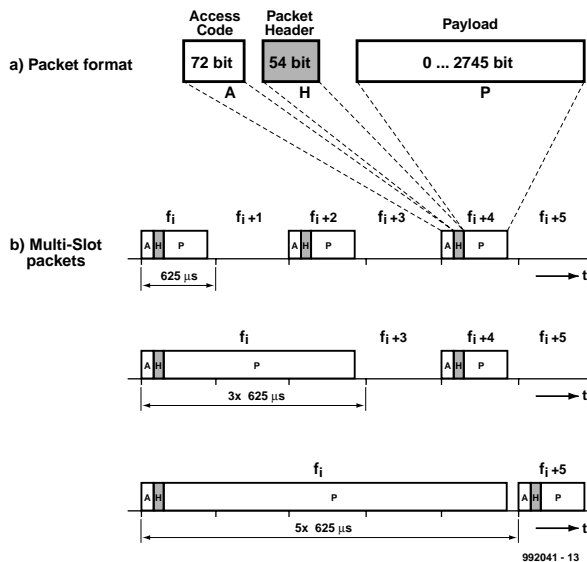


Figure 3. Format des paquets et paquets à plusieurs intervalles.

réseau et s'y synchronise avec le maître de ce réseau. L'ensemble de plusieurs réseaux de communication Bluetooth se nomme *réseau de diffusion* (figure 2). Comme les esclaves sont liés à l'horloge du maître, la voie multiplexée dans le temps empêche les collisions.

Bluetooth permet donc de former un picoréseau comportant jusqu'à 8 appareils. Le réseau de diffusion parent peut inclure jusqu'à 10 picoréseaux, ce qui signifie qu'il est possible de disposer de systèmes comportant jusqu'à 80 appareils Bluetooth dans un rayon de 10 mètres. Même si un appareil de chacun des 10 picoréseaux est actif en voie multiplexée dans le temps, le débit de transmission ne diminue que d'environ 10 % dans chaque picoréseau.

Modulation et protection contre les erreurs

La porteuse sautante à bande étroite de chaque canal est modulée par déplacement de fréquence gaussien (GFSK = *Gaussian Frequency Shift Keying*) et sa largeur de bande est 220 kHz (-3 dB) pour une excursion de

fréquence d'environ 150 kHz. Ce procédé de modulation très simple a été choisi pour pouvoir se servir de puces émettrices-réceptrices aussi peu coûteuses que possible. La puissance d'émission nominale de 1 mW (= 0 dBm) offre une portée normale d'au moins 10 mètres. Il existe des cas particuliers où il est permis d'atteindre une intensité de champ plus élevée à l'aide d'une émission de puissance de 100 mW (= +20 dBm) sur la bande ISM lorsqu'on se sert de la transmission par déplacement de fréquence ; on peut atteindre alors une portée de 100 mètres. La puissance d'émission du module radio Bluetooth s'adapte automatiquement –dans certaines limites– au trajet de communication.

La **protection contre les erreurs** de Bluetooth est aussi adaptative, ce qui signifie en l'occurrence que si la liaison est très bonne, la protection contre les erreurs est sacrifiée en faveur du débit. La correction d'erreur en aval, dite **FEC**, (*Forward Error Correction*) à deux étages n'intervient que lorsque la communication éprouve des difficultés. Elle se fait bien entendu au détriment du débit. Ce à quoi vient s'ajouter –hormis pour les canaux vocaux– un procédé

appelé **ARQ** (*Automatic Retransmission Query*), en d'autres termes une demande automatique de retransmission. Ce moyen permet de demander qu'un paquet soit retransmis. L'en-tête de chaque paquet est toujours protégé par une correction d'erreur en aval (**FEC**) donc, même si la correction FEC a été désactivée, il est possible de demander la retransmission d'un paquet par **ARQ** lorsque des perturbations apparaissent.

SCO et ACL

Le protocole de la bande de base de Bluetooth est prévu pour deux méthodes de transmission des données : point à point et point à multipoint.

