

ÉLECTRONIQUE et MICRO-INFORMATIQUE

www.elektor.fr

MONO-CARTE « HIGH-SPEED » à DS89C420



PAN, Processeur
d'Audio
Numérique



Gradateur
DMX

Autres capteurs
d'image

Boussole pour
RCX de Lego

EDiTS Pro
Upgrade du ESDL



Gradateur DMX

Pour applications semi-professionnelles

Projet : Andreas Kneer

Ce gradateur, qui peut être commandé par un signal DMX numérique, une tension de commande de 0 à 10 V ou manuellement, permet d'ajuster la luminosité individuelle de 1 à 4 projecteurs, d'une puissance de 1 kW chacun.



Le gradateur DMX est destiné aux applications semi-professionnelles. Il est commandé par un pupitre de mélange de lumière numérique DMX ou analogique. Il accomplit parfaitement sa tâche, qui est d'atténuer, mais se refuse aux fantaisies charmantes telles que les traînées lumineuses ou la conversion son en lumière. Par contre, le préchauffage des lampes dans la plage lumineuse de 0 à 100 % prolonge leur vie et offre un fonctionnement autonome à atténuation constante sans recours au pupitre de mélange de lumière. D'autres caractéristiques, telles que le choix du canal par commutateurs à molette et la linéarisation numé-

Caractéristiques

- Compatibilité avec les pupitres de commande de l'éclairage à signaux de commande DMXD512 ou 0 à 10 V
- Choix du canal grandement facilité par les commutateurs à molette
- Possibilité de brancher des projecteurs d'une puissance totale de 1000 W dans chaque canal
- Gradateurs par découpage de phase à triacs pour lampes à incandescence (halogène).
- Commande linéaire de la lampe par correction numérique de la caractéristique de l'angle de découpage de phase
- Préchauffage réglable de 0 à 100 %
- Compact, pouvant donc être monté facilement aux projecteurs, à leur mât ou sur une traverse
- Conception robuste et insensible aux parasites

rique de la plage de commande, complètent élégamment le cahier des charges.

Les rangées de projecteurs (*bar*) à 4 ou 6 projecteurs ne constituent pas une rareté dans le domaine de la technique de l'éclairage. Pour que les dimensions du gradateur présenté ici restent dans des limites raisonnables, on a choisi un fonctionnement monophasé. En partant d'une puissance de 1 kW par projecteur, il est possible d'en raccorder trois à une phase de 16 A. Il faut 2 gradateurs à 3 canaux chacun pour une rangée de six. Pour pouvoir monter le gradateur à proximité immédiate des projecteurs, on a conçu un boîtier métallique ressemblant à un bloc multiprises triple surdimensionné. Le boîtier sert en même temps de dissipateur thermique pour les triacs.

Comme il est de toute façon nécessaire de confier le décodage du signal DMX à un microcontrôleur, la majeure partie de toutes les fonctions d'atténuation est aussi réalisée par logiciel. Le 68HC11F1 utilisé ici comporte, outre 1 k de mémoire RAM interne, 512 octets d'EEPROM interne dans laquelle le logiciel a été « tassé » de façon à éviter l'utilisation d'une ROM externe. On économise temps et argent. La périphérie intégrée est imposante : elle comporte une interface série, un convertisseur A/D et la structure de temporisation du processeur, ce qui permet de réaliser un circuit et un logiciel tous deux très compacts.

Le schéma-bloc (figure 1) représente le fonctionnement du logiciel. Ce dernier se compose de trois tâches de priorités différentes.

Tâche DMX

DMX512 évince toujours plus le signal analogique 0 à 10 V. Le protocole DMX512 est électriquement identique à RS485, ce qui permet d'utiliser toutes les commandes de ligne RS485. L'appareil de commande envoie par la liaison bifilaire des trames de données de 8 bits utiles chacune avec un débit de 250 kbits/s. Le décodage des bits de données dans le récepteur peut être effectué par une interface UART couplée à un ordinateur. Une trame de données DMX contient 11 bits (ci-

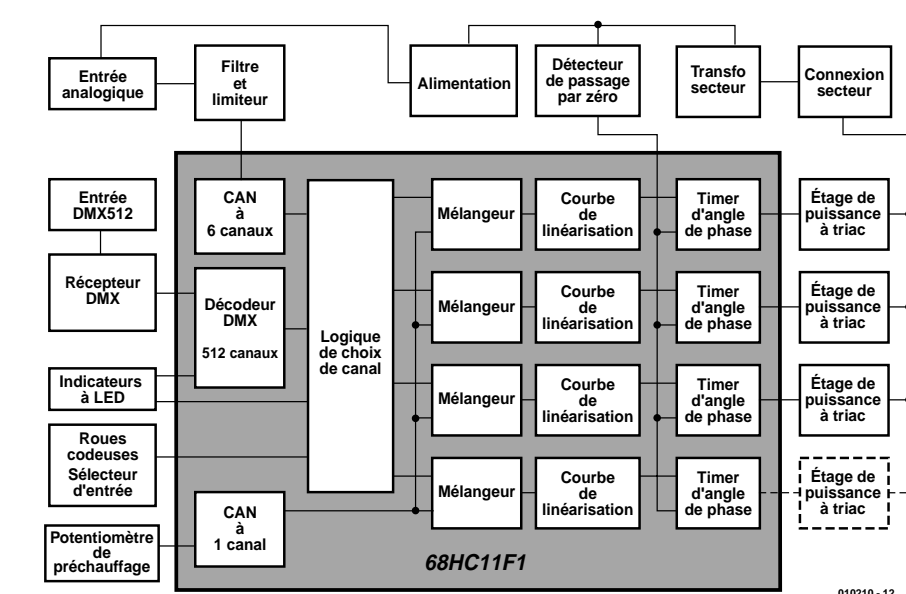


Figure 1. Schéma du gradateur DMX.

inclus le bit de départ et les 2 bits d'arrêt) et nécessite donc environ 44 μ s pour la transmission. La norme DMX prescrit de transmettre les octets de données à intervalles de 1 s au plus. Il n'est par contre pas fait mention d'une distance minimale. Lorsqu'un récepteur DMX veut recevoir de tous les émetteurs, il doit traiter toutes les trames de données quel que soit le laps de temps entre elles. Si le pupitre de mélange de lumière est très rapide et envoie les trames sans discontinuer, le microprocesseur ne dispose que de la durée d'une trame, 44 μ s, pour traiter les octets de données. De telles conditions requièrent un processeur puissant et un logiciel optimisé.

L'émetteur DMX commence la transmission par une interruption (*Break*, 88 μ s au moins dans l'état bas sur la ligne DMX+). La longueur « non conforme » de ce signal indique au récepteur que la transmission va commencer. Puis l'émetteur envoie l'octet de départ qui indique l'appareil auquel les données sont destinées. Les gradateurs DMX ne doivent réagir qu'aux données commençant par l'octet initial 0. L'octet initial permet donc théoriquement de transmettre 512 canaux d'atténuation plus 512 canaux pour scanner, 512 canaux pour lumières mobiles (*Moving Lights*), etc. Mais cette option est rarement utilisée. Après l'octet initial, l'émetteur com-

mence à transmettre successivement les canaux. Le dernier canal à transmettre peut, mais ne doit pas nécessairement, être le canal 512. Il est possible de provoquer un arrêt prématuré en envoyant un « Break » (point d'arrêt). On trouvera d'autres informations sur les particularités de la transmission DMX512 dans le numéro de mai 2001 d'Elektor.

Le programme de traitement (*Handler*) DMX du logiciel est conforme à la norme DMX512/1990 ; il lit tous les 512 canaux (si l'octet initial = 0) et mémorise les valeurs dans la RAM. Comme la tâche de traitement DMX ne dispose que de 44 μ s pour effectuer ces fonctions, sa priorité dans le système est la plus élevée. Le programme de traitement DMX est appelé dès qu'un octet est reçu dans le tampon UART.

Tâche principale

Le programme principal du logiciel évalue l'état des éléments de commande et pilote les diodes lumineuses. Les signaux d'entrée des gradateurs proviennent du convertisseur A/D interne qui reçoit les signaux analogiques par une douille DIN 8 broches d'un type très répandu dans la technique de l'éclairage ou directement des canaux DMX. Selon la position du sélecteur de canaux, la logique de sélection choisit 3 des 6 canaux analogiques ou 3 canaux DMX dont la luminosité est extraite du convertisseur A/D ou de la mémoire DMX intermédiaire. Pour finir, chaque valeur est comparée à la luminosité du préchauffage des lampes et la plus élevée des 2 valeurs comparées est convertie par une formule de linéarisation en une durée

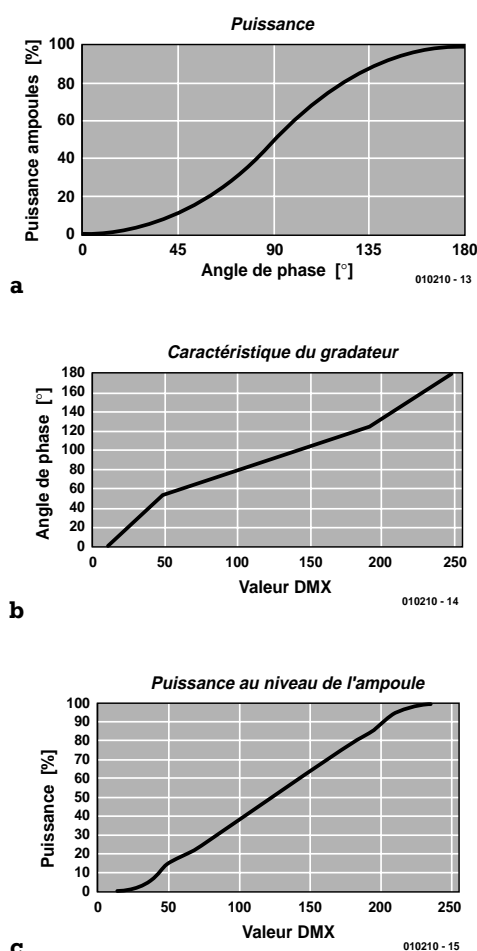


Figure 2 a) Puissance de la lampe en fonction de l'angle de phase.

b) Caractéristique permettant au logiciel de linéariser la puissance.

c) Résultat de la linéarisation : dépendance linéaire comportant des parties « droites » pour les valeurs DMX basses et élevées.

d'excitation des triacs.

N'oublions pas que la tension du réseau est sinusoïdale et non rectangulaire, donc que la puissance appliquée aux lampes n'est pas proportionnelle à l'angle de phase spécifié (**figure 2a**). Il est souhaitable que la caractéristique de commande soit linéaire pour que le régulateur du pupitre de commande d'éclairage puisse positionner avec exactitude la luminosité d'un projecteur même aux faibles valeurs. La caractéristique du gradateur est inversée précisément dans ce but (**figure 2b**).

La caractéristique de commande du gradateur est horizontale au début et à la fin. Ce facteur est important dans le cas de pupitres de mélange analogiques : il faut éviter en effet qu'une tension résiduelle de quelques millivolts fasse luire les lampes ou empêche d'atteindre une luminosité de 100 %. La correction

de la caractéristique permet d'atteindre un comportement presque linéaire de la puissance dans la plage des valeurs numériques (**figure 2c**). Les valeurs de la durée d'excitation obtenues en tenant compte de la correction de la caractéristique sont écrites dans les registres de temporisation du contrôleur.

Tâche de temporisation

L'excitation des triacs s'effectue par l'entremise des registres de comparaison de sortie (*Output compare*) du 68HC11. La tâche de temporisation est toujours appelée au moment du passage par zéro de la tension du réseau. Lors de l'appel, l'angle de phase est ajouté à la valeur courante du compteur sans fin du 68HC11 (incrémenté chaque μs) mémorisée temporairement, puis cette somme est écrite dans les registres « *Output compare* ». La valeur est alors comparée continuellement à la valeur actuelle du compteur sans intervention du logiciel. Le triac est activé dès que le registre « *Output compare* » est égal au compteur. Le triac est désactivé par la tâche de temporisation lors du passage suivant par zéro. Un signal PWM synchronisé sur la fréquence du réseau est donc engendré aux sorties du 68HC11.

Entrée analogique et entrée DMX

L'alimentation de la partie basse tension est représentée en haut à gauche du schéma (**figure 3**). Le fusible F2 constitue une protection contre le suréchauffement. Lorsque, pour quelque raison que ce soit, la température interne de l'air dépasse 95 °C, ce fusible coupe de façon permanente le circuit du gradateur. La limite de 95 °C vaut aussi pour le soudage, donc attention lors du montage ! La tension d'alimentation des pupitres de mélange analogiques est par ailleurs prélevée en amont du régulateur, protégée séparément par F3 (F3 est un fusible Polyswitch ou plus précisément un composant résistant à coefficient de température positif) et rendue accessible à l'extérieur par la douille DIN. Cela permet de raccorder aussi des pupitres de mélange très simples ne comportant qu'un régulateur de ten-

sion 10 V et un potentiomètre.

La tension secondaire redressée est prélevée à l'anode de D1. Comme D1 « élimine » le condensateur de lissage C16, le passage par zéro de la tension alternative nécessaire à la temporisation est encore présent en ce point. Ce passage est détecté par le circuit de comparaison comportant IC4.

Le comparateur lui-même, IC4.A, comporte une hystérésis produite par R21. En effet, les gradateurs souffrent souvent d'un problème : lorsque des charges importantes sont commutées à proximité du passage par zéro, l'inductance et la résistance des conducteurs provoquent un bref effondrement de la tension du réseau, ce qui conduit le détecteur à reconnaître comme passage par zéro ce qui n'en est pas un. Cet effet est atténué par une hystérésis. IC4.B tamponne et inverse le signal de sortie destiné à l'entrée d'interruption du contrôleur.

Un réseau constitué par des diodes doubles antiparallèles (D3 à D8), des diodes de Schottky (D10, D11), les condensateurs C9 à C14 et des résistances (R28 à R33, rangée R46 et R45) forme l'entrée analogique à 6 canaux du gradateur. Les circuits sont identiques pour les 6 canaux. Les résistances de protection forment un diviseur de tension qui réduit la plage de tension d'entrée de 0 à 10 V à 0 à 5 V. Le diviseur et le condensateur forment un filtre RC qui élimine les tensions parasites HF. Les doubles diodes limitent d'éventuelles sous-tensions et surtensions. Une de ces diodes est raccordée à la résistance R45 de 4,7 k Ω et aux diodes de Schottky D10/D11. Une tension (par rapport à la masse) d'environ 0,3 V et 4,7 V aux bornes de R45 est engendrée par celles des diodes. La tension à l'entrée du microcontrôleur ne peut partant pas dépasser 4,7 V plus la tension de seuil d'une diode, c'est-à-dire 5,4 V ou descendre au-dessous de 0,3 V – 0,6 V soit –0,4 V. Un connecteur d'entrée à picots sur la carte permet de raccorder la douille DIN 8 broches comme le montre la **figure 4**. Le convertisseur A/D comporte aussi un filtre passe-bas RC R43/C16 qui assure l'absence de perturbations dans la tension de référence de 5 V. L'entrée DMX512 est équipée de la commande de ligne RS485 bidirec-

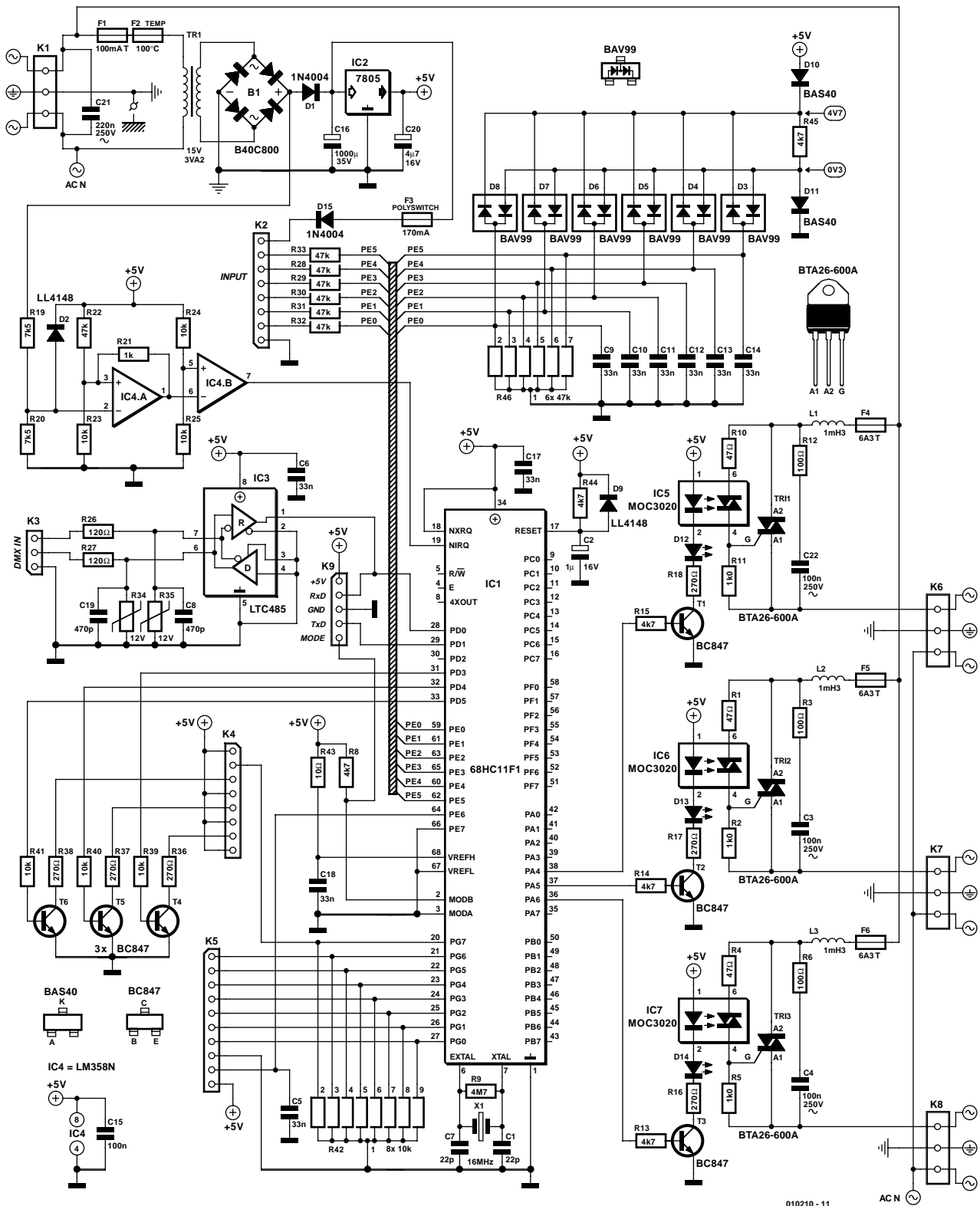


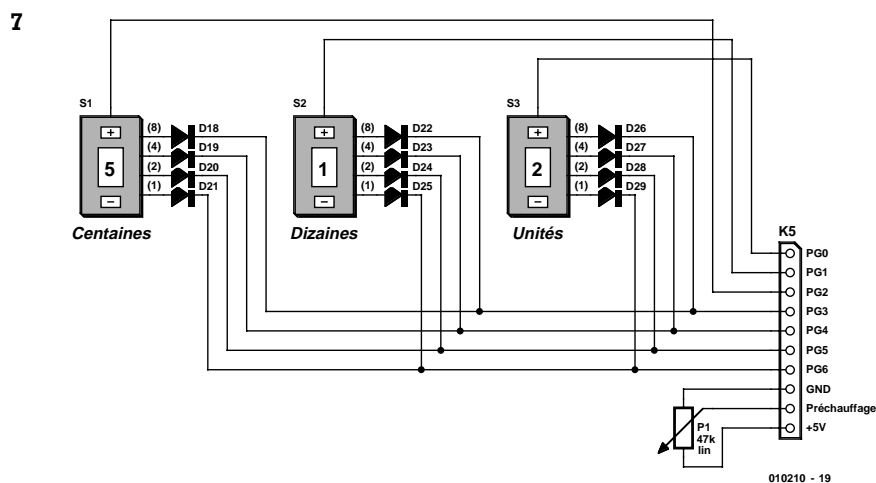
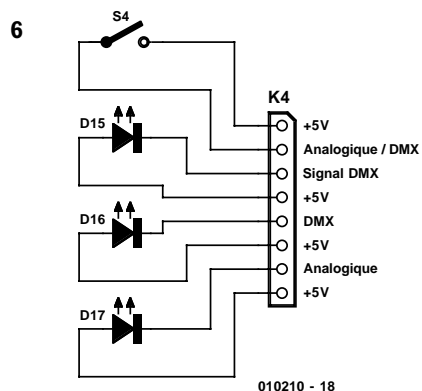
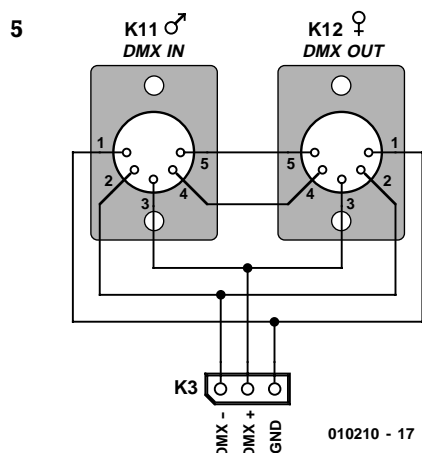
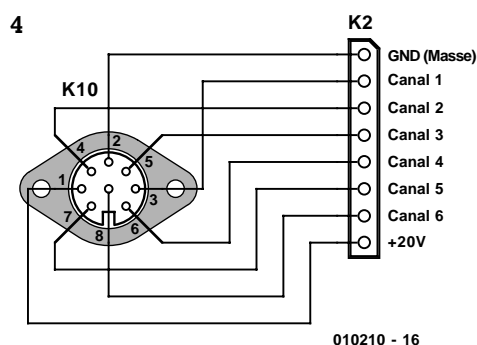
Figure 3. Circuit avec les entrées à gauche et l'électronique de puissance à droite. Le microcontrôleur assurant la communication est placé entre eux.

tionnelle LTC485. La fonction d'émission, qui n'est pas utilisée ici, est mise hors circuit. Les lignes d'entrée sont reliées au circuit intégré par un filtre passe-bas contre les parasites

HF. Des varistances 12 V pour véhicule automobile protègent les circuits d'entrée des surtensions. La sortie du récepteur est directement reliée à l'interface série du

68HC11. La liaison des douilles DMX avec K3 est illustrée dans la **figure 5**.

Trois LED pilotées depuis les broches PD3 à PD5 du port du contrôleur par un étage à transistor indiquent si l'entrée DMX ou l'en-



trée analogique est active. Les LED sont reliées à K4 tout comme l'inverseur DMX/analogique. L'état du commutateur est lu sur le port PG7. La **figure 6** montre la liaison des LED et du commutateur avec K4.

Les commutateurs à roue codeuse permettant de choisir le canal sont reliés à K5 (**figure 7**). Les commutateurs possèdent des sorties binaires branchées en parallèle qui sont directement découplées à l'emplacement des interrupteurs par des diodes 1N4148 externes. Lors de l'évaluation par le contrôleur, celui-ci effectue le multiplexage des 3 commutateurs par PG0 à PG3 et identifie la valeur choisie au moyen des lignes du port PG4 à PG6.

Les canaux analogiques sont aussi influencés par les commutateurs à molette. Si le canal 1 est choisi, la tension de commande sur PE0 détermine la luminosité de la lampe en PA6, PE1 assure la régulation de PA5 et PE2 celle de PA4. Si le canal 2 est choisi, la régulation est effectuée entre PE1 PA6, PE2 PA5, etc.

Le potentiomètre d'ajustement du préchauffage est raccordé aux 3 broches restantes de K5. La prise médiane du potentiomètre est directement reliée à l'entrée ADC du contrôleur (PE6). Un condensateur en parallèle se charge des perturbations éventuelles.

Sorties

Le processeur lui-même ne nécessite aucun montage externe particulier, hormis un oscillateur de

16 MHz et un circuit de réinitialisation (R44, C2 et D9). Une résistance R8 à MODB assure le redémarrage du processeur en mode « single chip » après réinitialisation. Nous pouvons donc nous concentrer sur les sorties de puissance.

Les 3 étages finals à triacs sont reliés à PA4 à PA6. Le circuit est le même pour tous les canaux.

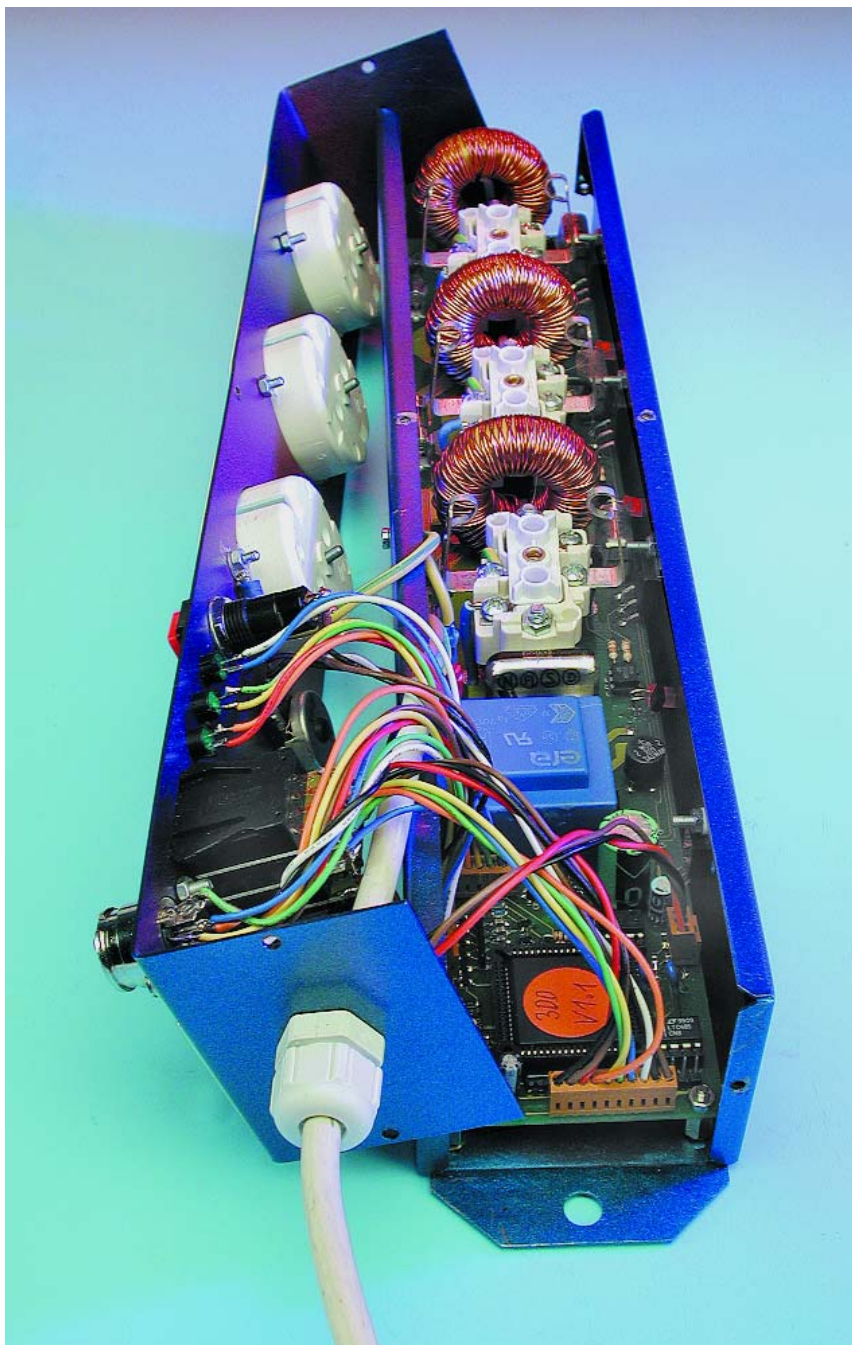
Les triacs utilisés sont des BTA26/600A. Ils sont conçus pour un courant maximum de 26 A. Si cela semble surdimensionné, il ne faut pas oublier que la résistance à froid des lampes à incandescence est très faible, donc que le courant initial est très élevé (jusqu'à 10 fois la valeur du courant nominal). Un grand nombre de lampes halogène présentent en outre la caractéristique déplaisante de provoquer un court-circuit lorsqu'elles rendent l'âme. Plus un triac est stable et plus il a de chances de survivre à la défaillance d'une lampe. Contrairement aux types TIC très répandus, les types BTA possèdent l'avantage d'avoir un boîtier isolé des sorties. Il n'est donc pas nécessaire de recourir à un montage isolé trop élaboré.

Chaque triac est commandé par un second triac intégré dans l'optocoupleur. Le transistor raccordé commute dès que le contrôleur positionne une sortie au niveau haut. Un courant passe par la LED de l'optocoupleur, le triac de l'optocoupleur est amorcé et le courant passe de la phase à la gâchette du triac de puissance. Celui-ci est alors également amorcé. L'optocoupleur provoque donc la fermeture du circuit qui permet au courant destiné à la lampe de passer.

L'angle de phase lors de la mise sous tension peut être tel que toute la tension du réseau est appliquée subitement à la lampe ce qui provoque un flanc de courant tout aussi abrupt. Mais les flancs de courant abrupts ne font aucun bien à la CEM. Une bobine en série avec la lampe limitera la montée rapide du flanc. Cette bobine est aussi le composant qui limite la puissance du gradateur à des lampes de 1 kW en limitant le courant à 4,5 A.

La tension qui augmente brusquement aux bornes d'un triac lors de la mise en marche peut causer sa

Figures 4 à 7. Raccordement des organes de commande à la carte.



perce. Un élément RC monté en parallèle sur le triac réduira la probabilité de ce genre de surprise déplaisante. L'élément RC protège aussi le triac des pointes de tension qui pourraient être engendrées par l'ouverture d'un circuit comportant des inductances et empêche tout amorçage intempestif.

Pour éviter de renvoyer au réseau les pointes de tension et de courant résiduelles, on montera un condensateur de 220 nF en parallèle sur l'entrée secteur. Se servir de condensateurs de déparasitage X agréés par un organisme de normalisation.

Montage

Le recours à des composants SMD pour la partie de commande (format 0805, faute de quoi 1206) permet de réduire la taille du gradateur. Les caractéristiques des transistors et diodes SMD indiqués dans la liste de pièces ne sont pas critiques (attention toutefois à D10 et D11) et ils peuvent être remplacés sans autre par des composants similaires de même brochage. Placez de préférence ces composants en premier sur la platine double face (représentée légèrement réduite dans la **figure 8**). Vous pouvez ensuite pla-

cer tous les autres composants, des plus petits aux plus grands. Veillez à ne pas déclencher le protecteur thermique en le soudant. La soudure SMD est un vrai trifouillage qu'il vaut mieux effectuer sous l'œil attentif d'une loupe éclairante.

Les triacs (ainsi que le régulateur de tension) doivent être refroidis. Un boîtier métallique constitue la solution la plus simple. Mais on cherchera en vain des boîtiers tout assemblés dans les catalogues d'électronique ; il faut mettre soi-même la main à la pâte. Les petits boîtiers de chez Teko, fort appréciés et relativement faciles à plier, constituent une bonne source d'inspiration. Une scie à chantourner permettra de découper des trous de la bonne taille dans le couvercle pour les prises. Il faut en outre, comme on le voit dans la photo du titre, percer des trous pour les 3 douilles, les commutateurs à molette, les 3 LED, l'inverseur DMX/analogique et le potentiomètre. Percez les trous de fixation des triacs (y compris leurs LED) et du régulateur de tension sur un des côtés. Contrairement aux types de triacs utilisés, le régulateur de tension ne doit pas être isolé du boîtier (par exemple par une plaque de mica).

Introduisez la platine partiellement équipée dans le boîtier terminé, vissez tout d'abord à fond les composants mentionnés en dernier au boîtier et fixez-les sur le dessus de la platine avec une gouttelette de soudure. Ils seront correctement soudés plus tard. On évite ainsi l'apparition de tensions mécaniques.

Les pistes qui passent sous les bobines doivent être séparées de celles-ci par un peu de bande isolante. Fixer les bobines à l'aide d'attaches de câble et d'un peu d'adhésif à chaud et les souder. Montez les prises à contact de protection sur des entretoises de 15 mm de haut pour qu'elles dépassent les bobines. Raccorder des fils de cuivre d'un diamètre de 1,5 mm aux bornes des prises, les faire passer par la carte et les souder.

Sécurité

Venons-en au chapitre sur la sécurité électrique que vous attendiez impatiemment. Comme toujours avec de l'électronique fonctionnant à la tension du réseau, et particulièrement dans le cas d'un appareil de classe I à boîtier métallique, la disposition sur la carte acquiert une grande importance. Il faut que les parties soumises à la tension du secteur et les pistes soient séparées les unes des autres et du boîtier par une distance minimale (3 mm) ; en outre, toutes les parties métalliques du boîtier avec lesquelles on pourrait venir en contact et toutes celles, comme les boutons métalliques, qui dépassent vers l'extérieur, doivent être raccordées à

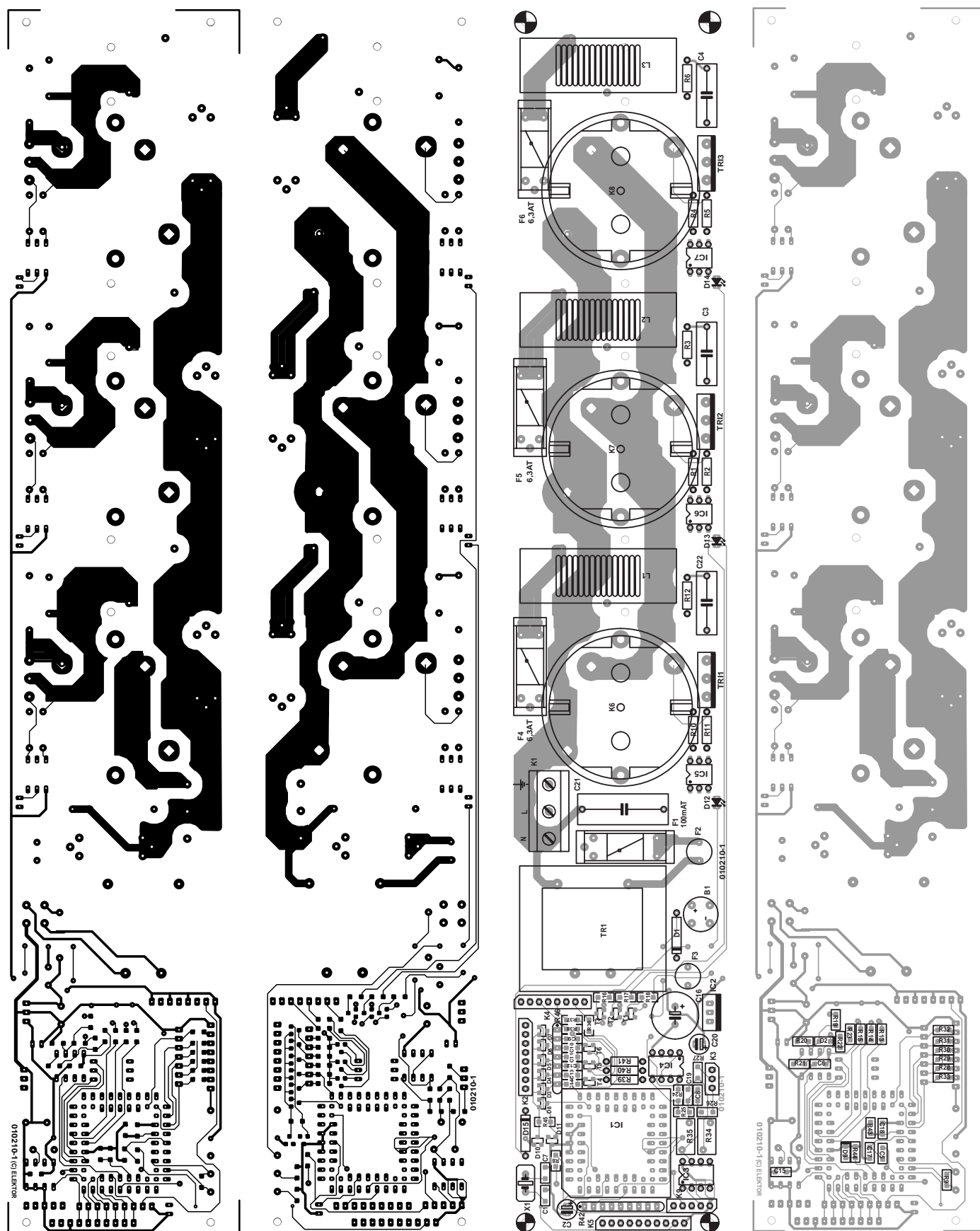


Figure 8. Tout sur une seule carte : schéma de montage et disposition de la carte d'atténuation à l'échelle 1 : 1.5 (67 %).

Liste des composants

(tous CMS 0805)

Résistances :

R1, R4, R10 = 47 Ω
 R2, R5, R11 = 1 k Ω
 R3, R6, R12 = 100 Ω
 R7 non implantée
 R8, R13 à R15, R44, R45 = 4 k Ω CMS
 R9 = 4 M Ω CMS
 R16 à R18 = 270 Ω CMS
 R19, R20 = 7 k Ω CMS
 R21 = 1 k Ω CMS
 R22, R28 à R33 = 47 k Ω CMS
 R23 à R25 = 10 k Ω CMS
 R26, R27 = 120 Ω CMS
 R34, R35 = varistor 12 V K14/12
 R36 à R38 = 270 Ω
 R39 à R41 = 10 k Ω
 R42 = réseau de 8 résistances de 10 k Ω
 R43 = 10 Ω CMS
 R46 = réseau de 6 résistances de 47 k Ω

Condensateurs :

C1, C7 = 22 pF CMS
 C2 = 1 μ F/16 V vertical
 C3, C4, C22 = 100 nF/250 V~ X2
 C5, C6, C9 à C14, C17, C18 = 33 nF CMS
 C8, C19 = 470 pF CMS
 C15 = 100 nF CMS
 C16 = 1 000 μ F/35 V vertical
 C20 = 4 μ F/16 V vertical
 C21 = 220 nF/250 V~ X2

Semi-conducteurs :

B1 = B40C800 modèle rond
 D1, D15 = 1N4004
 D2, D9 = 1N4148
 D10, D11 = BAS40
 D3 à D8 = BAV99
 D12 à D14 = LED
 IC1 = 68HC11 IF1N (Motorola programmé **EPS010210-41***)
 IC2 = 7805
 IC3 = LTC485 CN8

IC4 = LM358N
 IC5 à IC7 = MOC3020
 T1 à T6 = BC847

Divers :

F1 = fusible 100 mA retardé avec porte-fusible et couvercle plastique
 F2 = fusible thermique 100 °C (Conrad 534137)
 F3 = Polyswitch 170 mA (Conrad 551198)
 F4 à F6 = fusible 6A3T avec porte-fusible et couvercle plastique
 K1 = bornier encartable à 3 contacts au pas de 7,5 mm (RM7,5)
 K3 = embase autosécable à 1 rangée de 3 contacts
 K2, K4 = embase autosécable à 1 rangée de 8 contacts
 K5 = embase autosécable à 1 rangée de 10 contacts
 K6 à K8 = prise secteur encastrable (Conrad 620297)
 K9 = embase autosécable à 1 rangée de 5 contacts
 L1 à L3 = bobine torique 1 mH3, 4,5 A (Conrad 534366)
 TR1 = transformateur 15 V, 3VA2 (ou autre transformateur EL30/13,6)
 TR11 à TR13 = BTA26-600A (isolé !), 600 V, 25 A, 100 mA
 X1 = quartz 16 MHz
 Support PLCC à 68 contacts
 * = cf. site Elektor à www.elektor.fr

Périphérie :

K10 = embase BIN à 8 contacts pour montage châssis
 K11 = embase XLR mâle à 5 contacts pour montage châssis
 K12 = embase XLR femelle à 5 contacts pour montage châssis
 D15 à D17 = LED verte
 S1 à S3 = roue codeuse BCD
 D18 à D29 = 1N4148 (à souder directement sur S1 à S3)
 P1 = potentiomètre 47 k Ω linéaire
 S4 = interrupteur unipolaire

dès que l'appareil est alimenté par le secteur. La luminosité des lampes peut alors être réglée de 0 à 100 % en tournant le potentiomètre. Raccorder ensuite une source de tension de 0 à 10 V et placer le commutateur sur ANALOGIQUE pour que la LED analogique D17 s'allume. Positionner les commutateurs de codage sur 001. Si la LED clignotait encore, elle devrait maintenant cesser. Si l'on applique alors une tension au canal analogique 1, la lampe de la prise 1 s'allumera, au canal 2, la lampe de la prise 2 etc. Les commutateurs de codage déterminent le canal attribué à la prise analogique 1. Positionnons les commutateurs sur la valeur 3 ; cela signifie que la prise 1 est attribuée au canal 3, la prise 2 au canal 4 et la prise 3 au canal 5. La LED clignote en signe d'erreur si la valeur choisie dépasse 6.

Appliquons enfin un signal DMX ; la LED du signal DMX papillote au rythme de la transmission. Positionner le commutateur sur DMX pour commander les lampes par DMX. La LED analogique s'éteint et la LED DMX s'allume. Il devrait à présent être possible de commander la lampe de la prise 1 par le canal DMX défini par les commutateurs de codage et celle de la prise 2 par le canal suivant. La LED DMX clignote si les commutateurs de codage ont été positionnés sur une valeur DMX erronée (0 ou plus de 512).

Un dernier conseil avant de se mettre au travail : 1 kW est la charge la plus élevée par canal. Le fonctionnement du gradateur DMX a été testé par le laboratoire d'Elektor avec des spots de 100 W (approche qui ne requiert pas encore de refroidissement des triacs). Nous vous conseillons de procéder de la même façon. Vérifiez aussi la température du régulateur de tension lors des essais. Si la température de ce composant devient trop élevée, il ne reste plus qu'à utiliser un modèle plus puissant ou à monter un dissipateur thermique.

(010210)

demeure au conducteur de protection. La masse du circuit de la partie basse tension doit être aussi reliée au conducteur de protection. N'utilisez à aucun prix des triacs non isolés ! Ce serait théoriquement possible, mais l'isolation électrique nécessaire, composée de plaques de mica et de vis de plastique, EST ILLICITE. La carte est plus solidement immobilisée lorsque le cadre des commutateurs à molette est fixé à la face supérieure du boîtier. N'utiliser des vis de plastique que lorsque

la distance minimale d'isolation de 3 mm est respectée. N'oubliez pas de lire attentivement la page sur la sécurité de ce fascicule et suivez ses conseils (pur égoïsme : nous avons besoin de lecteurs d'Elektor en bonne condition !).

On peut passer à la mise en service lorsque l'appareil est définitivement installé et le câble secteur déchargé de toute traction conformément à la photo. Commençons par raccorder 3 lampes à incandescence. Une LED au moins doit s'allumer ou clignoter

Canal 4

Le circuit du gradateur a été conçu de façon à permettre la réalisation d'une version à 4 canaux pour des rangées de 4 luminaires. Le signal pour un éventuel quatrième étage final à triac est disponible à la broche PA3 (broche 39) du processeur. L'auteur peut fournir un circuit et la topologie de la carte pour la version 4 canaux. Prière de s'adresser à l'auteur (allemand) à l'adresse E-mail :

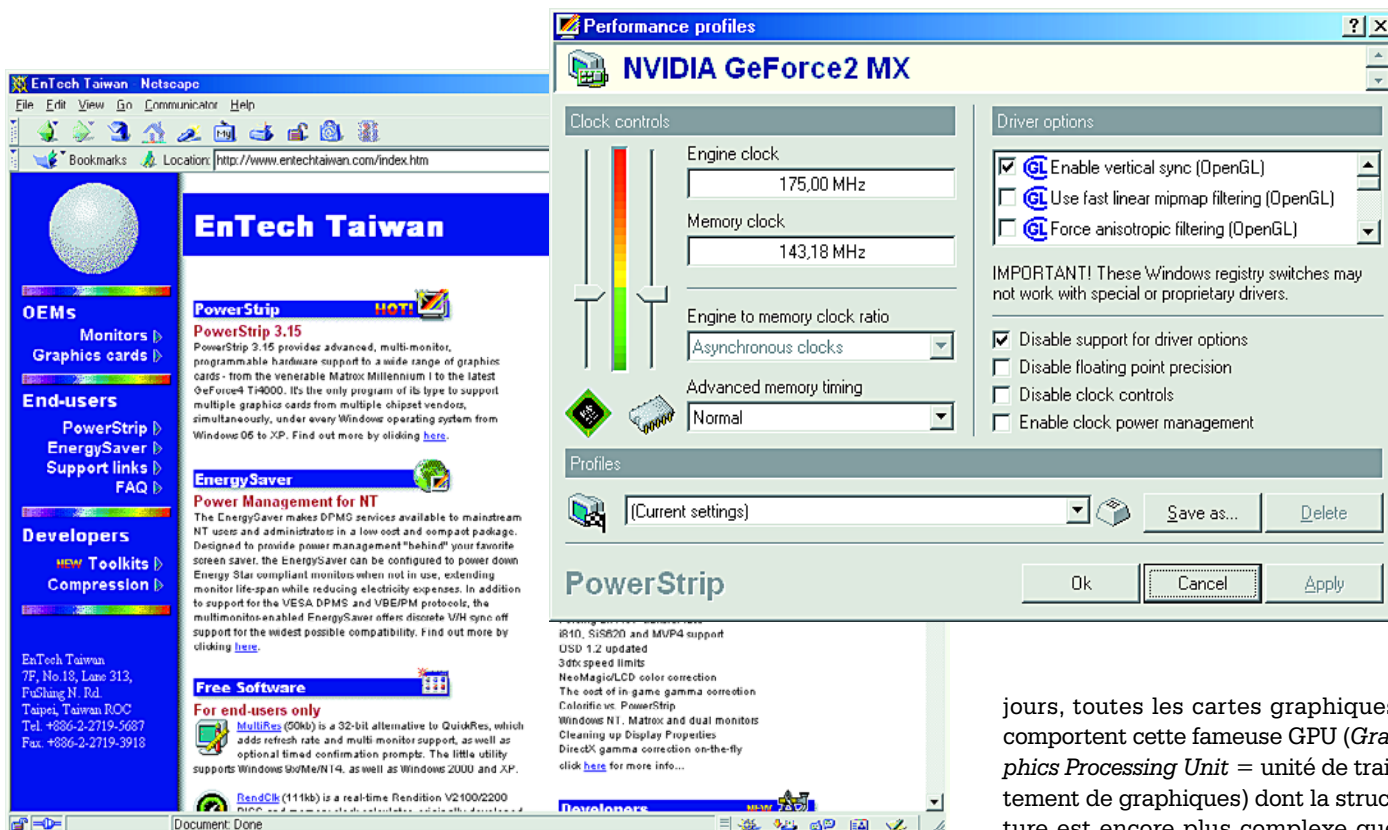
Akneer@aol.com

pour de plus amples renseignements.

Surcadençage des cartes graphiques

Gains de vitesse par logiciel

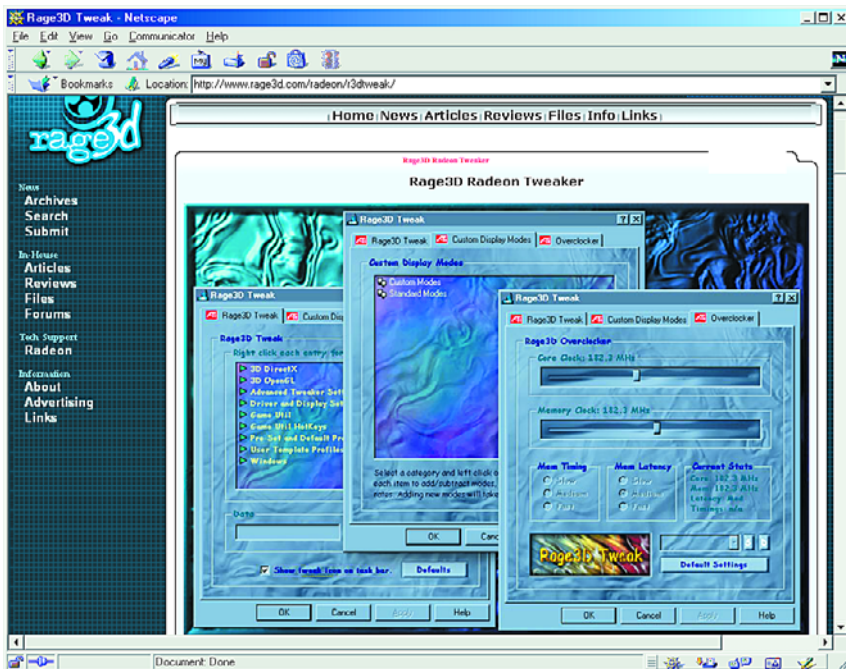
Les amateurs de jeux sur ordinateur l'ont toujours constaté : tant le processeur que la carte graphique de leur système ne sont jamais assez rapides. Les jeux les plus récents en particulier mettent à rude épreuve le processeur graphique, la GPU, présent sur la carte. L'apparition sur le marché du nouveau monstre de vitesse de nVidia, le GeForce 4, n'a fait qu'accentuer l'intérêt dans le surcadençage des cartes graphiques.



Tout ce que l'on demandait aux premières cartes graphiques était de convertir les informations devant constituer l'image à l'écran

en un format vidéo que le moniteur puisse utiliser. Mais cette époque est révolue depuis belle lurette. De nos

jours, toutes les cartes graphiques comportent cette fameuse GPU (*Graphics Processing Unit* = unité de traitement de graphiques) dont la structure est encore plus complexe que celle du processeur présent sur la carte-mère. La GPU remplit de nombreuses tâches et procède plus spécifiquement directement à nombre d'opérations graphiques sans l'in-



tervention de la CPU (l'unité centrale). Il est possible ainsi de créer des formes complètes, des ombres, et autres impacts de lumière à une vitesse telle que l'utilisateur ait, sur son écran, des mouvements de personnage et autres paysages parfaitement souples. Ceci est plus particulièrement important dans le cas des jeux et c'est bien là le domaine où la puissance de calcul des processeurs graphiques modernes est utilisée la plus efficacement.

Dans le cas d'une carte graphique, il est certains facteurs plus importants que d'autres pour la puissance de traitement : la vitesse du processeur (de la carte graphique), la vitesse de la mémoire et les fonctions additionnelles intégrées matériellement dans le noyau du processeur (le nombre de pipelines pour le calcul des formes géométriques par exemple). Il est impossible de modifier quoi que ce soit à ce niveau, ces caractéristiques matérielles étant fixées une fois pour toutes. Il est partant illusoire d'imaginer atteindre, avec une GeForce 2 voire 3, la puissance de calcul et les possibilités offertes par une GeForce 4. Il n'en reste pas moins qu'il existe des possibilités de peaufinage de certains facteurs, opération connue chez les spécialistes sous le terme de « *tweak* ». Il est en effet possible, relativement facilement, de jouer sur les fréquences de travail de la GPU et de la mémoire

(ces fréquences sont en effet indépendantes l'une de l'autre) et cela par le biais de l'un ou l'autre registre; d'un point de vue matériel (*hardware*), cela n'implique pas de modification. Il est en effet possible, à l'aide de programmes sophistiqués, de faire passer, par exemple, la vitesse du processeur graphique de 220 à 250 MHz. En fonction du système de refroidissement dont est doté la carte et la qualité de ses composants, il y a toujours une certaine marge de performances que l'« amateur de surcadencage » peut tenter d'utiliser. On a vite fait de voir que les choses tournent au vinaigre par l'apparition d'irrégularités dans l'image, voire la disparition totale de l'image. Dans la majorité des cas on pourra revenir à un fonctionnement normal par le choix d'une fréquence plus faible (celle prévue avant les modifications), mais il faut bien être conscient du fait que ces tentatives de surcadencage peuvent avoir des conséquences désastreuses si l'un des composants était endommagé par surchauffe et trépasserait prématurément. Tout le monde sait que l'on court des risques lors d'essais de surcadencage.

Il n'y a plus, aujourd'hui, que 2 grands fabricants de processeurs graphiques à fonctionnalité 3D, suivis, à distance respectueuse, par des joueurs de « Deuxième Division ». La majorité des amateurs de jeux sur

micro-ordinateur auront partant dans leur système une carte graphique dotée d'un processeur de nVidia (GeForce 2, 3 or 4) ou de ATI (Rage, Radeon, 7500, 8500). Dans ces conditions, notre quête à la recherche d'outils de « *tuning* » de la carte graphique recouvrira la part la plus importante du marché même si nous nous limitons à ces 2 fabricants. Il n'en reste pas moins qu'il existe certains programmes pour les marques moins représentées sur le marché comme nous le verrons un peu plus loin.

Nous commençons notre liste d'utilitaires de surcadencage par le fameux **Powerstrip** [1] de la firme taiwanaise EnTech. Ce programme offre à son utilisateur de nombreuses possibilités de modifications des paramètres de la carte graphique. Il supporte nombre de moniteurs et reconnaît les caractéristiques de la plupart des cartes graphiques, de la Matrox Millennium I aux toutes dernières Radeon 8500 et GeForce 4.

Une paire de régleurs à glissière permettent de jouer indépendamment sur la fréquence de la GPU et de la mémoire. Le programme offre la possibilité de créer des profils spécifiques aux applications, moniteur(s), correction de couleur et de réglage de vitesse. Il est même utilisable par le débutant qui n'a (encore ?) que peu d'expérience dans le domaine du surcadencage.

Les cartes GeForce extrêmement populaires ont des défenseurs fervents sur Internet, témoins les nombreux sites consacrés spécialement à cette famille de cartes. La plupart de ces sites comportent une section de téléchargement de programmes de « *tweaking* ». Nous sommes, dans le cadre de cet article, limité à un site sachant que les programmes proposés partout sont pratiquement les mêmes.

On trouve, sur **GeForce Zone** [2], une page de peaufinage comportant une dizaine de programme de surcadencage (tels que NVMax,

Adresses Internet :

- [1] Powerstrip (shareware):
www.entechtaiwan.com/ps.htm
- [2] GeForce Zone
www.geforcezone.com/tweaks.htm
- [3] Rage3D:
www.rage3d.com/radeon/
- [4] Grafikkarten Tools:
www.grafik.overclocker-tools.de/
- [5] Guru3D:
www.guru3d.com/files/
- [6] Tweak3D: GeForce3 cooling:
www.tweak3d.net/reviews/misc/gf3cooling/

Coolbits et Rivatuner), d'édition du BIOS présent sur la carte graphique et autres outils pratiques.

Le site le plus intéressant (à notre avis et jusqu'à plus ample informé) consacré aux cartes ATI est **Rage3D** [3]. Il reprend tout ce dont on peut avoir besoin avec les GPU modernes, de la Rage à la Radeon 8500. On y trouve les pilotes les plus récents et divers utilitaires. On découvrira, sous le bouton Tech Support, différents utilitaires de peaufinage tels que Rage3D Tweak, UniTuner, Radeon Tweaker et RadTweak.

On se voit confronté, sur le site allemand **Grafikkarten Tools** [4] (outils pour cartes graphiques), à une page massive proposant nombre d'utilitaires de surcadencage. On trouve à cet endroit un programme au moins, voire plus, pour toutes les puces connues, telles que Voodoo, S3, nVidia, Matrox, ATI et Kryo.

Sur **Guru3D** [5] aussi on trouve nombre d'auxiliaires pour les différents types de GPU, outre des utilitaires de surcadencage, également des pilotes et des programmes de test. La vitesse maximale à laquelle on peut espérer faire tourner une carte dépend souvent du système de refroidissement de la CPU et des circuits de mémoire. Si vous en avez le courage, vous pouvez doter le processeur et la RAM d'un dispositif de refroidissement addi-



tionnel, ce qui vous permettra de « tordre » (ce qui est d'ailleurs la signification première du verbe anglais *tweak*) quelques mégahertz supplémentaires.

Tweak3D [6] propose une description exhaustive sur l'art et la manière

de s'y prendre.

(025034)

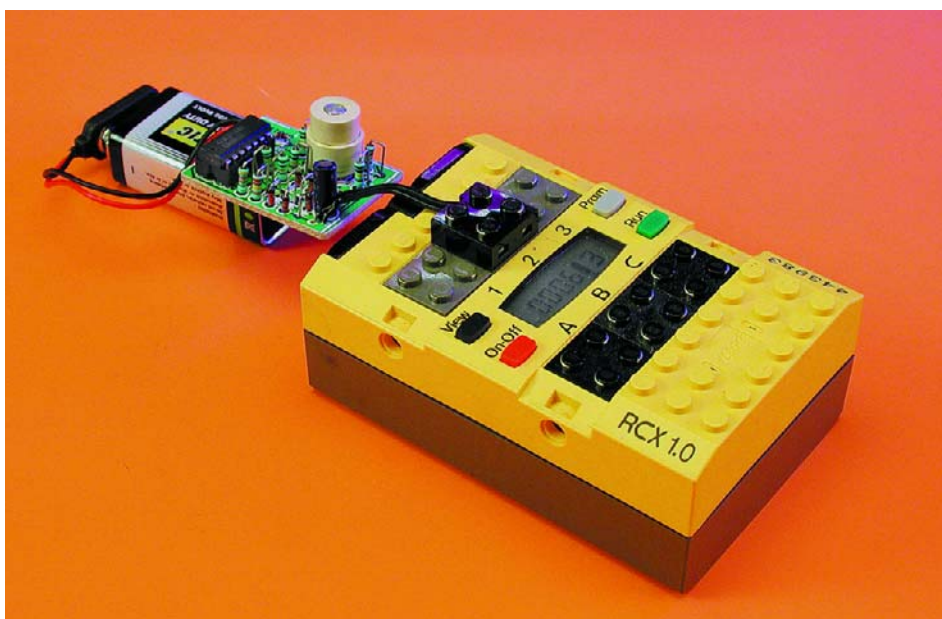
Boussole pour RCX de Lego

Plus jamais fausse route

un projet de Z. Otten

à la rédaction : Luc Lemmens

Mindstorms (remue-méninges), la série de robots désormais célèbre de Lego, a maintes fois inspiré Elektor. En l'an 2000, nous vous avons présenté, en feuilleton, cinq articles sur les tenants et les aboutissants du module RCX, un bloc de construction à microcontrôleur avec trois entrées pour capteurs et trois sorties de commande. Ce fut ensuite le tour d'une cellule photosensible, d'un capteur de proximité, d'un multiplexeur d'entrée, sans oublier l'interface I²C. Aujourd'hui, nous allons définitivement empêcher notre robot de perdre le nord.