La **transmission point à point** est baptisée **SCO** (*Synchronous Connection Oriented*), c'est-à-dire qu'elle est basée sur une liaison de type synchrone à commutation de circuit. Elle sert principalement à la transmission vocale. La liaison est symétrique en termes de débit dans les deux directions. La liaison en duplex intégral est établie par l'utilisation de deux intervalles de temps consécutifs correspondant aux deux sens de transmission des données. La transmission vocale est bien entendu beaucoup plus sensible aux délais que le transfert des données, ce pourquoi la demande de retransmission **ARQ** n'entre pas en ligne de compte. On fait appel par contre au processus **CVSD** (*Continuous Variable Slope Delta*) de codage de la parole qui se comporte honorablement lors d'erreurs de bits en se manifestant par un bruit de fond plus élevé. Le taux d'échantillonnage de la transmission vocale est celui de **RNIS** : 64 kilobits/s.

La **transmission point à multipoint** est asynchrone et ne dépend pas de la liaison (commutation par paquets). Dans le jargon de Bluetooth, il s'agit d'une liaison **ACL** (*Asynchronous Connectionless*). Ce mode de transmission permet au maître de s'adresser simultanément à plusieurs esclaves. Outre l'envoi d'informations à plusieurs esclaves, ce type de liaison est utilisé pour la transmission de paquets de données à un esclave spécifique. Pour améliorer le débit, un paquet peut utiliser plus d'un intervalle de temps de 625 SYMBOOL 109 (f "Symbol" \s 9 μs : 2 ou même 3 (figure 3). On interrompt en outre l'étalement du spectre par bonds de fréquence et on conserve la fréquence momentanée (f_i) pour les 3 ou 5 intervalles. Il est alors possible d'utiliser les autres intervalles normalement réservés au changement de fréquence pour la transmission des données. Pour resynchroniser, on rétablit la

FEC	Intervalles	Débit symétrique	Débit asymétrique
sans	1	2 x 172,8 kilobits/s	172,8 kBit/s + 172,8 kilobits/s
sans	3	2 x 384,0 kilobits/s	576,0 kBit/s + 86,4 kilobits/s
sans	5	2 x 432,6 kilobits/s	721,0 kBit/s + 57,6 kilobits/s
avec	1	2 x 108,8 kilobits/s	108,8 kBit/s + 108,8 kilobits/s
avec	3	2 x 256,0 kilobits/s	384,0 kBit/s + 54,4 kilobits/s
avec	5	2 x 286,7 kilobits/s	477,8 kBit/s + 36,3 kilobits/s

suite de sauts à la fréquence initialement prévue pour le quatrième (f_{i+4}) et le cinquième intervalle (f_{i+5}).

La transmission des données d'un esclave au maître ou à un autre esclave par l'entremise du maître doit être autorisée par celui-ci. La symétrie du débit dans les 2 sens des liaisons **ACL** est commandée par le maître. La transmission asymétrique est limitée à 721 kilobits/s dans un sens et à 57,6 kilobits/s dans le sens opposé. Ces 2 valeurs se réfèrent à la transmission à 5 intervalles non protégée par **FEC**. Il va sans dire que le débit diminue lorsque les mécanismes de correction **FEC (Forward Error Correction)** sont utilisés conjointement au processus **ARQ** cité plus haut. Un **FEC** au taux de 2/3 est utilisé. Dans le cas symétrique de 5 intervalles non protégés, on peut obtenir au plus 432,6 kilobits/s dans les 2 sens. Le **tableau 2** contient les autres débits pour la liaison **ACL**.

La transmission des données par les 2 types de liaison, **ACL** et **SCO**, fait appel à 16 types de paquets différents dont quelques-uns assurent des fonctions de commande. Chaque paquet comporte une zone d'identification de 72 bits, le code d'accès (*Access Code*), dérivée de l'adresse 48 bits du maître et incluant une protection d'erreur **FEC**. L'en-tête de 54 bits qui suit est aussi protégé par un **FEC** au taux de 1/3. Un paquet normal se termine par 2 745 bits utiles ou moins (**figure 3**). Les paquets de 3 et 5 intervalles peuvent transmettre un nombre proportionnellement plus élevé de bits utiles.

Bluetooth peut aussi chiffrer la transmission. L'authentification est basée sur une clé de 128 bits. L'utilisateur précise s'il désire que le chiffrement soit effectué dans un sens ou dans les deux. Ce paramétrage est mémorisé. Il est donc possible par exemple de spécifier exactement avec quels appareils un téléphone portable peut communiquer. Le contact avec son propre ordinateur portable ne subit par exemple aucune restriction tandis que l'accès à celui d'un collègue sera refusé.