On peut faire circuler un robot dans un environnement donné de différentes manières. Lui laisser trouver son chemin tout seul. Détecteurs de proximité et de contact permettront

au robot d'éviter les obstacles ou les collisions et finalement de se retrouver dans une situation inextricable. La panoplie Mindstorms contient les

capteurs nécessaires. Une deuxième possibilité, c'est de fixer au robot un but à atteindre, lui faire découvrir la route qui conduit à un endroit défini. Pensons à une ligne guide, comme sur le tapis joint à la boîte de construction.

On peut aussi programmer d'avance le parcours dans le RCX en prédisant le temps pendant lequel les différents moteurs devront fonctionner, après avoir évalué le trajet à parcourir au cours de ces périodes. L'inconvénient de cette méthode, c'est que le régime des moteurs va dépendre de la tension de batterie du RCX. Il diminuera au fil du temps et le robot s'éloignera de plus en plus du trajet prévu. Mais avec une boussole, il serait à tout moment capable de vérifier son orientation, la direction suivie et de la modifier au besoin. Dans cet article, nous allons voir comme il est facile de doter un robot Mindstorms d'une boussole. Nous avons choisi pour cette appli-

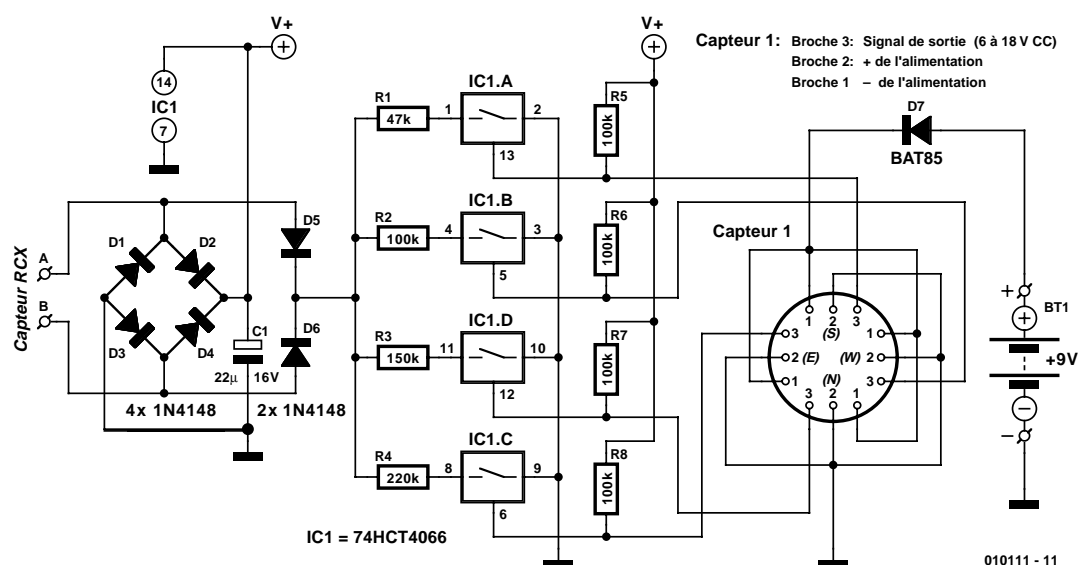


Figure 1. On trouve sur ce schéma quatre interrupteurs analogiques, quelques composants passifs et une boussole Pewatron du type 6945.

cation un capteur de Pewatron AG, un compas numérique du type 6945, et ce n'est pas la première fois qu'il apparaît dans un projet Elektor. En septembre 1996 déjà, nous avons présenté une boussole électronique basée sur le 6945. Ce capteur dispose d'un rotor miniature, d'un aimant et d'un circuit intégré particulier, à effet Hall, pour déterminer l'orientation par rapport au champ magnétique terrestre. Les quatre sorties numériques, dont seules deux sont actives à un moment donné, offrent une résolution de 45 degrés sur la rose des vents. Il faut que le capteur soit bien positionné à la verticale et tenu éloigné des champs magnétiques parasites. Pour un prix tournant autour de 50 €, on ne peut pas dire que le 6945 soit bon marché, aussi allons-nous le traiter avec les égards qui lui sont dus.

Le circuit

La **figure 1** vous présente le schéma de notre boussole pour RCX. Le circuit d'interface entre capteur et RCX s'alimente au module lui-même. C'est bien connu, le RCX est capable d'alimenter une sonde et d'échanger des données avec elle, rien que sur deux fils (cf. **figure 1**). La tension d'alimentation est présente la plupart du temps aux bornes A et B. Mais pendant de courtes périodes, elle est coupée pour procéder à la

mesure et c'est un condensateur électrolytique, C1, qui assure l'intérim. Le pont de diodes D1 à D4 autorise l'inversion des bornes A et B sans aucune conséquence. Au cours de la mesure, D5 et D6 jouent un rôle similaire pour la détermination de la résistance connectée aux entrées pour capteur du RCX.

Le capteur lui-même, notre 6945, serait trop gourmand (20 mA) et devra donc puiser son énergie à la pile de 9 V Bt1. Comme les bornes du RCX ne peuvent fournir que 10 mA, tout au plus, elles seraient mises à sec. La diode D7 protège le capteur d'une inversion de polarité dans la tension d'alimentation.

Le 6945 est équipé de quatre sorties à collecteur ouvert dont deux passent simultanément au niveau bas, selon l'orientation de la sonde. Les résistances d'excursion haute R5 à R8 adaptent les niveaux de sortie à la tension d'alimentation de l'interface, prélevée, nous l'avons dit, sur le RCX.

Les quatre sorties du 6945 commandent autant d'interrupteurs analogiques, contenus dans IC1, lesquels organisent des combinaisons de résistances en parallèle, avec R1 à R4, à l'entrée de sonde du RCX. Le **tableau 1** résume les différentes valeurs que le RCX va pouvoir mesurer, en fonction de l'orientation du capteur. Nous y trouvons les huit azimuts magnétiques correspondants.

Remarquons que cela ne représente qu'une petite partie du domaine de mesure du RCX, qui est capable d'une résolution à 1 024 niveaux.

On aurait pu appliquer directement les résistances R1 à R4 aux sorties du capteur et épargner ainsi IC1 et R5 à R8. Mais vu le prix de cette sonde et le laconisme du feuillet de caractéristiques sur sa capacité de charge en sortie, nous avons préféré jouer la sécurité.

Construction

Si c'est la compacité qui vous intéresse au premier chef, mieux vaut partir d'une platine perforée qui autorise les dimensions les plus faibles. Mais si vous êtes plus à l'aise à ce point de vue, la **figure 2** vous fournira un tracé adéquat. Il faut être attentif aux 4 ponts de câblage à proximité du capteur, en parti-

Tableau 1.

Valeurs de référence pour les résistances R1 à R4.

| Point cardinal | 3 | 6 | 9 | 12 | valeur RCX |
|----------------|---|---|---|----|------------|
| N | 1 | 1 | 1 | 0 | 863 |
| NE | 0 | 1 | 1 | 0 | 920 |
| E | 0 | 1 | 1 | 1 | 810 |
| SE | 0 | 0 | 1 | 1 | 856 |
| S | 1 | 0 | 1 | 1 | 818 |
| ZO | 1 | 0 | 0 | 1 | 830 |
| O | 1 | 1 | 0 | 1 | 787 |
| NO | 1 | 1 | 0 | 0 | 887 |

culier le plus minuscule qui lui se trouve en dessous du capteur, pratiquement au milieu. Il est plus pratique de le réaliser en fil de cuivre isolé et de le placer sur la face cuivrée, ceci pour permettre au capteur de reposer directement sur la platine.

Raccorder le capteur au RCX, nous l'avons fait à l'aide d'un cordon Lego standard coupé en deux. La pile, il faut la placer aussi loin que possible de la sonde, son boîtier métallique pourrait influencer la mesure.

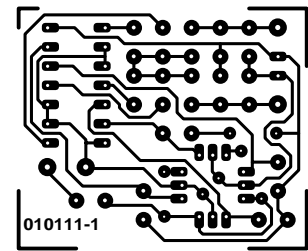
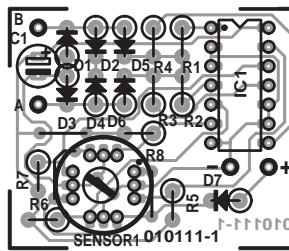


Figure 2. Par souci de compacité, les résistances et diodes se montent à la verticale. N'oubliez pas le pont de câblage sous le capteur.

Listage 1. Exemple de programme en Visual Basic.

```
Private Sub Command1_Click()
Const setpoint = 5
Const measure = 6
Const dif = 7

With RCX
.ComPortNo = 1 ' COM port 1
.LinkType = 0 ' 0 = IR
.PBrick = 1 ' 1 = RCX, 0 = CyberMaster
.InitComm ' Initialise the COM port

.SelectPrgm SLOT3
.BeginOfTask MAIN
.SetSensorType SENSOR_2, LIGHT_TYPE
.SetSensorMode SENSOR_2, RAW_MODE, 0

.SetVar setpoint, CON, 1 ' 1=N, 2=NE, 3=E, 4=SE, 5=S, 6=SW, 7=W, 8=NW
.SetVar measure, CON, 0
.SetVar dif, CON, 0

.StartTask 1
.EndOfTask 'Main

.BeginOfTask 1
.Loop CON, forever
'North
.If SENVAL, SENSOR_2, EQ, CON, 863
.SetVar measure, CON, 1
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 862
.SetVar measure, CON, 1
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 861
.SetVar measure, CON, 1
.EndIf
'North-East
.If SENVAL, SENSOR_2, EQ, CON, 920
.SetVar measure, CON, 2
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 919
.SetVar measure, CON, 2
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 918
.SetVar measure, CON, 2
.EndIf
'East
.If SENVAL, SENSOR_2, EQ, CON, 810
.SetVar measure, CON, 3
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 809
.SetVar measure, CON, 3
```

Liste des composants

Résistances :

R1 = 47 kΩ
R2, R5 à R8 = 100 kΩ
R3 = 150 kΩ
R4 = 220 kΩ

Condensateurs :

C1 = 22 μF/16 V radial

Semi-conducteurs :

D1 à D6 = 1N4148
D7 = BAT85
ICI = 74HCT4066

Divers :

Capteur I = Pewatron 6945
(Pewatron AG,
www.pewatron.com)
BTI = pile compacte 9 V +
connecteur à pression
2 câbles de connexion Lego
(coupé en deux) :
l'un d'une longueur de 26,6 cm
(n° commande Lego 5311)
l'autre d'une longueur de 9 cm
(n° commande Lego 5041)

Logiciel

Le programme témoin (cf. **listage 1**) pour notre boussole est rédigé en Visual Basic. On installe d'abord le logiciel fourni sur CD avec l'ensemble Mindstorms, ainsi que la bibliothèque SPIRIT.OCX qui permet de programmer le RCX et de le piloter à partir du langage évolué. Le code peut aussi, sans grande difficulté, s'adapter à d'autres langages, comme Delphi ou C++.

Commençons avec un formulaire vierge en Visual Basic et plaçons-y un bouton Spirit control et un Command. Un double clic sur ce bouton ouvre une fenêtre dans laquelle nous copions le listage 1. Lors de l'exécu-

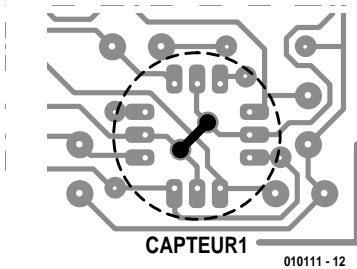


Figure 3. Ce dessin montre comment implanter le plus petit de ponts de câblage disposé côté « pistes » de la platine.

tion du logiciel en Visual Basic, l'application sera transmise par la tour au bloc RCX.

L'exemple choisi est tout simple. Le logiciel fixe une direction (normalement .SETVAR setpoint, CON, 1 ; ce dernier chiffre pouvant être compris entre 1 et 8, il détermine la direction choisie). Ensuite, on lit indéfiniment, en boucle, la valeur du capteur. Lorsque l'azimut correspond à la direction choisie au préalable, le vibreur du RCX émet un signal.

Les quatre lignes qui suivent l'instruction « With RCX » traitent de la communication entre PC et RCX. Elles n'ont aucune relation directe avec notre application. Après cela, on sélectionne le 3^e « slot » du programme dans le RCX, puis dans la tâche principale (*Main*), on règle la direction voulue pour travailler avec la boussole (*setpoint*), le registre destiné à mémoriser la valeur mesurée (*measure*) et l'on met à zéro la différence (*dif*) entre mesure et objectif. La dernière instruction de *Main* lance la tâche 1, celle qui mesure en boucle sans fin.

Chaque mesure de direction est exécutée trois fois de suite pour éviter la dispersion. Lorsque le résultat se situe dans les aires de vent d'un des secteurs, on transmet le numéro d'ordre de ce secteur.

Après les 24 comparaisons de la mesure, on vérifie si la valeur correspond à la direction choisie. Si c'est le cas (*dif*=0), le système entonne un cri de victoire, le *Playsystemsound*.

(010111)

Bibliographie :

Lego Robotics Invention System
Elektor 4 à 9 / 2000 et 11 / 2000.

```
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 808
.SetVar measure, CON, 3
.EndIf
'South-East
.If SENVAL, SENSOR_2, EQ, CON, 856
.SetVar measure, CON, 4
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 855
.SetVar measure, CON, 4
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 854
.SetVar measure, CON, 4
.EndIf
'South
.If SENVAL, SENSOR_2, EQ, CON, 818
.SetVar measure, CON, 5
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 817
.SetVar measure, CON, 5
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 816
.SetVar measure, CON, 5
.EndIf
'South-West
.If SENVAL, SENSOR_2, EQ, CON, 830
.SetVar measure, CON, 6
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 829
.SetVar measure, CON, 6
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 828
.SetVar measure, CON, 6
.EndIf
'West
.If SENVAL, SENSOR_2, EQ, CON, 787
.SetVar measure, CON, 7
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 786
.SetVar measure, CON, 7
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 785
.SetVar measure, CON, 7
.EndIf
'North-West
.If SENVAL, SENSOR_2, EQ, CON, 887
.SetVar measure, CON, 8
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 886
.SetVar measure, CON, 8
.EndIf
.If SENVAL, SENSOR_2, EQ, CON, 885
.SetVar measure, CON, 8
.EndIf

'The sensor value is stored in register 'measure'
'the target value in 'setpoint'
.SetVar dif, VAR, measure 'dif=measure-setpoint
.SubVar dif, VAR, setpoint
.If VAR, dif, EQ, CON, 0
.PlaySystemSound 0
.EndIf
.EndLoop
.EndOfTask
End With

End Sub
```

Bascom-51

Un nouveau compilateur BASIC pour la carte 89S8252 Flash

Burkhard Kainka

Il était possible, jusqu'à présent, d'utiliser la carte 89S8252 Flash d'Elektor avec, entre autres, l'interpréteur BASIC-52. Les amateurs du langage de programmation BASIC ont, cependant, une alternative : BASCOM-51, la nouvelle étoile au firmament des compilateurs.

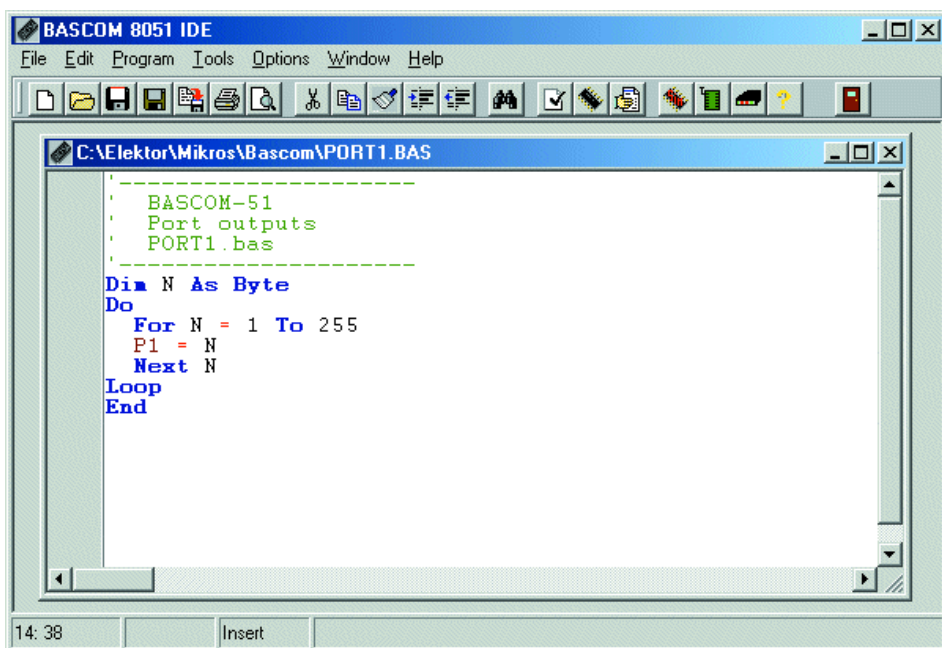


Figure 1. L'environnement de programmation.

Nous devons à Mark Alberts et à sa société néerlandaise (www.mcselec.com) la naissance de BASCOM, un compilateur développé à l'origine pour la famille AVR d'Atmel. Plus tard, il élargit cette famille de compilateurs par la version BASCOM-LT destinée tout particulièrement au petit contrôleur du type 89C2051. Plus tard, il en développa une nouvelle version, BASCOM-51, destinée aux dérivés plus puissants de la famille 8051. Il est possible de télécharger, depuis le site Internet de MSC-Electronics une version de démonstration de BASCOM-51, capable de

générer des programmes dont la taille est limitée à 2 Koctets, une taille respectable dans le monde des compilateurs, sachant que le code généré par BASCOM est très efficace (lire compact). Nous avons testé cette version de démonstration avec notre carte 89S8252 Flash, le résultat de ce test étant plus que satisfaisant, pourquoi ne pas dire simplement excellent !.

Le compilateur connaît un certain nombre de paramètres importants.

Sous le point de menu « Options/Compiler/Output » on choisit le type de fichier à générer en fin d'opération. Le type le plus important est le fichier Hex, sachant que ce type de fichier (*hexfile*) peut être transféré dans le contrôleur à l'aide d'un programme tel que MicroFlash ou AtmelISP. Le fichier de type Debug est très intéressant pour travailler avec le simulateur intégré de BASCOM. Le fichier de type Error sert à visualiser les messages donnant les erreurs ayant eu lieu en cours d'opération.

On trouve un autre paramétrage important sous le point de menu « Options/Compiler/Output/Misc », paramètre où se fait le choix du fichier Registre (*Register file*), 89s8252.dat dans le cas présent. Le compilateur connaît dès lors tous les registres à fonction spéciale (SFR = *Special Function Register*) de ce contrôleur.

La meilleure façon d'essayer ce compilateur est de lui faire exécuter un petit programme pour sorties de port rapides. Le premier point qui frappe est de constater que BASCOM se passe de numérotation de lignes d'instructions. Il s'agit partant d'un BASIC structuré. On utilise ici, au lieu du classique GOTO + numéro de ligne, des Do, While et autres instructions similaires.

BASCOM suit une tendance en cours de généralisation : les langages de

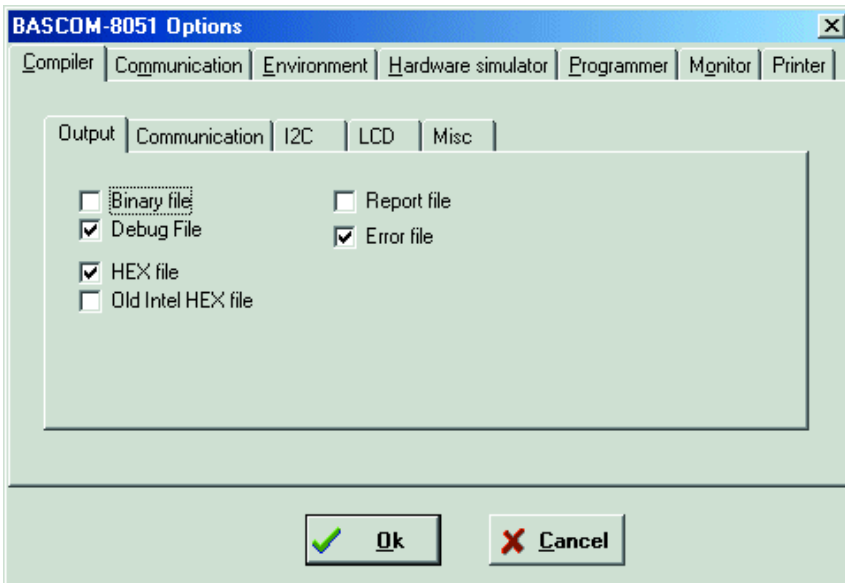


Figure 2. Le paramétrage du compilateur.

programmation tels que BASIC, C et Pascal se rapprochent de plus en plus.

Seconde constatation agréable, il faut déclarer le type de toutes les variables. Il est possible, par exemple, de travailler avec des octets (Bytes) ce qui nous rapproche très près de ce que l'on pourrait faire en assembleur. On peut en outre indiquer si les variables concernées doivent se trouver en mémoire interne ou externe. BASCOM-51 peut fort bien se débrouiller sans mémoire externe, ce qui est loin d'être évident dans le cas d'un compilateur.

```
' _____
'   BASCOM-51
'   Port outputs
'   PORT1.bas
' _____
Dim N As Byte
Do
```

NiliPascal

L'assembleur, le BASIC et C ne sont pas les seuls langages de programmation utilisables pour la programmation de microcontrôleurs. Le Pascal fait également partie de cette famille. Nous avons testé un Pascal connu, NiliPascal, sur la carte 89S8252 Flash et cela avec succès. Il est important de bien définir les paramètres corrects au niveau du compilateur :

| | |
|--------------|-----------|
| Codesegment: | \$0000 |
| Ramsegment: | \$0000 |
| Ramsize: | \$8000 |
| MHZ: | \$11.0592 |
| Baud(ser.): | 9600 |

C'est avec ce paramétrage que nous avons converti le programme en Pascal donné ci-après (lecture de l'état d'un port) :

```
{ Copyright by O'Neil Annand Valentin Som
Programme de test sous NiliPascal pour simulation sous NiliSim }
```

```
program port_test;

const port1_0 = $90;

var i:integer;

begin
  { mettre toutes les sorties à '1' pour la lecture d'un port }
  writeport ($FF,1);
  writeln ('Port1=',port(1));
  write ('Bit 0 de Port 1 est à ',bit(port1_0) ' logique ');
end.
```

Le fichier de type .hex obtenu en sortie a été chargé dans le contrôleur, le résultat de cette opération ayant été testé avec succès par le biais d'un programme de terminal. On peut partant affirmer que NiliPascal est utilisable lui aussi.

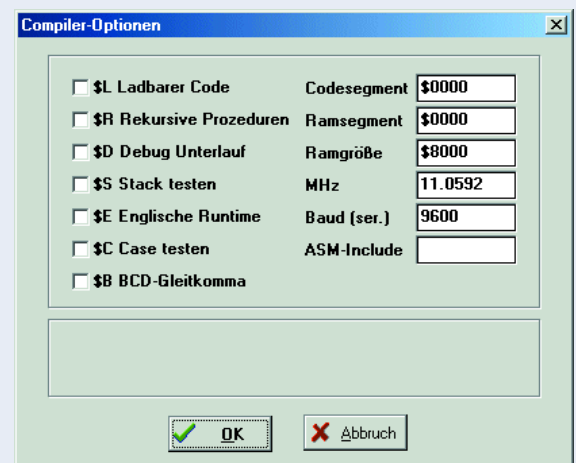
Sachez qu'il existe un ouvrage accompagné du logiciel NiliPascal :

Compilateur croisé PASCAL, O'Neil A. V. Som, Publitrone, ISBN – 2-86661-099-7

Autre ouvrage consacré au PASCAL digne d'intérêt :

Je programme en PASCAL les microcontrôleurs de la famille 8051 (80C537)

Bernd vom Berg, Peter Groppe, Publitrone, ISBN – 2-86661-098-9



```

For N = 1 To 255
  P1 = N
Next N
Loop
End

```

Avec chaque nouveau logiciel de programmation le premier aspect auquel on s'intéresse est bien évidemment la vitesse d'exécution que l'on peut espérer atteindre. Un oscilloscope branché à la ligne de port P1.1 donne une réponse rapide et indéniable : 30 μ s et pas plus pour chaque nouvelle sortie. À titre de comparaison, si l'assembleur est, avec 3 μ s seulement, 10 fois plus rapide, le compilateur-C READS-51 requiert lui, 125 μ s pour la même tâche, BASIC-52 ayant besoin lui de pas moins de 2 500 μ s. On le constate, BASCOM-51 est le vainqueur indiscutable dans la catégorie des compilateurs et interpréteurs que nous avons comparés. Autre avantage important par rapport à BASCI-52, les programmes générés sont du type à capacité de démarrage d'exécution automatique (*auto-start*).

Le compilateur dispose de plusieurs interfaces sérieles. Les instructions INPUT et PRINT transfèrent des données par le biais de l'UART (*Universal Asynchronous Receiver/Transmitter* = Émetteur/Récepteur Asynchrone Universel) interne du contrôleur. Le taux de transmission (*baudrate*) est à paramétrer sous le point de menu « Options/Compiler/Communication ».

```

' _____
' BASCOM-51
' Port inputs and outputs
' RS232.bas
' _____
Dim N As Byte

```

```

While 1 = 1
  Input N
  P1 = N
  N = P1
  Print N
Wend

```

Il est également possible d'utiliser, outre l'UART matériel, un UART logiciel, de sorte que la carte 89S8252 Flash d'Elektor peut également disposer de plusieurs interfaces. Le Listage ci-après montre comment utiliser ces UART, sachant que l'on utilise, pour le premier test, les mêmes contacts que pour l'UART que possède le processeur. Au niveau de l'UART logiciel on traite des octets individuels par le biais des instructions GET et PUT.

```

' _____
' BASCOM-51

```

Tableau 1. Comparaison Compilateur/Assembleur

| Language | Taille de mémoire requise | Durée de boucle | Auto-start |
|------------|---------------------------|-----------------|------------------------|
| Basic-52 | 8 K ROM, RAM | 2500 μ s | avec EEPROM uniquement |
| READS51 C | >4 K ROM, RAM | 125 μ s | oui |
| Assembleur | < 1 K ROM | 3 μ s | oui |
| BASCOM-51 | < 2 K ROM | 30 μ s | oui |

```

' Port input freq. divider
' Toggle.bas
' _____

```

```

While 1 = 1
  While P1.0 = 1
    Wend
    P1.1 = Not P1.1
    While P1.0 = 0
      Wend
    Wend
  Wend

```

BASCOM-51 est capable de traiter non seulement des octets de ports mais également des bits individuellement. L'exemple ci-après divise la fréquence disponible sur la ligne de port P1.0 et fournit la moitié de cette fréquence sur la ligne P1.1, ceci par inversion, lors de chaque flanc descendant, du bit P1.1. Le programme travaille partant en diviseur binaire. La fréquence d'entrée maximale est de 50 kHz.

```

' _____
' BASCOM-51
' Port outputs
' PORT1.bas
' _____

```

```

Dim N As Byte
Do
  For N = 1 To 255
    P1 = N
  Next N
Loop
End

```

Il est possible, si le besoin s'en fait sentir, d'intercaler des lignes d'assembleur. L'exemple ci-après illustre la génération d'impulsions rapides sur le port P1, impulsions se suivant à intervalles de 10 ms. La temporisation de durée plus importante se fait par BASIC, les processus à la chronologie critique en assembleur.

```

' _____
' BASCOM-51
' Ass. example Assem.bas

```

```

' mixed assembler and Basic
' _____
Print "pulses"

```

```

While 1 = 1
  $asm
    mov A , #00
    mov R1,#100
L1:
    mov P1,A
    inc a
    djnz R1,L1
  $end Asm
  Waitms 10
Wend
End

```

L'intérêt de BASCOM-51 ne se limite pas aux aspects évoqués ici. Il supporte ainsi différentes interfaces utilisables avec des périphériques plus spécifiques, affichages LCD à pilotage en mode d'opération ou 8 bits, bus I²C, bus monofilaire (1-Wire bus) et comme nous l'avons illustré plus haut, l'interface sériele logicielle. Ceci simplifie très notablement nombre de processus. Nous ne pouvons que recommander à tout utilisateur de contrôleurs de la famille 8051 ou AVR de s'intéresser à BASCOM d'un peu plus près.

(021003)

On pourra, pour de plus amples informations au sujet de BASCOM, se tourner vers le site de son auteur :

www.mcselec.com

autres sites où télécharger la version de démonstration gratuite :

www.elektronikladen.de/bascomavr.html
www.elektronikladen.de/bascom8051.html
[www.ckuehnel.ch \(allemand\)](http://www.ckuehnel.ch (allemand))
[www.ckuehnel.ch/ckskript/auswahl.htm \(allemand\)](http://www.ckuehnel.ch/ckskript/auswahl.htm (allemand))

L'un des endroits où commander BASCOM est le site Internet de Elektronik Laden (RFA) :
www.elektronikladen.de

Instruction

| | |
|-----------------------|--|
| SINCLUDE | Includes an ASCII file in the program at the current position. |
| SBAUD | Instruct the compiler to override the baud rate setting from the options menu. |
| SCRISTAL | Instruct the compiler to override the crystal frequency options setting. |
| SRAMSTART | Compiler directive to specify starting internal memory location. |
| SDEFAULT XRAM | Compiler directive to place each dimensioned variable as XRAM. |
| SLARGE | Instructs the compiler that LCALL statements must be used. |
| SLOAD | Instruct the compiler to generate code for 8-bit LCD displays attached to the data bus. |
| SNOBREAK | Instruct the compiler that BREAK statements must not be compiled. |
| SNOINIT | Instruct the compiler that no initialisation must be performed. |
| SNO\$P | Includes the compiler that the stack pointer must not be set. |
| SOBJ | Includes label objectcode. |
| SRAMSTART | Specifies the location of the external RAM memory. |
| SRAMSIZE | Specifies the size of the external RAM memory. |
| SRAMSTART | Specifies the location of the ROM memory. |
| SSERIALINPUT | Specifies that serial input must be redirected. |
| SSERIALINPUT2LCD | This compiler directive will redirect all serial input to the LCD display instead of echoing to the serial port. |
| SSERIALOUTPUT | Specifies that serial output must be redirected. |
| SSIM | Generates code without waiting loops for the simulator. |
| ABS() | Returns the absolute value of a numeric variable. |
| ALIAS | Indicates that the variable can be referenced with another name. |
| ASC() | Convert a string into its ASCII value. |
| BCD() | Converts a variable into its BCD value. |
| BITWAIT | Wait until a bit is set or reset. |
| BREAK | Generates a reserved opcode to pause the simulator. |
| CALL | Call and execute a subroutine. |
| CH0() | Convert a byte, integer/Word variable or a constant to a character. |
| CLS | Clear the LCD display and set the cursor home. |
| CONST | Declares a symbolic constant. |
| CONFIG TIMER0, TIMERN | Configure TIMER0 or TIMERN. |
| CONFIG LCD | Configure the LCD display. |
| CONFIG LCDBUS | Configures the LCD databus. |
| CONFIG BAUD | Configure the uP to select the intern baud rate generator (only 80535, 80537) |
| CONFIG IWMRE | Configure the pin to use for IWMRE statements. |
| CONFIG SDA | Overrides the SDA pin assignment from the Option Settings. |
| CONFIG SCL | Overrides the SCL pin assignment from the Option Settings. |

Signification

| | |
|-----------------------|--|
| SINCLUDE | Includes an ASCII file in the program at the current position. |
| SBAUD | Instruct the compiler to override the baud rate setting from the options menu. |
| SCRISTAL | Instruct the compiler to override the crystal frequency options setting. |
| SRAMSTART | Compiler directive to specify starting internal memory location. |
| SDEFAULT XRAM | Compiler directive to place each dimensioned variable as XRAM. |
| SLARGE | Instructs the compiler that LCALL statements must be used. |
| SLOAD | Instruct the compiler to generate code for 8-bit LCD displays attached to the data bus. |
| SNOBREAK | Instruct the compiler that BREAK statements must not be compiled. |
| SNOINIT | Instruct the compiler that no initialisation must be performed. |
| SNO\$P | Includes the compiler that the stack pointer must not be set. |
| SOBJ | Includes label objectcode. |
| SRAMSTART | Specifies the location of the external RAM memory. |
| SRAMSIZE | Specifies the size of the external RAM memory. |
| SRAMSTART | Specifies the location of the ROM memory. |
| SSERIALINPUT | Specifies that serial input must be redirected. |
| SSERIALINPUT2LCD | This compiler directive will redirect all serial input to the LCD display instead of echoing to the serial port. |
| SSERIALOUTPUT | Specifies that serial output must be redirected. |
| SSIM | Generates code without waiting loops for the simulator. |
| ABS() | Returns the absolute value of a numeric variable. |
| ALIAS | Indicates that the variable can be referenced with another name. |
| ASC() | Convert a string into its ASCII value. |
| BCD() | Converts a variable into its BCD value. |
| BITWAIT | Wait until a bit is set or reset. |
| BREAK | Generates a reserved opcode to pause the simulator. |
| CALL | Call and execute a subroutine. |
| CH0() | Convert a byte, integer/Word variable or a constant to a character. |
| CLS | Clear the LCD display and set the cursor home. |
| CONST | Declares a symbolic constant. |
| CONFIG TIMER0, TIMERN | Configure TIMER0 or TIMERN. |
| CONFIG LCD | Configure the LCD display. |
| CONFIG LCDBUS | Configures the LCD databus. |
| CONFIG BAUD | Configure the uP to select the intern baud rate generator (only 80535, 80537) |
| CONFIG IWMRE | Configure the pin to use for IWMRE statements. |
| CONFIG SDA | Overrides the SDA pin assignment from the Option Settings. |
| CONFIG SCL | Overrides the SCL pin assignment from the Option Settings. |

Syntaxe

| | |
|---|-----------------|
| SINCLUDE | file |
| SBAUD | = var |
| SCRISTAL | = var |
| SRAMSTART | = constant |
| SDEFAULT XRAM | |
| SLARGE | |
| SLOAD | = [bit] address |
| SNOBREAK | |
| SNOINIT | |
| SNO\$P | |
| SOBJ obj | |
| SRAMSTART | = [bit]address |
| SRAMSIZE | = [bit] size |
| SRAMSTART | = [bit] address |
| SSERIALINPUT | = label |
| SSERIALINPUT2LCD | |
| SSERIALOUTPUT | = label |
| SSIM | |
| var = ABS(var2) | |
| newvar ALIAS oldvar | |
| var = ASC(string) | |
| PRINT BCD(var), LCD BCD(var) | |
| BITWAIT x SET/RESET | |
| BREAK | |
| CALL Test [var1, varn]] | |
| PRINT CHR(var) s = CHR(var) | |
| CLS | |
| DIM symbol AS CONST value | |
| CONFIG TIMERx = COUNTER/TIMER, GATE = INTERNAL/EXTERNAL, MODE = 0/3 | |
| CONFIG LCD = LCDtype | |
| CONFIG LCDBUS = constant | |
| CONFIG BAUD = baudrate | |
| CONFIG IWMRE = pin | |
| CONFIG SDA = pin | |
| CONFIG SCL = pin | |

Instruction

| | |
|----------------------------|---|
| I2CSEND | Sends data to an I2C-device. |
| I2CSTART,I2CSTOP,I2C8BYTE, | |
| I2CW8YTE | |
| IDLE | Put the processor into the idle mode. |
| IF | Allows conditional execution or branching, based on the evaluation of a Boolean expression. |
| INCR | Increments a variable by one. |
| INKEY | Returns the ASCII value of the first character in the serial input buffer. |
| INP() | Returns a byte read from a hardware port or external memory location. |
| INPUTBIN | Read binary values from the serialport. |
| INPUTHEX | Allows input from the keyboard during program execution. |
| INPUT | Allows input from the keyboard during program execution. |
| LCD | Sends constant or variable to LCD display. |
| LDCHX | Sends variable in hexadecimal format to the LCD display. |
| LEFT() | Returns the specified number of leftmost characters in a string. |
| LEN | Returns the length of a string. |
| LOAD | Load specified TIMER with a value for autoreload mode. |
| LOCATE | Moves the LCD cursor to the specified position. |
| LOOKUP | Returns a value from a table. |
| LOOKUPSTR | Returns a string from a table. |
| LOW | Retrieves the least significant byte of a variable. |
| LOWERLINE | Reset the LCD cursor to the lowerline. |
| MAKEBD() | Convert a variable into its BCD value. |
| MAKEINT() | Compact two bytes into a word or integer. |
| MAKEDC() | Convert a BCD byte or Integer/Word variable to its DECIMAL value. |
| MID() | The MID function returns part of a string (a sub string). The MID statement replaces part of a string variable with another string. |
| MOD | Returns the remainder of a division. |
| NEXT | Ends a FOR...NEXT structure. |
| ON Interrupt | Execute subroutine when specified interrupt occurs. |
| ON Value | Branch to one of several specified labels, depending on the value of a variable. |
| OPEN - CLOSE | Opens and closes a device. |
| OUT | Sends a byte to a hardware port or external memory address. |
| PI,P3 | PI and P3 are special function registers that are treated as variables. |
| PEEK() | Returns a byte stored in internal memory. |
| POKE | Write a byte to an internal memory location. |
| POWERDOWN | Put processor into powerdown mode. |

Signification

| | |
|--------------|---|
| I2CSEND | Sends data to an I2C-device. |
| I2CSTART | generates an I2C start condition, I2CSTOP generates an I2C stop condition. |
| I2C8BYTE | receives one byte from an I2C-device, I2CW8YTE sends one byte to an I2C-device. |
| IDLE | Put the processor into the idle mode. |
| IF | Allows conditional execution or branching, based on the evaluation of a Boolean expression. |
| INCR | Increments a variable by one. |
| INKEY | Returns the ASCII value of the first character in the serial input buffer. |
| INP() | Returns a byte read from a hardware port or external memory location. |
| INPUTBIN | Read binary values from the serialport. |
| INPUTHEX | Allows input from the keyboard during program execution. |
| INPUT | Allows input from the keyboard during program execution. |
| LCD | Sends constant or variable to LCD display. |
| LDCHX | Sends variable in hexadecimal format to the LCD display. |
| LEFT() | Returns the specified number of leftmost characters in a string. |
| LEN | Returns the length of a string. |
| LOAD | Load specified TIMER with a value for autoreload mode. |
| LOCATE | Moves the LCD cursor to the specified position. |
| LOOKUP | Returns a value from a table. |
| LOOKUPSTR | Returns a string from a table. |
| LOW | Retrieves the least significant byte of a variable. |
| LOWERLINE | Reset the LCD cursor to the lowerline. |
| MAKEBD() | Convert a variable into its BCD value. |
| MAKEINT() | Compact two bytes into a word or integer. |
| MAKEDC() | Convert a BCD byte or Integer/Word variable to its DECIMAL value. |
| MID() | The MID function returns part of a string (a sub string). The MID statement replaces part of a string variable with another string. |
| MOD | Returns the remainder of a division. |
| NEXT | Ends a FOR...NEXT structure. |
| ON Interrupt | Execute subroutine when specified interrupt occurs. |
| ON Value | Branch to one of several specified labels, depending on the value of a variable. |
| OPEN - CLOSE | Opens and closes a device. |
| OUT | Sends a byte to a hardware port or external memory address. |
| PI and P3 | PI and P3 are special function registers that are treated as variables. |
| PEEK() | Returns a byte stored in internal memory. |
| POKE | Write a byte to an internal memory location. |
| POWERDOWN | Put processor into powerdown mode. |

Syntaxe

| | |
|--|---|
| I2CSEND | slave, var I2CSEND slave, var , bytes |
| I2CSTART, I2CSTOP, I2C8BYTE | var, 8/9, I2CW8YTE val |
| IDLE | |
| IF | expression THEN, [ELSEIF expression THEN], [ELSE], END IF |
| INCR | var |
| var = INKEY | |
| var = INP(address) | |
| INPUTBIN var1 [var2] | |
| INPUTHEX [" prompt"], var [, varn] [NOECH0] | |
| INPUT [" prompt"], var [, varn] [NOECH0] | |
| LCD x | |
| LDCHX | var |
| var = Left(var1, n) | |
| var = LEN(string) | |
| LOAD TIMER, value | |
| LOCATE y, x | |
| var = LOOKUP(value, label) | |
| var = LOOKUPSTR(value, label) | |
| var = LOW(s) | |
| LOWERLINE | |
| var1 = MAKEBD(var2) | |
| varn = MAKEINT(var, n) | |
| var1 = MAKEDC(var2) | |
| var1 = MID(var1, st [, l]) MID(var, st [, l]) = var1 | |
| ret = var1 MOD var2 | |
| NEXT [var] | |
| ON interrupt label [NO\$AVE] | |
| ON var [GOTO] [GOSUB] label [, label2] | |
| OPEN "device" for MOD As #channel, CLOSE #channel | |
| OUT address, value | |
| Px = var, var = Px | |
| Px = PEEK(address) | |
| POKE address, value | |
| POWERDOWN | |

Instruction

CONFIG DEBOUNCE
CONFIG SPI
CONFIG LCDPIN
CONFIG WATCHDOG
COUNTERx

CPEEK()

CURSOR

DATA

DEBOUNCE

DECR

DECLARE SUB

DEFINT, DEFBIT, DEFBYTE, DEFWORD

DEFLCDCHAR

DELAY

DIM

DISABLE

DISPLAY

DO...LOOP

ELSE

ENABLE

END

END IF

ERASE

EXIT

FOR

FOURTHLINE

FUSING

GETRC

GETICS

GOSUB

GOTO

HEX()

HEXVAL()

HIGH

HOME

ICRECEIVE

Signification

Configures the delaytime for the DEBOUNCE statement.

Configures the SPI related statements.

Override the LCD-options to store the settings in your program.

Configures the watchdog timer from the AT89C8252

Set or retrieve the COUNTER0 or COUNTERn variable. For 8052 TIMER2 compatible chips, COUNTER2 can be used too.

Returns a byte stored in code memory.

Set the LCD Cursor State.

Specifies values to be read by subsequent READ statements.

Debounce a port pin connected to a switch.

Decrements a variable by one.

Declares a subroutine.

Declares all variables that are not dimensioned of the DefXXX type.

Define a custom LCD character.

Delay program execution for a short time.

Dimension a variable.

Disable specified interrupt.

Turn LCD display on or off.

Repeat a block of statements until condition is true.

Executed if the IF-THEN expression is false.

Enable specified interrupt.

Terminate program execution.

End an IF...THEN structure.

Erases a variable so memory will be released.

Exit a FOR...NEXT, DO...LOOP...WHILE...WEND or SUB...END SUB.

Execute a block of statements a number of times.

Reset LCD cursor to the fourth line.

Formats a floating-point value.

Retrieves the value of a resistor or a capacitor.

Retrieves a ICS infrared code and subaddress.

Branch to and executes subroutine.

Jump to the specified label.

Returns a string representation of a hexadecimal number.

Convert string representing a hexadecimal number into a numeric variable.

Retrieves the most significant byte of a variable.

Place the cursor at the specified line at location I.

Receives data from an I2C serial device.

Syntaxe

CONFIG DEBOUNCE = time

CONFIG SPI = S0FT, DIN = PIN, DOUT = PIN, CS = PIN, CLK = PIN

CONFIG LCDPIN, DB4 = ..., DB5 = P1.2, DB6 = P1.3, DB7 = P1.4, E = P1.5, RS = P1.6

CONFIG WATCHDOG = time

COUNTERX = var or var = COUNTERX

var = CPEEK(address)

CURSOR ON / OFF BLINK / NOBLINK

DATA var [, varn]

DEBOUNCE Pxy, state, label [, SUB]

DECR var

DECLARE SUB TEST[(var as type)]

DEFBIT b, DEFBYTE c, DEFINT i, DEFWORD x

DEFLCDCHAR char1,r2,r3,r4,r5,r6,r7,r8

DELAY

DIM var AS [XRAM/IRAM] type

DISABLE interrupt

DISPLAY ON / OFF

DO statements LOOP [UNTIL expression]

ELSE

ENABLE interrupt

END

END IF or ENDIF

ERASE var

EXIT [FOR] [DO] [WHILE] [SUB]

FOR var = start TO/DOWNTO end [STEP value]

FOURTHLINE

var = Fusing(source, mask)

var = GETRC(pin)

GETICS(address, command)

GOSUB label

GOTO label

var = Hex(x)

var = HEXVAL(x)

var = HIGH(s)

HOME UPPER / LOWER / THIRD / FOURTH

ICRECEIVE slave, var I2CRECEIVE slave, var I2RW, I2R

Instruction

CONFIG DEBOUNCE
CONFIG SPI
CONFIG LCDPIN
CONFIG WATCHDOG
COUNTERx

CPEEK()

CURSOR

DATA

DEBOUNCE

DECR

DECLARE SUB

DEFINT, DEFBIT, DEFBYTE, DEFWORD

DEFLCDCHAR

DELAY

DIM

DISABLE

DISPLAY

DO...LOOP

ELSE

ENABLE

END

END IF

ERASE

EXIT

FOR

FOURTHLINE

FUSING

GETRC

GETICS

GOSUB

GOTO

HEX()

HEXVAL()

HIGH

HOME

ICRECEIVE

Instruction

PRINT
PRINTBIN
PRINTHEX
PRIORITY
READ
REM
RESET
RESTORE
RETURN
RIGHT()
ROTATE
SELECT

SET

SHIFTCURSOR

SHIFTLCD

SOUND

SPACE()

SPIN

SPIOUT

START

STOP

STOP TIMERx

STR()

STRING()

SUB

SWAP

THIRDLIN

UPPERLINE

VAL()

VARPTR()

WAIT

WAITKEY

WAITMS

WHILE...WEND

Syntaxe

PRINT var ; "constant"

PRINTBIN var [,varn]

PRINTHEX var

PRIORITY.SET / RESET interrupt

READ var

REM or *

RESET bit, RESET varx

RESTORE label

RETURN

var = RIGHT(var[,st])

ROTATE var, LEFT/RIGHT [, shifts]

SELECT CASE var, CASE test1 : statements, [CASE test2 : statements],

CASE ELSE : statements, END SELECT

SET bit, SET varx

SHIFTCURSOR LEFT / RIGHT

SHIFTLCD pin, pdclock, var, option, SHIFTOUT pin, pdclock, var, option

SOUND pin, duration, frequency

var = SPACE(x)

SPIN var, bytes

SPIOUT var, bytes

START timer

STOP

STOP timer

var = Str(x)

var = STRING(n, n)

SUB Name [(var)]

SWAP var1, var2

THIRDLIN

UPPERLINE

var = Val(s)

var = VARPTR(var2)

WAIT seconds

var = WAITKEY

WAITMS ms

WHILE condition statements WEND

(on parle d'adresses-miroir). Cette caractéristique interdit malheureusement l'utilisation, simultanément, du même domaine par une EEPROM.

Les accès en écriture ne sont autorisés que pour A0=0, sachant qu'à A0=1, l'affichage met ses données sur le bus. Comme on a procédé à l'interconnexion en fonction logique NON-ET (NAND) des lignes \overline{RD} et \overline{WR} (ce qui se traduit, en raison de l'inversion, par une fonction WR OU RD), une instruction d'écriture vers une adresse de lecture se traduirait par un conflit de bus. On dispose ainsi, pour la commande, des adresses suivantes :

8000_{HEX} : Écriture d'instruction
 8001_{HEX} : Lecture d'instruction
 8002_{HEX} : Écriture de données
 8003_{HEX} : Lecture de données

On a un miroir de ce domaine jusqu'à FFFF_{HEX}. On pourrait partant fort bien utiliser également, par exemple, le domaine allant de F000_{HEX} à F003_{HEX}. L'affichage connaît un nombre important d'instructions. Il est fait une différence au niveau du type d'instruction, qui possèdent un nombre caractéristique de bits mis à zéro dans les positions de poids fort.

| Fonction | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------------------|--|--------------------|-----------|----|----|---------|----|---|
| Effacer l'affichage | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Curseur en position « home » | 0 | 0 | 0 | 0 | 0 | 0 | 1 | x |
| Décaler | 0 | 0 | 0 | 0 | 0 | 1 | ID | S |
| | (ID= 1/0: droite/gauche, S= 1/0: sans/avec texte) | | | | | | | |
| Affichage, curseur | 0 | 0 | 0 | 0 | 1 | D | C | B |
| | (D,C,B= 1/0: Affichage, curseur, clignotement On/Off) | | | | | | | |
| Décaler | 0 | 0 | 0 | 1 | SC | RL | x | x |
| | (SC= 1/0: curseur texte vers G/D = 1/0: gauche/droite) | | | | | | | |
| Initialisation | 0 | 0 | 1 | DL | N | x | x | x |
| | (DL= 1/0: bus 8/4 bits, N= 1/0: une ligne ou les deux) | | | | | | | |
| Générateur de caractères | 0 | 1 | Caractère | | | Colonne | | |
| Position du curseur | 1 | Adresse de mémoire | | | | | | |

Il est possible de procéder à relecture du registre d'instruction de l'affichage. Il fournit alors comme information l'état de son indicateur d'occupation, BF (*Busy Flag*) et la position du curseur à cet instant. Il faut impérativement, avant tout

accès pour une opération d'écriture, vérifier l'état de l'indicateur BF. Ce n'est qu'à condition que BF soit au niveau bas que l'on pourra écrire une instruction ou un octet de donnée dans l'affichage, sachant que l'on court sinon le risque de destruction du circuit du contrôleur de l'affichage.

| | | |
|----------------|----|--|
| Système prêt ? | BF | Adresse mémoire (=position du curseur) |
|----------------|----|--|

Il faut commencer, après mise sous tension, par écrire quelques octets d'initialisation dans le registre d'instruction. À titre d'exemple :

| | |
|---|----------------------|
| Initialisation avec bus de données à 8 bits et 2 lignes | 0011 1000 = 38h = 56 |
| Activer affichage, éteindre curseur | 0000 1100 = 0Ch = 12 |
| Effacer l'affichage | 0000 0001 = 01h = 1 |

L'affichage comporte un indicateur de donnée interne auquel sont attribuées les différentes positions de caractères. Dans le cas d'un affichage à 2 lignes de 16 caractères cela se traduit par la cartographie suivante :

Ligne 1 : Adresse 00_{HEX} à 0F_{HEX}
 Ligne 2 : Adresse 40_{HEX} à 4F_{HEX}

d'une position lors de l'écriture d'un caractère, mais il est possible, par la saisie d'une adresse donnée, de le mettre à une position choisie.

Le petit programme écrit en BASIC-52 listé ci-après constitue un premier exemple de pilotage rudimentaire. Le programme ne procède pas à une interrogation de l'indicateur BF. Cela ne pose pas de problème tant que le

pilotage se fait depuis un programme purement BASIC en raison de la relative lenteur de ce langage.

Pilote de LCD pour BASIC-52

En BASIC-52, l'affichage de caractères individuels n'est guère confortable. L'interpréteur prévoit cependant une extension prenant la forme d'un pilote (*driver*) propre permettant de contourner les instructions PRINT. L'utilisateur devra à cet effet créer une routine de sortie chargée de lire les caractères depuis le registre R5 (banc 0) et de les transmettre à la sortie, l'affichage LCD dans le cas présent. Il est possible, par l'instruction de sortie utilisateur (*User Output*) UO 1, de détourner la sortie vers un pilote de son cru. Lors de l'exécution d'une instruction PRINT, le système saute, pour chacun des caractères, à l'adresse 4030_{HEX}, adresse à laquelle doit se trouver une instruction de saut vers le programme-pilote de sortie.

L'instruction PRINT@ offre une autre possibilité d'utiliser un pilote-maison. Dans ce cas-là on saute à l'adresse 403C_{HEX}, adresse à laquelle on doit trouver, comme dans le cas précédent, une instruction de saut vers la routine de sortie.

Le pilote-maison ne sera activé que lorsque le bit 39 de la RAM interne (bit 7 à l'adresse 24_{HEX}) se trouve positionné (mis à « 1 »). Ceci pourra se faire, par exemple, après la routine d'initialisation de l'affichage qui est elle indispensable de toutes façons.

L'extension décrite plus loin intègre le pilote de LCD ainsi que les autres extensions d'instruction décrite ci-après. BASIC-52 met déjà à disposition de nombreuses aides pour pouvoir se doter d'extensions externes. Les sous-programmes en langage machine doivent comporter les lignes suivantes : Tous les registres, exception faite des 8 registres du

banc 3 (18_{HEX} à 1F_{HEX}) et l'accu et le pointeur de données doivent rester inchangés. L'utilisateur a le droit d'utiliser à ses propres fins la totalité du banc 3.

Le pilote de LCD sauvegarde l'accu et le pointeur de données sur la pile de manière à les restaurer avant le saut de retour (en fin de routine). Le registre 4 du banc 3 contient la position actuelle du curseur. Lors d'un caractère retour chariot (CR = *Carriage Return*) le curseur est positionné au début de la ligne en cours. Dans le cas d'un affichage à 2 lignes l'utilisateur pourra faire en sorte que le curseur vienne se positionner au début de la seconde ligne (position 64 = 40_{HEX}). On s'assure, lors de chaque accès de donnée sur l'affichage, par le biais de la routine Busy, que le système est paré.

Il faut, avant la première mise en oeuvre du pilote, préparer l'affichage par un appel à la routine d'initialisation (CALL 4250h). BASIC-52 permet à l'utilisateur de prévoir de telles extensions de fonctions par le biais d'instructions d'appel spéciales, CALL 00, CALL 01, CALL 03 et ainsi de suite jusqu'à CALL 127, instructions qui sont dérivées automatiquement vers les adresses 4100_{HEX}, 4102_{HEX}, 4104_{HEX} jusqu'à 41FE_{HEX}. Les instructions de saut correspondantes envoyant aux adresses effectives permettent un appel simplifié de l'initialisation du LCD (CALL 00) et de la routine du curseur. Le curseur doit être positionné par le biais d'un paramètre supplémentaire :

```
CALL 427FH 0
ou, en raccourci
CALL 01 0
```

La routine du curseur traite une expression placée derrière l'adresse de CALL, expression qui pourra même comporter une opération mathématique telle que CALL 01 (64+4*N).

Le pilote d'affichage doit se trouver en RAM à partir de l'adresse 4030_{HEX}. Il n'y a pas de problème à ce niveau sachant que la totalité de la RAM de la mono-carte à 89S8252 Flash sert également de mémoire de programme. Il faudra cependant, par l'instruction MTOP=8191, limiter quelque peu le domaine de la RAM BASIC pour éviter tout conflit avec la gestion de mémoire propre au système. Le programme BASIC **LCD2.BAS** met le pilote LCD à poste en RAM.

Un RUN se traduit par le chargement du pilote en RAM et son initialisation. On pourra ensuite supprimer le programme de charge-

Listage 1. Commande de l'affichage en Basic-52

```
1 REM LCD data output (LCD.BAS)
2 REM data 8002h, commands 8000h
10 STRING 80,16
20 GOSUB 1000
30 $(0)="Elektor 89S8252 "
40 GOSUB 1300
50 $(1)="LCD 2 * 16 "
60 GOSUB 1400
70 END
1000 REM LCD-Reset
1010 XBY(8000H)=56
1020 XBY(8000H)=12
1030 XBY(8000H)=1
1040 RETURN
1100 REM cursor position 0
1110 XBY(8000H)=128
1120 RETURN
1200 REM output
1210 XBY(8002H)=A
1220 RETURN
1300 REM line 1
1310 XBY(8000H)=128
1320 FOR I=1 TO 16 : XBY(8002H)=ASC$(0),I : NEXT
1330 RETURN
1400 REM line 2
1410 XBY(8000H)=192
1420 FOR I=1 TO 16 : XBY(8002H)=ASC$(1),I : NEXT
1430 RETURN
```

Listage 2. Le pilote d'affichage

```
;Basic 52 extension LCD driver (Basic52_LCD.asm)
4030 .org 4030H
4030 02 42 00 ljmp LCD ;UO 1
403C .org 403CH
403C 02 42 00 ljmp LCD ;PRINT@
4100 .org 4100H
4100 41 50 ajmp LCDInit ;CALL 0
4102 41 7F ajmp CURSOR ;CALL 1

4200 .org 4200H
4200 C0 83 LCD push DPH
4202 C0 82 push DPL
4204 C0 E0 push ACC
4206
4206 51 2E acall Busy
4208 E5 1C mov A,28 ;Cusor position
420A 44 80 orl A,#128
420C 90 80 00 mov DPTR,#8000H
420F F0 movx @DPTR,A ;set Cursor
4210 51 2E acall Busy
4212 E5 05 mov A,05 ;get char
4214 B4 0D 02 cjne A,#0DH,J1 ;= CR?
4217 80 1D sjmp CR
4219 B4 0A 02 cjne A,#0AH,J2 ;= LF?
421C 80 09 sjmp End
421E 90 80 02 J2 mov DPTR,#8002H
4221 F0 movx @DPTR,A ;output char
4222 E5 1C mov A,28 ;bank 3, R4, cursor
pos.
4224 04 inc A
4225 F5 1C mov 28,A ;inc cursor
4227 D0 E0 End pop ACC
4229 D0 82 pop DPL
422B D0 83 pop DPH
422D 22 ret
422E
422E 90 80 01 Busy mov DPTR,#8001H
4231 E0 movx A,@DPTR
4232 20 E7 F9 jb ACC.7,Busy
4235 22 ret
4236
4236 51 2E CR acall Busy
```



```

4238 74 20          mov     a,#32
423A 90 80 02       mov     DPTR,#8002h
423D F0            movx    @DPTR,a
423E E5 1C         mov     A,28
4240 04           inc     A
4241 F5 1C         mov     28,a
4243 54 3F         anl     A,#63
4245 B4 28 EE      cjne    A,#40,CR
4248 E5 1C         J3      mov     a,28
424A 54 C0         anl     a,#192
424C F5 1C         mov     28,a
424E 80 D7         sjmp    End
4250
4250 C0 83         LCDInit push    DPH
4252 C0 82         push    DPL
4254 C0 E0         push    ACC
4256
4256 51 2E         acall    Busy
4258 90 80 00      mov     DPTR,#8000H
425B 74 38         mov     A,#38H
425D F0            movx    @DPTR,A
425E 51 2E         acall    Busy
4260 90 80 00      mov     DPTR,#8000H
4263 74 06         mov     A,#06H
4265 F0            movx    @DPTR,A
4266 51 2E         acall    Busy
4268 90 80 00      mov     DPTR,#8000H
426B 74 0C         mov     A,#0CH
426D F0            movx    @DPTR,A
426E 51 2E         acall    Busy
4270 90 80 00      mov     DPTR,#8000H
4273 74 01         mov     A,#01H
4275 F0            movx    @DPTR,A
4276 D2 27         setb    027H      ;PRINT@ enable
4278
4278 D0 E0         pop     ACC
427A D0 82         pop     DPL
427C D0 83         pop     DPH
427E 22           ret
427F
427F C0 E0         CURSOR push    ACC
4281 74 39         Mov     A,#57
4283 12 00 30      lcall    030h
4286 74 01         Mov     A,#1
4288 12 00 30      lcall    030h
428B E9           mov     A,R1
428C F5 1C         mov     28,A
428E D0 E0         pop     ACC
4290 22           ret

```

Listage 3. Programme de chargement du pilote d'affichage

```

3999 REM LCD driver          (LCD2.bas)
4000 MTOP=8191
4029 REM output jumps
4030 DATA 002H,042H,000H,000H,000H,000H,000H,000H
4038 DATA 000H,000H,000H,000H,000H,000H,000H,000H
4039 FOR N=0 TO 15 : READ D : XBY(04030H+N)=D : NEXT N
4099 REM CALL jumps
4100 DATA 041H,050H,041H,07FH,061H,000H,000H,000H
4101 FOR N=0 TO 7 : READ D : XBY(04100H+N)=D : NEXT N
4199 REM LCD
4200 DATA 0C0H,083H,0C0H,082H,0C0H,0E0H,051H,02EH
4208 DATA 0E5H,01CH,044H,080H,090H,080H,000H,0F0H
4210 DATA 051H,02EH,0E5H,005H,0B4H,00DH,002H,080H
4218 DATA 01DH,0B4H,00AH,002H,080H,009H,090H,080H
4220 DATA 002H,0F0H,0E5H,01CH,004H,0F5H,01CH,0D0H4228 DATA
0E0H,0D0H,082H,0D0H,083H,022H,090H,080H
4230 DATA 001H,0E0H,020H,0E7H,0F9H,022H,051H,02EH
4238 DATA 074H,020H,090H,080H,002H,0F0H,0E5H,01CH
4240 DATA 004H,0F5H,01CH,054H,03FH,0B4H,028H,0EEH
4248 DATA 0E5H,01CH,054H,0C0H,0F5H,01CH,080H,0D7H
4250 DATA 0C0H,083H,0C0H,082H,0C0H,0E0H,051H,02EH

```

ment voire y ajouter le programme utilisateur. La mise en oeuvre du pilote de LCD n'a rien de difficile. Dans le cas normal on a détournement de la sortie par l'instruction UO 1. Dès lors, toutes les instructions PRINT on trait à l'affichage. Il est en outre possible, avec CALL 01, de modifier la position du curseur. Le programme **LCDDEMO.BAS** donné en exemple visualise l'état du port P1 à l'affichage. On a ensuite affichage d'un texte. À chaque fin de ligne l'instruction PRINT fait repasser le curseur en début de la ligne en cours.