Bluetooth peut offrir un canal de données **ACL** asynchrone ou jusqu'à 3 canaux **SCL** synchrones et également un canal de données asynchrone avec un canal vocal parallèle de 64 kilobits/s.

États du picoréseau

Les modules de Bluetooth qui font partie d'un picoréseau sont en mode d'attente et cherchent la présence de transmissions à proximité toutes les 1,28 secondes. Il leur faut vérifier

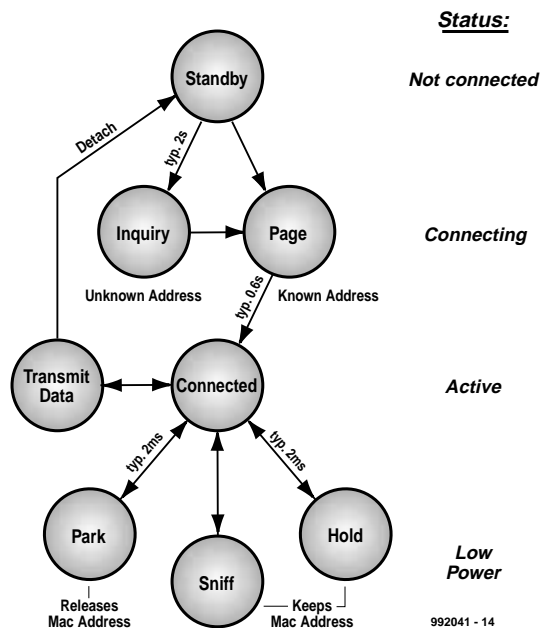


Figure 4. Diagramme d'état d'un module Bluetooth.

32 « porteuses d'éveil » parmi les 79 fréquences. La France, l'Espagne et le Japon limitent les porteuses d'éveil à 16 d'un total de 23 fréquences.

Un module Bluetooth qui ne communique pas encore n'envoie qu'un appel par les porteuses d'éveil auquel répondent les autres radios Bluetooth qui se trouvent à proximité. L'établissement initial de la liaison par les deux appareils est suivi de celui d'un picoréseau dont l'appelant sera le maître. Son adresse définit la suite de sauts de ce picoréseau. L'esclave initial et les appareils raccordés par la suite à ce picoréseau se synchronisent sur la fréquence d'horloge du maître.

Le **figure 4** reproduit le diagramme d'état d'un module Bluetooth : il passe du mode d'attente (*Standby*) à l'état d'interrogation (*Inquiry*) lors de la détection d'un second appareil et à une diffusion, puis à une demande d'adresse. Une fois l'adresse de l'autre module obtenue ou si elle était déjà connue en mode d'attente, l'appareil passe en mode d'appel (*Page Mode*). La liaison sera établie (état : *Connected*) au bout d'environ 0,6 secondes. Tout est prêt pour la transmission des données (*Transmit Data*). Après l'échange de données, le module Bluetooth peut retourner en mode d'attente ou adopter l'un des trois états à faible consommation ci-dessous.

On a tout d'abord un mode de maintien (*Hold*) au cours duquel l'appareil reste actif dans le picoréseau. Lorsque le temporisateur interne de l'esclave vient à expiration, l'esclave s'annonce brièvement au maître avant de recom-

mencer le compte à rebours. Un esclave en mode de maintien peut passer immédiatement à la transmission des données. Le maître peut forcer l'esclave à se mettre en mode de maintien, mais l'esclave peut aussi lui en faire la demande.

En mode « renifleur » (*Sniff*), l'esclave est programmé pour se mettre périodiquement à l'écoute du picoréseau afin de déterminer si ce dernier désire lui envoyer des données.

Le dernier mode à faible consommation fait appel au parcage des esclaves (*Park*). Dans ce cas, l'esclave se retire du picoréseau et rend son adresse MAC (*Media Access Control*, 0 à 7, ce qui correspond à 8 appareils par picoréseau). Il ne fait alors plus rien d'autre que de se synchroniser à longs intervalles avec le maître de son picoréseau.