Le programme visualise, en temps réel, l'état du port 1 à l'affichage :

PORT P1 = 255

Le pilote LCD, une fois chargé, reste en RAM, même si, après un NEW, on charge un nouveau programme. On pourra continuer d'utiliser l'affichage. Le transfert d'opérations d'impression (Print) vers l'affichage en particulier constitue une simplification notable.

Pilotage de LCD en C

Un pilotage par le biais de Reads51 requiert des fonctions en assembleur vu qu'il faut accéder à de la mémoire externe.

Heureusement que du code assembleur en ligne (*Assembler-Inline-Code*) ne pose pas de problème. Le **listage 5** met à disposition les fonctions décisives LCDinit, LCDcursor et LCDwrite. Le code assembleur tout ce qu'il y a de plus orinaire, prend tout simplement place entre les repères #asm et #endasm. Il est également possible, de ce fait, d'utiliser le compilateur pour des exercices d'assembleur pur. Il faudra se rappeler que, par rapport à TASM, l'écriture des nombres hexadécimaux se fait différemment (0x8000 au lieu de 8000h).

La construction de fonctions en assembleur est exactement la même que pour les fonctions en C; il est possible, depuis des fonctions en C, de les appeler sous leur dénomination propre. Le bloc assembleur ne se termine pas par une instruction de saut ret sachant que le compilateur C se charge de cette opération lors de la conversion de la fonction.

Dans les cas requérant le transfert d'une variable vers la fonction, il faut s'attendre à un peu plus de complexité sachant qu'il faut la transférer vers l'accumulateur. Reads51 connaît les variables systèmes spéciales, BPL et BPH, pour la gestion d'une pile logicielle interne, qui permet également un transfert de paramètres. Un paramètre transmis à une

Listage 4. Test du pilote de LCD

```
100 REM driver use (LCDDEMO.BAS)
110 UO 1 : REM User Output enable
120 CALL 01H 0 : REM Cursor to
    line 1
130 PRINT " PORT P1 = "
140 CALL 01H 64 : REM Cursor to
    line 2
150 PRINT PORT1
160 GOTO 150
```

```
4258 DATA 090H,080H,000H,074H,038H,0F0H,051H,02EH
4260 DATA 090H,080H,000H,074H,006H,0F0H,051H,02EH
4268 DATA 090H,080H,000H,074H,00CH,0F0H,051H,02EH
4270 DATA 090H,080H,000H,074H,001H,0F0H,0D2H,027H
4278 DATA 0D0H,0E0H,0D0H,082H,0D0H,083H,022H,0C0H
4280 DATA 0E0H,074H,039H,012H,000H,030H,074H,001H
4288 DATA 012H,000H,030H,0E9H,0F5H,01CH,0D0H,0E0H
4290 DATA 022H
4291 FOR N=0 TO 144 : READ D : XBY(04200H+N)=D : NEXT N
4292 CALL 00
```

Listage 5. Le module LCD.C

```
// ----- READS51 generated header -----
// module : C:\Rigel\Reads51\Work\LCD\LCD.c
// created : 10:46:46, Tuesday, March 05, 2002
// -----
```

```
#define data_write 0x8002;
#define data_read 0x8003;
#define cmd_write 0x8000;
#define cmd_read 0x8001;
```

```
#include <Sio51.h>
```

```
void LCDinit(void){
//
//
#asm
    lcall Busy
    mov DPTR,#cmd_write
    mov A,#56
    movx @DPTR,A
    lcall Busy
    mov DPTR,#cmd_write
    mov A,#12
    movx @DPTR,A
    lcall Busy
    mov DPTR,#cmd_write
    mov A,#0x0C
    ;movx @DPTR,A
    lcall Busy
    mov DPTR,#cmd_write
    mov A,#1
    movx @DPTR,A
#endasm
}
```

```
#asm
    Busy:
    mov dptr, #cmd_read
    movx A, @dptr
    jb ACC.7,Busy
    ret
#endasm
```

```
void LCDwrite(unsigned char dat){
//
//
#asm
    lcall Busy

    mov a, BPL ;load dat to a
    add a, #0xFA
    mov dpl, a
    mov a, BPH
    addc a, #0xFF
    mov dph, a
    movx a, @dptr

    mov dptr, #data_write
```

```
    movx @dptr, a
```

```
#endasm
}
```

```
void LCDcursor (unsigned char pos)
```

```
{
#asm
    lcall Busy

    mov a, BPL ;load pos to a
    add a, #0xFA
    mov dpl, a
    mov a, BPH
    addc a, #0xFF
    mov dph, a
    movx a, @dptr

    mov dptr, #cmd_write
    orl a, #0x80
    movx @dptr, a
#endasm
}
```

```
main()
{
    char dat;
    unsigned char pos;
    unsigned char n;

    InitSerialPort0(DEF_SIO_MODE);
    LCDinit();
    while (1)
    {
        pos=0;
        LCDcursor(pos);
        for(n=0; n<16; n++)
            LCDwrite (32);
        LCDcursor(pos);
        while (pos < 16)
        {
            dat=getc();
            if (dat == 13) pos=16;
            else LCDwrite(dat);
            pos=pos+1;
        }
        pos=64;
        LCDcursor(pos);
        for(n=0; n<16; n++)
            LCDwrite (32);
        LCDcursor(pos);
        while (pos < 80)
        {
            dat=getc();
            if (dat == 13) pos=80;
            else LCDwrite(dat);
            pos=pos+1;
        }
    }
    while (1);
}
```

fonction se trouve à une adresse dont la position se situe 6 points en-dessous de l'adresse de pile au moment de l'appel de la fonction. Il est possible, depuis cet endroit, de transférer (charger) dans l'accumulateur un octet y ayant été transféré. Contrairement à ce qui est le cas sous BASIC-52, le langage C est tellement rapide qu'il faut impérativement se demander si l'affichage LCD est prêt à recevoir une nouvelle instruction. C'est là la fonction du sous-programme écrit en assembleur Busy. IL y est fait appel avant chaque nouvelle instruction et avant toute opération d'écriture.

Le programme C principal crée un affichage sur lequel il est possible

d'écrire par l'intermédiaire de l'interface série. Un terminal quelconque sert au transfert du texte qui apparaît ensuite sur l'affichage. On travaille avec 2 lignes qui sont écrites successivement. À chaque fois que le curseur arrive à la fin d'une ligne, l'autre ligne est effacée complètement, pour recevoir la suite du texte à partir de la position la plus à gauche de la ligne.

(010208-7)

En guise de conclusion

Ce dernier article termine notre cours « Microcontrôleurs ». Nous y avons décrit les éléments les plus importants tant matériels que logiciels. Tout au long du cours nous

avons eu des réactions de lecteurs dont certaines se sont même traduites par une réorientation du contenu prévu à l'origine. Nous les en remercions. Il n'est pas possible, bien entendu, d'aborder tous les aspects du thème « Microcontrôleurs ». Le nombre d'applications et d'idées restant ouvertes ne cesse de croître.

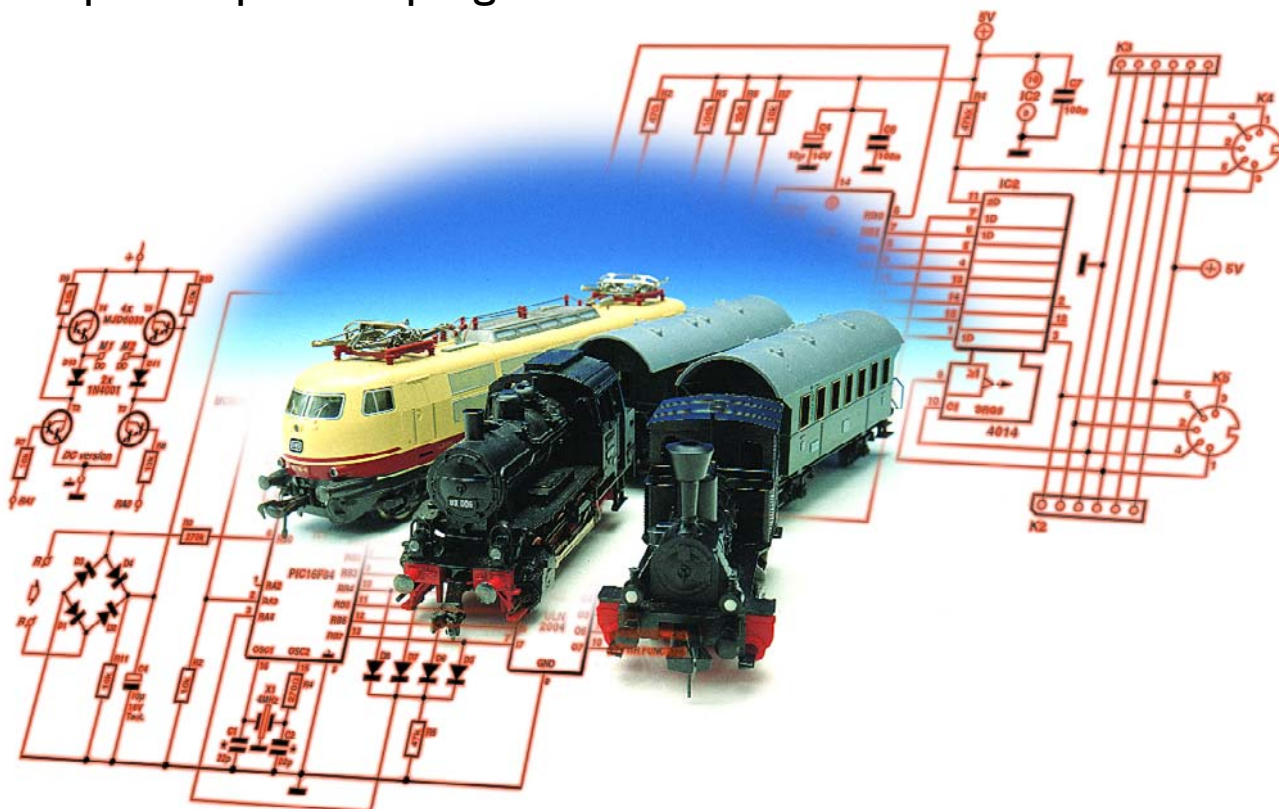
Il est dans nos intentions de vous proposer, à intervalle plus ou moins régulier, l'une ou l'autre application reposant sur la carte 89S8252 Flash. Cette mono-carte est devenue, en quelques mois, une plate-forme permettant de réaliser des développements très divers. Si vous avez effectué une application ou envisagez dans les prochaines semaines de réaliser un montage basé sur cette carte, vous pourrez participer à un concours Elektor dont nous donnerons le règlement dans le numéro double de Juillet/Août.

EDiTS Pro Upgrade du ESDL

Avec upgrade du répondeur

Henk Prince

Après annonce d'une nouvelle version du logiciel tournant sous PC et du contrôleur du système de pilotage de réseaux ferroviaires EDiTS Pro (*Elek-tor Digital Train System*) il fallait bien évidemment s'attendre à voir arriver une option de réactualisation (*upgrade*) de l'ESDL (le super-décodeur de locomotive). Cela fait près de 3 ans (cf. [1] de la bibliographie) que nous avons présenté le super-décodeur de locomotive dans l'un de nos numéros. La nouvelle version décrite ici pourra, dès à présent, remplacer l'ancienne sachant qu'elle offre un certain nombre de possibilités nouvelles telles qu'une option de programmation du tableau de vitesse.



Sur la nouvelle version du **Edits Pro Super-Décodeur** de Locomotive (ESDL pour les intimes), l'auteur ne s'est pas contenté d'ajouter de nouvelles fonctionnalités de support pour la nouvelle version du logiciel pour PC. Il a remis sur le métier un certain nombre des aspects de base. Il devient possible maintenant de tenir compte du type de moteur équipant la locomotive. Comme vous n'êtes pas sans le savoir, il existe des tracteurs à moteur à courant alternatif et à courant continu.

Les caractéristiques d'un moteur continu ne sont pas celles d'un moteur alternatif, de sorte que qu'une locomotive à moteur continu, ira, à une position faible du régleur, plus vite que ne le ferait son homologue à courant alternatif. Le nombre d'amateurs de moteurs à courant continu ne cessant de croître, nous avons fait en sorte qu'il soit possible d'adapter le tableau de vitesse que comporte le décodeur aux caractéristiques du moteur.

Le tableau de vitesse

La vitesse du moteur se laisse régler en 16 pas. Chaque position du régleur correspond à un emplacement du tableau se trouvant dans l'EEPROM du décodeur. Ce tableau comporte des valeurs qui servent à déterminer la vitesse de déplacement de la locomotive. On a, en principe, la valeur 0 dans l'emplacement 0 (position la plus basse du régleur) et à l'emplacement 15 la valeur la plus élevée, 0FF_{HEX}.

À l'origine, le décodeur a été conçu

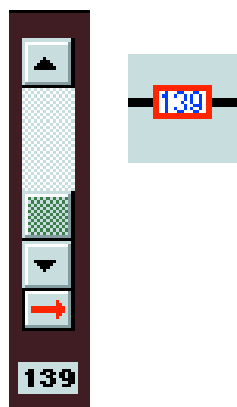


Figure 2. Régleur de vitesse pour une locomotive et un bouton de répondeur se trouvant à l'adresse 139.

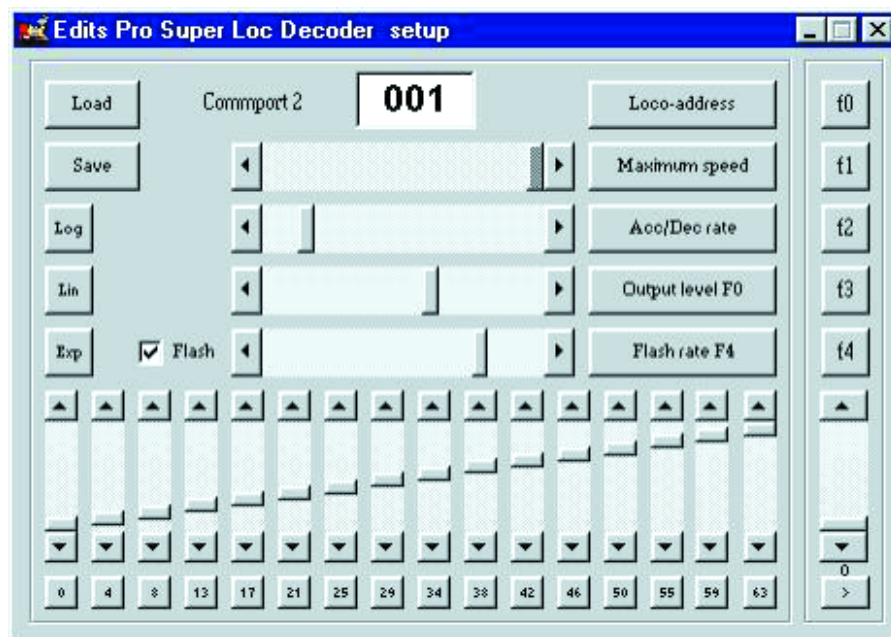


Figure 1. La fenêtre du programme permettant la programmation de cette nouvelle version super-décodeur de locomotive.

pour des moteurs à courant alternatif de Märklin et le tableau d'origine du composant a été paramétré de manière à ce que lorsque le régleur se trouve en position basse la tension appliquée au moteur soit dès le départ un peu plus importante.

Les locomotives de marques utilisant un moteur continu comportent dans la plupart des cas un aimant permanent et d'une combinaison d'engrenages donnant un rapport de conversion plus élevé. Ceci explique que leur couple de démarrage soit meilleur et qu'elles se mettent, lorsque le régleur se trouve à une position basse, à rouler trop vite (pour la position de régleur concernée). Il est possible de compenser partiellement cette dérive en abaissant la vitesse maximale, cette approche étant cependant loin d'être idéale. Ceci explique notre décision d'une modification du progiciel (*firmware*) qui permette de changer le contenu du tableau.

Nous avons tiré parti du fait, qu'en vue de programmer les paramètres du décodeur, que cela était possible à l'aide de moyens simples. L'inconvénient de cette approche est que l'on ne peut utiliser que les « trits » du format Motorola (sur lequel le protocole est basé). Si l'on sait que l'on utilise 2 trits par instruction, on en déduit que le nombre maximal d'instructions

possible est de 3^2 soit 9. 7 d'entre elles sont déjà utilisées pour les paramètres standard (**tableau 1**). Il nous en reste partant 2 de disponibles pour la programmation du tableau de vitesse. Vu que nous utilisons le format Motorola standard cette opération pourra se faire à l'aide du programmeur manuel décrit dans l'article cité en référence [1].

Nous sommes parfaitement conscients du fait que la programmation de la totalité du tableau est une opération de longue haleine. Pour cette raison, nous avons créé un petit programme auxiliaire disponible tant sur le site de EDITS-Pro (cf. la référence [4] en fin d'article) que sur le nôtre (www.elektor.fr, domaine de téléchargement, juin 2002, nr 020094-11).

La programmation

Nous allons voir d'un peu plus près la procédure de programmation du tableau (mais également du reste des paramètres) pour donner aux amateurs la possibilité de programmer le décodeur par le biais d'un programmeur qu'ils auraient en leur possession. Comme, nous supposons que vous le savez, l'information véhiculée par les rails prend la forme d'un train de trits successif, de 4 trits d'adresse de locomotive suivis par 1 trit pour l'éclairage avant et de 4 bits constituant les données de vitesse.

Nous avons donné, dans les différents articles de 1999 consacrés au sujet, les détails du protocole Motorola. Notons que vous pouvez également les trouver dans l'ouvrage de che-

Tableau 1. Programmation des paramètres

9 8 7 6 5 4 3 2 1 Trit

| | |
|---------|---|
| LLLL 0 | programmation de l'adresse de locomotive (LLLL = 1 à 255) |
| MMM 0 1 | programmation de la vitesse maximale (MMM = 0 à 15) |
| AAA 0 X | programmation de l'inertie d'accélération et de freinage (AAA = 0 à 15) |
| LLLL 0 | programmation de l'adresse de locomotive |
| SSS 1 1 | programmation de l'intensité lumineuse de la fonction 0 (SSS = 0...15) |
| KKK 1 X | programmation de la fréquence de clignotement de la fonction 4 (KKK = 0...26) |
| LLLL 0 | programmation de l'adresse de locomotive |
| CCC X 1 | nouveau : programmation de l'adresse du tableau (CCC = 0 à 15) |
| EEE X X | nouveau : programmation de la valeur de vitesse (EEE = 0 à 255) |
| Où : | trit 0 = 00 binaire, trit X = 01 binaire, trit 1 = 11 binaire; trit 1 à 4 = adresse de locomotive 79 |

vet de nombreux amateurs d'EDiTS, dont les coordonnées sont donnée en référence [2].

Le paramétrage du ESDL s'est vu réserver une adresse fixe, 79, adresse qui occupe bien évidemment 4 trits. Les 5 trits restants servent à l'envoi de l'information de paramétrage avec, par exemple, l'adresse de locomotive, la vitesse maximale, l'inertie de démarrage et de freinage, etc. Le décodeur doit avoir reçu chaque instruction correctement au minimum 10 fois avant qu'elle ne soit acceptée comme valide. Dans la pratique cela signifie que si l'on place une instruction pendant 2 secondes sur les rails les données se

trouvent bien écrites dans la mémoire.

Le **tableau 1** récapitule toutes les possibilités de paramétrage.

La procédure de programmation du tableau n'a rien de bien sorcier : on commence par programmer l'adresse de l'emplacement où nous désirons modifier une valeur. Ensuite, nous programmons la valeur requise au dit emplacement. Il est possible de cette manière de paramétrer un à un les différents emplacements.

Le programme

L'auteur a bien évidemment fait de son mieux pour se simplifier la vie, raison pour laquelle il a écrit un programme permettant de paramétrer le ESDL. Ce logiciel, qui ne peut malheureusement être utilisé qu'en combinaison avec le contrôleur d'EDiTS Pro, simplifiera la vie à nombre d'amateurs de ce modèle de réseau ferroviaire. Les possesseurs d'un autre type de contrôleur devront écrire leur propre programme.

Lors du démarrage nous voyons apparaître une fenêtre qui visualise l'ensemble des éléments. Certains de boutons sont familiers puisque ce sont ceux de la fenêtre de paramétrage de EDiTS Pro, à savoir adresse de locomotive, vitesse maximale, inertie au démarrage et au freinage, intensité de l'éclairage et fréquence de clignotement. Sur la gauche de ces boutons nous trouvons les régleurs servant à la définition de ces éléments.

La série de régleurs servant à la définition des valeurs du tableau située

juste en-dessous saute immédiatement aux yeux. Chaque régleur correspond à un emplacement du tableau et à une position du régleur de vitesse. Il est également possible de lire la position du régleur sur le bouton numérique placé juste en-dessous. Dès que l'on actionne le bouton (par un souris-gauche) on aura programmation de la valeur du régleur correspondant dans la mémoire du décodeur. Il est possible, de cette manière, de passer en revue successivement tous les emplacements.

Il a été prévu, en raison de l'importance de la quantité d'informations accumulées pour chaque locomotive et de l'impossibilité, à posteriori, de relire les données envoyées vers le contrôleur, une option de sauvegarde (Save) des données vers le disque dur ainsi d'ailleurs qu'une possibilité de leur chargement (Load). Les paramètres sont sauvegardés dans un petit fichier dont la dénomination est celle du numéro apparaissant dans la fenêtre du champ de saisie supérieur doté de l'extension .stp. Tout à gauche, en-dessous des boutons Save et Load, nous découvrons 3 boutons pour un certain nombre de valeurs de tableau standard : logarithmique, linéaire et exponentiel. Elles pourront servir de base de départ pour la création d'un tableau répondant à vos souhaits spécifiques.

Le programme utilise le même fichier (com.inf) pour le paramétrage du port COM qu'EDiTS Pro et démarrera partant avec le même port qu'EDiTS Pro, à condition que ce programme de paramétrage se trouve dans le même répertoire.

Il est possible, si on le désire, d'opter pour un port COM différent par un double clic sur le texte. Le logiciel n'existe qu'en anglais.

Attention : lors de la programmation du décodeur, toutes les locomotives se trouvant sur les rails se voient attribuer les nouveaux paramètres. La programmation ne tient partant pas compte de l'adresse de locomotive visualisée à l'écran.

On trouve, pour terminer, sur la droite, un certain nombre d'organes de commande permettant de tester les paramètres tout juste définis. C'est la locomotive dont l'adresse est visualisée à cet instant dans le

Tableau 2. Adresses dans le répondeur.

| Adresse de loco | Bouton de répondeur | bit 7 |
|-----------------|---------------------|-------|
| sans IR | vierge (00) | |
| 0 (*1) | | |
| 1 | 1 | 1 |
| ... | ... | ... |
| 78 | 78 | 1 |
| 79 (*1) | interdit | - |
| 80 | 80 | 1 |
| 81 | 81 | 1 |
| 82 | 82 | 1 |
| 83 | 83 | 1 |
| ... | ... | ... |
| 127 | 127 | 1 |
| 128 (*2) | interdit | 0 |
| 129 | 129 | 0 |
| ... | ... | ... |
| 254 | 254 | 0 |
| 255 | 255 | 0 |

(*1 Cette adresse est bloquée dans le ESDL

(*2 La locomotive roule bien à cette adresse, mais le bouton de répondeur n'est pas activé

champ de saisie de l'adresse de locomotive qui réagira à ces impulsions. Il est possible ainsi de commander la vitesse, la direction et les 5 fonctions. Autres adaptations

Avec la dernière version du logiciel de pilotage d'EDiTS Pro, le domaine des adresses de locomotive a augmenté très sensiblement pour aller jusqu'à 255. Ceci implique comme conséquence qu'il faut que tant le décodeur de locomotive que le répondeur d'adresse soient prévus pour cela. Bien que le ESDL ait été prévu à l'origine pour une plage d'adresses allant de 1 à 81, il n'en est pas moins possible, avec l'ancienne version, d'utiliser des adresses supérieures à 81. Puisque nous étions en plein processus de réactualisation (*update*) nous avons fait en sorte que le nouveau décodeur soit utilisable sur la totalité du domaine.

Il semblerait en outre qu'un certain nombre d'utilisateurs aient réussi à programmer une locomotive à l'adresse 79. Le résultat de cette opération est une locomotive qui circule mais qu'il est devenu impossible de paramétrer à une autre adresse. Ce nouveau progiciel bloque cette adresse « maudite ».

Répondeur

Il faut bien entendu que répondeur d'adresse de locomotive [2] soit en mesure de reconnaître également

les adresses les plus élevées. Dans la version d'origine, on a mis, dès la réception de l'information infrarouge (IR), des données décodées sur le bus. Ce n'est qu'après fermeture du rail de contact que le bit 7 passe au niveau haut et qu'EDiTS Pro procédera à la lecture des données. Cela signifie que 129 (81_{HEX}) correspond à l'adresse de locomotive 1, 130 (82_{HEX}) à l'adresse 2 et ainsi de suite. Avec la nouvelle version, les choses changent. Le répondeur ne mettra les données sur le bus qu'après fermeture du rail de contact. Nous avons, de manière à garder une compatibilité descendante avec la version précédente, laissé le domaine des adresses faibles tel qu'il était mais en l'augmentant jusqu'à 127 (0FF_{HEX}) domaine auquel nous avons ajouté le reste (adresses de locomotive 128 à 255, 0_{HEX} à 07_{HEX}). Le résultat de cette opération est qu'EDiTS Pro voit l'adresse de locomotive 128 (80_{HEX}) comme étant l'adresse 00 et partant comme une locomotive ne disposant pas de dispositif de réponse (répondeur) IR. Le **tableau 2** qui indique également les exceptions devrait éclairer « votre lanterne ».

Signalons, en guise de conclusion, qu'il est possible d'utiliser conjointement le ESDL et le répondeur, et cela même avec l'ancienne version d'EDiTS Pro. Sachez cependant que

Littérature

- [1] Elektor
- [2] EDiTS Pro : répondeur d'adresse de locomotive, Elektor n° 258, décembre 1999, page 66 et suivantes
- [3] EDiTS Pro, pilotage par ordinateur de modèle réduit ferroviaire, Publitronic, ISBN 2-86661-124-1

Adresses Internet :

Groupe utilisateurs EDiTS Pro sur Yahoo :

fr.groups.yahoo.com/group/editspro

Adresse du site EDiTS Pro de l'auteur :

www.gironet.nl/home/editspro

seule la nouvelle version, la V1.2, met à disposition le domaine d'adresse additionnel. Vu que seul le progiciel (le contenu du composant programmé) a changé, les platines et du décodeur et du répondeur n'ont pas à subir de modification.