Applications intéressantes de Bluetooth

Bluetooth permet en premier lieu de remplacer le câblage entre des ordinateurs, portatifs ou non, et les imprimantes, numériseurs (*scanner*) et réseaux locaux **LAN (Local Area Network)** par une transmission sans fil. Le clavier, la souris et la manette de jeu ou la boule de commande ne doivent donc plus nécessairement se trouver dans la ligne visuelle de l'ordinateur. L'ordinateur et le téléphone portable ouvrent la voie à d'autres applications. Il fallait jusqu'ici penser à envoyer le courrier électronique écrit sur un ordinateur portable à bord d'un avion après

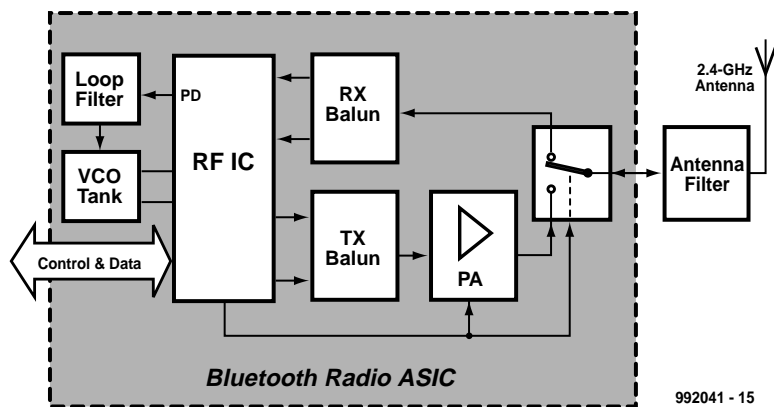


Figure 5. Diagramme fonctionnel d'un module radio Bluetooth.

l'atterrissage. Plus maintenant. Dès que le téléphone portable est remis en marche après la descente d'avion, l'ordinateur portable s'y connecte par l'intermédiaire de Bluetooth et expédie le courrier. La puissance d'émission de 100 mW au plus du module radio Bluetooth permet de l'utiliser aussi à bord d'un avion.

Le câblage entre un dispositif mains libres et un téléphone mobile est aussi destiné à disparaître. Le téléphone mobile se trouve dans la poche et est connecté sans fil au casque microtéléphone du dispositif mains libres. Un tel dispositif mains libres contribue aussi à la sécurité des conducteurs de véhicules automobiles. On peut enfin équiper un ordinateur portable de haut-parleurs et d'un microphone. Bluetooth permet au téléphone portable qui se trouve dans la mallette de recevoir sans fil les signaux vocaux. Un seul appareil, l'ordinateur portable, transfère la voix, les données et les graphiques. La synchronisation automatique de l'ordinateur portable avec l'ordinateur de

bureau et l'assistant numérique personnel PDA (*Personal Digital Assistant*) par Bluetooth est une idée digne d'attention. Si les appareils sont proches l'un de l'autre, ils peuvent échanger des dates d'échéance, du courrier électronique et des adresses pour que tous soient également à jour. Il serait même possible de se servir du téléphone portable et de Bluetooth pour permettre à une entreprise de mettre à jour ou de modifier automatiquement la liste d'échéances d'un technicien du service extérieur. Les participants à des réunions et entretiens y amèneront dorénavant leur ordinateur portable équipé d'une interface Bluetooth et se mettront spontanément en réseau. On échangera des graphiques, du texte ou des données ou on actionnera les appareils de projection, le tout sans fil. L'accès sans fil à Internet par Bluetooth donnera une plus grande liberté de mouvement. Les pages Internet arriveront sans fil jusqu'à l'ordinateur portable par le téléphone portable, le modem local ou le réseau local de l'entreprise.

Au bureau comme à la maison, un téléphone portable Bluetooth peut commuter automatiquement du mode GSM au mode DECT dès qu'il a établi le contact avec la station domestique de base du téléphone sans fil. Un seul téléphone, donc, pour toutes les communications. L'électronique de divertissement n'est pas en reste de suggestions : Bluetooth pourrait servir de liaison sans fil avec une caméra vidéo ou un appareil photo. La caméra transmet l'image par téléphone portable au réseau de distribution ou l'enregistre sans fil sur un ordinateur portable. Il serait même possible d'envoyer des cartes postales électroniques sous forme d'images fixes au téléphone portable par un trajet Bluetooth, puis de là au destinataire. Bluetooth peut aussi remplacer la télécommande d'installations stéréo et de téléviseurs. Plus besoin de liaison visuelle, contrairement à la télécommande à infra-rouges.