(020094)

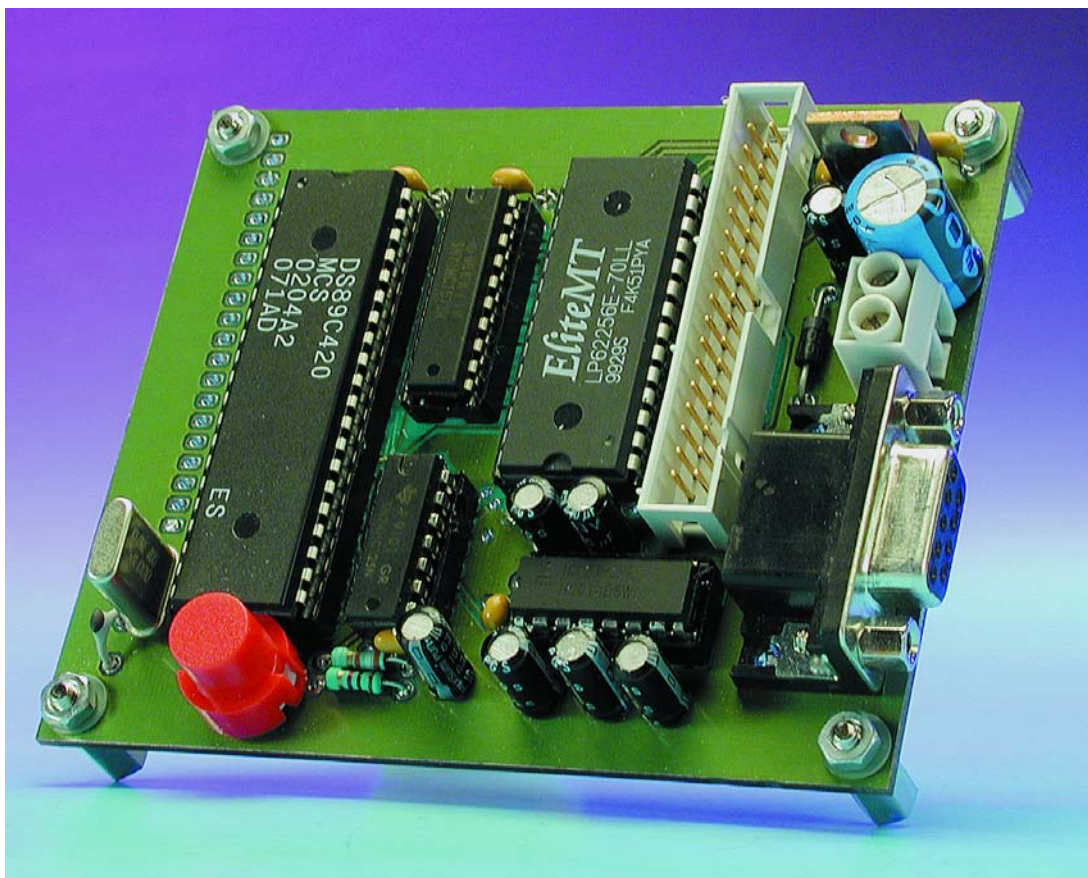
*On notera que cette nouvelle mouture de PIC s'est vu attribuer une autre dénomination, **EPS020094-41** pour le PIC du super-décodeur de locomotive et **EPS020095-41** pour le PIC du répondeur. Notez qu'il existe des offres spéciales pour 5, 10 ou 20 exemplaires (cf. site Elektor à l'adresse : www.elektor.fr et la liste EPS du mois prochain).*

Mono-carte « high-speed » à DS89C420

Capable de vitesses de traitement jusqu'à 33 MIPS !

Projet : Luc Lemmens

Au premier coup d'oeil (superficiel), le matériel de ce projet n'a rien de bien innovateur : il semble s'agir d'une mono-carte à 8051 relativement standard où seule est notable l'absence de support pour EPROM. Même la mémoire de programme Flash dont est doté le microcontrôleur utilisé ici n'a rien de révolutionnaire. C'est le DS89C420 de Dallas Semiconductor qui constitue la nouveauté de ce projet.



Il est en effet, à la même fréquence d'horloge, 12 fois plus rapide que le 8051 d'origine, ce qui lui permet de prétendre, à sa fréquence d'horloge maximale de 33 MHz, atteindre des vitesses de traitement de 33 MIPS !

Dans notre numéro de décembre de l'an passé [1] nous vous avons proposé une mono-carte à microcontrôleur utilisant un microcontrôleur Flash d'Atmel. Comme nous ne faisons pas d'ostracisme, c'est aujourd'hui au tour de Dallas Semiconductor d'avoir l'honneur de nos colonnes. Un menu réchauffé ? Oui et non. Oui, il s'agit en effet à nouveau d'un microcontrôleur doté de

mémoire Flash possédant de grandes similitudes avec son vieil ancêtre qu'est le 8051. Non : par l'apport de modifications au noyau du processeur, l'augmentation très importante de sa vitesse et l'adjonction de nouvelles caractéristiques à un 8051 standard, Dallas vient de mettre sur le marché un microcontrôleur 8 bits capable de tenir tête, sans problème, aux microcontrôleurs 16 bits plus modernes !

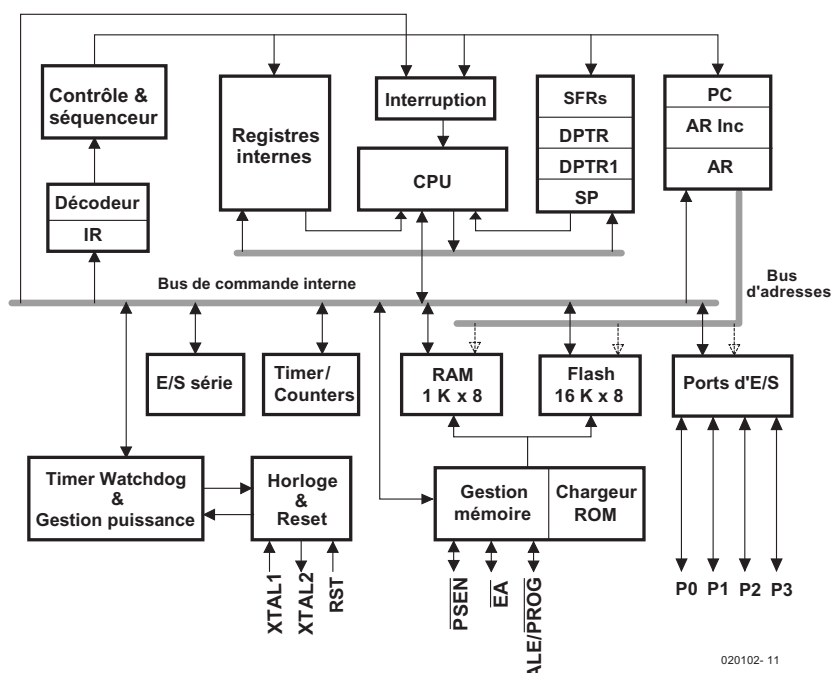
S'il est vrai que 33 MHz est une fréquence d'horloge plus que respectable pour un microcontrôleur quel qu'il soit, Dallas n'a pas atteint les 50 MHz dont ils rêvaient (c'était en effet le nombre magique qui était visé lors de l'annonce anticipée du DS98C420). Mais même à ces 33 MHz il n'en est pas moins 12 fois plus rapide qu'un 8051 standard tournant lui aussi à 33 MHz.

Signalons au passage que l'un des exemplaires de ce contrôleur dont nous disposons au laboratoire travaillait bien, à température ambiante, à 50 MHz, mais comme le fabricant doit pouvoir garantir cette fréquence maximale sur une plage de température allant de 0 à 70 °C et pour tous ses composants, il s'est résigné à parler de 33 MHz. Rien n'interdit partant aux amateurs de surcadencage de tenter leur chance...

Dallas Semiconductor est loin d'être

le seul fabricant de microcontrôleurs à broder sur le concept au succès indéniable qu'est celui du 8051. Nous vous avons présenté, au cours des ans, nombre de projets utilisant l'un ou l'autre microcontrôleur dérivé de ce contrôleur « progéniteur prolifique ». Il s'agissait dans la plupart des cas de contrôleurs dotés de périphériques additionnels tels que CAN (Convertisseur Analogique/Numérique), chien de garde (*watchdog*), unités de comparaison et capture (*compare&capture*), etc. Le DS89C420 (cf. le synoptique de la **figure 1**) ne respecte pas totalement ce moule. Même le noyau du processeur a été sujet à modifications, améliorations et autres extensions.

Toutes ces modifications sont rendues possibles par la structure de l'architecture de type 8051, en raison en particulier de la présence, dans le microcontrôleur, de registres spécifiques dits SFR (*Special Function Registers* = registre à fonction spéciale comme vous auriez pu le traduire sans être un expert de la langue de Shakespeare). Le domaine de RAM allant de l'adresse 80_{HEX} à 0FF_{HEX} est réservée à ces registres, mais sur le premier 8051 seule une partie de ces adresses était effectivement utilisée. Tout comme nombre d'autres descendants de ce composant, le DS89C420 met à contribution



020102-11

Figure 1. Synoptique de la structure interne du DS89C420.

| 8 OCTETS | | | | | | | |
|----------|-------|--------|---------|--------|------|--------|-------|
| F8 | | | | | | | |
| F0 | B | EIP1 | EIP0 | | | | |
| E8 | EIE | | | | | | |
| E0 | ACC | | | | | | |
| D8 | WDCON | | | BPA1 | BPA2 | | |
| D0 | PSW | | | | | FCNTL | FDATA |
| C8 | T2CON | T2MOD | RCAP2L | RCAP2H | TL2 | TH2 | |
| C0 | SCON1 | SBUF1 | ROMSIZE | BP2 | PMR | STATUS | TA |
| B8 | IP0 | SADEN0 | SADEN1 | | | | |
| B0 | P3 | IP1 | | | | | |
| A8 | IE | SADDR0 | SADDR1 | | | | |
| A0 | P2 | | | | | | |
| 98 | SCON0 | SBUF0 | | | | ACON | |
| 90 | P1 | EXIF | | | | | CKMOD |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON |
| 80 | P0 | SP | DPL | DPH | DPL1 | DPH1 | DPS |
| | | | | | | | PCON |

↑ BIT ADRESSABLE

020102 - 12

Figure 2. Tableau récapitulatif des SFR intégrés dans le DS89C420.

les adresses de registre « libres » pour gagner en fonctionnalités. Tous les extras ont été positionnés dans cette zone de mémoire de manière à permettre d'entrer en contact avec des périphériques additionnels ou de nouvelles fonctions par des opérations de lecture et d'écriture dans la mémoire du microcontrôleur. Ainsi, par exemple, un port sériel supplémentaire possède sa propre adresse de registre pour le tampon et un registre pour le paramétrage de ce port. Cette approche permet d'ajouter des fonctionnalités sans avoir à étendre le set d'instructions.

Un coup d'oeil au schéma de la **figure 2** nous apprend que ce nouveau rejeton de Dallas ajoute pas moins de 33 (!) registres SFR aux 21 que possède déjà, à l'origine, le 8051. Nous n'allons pas vous ennuyer en procédant à une énumération exhaustive de tous ces registres additionnels dont nous laisserons également le fonctionnement dans le vague. Nous limiterons notre intérêt aux nouveautés méritant d'être soulignées. Signalons la présence, sur le site Internet de Dallas (www.dalsemi.com) d'un *User's Guide* de près de 200 pages et d'une fiche de caractéristiques de 60 pages, documents ayant trait au dit microcontrôleur. Leur lecture vous révélera tous les secrets du DS89C420.

Le gain de vitesse est dû principalement à la structure totalement nouvelle du noyau (*core*) du contrôleur qui permet l'exécution d'un grand nombre des instructions en un seul cycle d'horloge. Notre vieux 8051 possède un

cycle-machine comportant 12 cycles d'horloge. Ceci suffit à expliquer que le DS89C420 soit, à fréquence d'horloge identique, 12 fois plus rapide que le 8051 et 4 fois plus rapide que les autres contrôleurs grande vitesse de la maison Dallas. De plus, le contrôleur intègre un certain nombre de périphériques et se targue de caractéristiques que ne possède pas le 8032 : 16 Koctets de mémoire de programme Flash, 1 Koctet de RAM, 3 temporisateurs/compteurs (*timer/counter*) à 16 bits, 2 UART (*Universal Asynchronous Receiver/Transmitter* = Émetteur/Récepteur Asynchrone Universel), 2 pointeurs de données (*datapointer*), un temporisateur chien de garde (*watchdog-timer*), 5 priorités d'interruption et 256 octets de RAM pour la pile et les variables.

Le DS89C420 possède également un mode de gestion de la consommation (*Power Management Mode*) qui permet une division dynamique par 1 024 de la fréquence d'horloge interne. Comme la consommation de courant du contrôleur est proportionnelle à la fréquence d'horloge à laquelle il travaille, ce mode permet des économies d'énergie notables aux instants de faible activité. Une simple interruption ou le déclenchement d'activités au niveau du port

sériel suffit à faire passer le contrôleur à sa pleine vitesse.

Le DS89C420 existe en 3 types de boîtier, DIP, PLCC et TQFP; il est totalement compatible broche à broche avec d'autres microcontrôleurs de la famille 8051. En ce qui concerne la programmation, ce contrôleur est totalement rétro-compatible, mais il serait dommage, pour de nombreuses applications, de ne pas mettre à profit les nouvelles possibilités offertes par ce composant. Dans ce premier article nous allons, avant d'aborder les étapes de description physique de la carte et de sa réalisation, nous intéresser aux nouvelles spécificités du DS89C420. Nous supposons ici que l'architecture du 8051 standard vous est connue.

Compatibilité

Il est parfaitement possible d'utiliser ce contrôleur en tant que remplaçant d'un 8051 monté dans une application existante, sachant qu'il en possède également la totalité du set d'instructions. En règle générale cette substitution ne devrait pas avoir de conséquence néfaste sur le fonctionnement du montage, mais la vitesse d'exécution devrait être sensiblement plus élevée. Il n'en reste pas moins qu'il y a un certain nombre de points auquel il faudra faire attention si l'on envisage une telle substitution.

Les temporisateurs/compteurs

Le DS89C420 possède, nous le disions, 3 temporisateurs (*timer*) à 16 bits qui travaillent selon le mode d'origine, à savoir 12 cycles d'horloge par cycle-machine, de sorte que la chronologie reste compatible avec celle du 8051. Il est heureusement possible, si on le désire, de les paramétrer individuellement de manière à utiliser 1 cycle d'horloge par cycle-machine.

Set d'instructions et chronologie

Les instructions remplissent exactement les mêmes fonctions que leurs homologues du 8051; les effets au niveau des bits, indicateurs (*flag*) et état du microcontrôleur sont restés inchangés. La chronologie des instructions a elle cependant changé, fait prenant une importance capitale

dans le cas de processus à la chronologie critique.

La chronologie relative de certaines instructions a été modifiée elle aussi. Avec l'architecture d'origine, les instructions « **MOVX A,@DPTR** » et « **MOV direct,direct** » duraient toutes deux 24 cycles d'horloge, ce qui revenait à 2 cycles-machine et étaient partant d'une durée identique. Dans le cas du DS89C420 la première des instructions en question ne dure plus que 2 cycles-machine soit 2 cycles d'horloge, alors que la seconde en dure 3. Il faudra partant étudier le set d'instructions avec soin lorsque la chronologie est un facteur décisif du fonctionnement correct d'un programme.

Programmable sérielement

Il est très facile de programmer le contrôleur « en circuit » c'est-à-dire lorsqu'il se trouve implanté sur le montage. IL suffit pour cela, comme nous le verrons dans le schéma donné dans le second article (à publier dans le numéro de septembre), il suffit de rajouter à l'électronique de base une triplette de tampons 3 états destinés à mettre les signaux RESET, EA et PSEN simultanément à l'état requis. Avec ce paramétrage, on adresse un secteur spécifique de la ROM du contrôleur dans lequel se trouve un programme moniteur qui est lancé. Il devient possible à partir de là, par le biais d'une liaison série vers un PC, de transférer des données vers la mémoire de programme en Flash du contrôleur et d'y chercher des données en vue de les lire. Le système de développement idéal en fait puisque l'on peut y développer et tester des applications destinées, ultérieurement, à « finir » dans un 8032 par exemple. Il va sans dire que dans ce cas-là il ne saurait être question de mettre à contribution les nouvelles possibilités, ô combien attrayantes, qu'offre le DS89C420 (vu que le 8032 ne serait pas en mesure de les comprendre). Signalons en passant que ce contrôleur est également programmable parallèlement par le biais de programmeurs d'EPROM ou de Flash standards.

La cartographie mémoire

Le DS89C420 possède les 3 domaines de mémoire que l'on différencie sur le 8051 (cf. **figure 3**) : les registres, la mémoire de programme et la mémoire de données. Tous les registres se trouvent intégrés au microcontrôleur, les 2 autres domaines pouvant être eux soit intégrés totalement au microcontrôleur, soit totalement externes, soit mixtes (partie en interne et partie en externe).

Le contrôleur dispose de 16 Koctets de mémoire de programme en Flash et de 1 Koctet de mémoire de données.

Un aspect totalement innovant est de pouvoir basculer de la mémoire de données pour en faire de la mémoire de programme ! Cela permet, par exemple, de saisir, en cours d'exécution d'un programme, un nouveau sous-programme par le biais du port série ou depuis de la mémoire externe, de l'écrire dans la mémoire de données, basculer ensuite cette partie en mode

mémoire de programme pour procéder à l'exécution de cette nouvelle partie de code. La mémoire Flash située entre 400_{HEX} et 7FF_{HEX} est, dans ce cas-là, désactivée et « remplacée » par un bloc de taille identique sis en RAM.

Bien que cette RAM fasse, physiquement, partie de la mémoire interne, elle est adressée comme de la mémoire externe par le biais d'instructions MOVX.

Les domaines d'adressage de la mémoire de programme et de celle de donnée ont tous deux une taille de 64 Koctets. Dès que l'on a sortie du domaine d'adresses du microcontrôleur, le système adresse automatiquement de la mémoire externe. La question que l'on vient alors à se poser est : mais où donc va-t-on trouver, par exemple, de la RAM statique suffisamment rapide pour pouvoir suivre ce microcontrôleur impatient ? Il n'y a pas d'inquiétude à avoir à ce sujet; lorsqu'il adresse de la mémoire externe, le DS89C420 calme le jeu, d'autant plus qu'il est même possible à l'utilisateur, de paramétrer la longueur des signaux WR, RD, PSEN et ALE de manière de permettre à un périphérique plus lente de « dialoguer » avec le système. Notons cependant que ce point spécifique

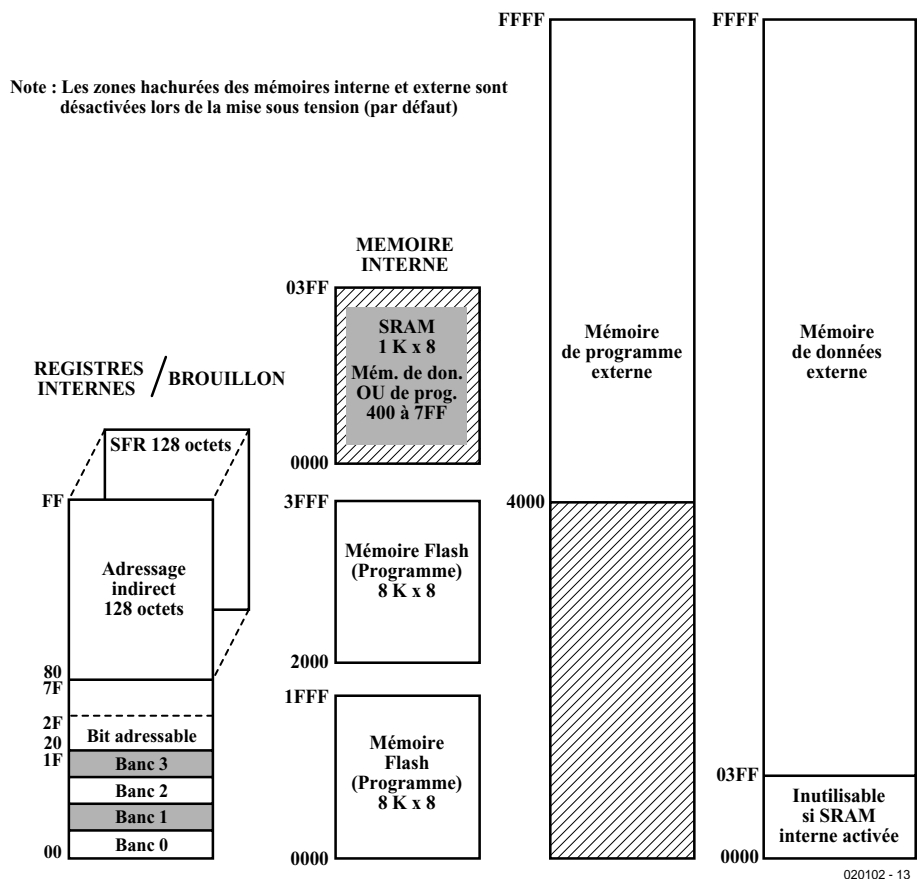


Figure 3. La cartographie du 89C420 est identique à celle du 8051 d'origine.

est plus délicat qu'il n'y paraît à première vue et qu'il demande un examen soigneux de la chronologie et du DS89C420 et de la mémoire qui y est couplée. Dallas Semiconductor a écrit un mode d'emploi décrivant dans le détail la façon de procéder, note citée en référence [2].

Il est possible, en cours d'exécution d'une application, de modifier l'adresse la plus élevée de la mémoire de programme interne. Il devient possible ainsi de désactiver une partie de cette mémoire pour libérer (momentanément) ce domaine d'adresse à l'intention de mémoire de programme externe.

Il est en outre possible d'adresser la mémoire externe en mode « pages ». Cela signifie que l'on n'utilise pas le bus d'adresses/de données multiplexé du port 0, mais d'un bus d'adresses multiplexé. Dans ce cas-là, l'octet d'adresse de poids fort transféré dans le verrou (*latch*). Tant que cet octet ne change pas, on ne modifie que l'octet d'adresse de poids faible. Il est possible de sauter le cycle de verrouillage ce qui permet une lecture ou écriture plus rapide de la mémoire externe. Il faut cependant être conscient du fait que ces mode requièrent des adaptations par rapport à la circuiterie standard.

Pointeurs de données DPTR et DPTR1

Lors de l'exécution d'instructions MOVX (lecture/écriture de la mémoire de programme externe), le 8051 utilise un registre pointeur de données, le DPTR (*Data Pointer Register*), capable de recevoir une adresse à 16 bits. Cette adresse peut concerner un emplacement dans la mémoire de données mais peut également être une adresse de matériel externe. Lorsque l'on a besoin de transférer des données d'une adresse à une autre on aura besoin d'un pointeur pour la source et d'un pointeur pour la destination. Ceci explique que le DS89C420 soit doté de 2 de ces pointeurs de registre, ce qui permet un « transbordement » bien plus efficace des données.

Un registre à fonction spéciale, un DPS (*Data Pointer Select*), est couplé à ces pointeurs, registre qui permet à l'utilisateur de procéder, automatiquement, à toutes sortes de manipulations au niveau des pointeurs de données.

Le bit DPS.0 (c'est-à-dire le bit de poids faible du DPS) définit le pointeur actif. Partant, il n'a pas été ajouté d'instruction au set d'instruction standard du 8051 pour permettre le traitement à l'aide du nouveau pointeur de données supplémentaire. La valeur par défaut de ce bit est « 0 », ce qui signifie que l'on a sélection de l'« ancien » pointeur de données. Un autre bit de ce registre (*Toggle Select*, DPS.5)

pourra permettre un basculement automatique du bit DSP.0, de manière à pouvoir passer automatiquement d'un pointeur à l'autre après une instruction ayant trait au DPTR.

Le contrôleur dispose en outre, pour simplifier et accélérer les choses, d'un mécanisme d'incrément et de décrémentation pour les 2 pointeurs de données, système activé par le bit *Auto Increment/Decrement Enable* (bit 4 du registre).

En résumé : il est possible, à condition de positionner les bits de commande adéquats, de ramener le déplacement d'un bloc de données à des instructions de lecture et d'écriture. L'un des pointeurs pointe vers l'emplacement de mémoire de la source, l'autre vers celui de la destination. Le microcontrôleur se charge de la sélection du pointeur correspondant et en modifie le contenu à souhait. Si ce concept est connu de ceux d'entre nos lecteurs qui ont affaire à des PNS (**P**rocesseur **N**umérique de **S**ignal), il est relativement innovateur dans le domaine des microcontrôleurs.

Le matériel (hardware)

Le schéma de ce projet ne présente que fort peu de surprises : on y découvre le microcontrôleur, le verrou d'adresse, la RAM, l'interface série, l'alimentation et les connecteurs d'extension. Ce type de circuiterie vous a été présenté à différentes reprises dans Elektor. Nous vous le présenterons dans le numéro de septembre. La seule nouveauté est un circuit intégré disposant de tampons à sorties 3 états qui permettent, par le biais de la ligne DTR de l'interface RS-232, de mettre le contrôleur dans un mode spécial. Lorsque les lignes \overline{EA} et \overline{PSE} du DS89C420 sont forcées au niveau bas (actives) et RESET au niveau haut au travers de ces tampons, on a, nous l'évoquions plus haut, lancement d'un programme moniteur situé sur la puce du contrôleur. Ce moniteur se trouve dans une partie spéciale de la ROM du composant, domaine dont l'accès ne peut se faire que si les 3 lignes concernées ont été mises aux niveaux indiqués plus haut, ce qui permet de ne pas avoir à lui consacrer de mémoire de programme spécifique.

L'aspect « logiciel »

Le site FTP de Dallas Semiconductor (<ftp://ftp.dalsemi.com/pub/microcontroller/>) met à disposition 2 téléchargements gratuits : *Microcontroller Toolkit* (MTKbeta.ZIP) et *Loader420* (loader420.ZIP), programmes qui permettent de faire passer la mono-carte à DS89C420 en mode moniteur et ainsi de communiquer avec le microcontrôleur. Le premier programme mentionné est « habillé » par Windows, mais présente en fait les mêmes fonctionnalités que le second. La fonction la plus importante est la programmation de la mémoire Flash interne du DS89C420, de sorte qu'il est possible, sans trop de difficultés, de transférer ses propres applications dans le contrôleur et de les adapter si nécessaire. Il est en outre possible d'examiner et de modifier les emplacements de mémoire, les ports, les registres et tutti-quanti. On se trouve partant devant un système de développement complet ! La mémoire de programme est reprogrammable ce qui permet de modifier aisément son contenu au cours du développement d'un prototype et cela même pour des projets où l'on utilisera, lors de la production finale, un microcontrôleur du type 8051 (ou similaire).

(000120-1)

Dans le prochain article (à paraître dans le numéro suivant le numéro double d'été, septembre) nous entrerons dans le détail du schéma de cette mono-carte à Ds89C420, microcontrôleur haute vitesse.

Bibliographie

- [1] Carte 89S8252 Flash, Elektor n° 282, décembre 2001, page 20 et suivantes
- [2] Application Note 26, Ultra High-Speed Microcontroller Memory Interface Timing.

Lampe de poche à LED

Les LED blanches ont le vent en poupe

Burkhard Kainka

On les trouve partout les nouveaux dispositifs d'éclairage minuscules à LED dont l'intensité lumineuse et la faible consommation de courant sont étonnantes. S'il vous venait à l'idée de réaliser vous-même un tel montage vous verrez que cela est extrêmement simple et ne requiert pas de composant exotique.



Lumineuses, durables, faibles consommatrices, voici quelques-uns des qualificatifs des LED blanches. Elles présentent cependant, comparées aux LED qui les ont précédé, qu'elles soient de couleur rouge, jaune ou verte, un petit inconvénient : il leur faut une tension de service plus élevée, à savoir de l'ordre de 3,6 V. Ceci explique que les lampes de poche à LED requièrent dans la plupart des cas, 3 cellules de pile, de manière à disposer de quelque 4,5 volts. Le présent montage prouve que l'on peut également se tirer d'affaire à l'aide d'une seule cellule. La LED n'est pas alimentée directement, encore qu'au travers d'une résistance de limitation de courant, par la source de tension mais par un

petit convertisseur de tension. On peut, de ce fait, se débrouiller avec une seule pile de 1,5 V.

Avec convertisseur de tension

Le convertisseur de tension de la **figure 1** se compose d'un multivibrateur astable, sortant quelque peu de l'ordinaire, et d'une bobine.

Lors de l'activation du bouton S1, le transistor T1 se voit alimenté en courant de base au travers des résistances R2 et R3, de sorte qu'il entre en conduction, partielle du moins.

En raison de la présence de R2, la tension de collecteur se voit maintenue à une valeur légèrement inférieure à la tension de base. Cette situation permet à T2 de devenir également partiellement conducteur. Le bruit intrinsèque atteint alors un niveau suffisant pour faire entrer T1 encore plus en conduction et bloquer totalement T2. Ce processus vaut également dans l'autre sens, vu que l'on a, par le biais de C1, un couplage de ce changement d'état ce qui se traduit par une entrée en oscillation de l'électronique.

Nous avons remplacée la résistance de collecteur classique au niveau de T2, par une inductance fixe de 470 μH . Lorsque T2 est passant il circule un courant à travers L1 où l'on a accumulation d'énergie sous la forme d'un champ magnétique.