Premiers composants Bluetooth

En collaboration avec Symbionics, Ericsson a réalisé un kit de développement Bluetooth contenant une abondante documentation et un logiciel d'assistance à la conception. Le processeur de la bande de base est fourni par VLSI et provient de la famille Bluetooth VWS26000 [6]. Le module radio (figure 5) est un hybride conçu par la firme Ericsson elle-même (PBA 313) [4]. Philips Semiconductors offre aussi un kit pour réalisateurs Bluetooth : le circuit intégré spécial de la bande de base est aussi un ASIC (*Application Specific Integrated Circuit*) de VLSI. Le module radio contient l'émetteur-récepteur Bluetooth UAA-3558. Le kit comporte deux cartes-filles Bluetooth identiques permettant d'établir une première liaison de communication. La figure 6 illustre la structure typique d'un module Bluetooth. L'entreprise Cambridge Silicon Radio [7] se consacre aux composants Bluetooth sur une seule puce avec module radio intégré. Les ASIC BlueCore™01, BlueCore™02 et BlueCore™03 acquerront par étapes un nombre croissant de fonctions d'un modem Bluetooth. Un module complet devrait avoir la taille d'un timbre-poste. Pour compléter le tableau, de nombreux fabricants de semi-conducteurs tels que Temic, Philips et Motorola offrent des C.I. émetteurs-récepteurs 2,4 GHz spécialement conçus pour les modules radio de Bluetooth. Tous les modules Bluetooth doivent être soumis à l'homologation du groupe SIG Bluetooth pour en assurer la compatibilité.

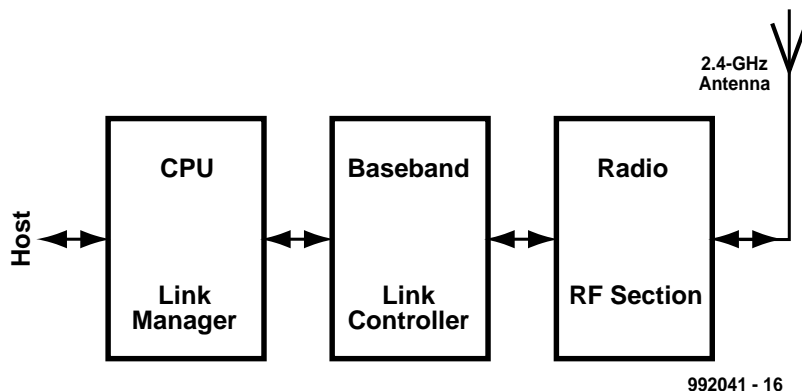


Figure 6. Diagramme fonctionnel d'un module Bluetooth.

Les concurrents de Bluetooth

Le concept de liaison hertzienne à courte distance illustré par Bluetooth a suscité des concurrents. On se sert déjà depuis un certain temps de trajets parcourus par des infra-rouges selon le procédé **IrDA** (*Infrared Data Association*) pour transmettre des données entre un PC et ses périphériques. Mais les infrarouges ne peuvent fonctionner qu'à vue et leur portée n'est au mieux que de quelques mètres pour un angle d'ouverture ne dépassant pas 60°. Bluetooth peut littéralement traverser les murs. Pour une comparaison entre IrDA et Bluetooth, voir [8]. Aux USA, plusieurs entreprises ont été créées pour établir **HomeRF** [9], une norme similaire à Bluetooth. Le protocole d'accès sans fil partagé (**Shared Wireless Access Protocol**, autrement dit **SWAP**) élaboré par le groupe de travail **HRFWG** (*HomeRF Working Group*) permet d'utiliser 127 appareils par réseau, portée

50 mètres. Il est intéressant de constater qu'on retrouve quelques fondateurs de Bluetooth parmi ceux du groupe **HRFWG** : Ericsson, IBM et Intel y jouent un rôle déterminant, aux côtés de Compaq, Hewlett-Packard, Microsoft, Motorola et Philips. Ce groupe ne compte toutefois que quelques centaines d'entreprises alors que le **SIG Bluetooth** peut se targuer de compter plus de 1 000 intéressés. Il faut en outre déboursier 500 \$ de droits d'inscription pour avoir accès au cahier des charges de **HomeRF**. La norme **DECT** de téléphonie sans fil est aussi candidate pour les réseaux locaux sans fil. Elle fonctionne dans un domaine de fréquence dédié de 1,8 GHz et sa portée est plus étendue (jusqu'à 500 mètres) en vertu de la puissance d'émission plus élevée. **DECT** et **Bluetooth** atteignent le même score dans la transmission vocale sans fil et dans la transmission des données à faible débit (64 kilobits/s max.), mais Bluetooth peut