Lorsque, ensuite, T2 bloque, le champ magnétique a pour conséquence une poursuite de la circulation du courant à travers L1, situation qui ne peut cependant se produire qu'à condition d'augmenter quasi-instantanément la tension de collecteur. Le courant ne circule cependant pas à travers le transistor bloqué, mais à travers la LED. À ce niveau, la LED limite l'impulsion de tension inductive, la tension de sortie s'adaptant cependant automatiquement à la tension directe de quelque 3,5 V. Le seul point auquel il faudra faire

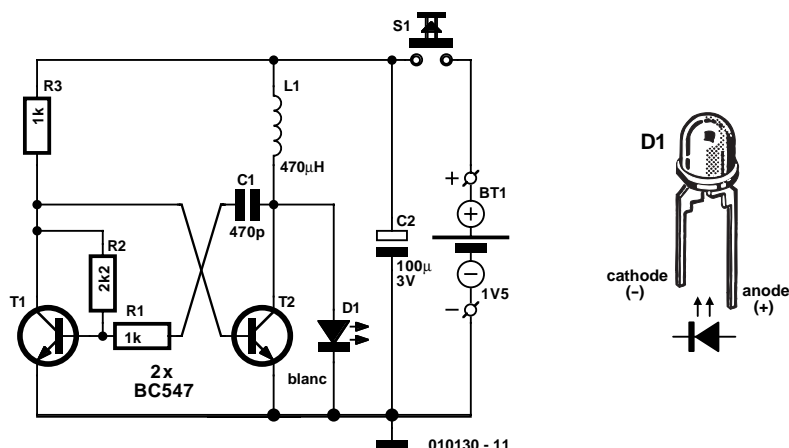


Figure 1. Le multivibrateur rehausse la tension appliquée à la LED

attention lors du dimensionnement des composants est que l'oscillation du multivibrateur soit suffisamment rapide pour éviter que l'énergie stockée dans la bobine n'ait déjà « disparu » avant que le basculement n'ait eu lieu.

Une bobine pleine d'énergie

La courbe supérieure de l'oscillogramme de la **figure 2** montre l'évolution de la tension aux bornes de la LED (1 V/div), la courbe du bas rend

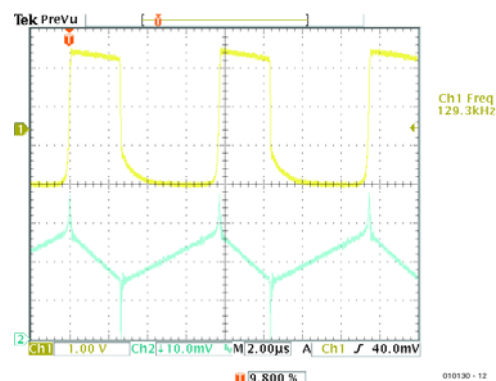


Figure 2. Oscillogramme de la tension aux bornes de la LED (1 V/div).

l'intensité du courant traversant l'inductance. La fréquence se situe aux alentours de 130 kHz. Les convertisseurs de tension « ordinaires » comportent le plus souvent un redresseur à la sortie. Ce dispositif n'est pas nécessaire ici vu que la LED régule elle-même le courant qui la traverse.

Le convertisseur de tension draine un courant de quelque 20 mA de la pile de 1,5 V. Dans ces conditions notre lampe de poche est moins gourmande que si elle était équipée d'une ampoule à incandescence miniature (mignonnette) classique. Si l'on table sur une capacité de pile de 2 000 mAh dans le cas d'une cellule alcaline, la durée de vie de la pile est de l'ordre de 100 heures. En effet, le convertisseur continue de fonctionner même lorsque la tension fournie par la pile est tombée en-dessous de 1,0 V. Ainsi, contrairement à ce qui est le cas avec une lampe de poche standard, on ne court pas le risque de se trouver brusquement dans le noir le plus complet, il reste largement le temps de se fouiller les poches à la recherche d'une pile de recharge. Notons qu'il est de ce fait également possible d'utiliser des piles dont la tension serait trop faible pour alimenter une lampe de poche à ampoule à incandescence. L'environnement vous en sera reconnaissant... Il est partant également possible d'envisager une alimentation par accumulateur CdNi de 1,2 V de tension. À une tension de 1,2 V, le circuit ne consomme que 17 mA. Il va sans dire que ces valeurs sont celles que nous avons mesurées sur notre prototype mais qu'elles peuvent varier d'une réalisation à l'autre en fonction de la qualité des composants utilisés et de leurs tolérances.

Mini-boîtier

Nous avons choisi, pour notre mini-lampe de poche à LED, un minuscule boîtier (utilisable

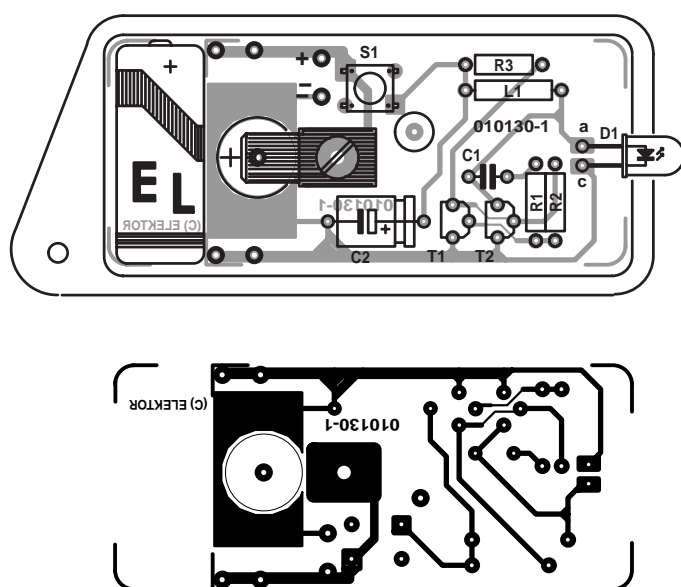


Figure 3. La platine laisse suffisamment de place pour une alimentation soit par « grosse » pile R1 soit par pile-bouton.

Liste des composants

Résistances :

R1, R3 = 1 kΩ
R2 = 2kΩ

Condensateurs :

C1 = 470 pF
C2 = 100 µF/3 V axial

Semi-conducteurs :

D1 = LED blanche
T1, T2 = BC548C, BC549C, BC550C

Divers :

L1 = self 470 µH
S1 = bouton-poussoir
Pile*
boîtier Box UM14
Matériel de fixation pour la pile

en tant que porte-clés) et dessiné une platine aux dimensions suffisamment compactes. La **figure 3** vous en propose le dessin des pistes et la sérigraphie de l'implantation des composants. Le boîtier est assez grand pour recevoir la source d'énergie, la pile. Notons que nous vous proposons un choix entre 2 types de cellules : si l'on utilise une « grande » pile (R1, Lady, AAA et tous les modèles de dimensions comparables dont le diamètre ne dépasse pas 12 mm et la longueur 30 mm) il peut s'avérer nécessaire de devoir limer légè-

rement l'intérieur du boîtier voire le doter d'un joint d'étanchéification en caoutchouc pris entre les 2 demi-coquilles. Mais on peut également utiliser une pile-bouton du type 675 (pile pour accessoires auditifs fournissant une tension de 1,4 V et présentant une capacité de 500 mAh). Il faudra pour cela percer un orifice dans la platine, souder, comme l'illustre la photo en tête d'article, une languette métallique côté

« pistes » et placer, côté « composants », un contact AMP (automobile) de manière à établir un bon contact durable.

(010130)

La réalisation ayant servi de base à ce montage provient du « coin du bricoleur » Internet de l'auteur sis à l'adresse :

<http://home.t-online.de/home/B.Kainka>

Oscilloscope et générateur de fonctions pour le PC

Appareillage de mesure à connexion Centronics

Harry Baggen

Nombre d'entre nos lecteurs connaissent sans doute le nom de Velleman, société célèbre pour les nombreux kits de montages électroniques qu'elle propose. La palette de produits et réalisations dont elle peut se targuer est encore plus riche : en effet, elle propose différents appareils de mesure. Nous avons examiné d'un peu plus près 2 de ces appareils : un oscilloscope numérique à mémoire et un générateur de fonctions pour le PC.

Comme nous le disions dans l'introduction, le nom de Velleman est presque synonyme de kits de montages électroniques bien pensés

disponibles dans nombre de magasins proposant ce type de réalisations. Depuis un certain temps déjà,

Velleman ne se limite plus uniquement aux kits de montages, leur assortiment s'étant énormément enrichi au point que l'on vous propose pratiquement tout ce que l'on peut trouver dans le commerce ayant trait à l'électronique, des globes à miroirs disco aux appareils de mesure... C'est dans cette catégorie que se classent les 2 appareils que nous allons passer au peigne fin.

Il s'agit d'un générateur de fonctions baptisé PCG10 (et K8016 dans sa version en kit) et d'un oscilloscope numérique à mémoire CS500 (dont il existe une version mono-canal en kit, le K8031). Ces 2 appareils ont été conçus en vue d'une utilisation en combinaison avec un PC. On peut partant difficilement être étonné de se trouver en présence de coffrets relativement plats de couleur grise dotés de quelques embases de connexion; la commande et la lecture de ces appareils se fait à 100% par le biais de l'écran du PC auquel sont connectés les appareils. Dans ce but,



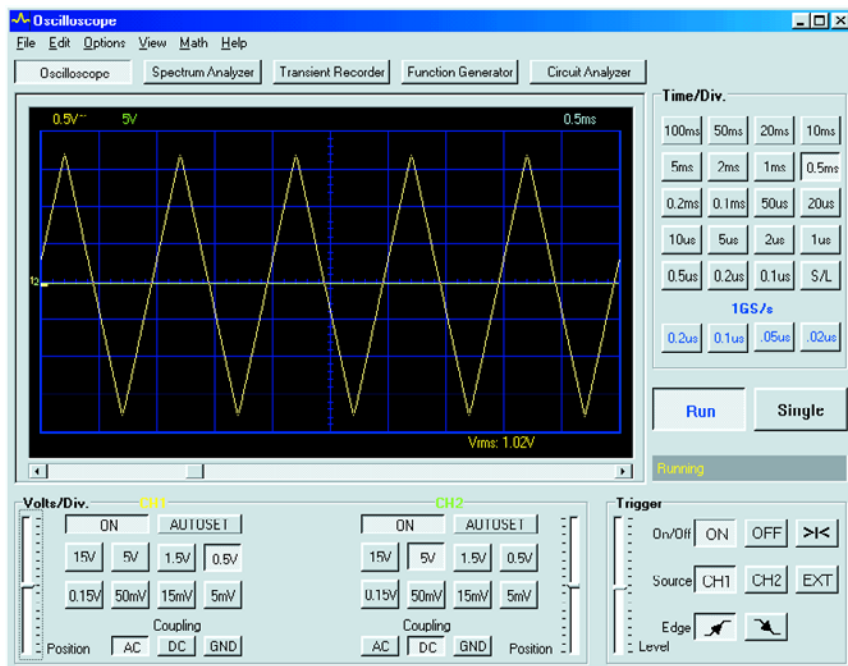


Figure 1. La fenêtre de l'oscilloscope offre toutes les possibilités de réglage que l'on trouve sur un oscilloscope bi-canal classique.

ces 2 appareils sont dotés d'une interface Centronics à isolation galvanique. Si l'on a l'intention d'utiliser les 2 appareils simultanément (ce qui paraît logique à première vue), on sera heureux d'apprendre que le générateur de fonctions comporte une possibilité d'insertion dans une boucle de sorte que l'on pourra

connecter les 2 appareils à un seul port Centronics. Commençons par passer en revue les caractéristiques techniques : les plus marquantes de ces 2 appareils :

Générateur de fonctions pour PC PCG10 :

– Plage de fréquence : 0 (CC) à

1 MHz

- Synthèse numérique directe (DDS = *Direct Digital Synthesis*) d'onde à 32 K de mémoire de forme d'onde, fréquence d'échantillonnage de 32 MHz
- Précision de la fréquence choisie : 0,01%
- Amplitude de sortie : 0,1 à 10 V_{cc} (impédance de sortie de 50 Ω)
- Plage de réglage d'offset : de -5 à +5 V
- Résolution de la forme de signal : 8 bits
- Formes d'onde disponibles : sinus, carré, triangle et forme d'onde définie par l'utilisateur
- Possibilité de mise hors-fonction du filtre de sortie

Oscilloscope numérique à mémoire PCS500 :

- Base de temps : de 100 ms à 10 ns/div
- Sensibilité d'entrée : 5 mV à 15 V/div
- Fréquence d'échantillonnage : 50 MHz
- Fréquence d'échantillonnage pour signaux continus : 1 GHz
- Nombreuses possibilités de déclenchement (flanc, niveau, source)
- Fonction d'autoset et de pré-déclenchement
- Marqueurs de chronologie, de fréquence et de tension
- Interpolation linéaire ou arrondie
- Longueur d'enregistrement : 4 096 échantillons par canal
- Fonctions additionnelles : analyseur de spectre et enregistreur de transitoires.

La pratique

Pour pouvoir examiner leur comportement en pratique, nous avons connecté les 2 appareils à un PC doté d'un Pentium. Sachant que l'appareil requiert un port Centronics il vous faudra soit connecter votre imprimante à l'interface USB (si elle est prévue pour cela à l'origine, soit acheter un câble pour imprimante assurant la conversion Centronics -> USB, soit encore enficher dans votre PC une carte (PCI ou ISA) de manière à le doter d'un second port Centronics. Dommage que ces appareils de mesure n'utilisent par l'interface USB, leur mise en oeuvre aurait été bien plus pratique (une idée pour la prochaine génération ?).

Ceci fait, les appareils peuvent être connectés. On commence par brancher le générateur au PC, l'oscilloscope étant ensuite relié à son tour au générateur. Les câbles de connexion nécessaires sont fournis avec les appareils. Il faudra bien faire attention lors du branchement des adaptateurs d'alimentation. L'oscilloscope travaille à 9 V, le générateur de fonctions à 12 V. Nous n'avons bien entendu pas

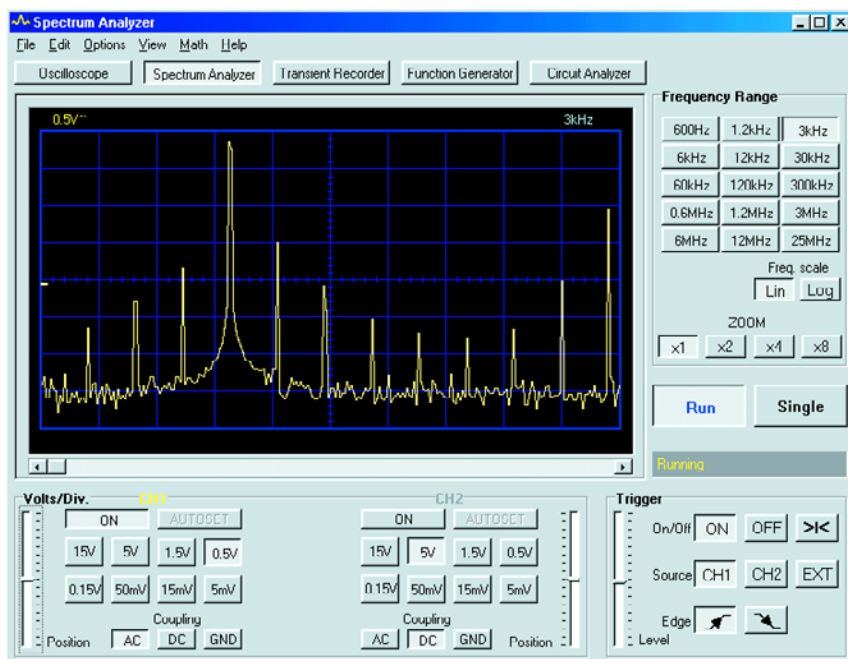


Figure 2. L'analyseur de spectre visualise la répartition des fréquences constituant le signal d'entrée.

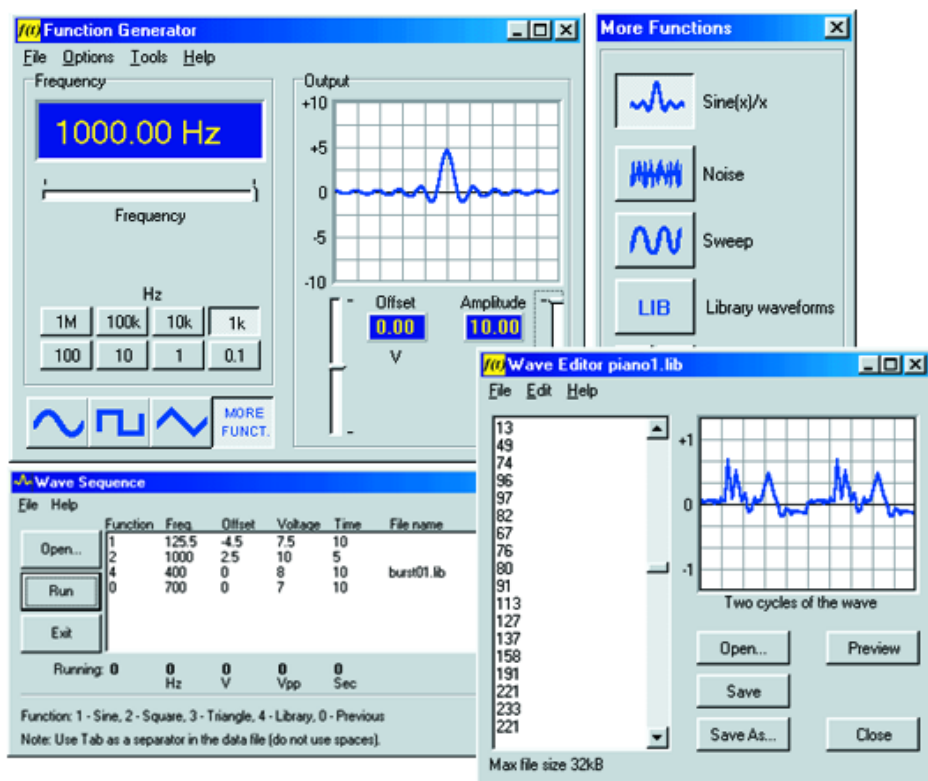


Figure 3. Le générateur offre différents extra tels que plusieurs formes d'onde préprogrammées, un séquenceur de train d'impulsions (wave sequencer) et un éditeur de train d'impulsions (wave editor).

essayer de voir ce qui se passe si l'on intervertit les adaptateurs...

Il est temps ensuite d'installer le programme correspondant, PC-Lab 2000. Ce logiciel tourne sous la majorité des versions Windows connues actuellement et l'installation sur notre PC-cible s'est faite sans le moindre problème. Le site Internet de Velleman propose les réactualisations les plus récentes.

Après lancement du programme on voit apparaître une fenêtre dans laquelle il faut cocher l'appareillage connecté au système et l'adresse à laquelle se trouve le port imprimante. On verra ensuite apparaître à l'écran la fenêtre de l'oscilloscope.

On pourra, pour les premiers essais, utiliser les sondes de mesure fournies, mais il est préférable, si l'on veut s'essayer à de « vraies » mesures, d'acheter de vraies sondes sachant que les câbles accompagnant ne sont rien de plus qu'un morceau de câble doté à l'une de ses extrémités d'un connecteur BNC et d'une paire de conducteurs terminés par des pinces crocodile à l'autre.

Dès l'application d'un signal l'oscilloscope propose, après action sur le bouton Autoset, une image parfaitement utilisable. Tant la sensibilité d'entrée que le calibre de l'échelle de temps sont ajustés automatiquement par cette fonction de sorte que l'on a affichage

d'un signal parfaitement lisible et ce d'une longueur de plusieurs périodes. Rien n'interdit bien évidemment non plus d'opter pour un réglage manuel comme on le ferait sur un oscilloscope classique. La fonction de mémoire permet l'enregistrement facile d'un signal unique et son affichage ultérieur.

Les marqueurs sont d'un usage très pratique, de même que la possibilité d'afficher à l'écran soit la valeur efficace du signal soit sa valeur en dBm. Les 4 calibres de durée les plus courts, de 200 à 20 ns/div n'ont malheureusement qu'un usage limité vu qu'ils font appel à une technique qui consiste à échantillonner un signal périodique à plusieurs moments successifs. On atteint de cette façon un pseudo-taux d'échantillonnage de 1 GHz, mais cette approche n'est utilisable correctement qu'avec des signaux ne variant pas dans le temps. Comment peut-on être sûr que tel sera le cas lors d'une mesure. Avec la fréquence d'échantillonnage normale de 50 MHz, cet oscilloscope est parfaitement utilisable pour des signaux allant jusqu'à 5 voire 10 MHz,

cette limite étant fonction de la mesure dans laquelle on doit encore pouvoir juger de la forme d'onde.

Une action sur le bouton permet de passer à la fenêtre de l'analyseur de spectre grâce auquel on peut voir la répartition en fréquence du signal, ce qui permet, par exemple, d'observer les produits de mélange. Il y a cependant quelques détails qui entravent le confort d'utilisation à la lecture. Mentionnons pour commencer une échelle de fréquence bizarre, à nos yeux (multiples de 3 et de 12, alors que l'on s'attendrait plutôt à 3 et 10; l'écran est en effet subdivisé en 10 carrés). La structure de l'échelle verticale est plus gênante elle. Il est impossible de la faire passer en un mode logarithmique, ce type d'échelle étant extrêmement utile lorsque l'on travaille avec l'affichage du résultat d'une FFT (transformée rapide de Fourier). D'autant plus qu'il est parfaitement possible, en ce qui concerne l'échelle horizontale, de sélectionner une échelle logarithmique. En dépit de cela, on découvre un certain nombre d'options fort intéressantes, le choix d'un type de fenêtre FFT donné par exemple et la présence, ici encore, de marqueurs fort pratiques (où le niveau est affiché en dB !).

L'enregistreur de transitoires est une autre fonction disponible. L'écran de cette fonction ressemble beaucoup à celui de l'oscilloscope à ceci près que l'échelle de temps est bien plus étalée (jusqu'à 2 000 s/div). Il est possible ainsi de suivre des processus de très longue durée (sur plus d'un an le cas échéant).

Les échantillons pris à l'aide de l'oscilloscope peuvent être stockés dans un fichier de texte pour être rappelés et être visualisés ultérieurement.

L'utilisation du générateur de fonctions se fait par le biais de l'ouverture d'une nouvelle fenêtre. Cette fenêtre permet non seulement d'ajuster la taille du signal, sa forme et sa fréquence, mais en outre de visualiser la forme d'onde du signal. La présentation est très lisible de sorte qu'il est facile d'ajuster la fréquence par action sur un régleur à glissière voire par l'entrée de la valeur exacte de la fréquence désirée. On dispose d'une option similaire pour le réglage de la tension de sortie et la valeur d'offset. L'appa-

reil » offre une option, outre les 3 boutons de forme d'onde sinus, carré et triangle, de bouton « More funct. » qui permet de choisir entre une fonction $\sin x/x$, du bruit, un balayage voire entre différentes formes d'ondes disponibles dans une bibliothèque. Un éditeur permet même de créer ses propres formes d'ondes.

Malheureusement, la plage de fréquences du générateur ne va pas au-delà de 1 MHz, limitation due au principe mis en oeuvre. La synthèse numérique directe d'ondes utilisée ici requiert une fréquence d'échantillonnage relativement élevée (32 MHz dans le cas présent) et elle doit être rehaussée proportionnellement lorsque l'on veut travailler à des fréquences de sortie plus élevées. Les avantages de cet appareil sont, indubitablement, une précision de la fréquence de sortie, la flexibilité du signal de sortie avec ses nombreuses formes d'onde ainsi que les possibilités de programmation.

Il est un point auquel un utilisateur potentiel devra bien faire attention : ne pas refermer la fenêtre du générateur de fonctions en cours d'utilisation sachant que si on le fait quand même, le générateur de fonctions est mis hors-fonction ce qui se traduit par la disparition du signal de sortie. Notre conseil : gardez la LED Ready du générateur bien à l'oeil, vu qu'elle s'allume dès que l'on a un signal en sortie.

Le plotter de Bode est une possibilité de test très intéressante de cette combinaison générateur de fonctions + oscilloscope. Il est en effet extrêmement simple de mesurer la caractéristique de transfert d'un filtre ou d'un amplificateur par la simple connexion de la sortie du générateur sur l'entrée du montage à tester, l'entrée de l'oscilloscope étant elle connectée à la sortie de l'électronique en question. Le reste va pratiquement tout seul, une fois que l'on a défini la plage et la sensibilité. On peut, dans le cas présent, opter pour une échelle logarithmique tant pour l'amplitude que pour la plage de fréquences, ce qui est très pratique !

La version de logiciel la plus récente offre une nouvelle possibilité fort intéressante : commander le générateur soit par le biais d'un fichier texte de son cru soit par l'in-

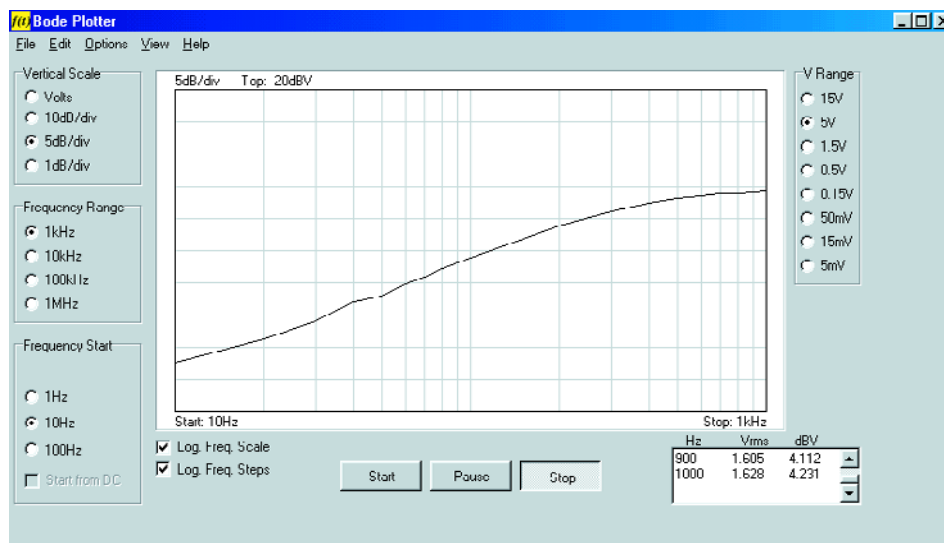


Figure 4. Un traceur de Bode permet une mesure simple de la caractéristique de réponse en fréquence d'un circuit quelconque.

termédiaire de l'interface RS-232 de l'ordinateur.

Conclusion

Si vous envisagez d'utiliser votre PC pour effectuer des mesures sur de l'électronique, la combinaison d'appareils décrite ici offre des possibilités très intéressantes. Cette combinaison convient également à des mesures à faire sur le terrain pour peu qu'on l'associe à un ordinateur portable. L'enregistreur de transitoires en particulier peut être d'un secours indéniable pour ce genre d'utilisation. Il n'en reste pas moins qu'il vous faudra trouver une prise secteur dans les alentours pour l'alimentation des appareils de mesure. Le générateur de fonctions remplit parfaitement, sans jeu de mot, sa fonction et peut se targuer de nombreuses possibilités, bien que sa plage de fréquences soit quelque peu limitée. C'est le domaine d'application qui décidera, oui ou non, s'il fait l'affaire. Prix du PCG10, 179,95 €, la version à réaliser soi-même, le K8016, coûtant 136,22 €.

La bande passante de l'oscilloscope devrait elle, au contraire, être suffisante pour la majorité des applications, ce qui ne nous empêche pas de souhaiter quelques améliorations au niveau du logiciel de l'analyseur de spectre. Velleman nous a fait part d'une réactualisation, à intervalle plus ou moins régulier du logiciel, en

le dotant d'améliorations qui peuvent aussi avoir été exprimées sous la forme de souhaits de ses utilisateurs.

Tous comptes faits, on se trouve ici en présence d'un set d'appareils de mesure attrayant. Le prix de l'oscilloscope surtout est relativement élevé, pas tant si on le compare à celui d'autres oscilloscopes pour PC concurrents vu que l'on trouve, pour la somme concernée, l'un ou l'autre oscilloscope classique parfaitement fonctionnel. Nous aurions, vu le prix du PCS500, aimé le voir doté d'une paire de vraies sondes de mesure. Si vous pouvez vous contenter d'un oscilloscope plus simple, il se pourrait que l'oscilloscope mono-canal K8031 soit exactement ce qu'il vous faut. Il est moins cher, mais travaille également à une fréquence d'échantillonnage plus faible. Qui sait, chers concepteurs, nous aurons peut-être l'occasion, la prochaine fois, de travailler avec une interface USB. Elle est sensiblement plus flexible et permet de garder le port Centronics du PC à la disposition de l'imprimante. Prix du PCS500A 495 €, le mono-canal 12 MHz à réaliser soi-même, le K8031 ne coûtant lui que 119.95 €.

(020080)

Pour plus d'informations, Velleman dispose d'un site Internet à l'adresse :

www.velleman.be

On trouvera, à l'adresse :

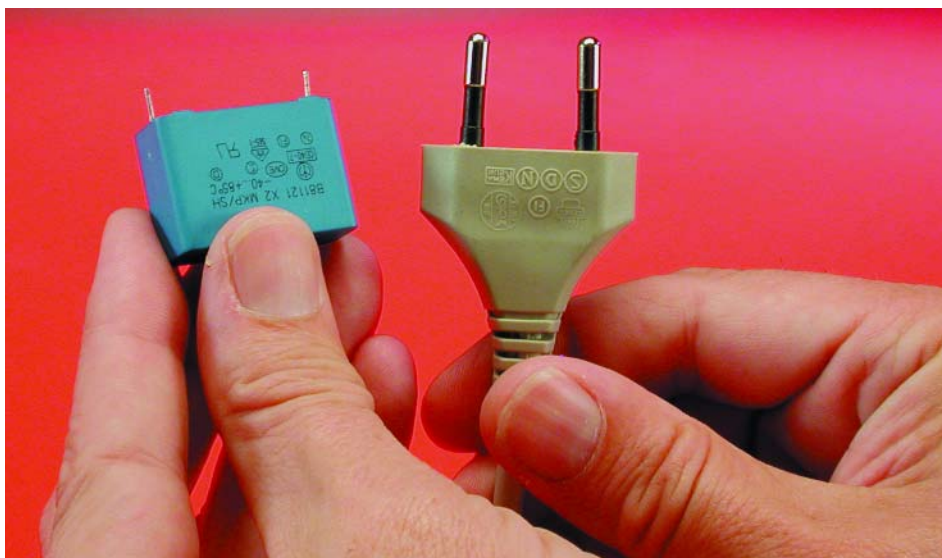
www.velleman.be/company/download.asp

les réactualisations et une version de démonstration du programme PC-Lab 2000.