Le roi à la dent bleue et ses amis

Le nom « Bluetooth » a été choisi en l'honneur du roi danois Harald Blaatand (en français : Harald à la dent bleue, 910 – 986), qui a christianisé et uni la plus grande partie de la Scandinavie. La figure (source : ROHDE&SCHWARZ), représentant une pierre runique découverte dans sa capitale Jelling (Jylland central), indique clairement qu'il fut un des premiers adeptes des communications sans fil entre téléphone et ordinateur portable...



Le choix du nom se rapporte bien entendu aux deux instigateurs principaux du système, Ericsson Mobile Communications (Suède) et Nokia Mobile Phones (Finlande) qui ont défriché le domaine des réseaux locaux sans fil (**Wireless LAN**, **Local Area Network**). Le nouveau roi à la dent bleue s'est fait entre-temps beaucoup d'amis, comme le montre la liste de membres du **SIG Bluetooth** ci-dessous (extrait, liste intégrale en [1]) :

AKG Acoustics, Alcatel, Analog Devices, AMD, Bang & Olufsen, Boeing, Bosch, Casio, Compaq, Dell, Fujitsu, Grundig, Hagenuk, Hewlett-Packard, ICO, LEGO, LG Electronics, Logitech, Lucent, Mitsubishi, Motorola, NCR, National Semiconductor, Philips, Psion Computer, Qualcomm, Rohde & Schwarz, Samsung, Sennheiser, Seiko Epson, Sharp, Siemens, TDK, Telia, Temic, Texas Instruments, VLSI Technology, Volvo, 3Com.

Bibliographie et liens

- [1] www.bluetooth.com
- [2] www.bluetooth.net
- [3] www.zdnet.co.uk/news/specials/1999/04/bluetooth/
- [4] <http://bluetooth.ericsson.se/default.asp>
- [5] www.intel.com/mobile/bluetooth/
- [6] www.vlsi.com
- [7] www.cambridgesiliconradio.com/
- [8] Comparaison IrDA – Bluetooth : www.countersys.com/tech/bluetooth.htm
- [9] HomeRF : www.homerf.org
- [10] HomePNA : www.homepna.org
- [11] ADSL – Accès haute vitesse à Internet, ELEKTOR 10/99, p. 28 et suiv.
- [12] www.bluetooth.rsd.de/ et www.bluetooth-testing.com

atteindre 721 kilobits/s. Un consortium gravitant autour de 3Com, IBM, Intel, AMD, Compaq, Hewlett-Packard, Cisco et Lucent « pousse » la mise en réseau par lignes téléphoniques. La méthode se dénomme **HomePNA**, c'est-à-dire **Home Phoneline Networking Alliance** [10] et recevra certainement un accueil particulièrement favorable aux États-Unis où il est coutume d'installer une prise téléphonique dans chaque pièce. « HomePNA » fonctionne dans le domaine de fréquence qui suit celui de **ADSL** (**Asymmetric Digital Subscriber Line**, [11]), entre 6 et 10 MHz. On tente aussi d'accomplir une mise en réseau domestique en ayant recours aux divers procédés de communication par l'entremise des lignes électriques. Un avantage : chaque pièce est équipée de prises. Mais les fortes perturbations qui règnent sur les lignes électriques limitent fortement le débit.

Perspectives d'avenir

Bluetooth semble en voie de s'imposer rapidement et partout comme système de réseau local sans fil. Outre les applications d'envergure de transmission des données, il sera certainement utilisé dans l'électronique de divertissement et dans les véhicules automobiles. Maintenant que Ericsson et Nokia ont présenté des projets pilotes, on peut s'attendre à voir apparaître cette année de nombreuses applications de Bluetooth. Le salon CeBIT 2000 devrait nous en apprendre davantage. Les premiers systèmes Bluetooth feront prochainement leur apparition sur le marché.

(992041)