Un condensateur en résistance au courant alternatif

Alimentation sans transformateur

On se passe, sur certains projets, que ce soit pour des raisons de place ou pour toute autre raison, de transformateur d'alimentation auquel on substitue une « résistance de limitation CA » qui prend la forme d'un condensateur. Il nous arrive de recevoir, de temps à autre, une lettre d'un lecteur nous demandant comment cela fonctionne et comment en calculer la valeur.



Il arrive, lorsque l'on a besoin, pour l'alimentation d'un circuit (ou sous-ensemble) électronique, de quelques milliampères seulement fournis par le secteur, que l'on opte pour une solution connue, celle du « sans tranfo », approche faisant appel à un condensateur qui remplit une fonction de « résistance de limitation de courant alternatif ». Il faut reconnaître que cette méthode n'est utilisée que

très occasionnellement vu que le gain d'espace se fait au détriment de la sécurité d'entrée en contact. En effet, l'alimentation se trouve en liaison directe avec le secteur de sorte qu'il est pour le moins délicat de parler de solution élégante. Quoi qu'il en soit, nous savons tous que chaque médaille a son revers et

que tout avantage est bardé de son inconvénient, ce qui ne nous empêche pas de constater que cette « résistance de limitation CA » peut constituer, dans certains cas, une option de solution très pratique.

L'un des derniers montages publiés dans Elektor dans lequel nous avons utilisé une alimentation sans transfo de ce genre a été la temporisation de mise en fonction du Crescendo. La **figure 1** reproduit le schéma du dit montage.

Il n'est pas nécessaire de connaître et comprendre le fonctionnement de cette électronique, l'important est de constater que le relais 24 V Re1 est alimenté par le secteur et ce en l'absence de transformateur. L'embase K1 est en effet reliée à la tension du secteur.

Dans le présent circuit, c'est le condensateur C1 qui fait office de résistance de limitation CA. Les résistances R1 à R3 peuvent porter à confusion, mais, en ce qui concerne ce mode de fonctionnement, leur importance est secondaire. R1 et R2 permettent à C1 de se décharger lors

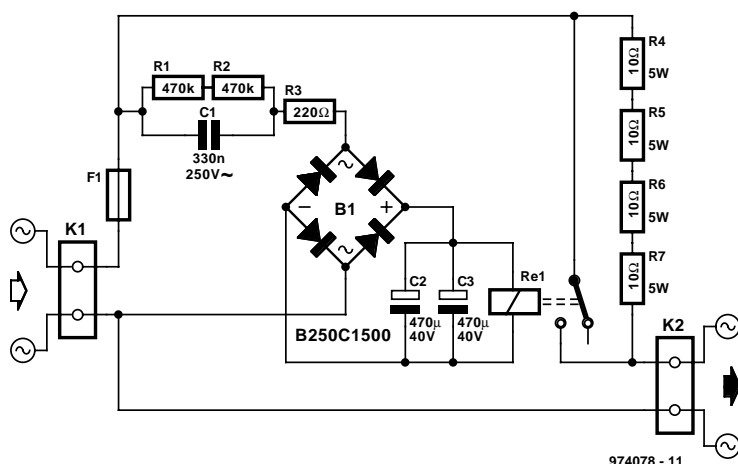


Figure 1. Dans ce circuit, le relais Re1 est alimenté directement par le secteur au travers de la « résistance de limitation CA » C1.

de la coupure de la tension d'alimentation, R3 servant à limiter le courant lors de la mise sous tension. L'abaissement de tension requise, à savoir de 230 à 24 V, est uniquement l'affaire de C1 et de lui seul.

Comment cela fonctionne-t-il ?

Il nous faut faire un rien de théorie de base. Le petit schéma de la **figure 2** sert à illustrer le principe de fonctionnement d'une telle résistance de limitation AC. Notre électronique se compose d'une pile, de 2 boutons-poussoirs, d'un condensateur et d'un instrument à bobine mobile.

Lors d'une action sur le bouton-poussoir S1 le condensateur se charge et on a circulation brève d'un courant à travers l'instrument M. Son aiguille accuse un débattement momentané avant de revenir à zéro. Si l'on relâche S1 et que l'on appuie ensuite sur le bouton S2, le condensateur se décharge ce qui se traduit par un débattement momentané, et ce dans l'autre sens, de l'aiguille de l'instrument M.

Poursuivons notre démonstration et remplaçons la pile par une source de tension alternative. Comme on se trouve en présence d'une tension qui ne cesse de changer de polarité, le condensateur se verra, en cas d'action sur S1, passer successivement en charge, en décharge, en charge à polarité inverse, en décharge et ainsi

de suite. L'instrument à bobine mobile M ne cessera de visualiser un courant, et ce bien que le dit courant ne cesse de changer de sens de circulation au rythme de la fréquence de la tension alternative. Si nous remplaçons l'instrument à Bobine mobile par un ampèremètre alternatif (un multimètre placé en calibre courant alternatif), l'instrument affichera un débattement constant. Cela permet de déduire que le condensateur C se comporte en fait comme une résistance. Cette résistance apparente au courant alternatif est appelée impédance du condensateur. La dite impédance varie en fonction de la fréquence et se laisse déterminer à l'aide de la formule suivante :

$$X_C = 1 / (2\pi fC)$$

Dans cette formule, C représente la capacité du condensateur exprimée en farads (F), f étant la fréquence; elle est exprimée en hertz (Hz).

Exemple

Comment pouvons-nous, dans la pratique, calculer la valeur d'une telle résistance de limitation AC ? Revenons pour cela au schéma de la figure 1. Le point de départ de nos calculs est qu'il nous faut un relais 24 V et que partant, la chute de tension aux bornes du condensateur C1 doit être de l'ordre de 200 V.

Il nous faut en outre connaître le courant qui traverse la bobine du relais; pour le type de relais concerné ce

Condensateurs X et Y

Il s'agit, dans le cas de ces condensateurs, de condensateurs de déparasitage (secteur) spéciaux des classes X et Y selon les normes IEC et qui se distinguent par une meilleure résistance aux impulsions et leur robustesse. IEC ne limite pas la capacité maximale des condensateurs X, mais leur implantation n'est permise qu'aux endroits où leur endommagement (court-circuit) ne risque pas de conduire à une situation dangereuse (choc électrique).

Les condensateurs d'antiparasitage de la classe Y peuvent eux en revanche être implantés à des endroits où leur défectuosité pourrait entraîner des risques de choc électrique. Ceci explique que ce type de condensateur soit lui limité à une valeur de capacité faible. Les condensateurs Y ont une tension de service de 250 V_{eff} et peuvent se targuer d'une meilleure sécurité électrique et mécanique. Les condensateurs X et Y sont subdivisés en 3 catégories (X1 à X3 et Y1 à Y3), la catégorie 1 respectant les exigences les plus sévères, le cahier des charges de la catégorie 3 étant, en ce qui concerne la résistance impulsionnelle et la durée de vie, le moins exigeant. Les condensateurs X1 sont prévus pour un fonctionnement continu sans surveillance, les condensateurs X3 étant eux utilisables sur des appareils qui ne sont reliés au secteur qu'en cours de fonctionnement tels que petits appareils ménagers et petits outils électriques. La résistance impulsionnelle des condensateurs X1 est de 4 kV (4 000 V), celle des condensateurs Y1 de 8 kV.

courant est de 20 mA.

Il est possible, en faisant appel à la loi d'Ohm, de calculer l'impédance que doit présenter le condensateur en se basant sur la valeur de la résistance équivalente aux bornes de laquelle on a, avec un courant de

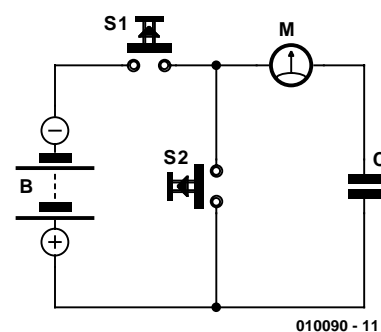


Figure 2. Une électronique simple permet de mettre le principe de fonctionnement à l'épreuve de la réalité.

20 mA, une chute de 200 V.

Elle devra être de l'ordre de $200 \text{ V} / 200 \text{ mA} = 10\,000 \, \Omega$, soit $10 \text{ k}\Omega$.

Une fois cette valeur déterminée, nous pouvons, à l'aide de la formule donnée plus haut, $X_C = 1 / (2\pi fC)$

calculer la valeur du condensateur. Nous entrons 50 Hz comme valeur de la fréquence du secteur ce qui nous donne

$$C = 1 / X_C 2\pi f$$

Si l'on remplace les termes connus cela nous donne :

$$C = 1 / 10\,000 \, \Omega * 2 * 3,14 * 50 \text{ Hz}$$

soit

$$C = 3,18 * 10^{-7} \text{ F}$$

Soit $0,318 \, \mu\text{F}$.

Un coup d'oeil à la figure 1 montre que nous avons opté pour une valeur de 330 nF , valeur normée la plus proche de la valeur théorique calculée.

La plage de tolérance de ce type de condensateurs est de $\pm 20 \%$ de sorte que la légère différence de la valeur utilisée par rapport à la valeur calculée n'a rien d'inquiétant. Ceux d'entre nos lecteurs que pourrait choquer notre manière de calculer, à savoir identifier la chute de tension aux bornes de la résistance capacitive X_C à celle prenant place aux bornes d'une résistance ohmique, ont, en principe raison. Dans la pratique, cette simplification amène très exactement au même résultat, c'est-à-dire à la même valeur pour le condensateur de limitation de courant et à la tension correcte aux bornes du relais. Si vous avez des doutes à ce sujet, rien ne vous interdit de recalculer les valeurs.

Il faut bien entendu en outre que le condensateur utilisé soit en mesure de supporter des tensions alternatives de 250 V, sachant de plus qu'il doit être du type X2 vu qu'il est utilisé dans une application travaillant avec la tension du secteur.

Limitations ?

Nous le disions tout en début d'article, le domaine privilégié de l'utilisation d'une résistance de limitation CA est celui où le courant ne dépasse pas quelques milliampères. cela signifie-t-il que, pour une raison ou une autre, ce dispositif ne peut pas être utilisé à des intensités de courant plus élevées ?

En principe non. Le condensateur de limitation travaille tout aussi bien à des courants importants qu'à des courants plus faibles. Tout ce qui change en fait est la taille de plus en plus importante du condensateur de sorte qu'à moment ou à un autre il devient plus raisonnable d'opter pour un transformateur. Ce point de changement d'approche se trouve,

en pratique, lorsque le courant atteint de l'ordre de 100 mA. Un condensateur X2 de $1,5 \, \mu\text{F} / 250 \text{ V}$ a pratiquement les mêmes dimensions qu'un transformateur secteur de $2 \times 9 \text{ V} / 100 \text{ mA}$. Dans le cas de courants plus importants encore, un condensateur électrochimique film

ou bipolaire haute tension risque même d'avoir un encombrement supérieur à celui d'un transformateur correspondant. Il sera préférable, ne serait-ce que pour des raisons de sécurité électrique, d'opter pour un transformateur.

(010090)

Théorie et pratique

Lors de notre exemple de calcul pratique de la valeur du condensateur nous avons supposé que la tension aux bornes du condensateur de limitation était égale à la tension du secteur diminuée de la tension de relais et valait partant 220 V ($224 - 24 = 200 \text{ V}$). Théoriquement cela est incorrect vu que la résistance capacitive X_C est une résistance apparente. De ce fait la résistance U_C aux bornes du condensateur est déphasée de 90° par rapport à la tension U_R aux bornes de la résistance, elle traîne de 90° par rapport à la tension sur la résistance. Sur le diagramme, les 2 tensions partielles U_R et U_C se forment un triangle rectangle (à angle droit). La tension totale (l'hypoténuse du triangle) ne s'obtient pas par addition des 2 tensions partielles mais requiert d'utiliser le vieux théorème de Pythagore pour son calcul :

$$U^2 = U_R^2 + U_C^2$$

On en dérive la tension aux bornes du condensateur

$$U_C = \sqrt{U^2 - U_R^2}$$

Nous donnons à U la valeur de la tension secteur de 230 V et pour U_R à nouveau 24 V , ce qui nous donne pour U_C :

$$U_C = \sqrt{230^2 - 24^2}$$

Et, si l'on prend la peine d'utiliser sa calculette on découvre, aux bornes du condensateur, la valeur étonnante de $228,7 \text{ V}$!

Si nous recalculons à partir de là la valeur du condensateur nous arrivons bien évidemment à un résultat légèrement différent de celui fourni par la méthode « pratique » :

$$X_C = U_C / I = 228,7 \text{ V} / 20 \text{ mA} = 11\,435 \, \Omega$$

$$C = 1 / X_C 2\pi f$$

Si nous entrons les valeurs :

$$C = 1 / 11\,435 \, \Omega * 2 * 3,14 * 50 \text{ Hz}$$

Ce qui nous donne :

$$C = 2,81 * 10^{-7} \text{ F} = 281 \text{ nF}$$

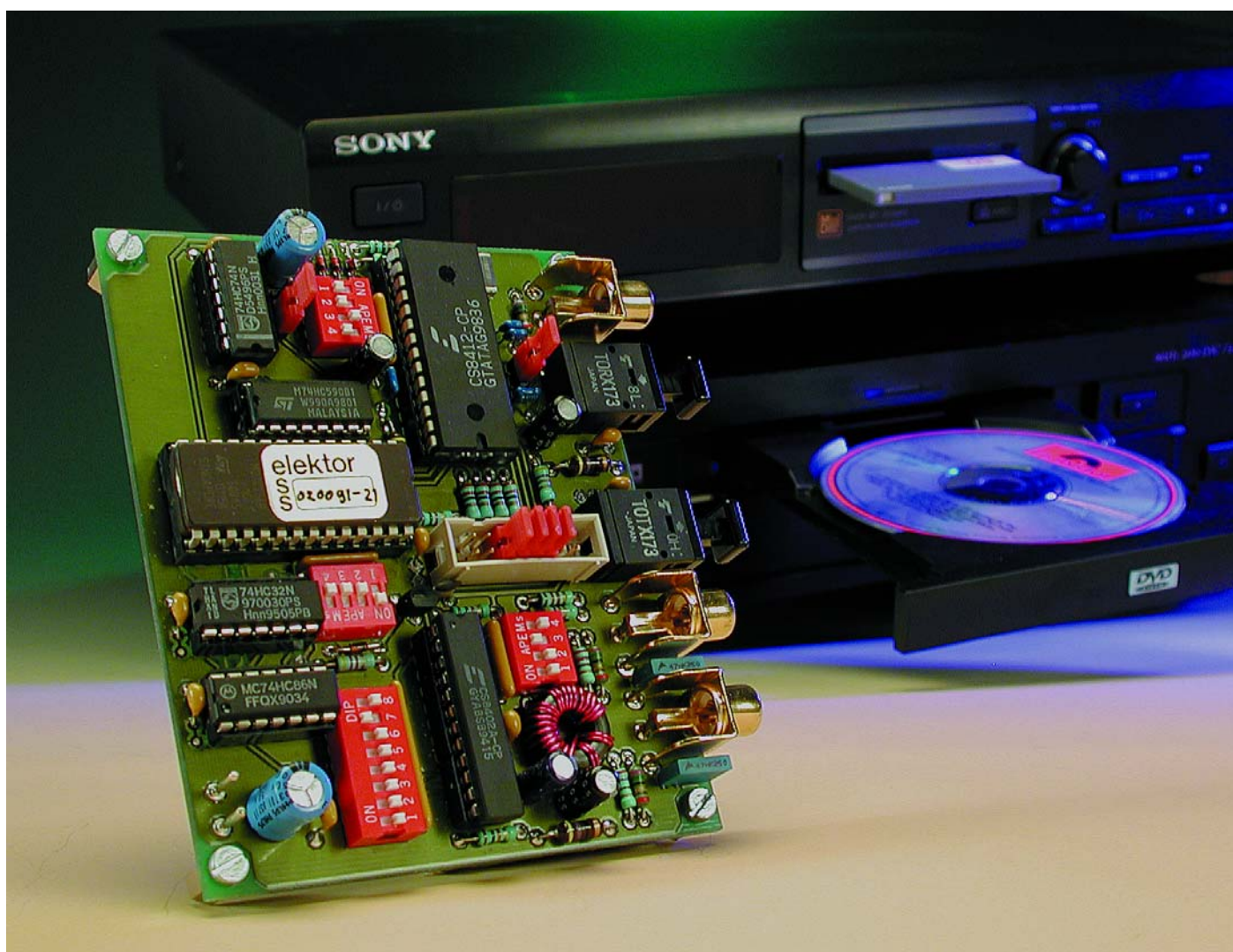
Dans la pratique, cela ne change en fait rien à la valeur du condensateur du circuit. La valeur normée la plus proche reste en effet, dans la série E6, toujours 330 nF et la tolérance par rapport à la valeur calculée est toujours inférieure à la tolérance du condensateur ($\pm 20\%$).

En conclusion : si l'on veut, à l'aide d'un condensateur de limitation, produire, à partir de la tension du secteur une tension d'alimentation faible et ce sans faire appel à un transformateur, la méthode de calcul simple proposée suffit largement.

PAN, Processeur d'Audio Numérique

...et pas seulement pour les « copybits » !

C'est sur le trajet en S/PDIF entre deux appareils audionumériques (lecteur CD, enregistreur CD ou lecteur/enregistreur MD), que vous pourrez intercaler ce montage, pas uniquement pour juguler le copybit, mais pour modifier quantité de choses dans le signal audionumérique.



Le présent montage est destiné uniquement à l'enregistrement et au traitement d'œuvres musicales personnelles ou la copie numérique à usage privé, comme la copie protégée de DVD. La Rédaction ne désire en aucune manière favoriser un quelconque usage illégal de ce montage, par lequel les droits d'auteurs seraient bafoués. Les systèmes de protection contre le piratage tels que Cactus Data Shield, SafeAudio et Key2Audio font normalement usage des propriétés du lecteur, lesquelles ne peuvent être modifiées par le montage décrit ici.

À la différence des montages similaires publiés jusqu'à présent dans Elektor, cette nouvelle réalisation est bien plus qu'un simple « étrangleur » de copybit (*copybit-killer*). Il s'agit cependant de son application primaire, surtout depuis que nous avons constaté que, sur les DVD, le système SCM (*Serial Copy Management* = interdiction de copie en série) interdit même l'unique copie permise par la législation de nombreux pays européens. Le copybit constitue, pour de nombreux audiophiles désireux de faire leurs propres compilations ou de traiter des

signaux d'audio numérique, un boulet dont ils se seraient fort bien passé. Le présent montage à d'autres atouts dans sa manche. Ainsi par exemple la possibilité, éventuellement par recours au TNS (traitement numérique du signal, l'équivalent de DSP, *Digital Signal Processing*) d'intervenir sur les données audio. En outre, il peut passer du coaxial à la fibre optique et inversement, petit supplément qui rend l'appareil encore plus universel.

En ce qui concerne les manipulations des données audionumériques, ce n'est pas uniquement le copybit que l'on peut modifier, mais le contenu quasi-complet du registre d'état de canal (CS = *Channel Status*) dont le copybit fait partie. Seules les informations de fréquence d'échantillonnage, d'accentuation (*emphasis*) et de longueur de mot resteront intactes. Il est même possible de modifier le bit-utilisateur (*user-bit*) et le bit de validité (*vality-bit*). D'ailleurs, le bit utilisateur, on peut penser à l'effacer purement et simplement. Dans le format grand-public (*consumer*), on utilise les 2 canaux, ce qui se traduit par un flux de données de bits-utilisateurs de 88 200 bits/s dans le cas d'un CD. La fonction principale de ce circuit est l'adaptation du bit d'état de canal (CS). Il s'agit là d'un bit distinct tant dans la sous-trame 1 (*sub-frame*) que dans la sous-trame 2. En pratique cependant, les 2 blocs de donnée concernés se retrouvent avec la même information. Nombre de récepteurs ne contrôlent souvent que l'une des 2 sous-frames. On détermine ou modifie partant les données reçues pour les 2 sous-frames simultanément. La seule limitation du présent montage est que cette opération se fait au niveau du bloc d'état de canal (*channel-status block*) de 192 trames. Les données du canal-utilisateur ne présentent en

effet pas de correspondance avec la taille de ce bloc. Nous vous renvoyons, pour plus d'informations sur le sujet, à la spécification grand-public IEC60958-3. Le bit de validité est lui aussi présent dans chaque trame et pourra être modifié si nécessaire.

Le concept

C'est à dessein que nous avons opté pour une approche de concept la plus universelle possible de manière à offrir, à ceux d'entre nos lecteurs qui voudraient tenter leurs propres expériences, le maximum d'ouvertures. Un coup d'oeil au schéma complet de la **figure 3** montre que l'on se trouve en présence d'une électronique relativement complexe mais dont la majeure partie ne comporte qu'un petit nombre de circuits intégrés. Nous pouvons dire, si tant est que nous n'entrions pas dans les détails, que nous nous trouvons en présence d'une combinaison constituée d'un récepteur S/PDIF (*Sony/Philips Digital Interface Format*) intégré, IC1, et de l'émetteur correspondant, IC2, composants entre les sorties et entrées d'état de canal desquels (lignes CI, UI et VI de IC1 et VO, CO et UO de IC2) on trouve un compteur binaire, IC6 et une EPROM, IC9. C'est en fait ce dernier composant qui constitue le coeur et le cerveau de notre Processeur Audio Numérique, PAN pour les intimes. L'adjonction de 4 interrupteurs DIP (3 quadruples et 1 octuple) au système d'adressage de l'EPROM offre une possibilité de choix entre 16 tableaux possibles. Le **tableau 1** donne le paramétrage par défaut des 4 interrupteurs DIP, positions de contacts qui ont pour effet de faire travailler le PAN en supprimeur de copybit. Seuls les 3 premiers tableaux ont alors encore une fonction.

Récepteur et émetteur S/PDIF

IC1 et IC2 sont des circuits intégrés faisant partie du catalogue de Crystal Semiconductor (adresse Internet : www.cirrus.com) : le récepteur CS8412-CP et l'émetteur CS8402A-CP. De par l'utilisation de ces composants, le PAN convient pour des fréquences d'échantillonnage allant de 32 à 48 kHz. Nous avons déjà, par le passé, utilisé ces 2 circuits intégrés, raison pour laquelle nous vous renvoyons, pour de plus amples informations à leur sujet, à leur fiche de caractéristiques d'une part et aux montages donnés en références [1] et [2] de la bibliographie en fin d'article. Nous nous limiterons partant à une description succincte.

Le CS8412 est un circuit intégré CMOS spécialement conçu pour la réception et le décodage de données audio aux formats les plus courants de AES/EBU, IEC958, S/PDIF et

Tableau 1.

Paramétrage par défaut de tous les contacts des interrupteurs DIP et des cavaliers de court-circuit.

| | | |
|-------|-----------|----------|
| S1 : | S1-1 OFF | |
| | S1-2 OFF | |
| | S1-3 ON | |
| | S1-4 OFF | |
| S2 : | S2-1 OFF | |
| | S2-2 OFF | |
| | S2-3 ON | |
| | S2-4 N.C. | |
| S3 : | S3-1 OFF | |
| | S3-2 OFF | |
| | S3-3 OFF | |
| | S3-4 OFF | |
| | S3-5 OFF | |
| | S3-6 OFF | |
| | S3-7 ON | |
| | S3-8 OFF | |
| S4 : | S4-1 ON | |
| | S4-2 OFF | |
| | S4-3 OFF | |
| | S4-4 OFF | |
| JPI : | coax | |
| JP2 : | 256 | |
| K2 : | 5-6 | cavalier |
| | 7-8 | cavalier |
| | 9-10 | cavalier |
| | 11-12 | cavalier |

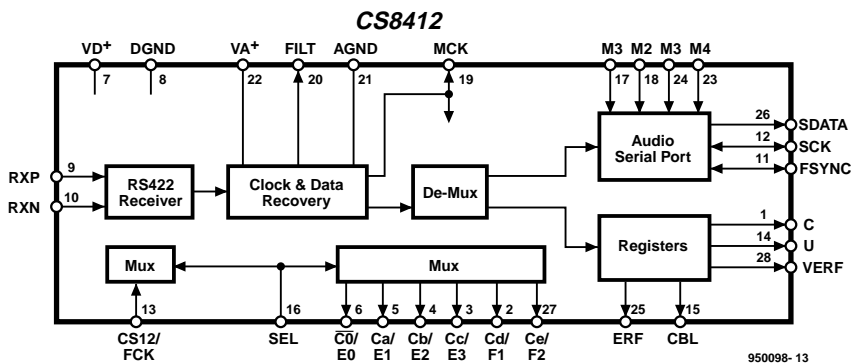


Figure 1. Structure interne du récepteur S/PDIF du type CS8412.

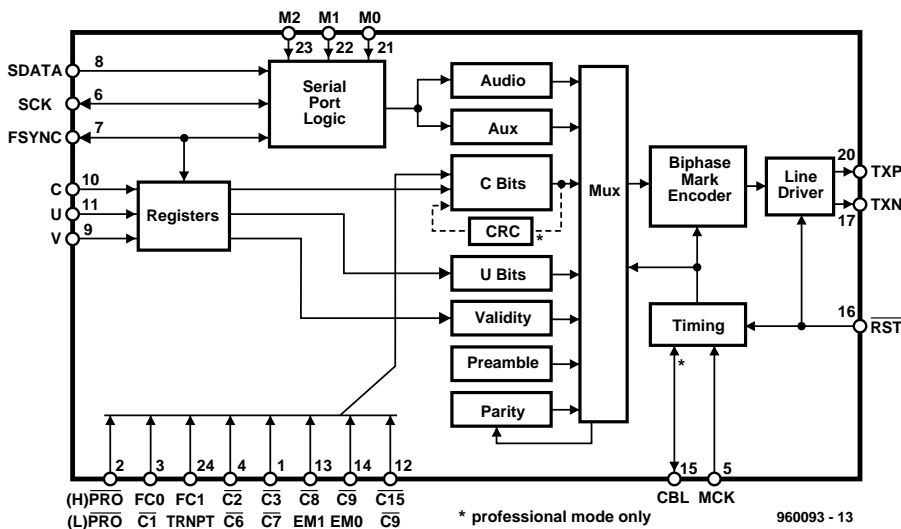


Figure 2. Les blocs de ce synoptique retraduisent la structure interne du CS8402A.

EIAJ CP340. Ce circuit intégré dont Crystal dit qu'il est un DAIR (*Digital Audio Interface Receiver*) reçoit le signal par le biais d'une interface série RS-422 et en dérive les signaux d'horloge et de synchronisation. Le système supporte un suréchantillonnage de 256x vu que le signal de sortie est doté d'une fréquence d'horloge 256 fois supérieure à la fréquence d'échantillonnage utilisée. Ce composant puissant se charge en outre du démultiplexage des informations audio et numérique. Le CS8412 démultiplexe directement les données de canal (*channel*), utilisateur (*user*) et de validité (*validity*) et les envoie vers les sorties sérieles, dont certaines sont dédiées aux bits les plus importants de l'état de canal (C, U et VERF).

Les données audio sont transférées vers l'extérieur par le biais d'un port audio sériel qui supporte pas moins de 14 formats. Ces don-

nées sont disponibles sur les broches SCK, FSYNC et SDATA, sachant que 4 broches de commande, M0 à M3, permettent de définir la configuration de ces broches de sortie.

Comme le montre un coup d'oeil au schéma, l'organe de paramétrage de ces broches prend la forme d'un quadruple interrupteur DIP, S1. Nous vous proposons, en figure 1, le synoptique (simplifié) du CS8412.

Nous en arrivons au CS8402A, le DAIT (*Digital Audio Interface Transmitter*). Comme le donne à penser sa dénomination, ce circuit intégré a été conçu spécialement pour coder et envoyer, d'où son titre de « *transmitter* », les données audio en respectant les standards de l'interface évo-

quée plus haut.

Ce circuit intégré offre une possibilité de paramétrage des bits de canal d'état les plus importants par le biais de 7 entrées. On les retrouve, en bas à gauche du synoptique du CS8402A représenté en figure 2. Si l'on revient au schéma complet du montage en figure 3 on constate que ce paramétrage se fait par le biais d'un octuple interrupteur DIP, S3. Chacune de ces 7 entrées remplit une double fonction, l'état de la broche 2 (PRO) se chargeant de cette sélection. Cette entrée (commandée au travers du contact 8 de S3, d'où sa dénomination S3-8) sert en effet à faire fonctionner le circuit intégré soit en mode « Professionnel » (*professional*) soit en mode « grand-public » (*consumer*); quel que soit le mode adopté, les données se voient codées en respect des normes valant pour chacun d'entre eux. En mode « Professionnel », le contact S3-8 est alors ouvert, il est possible de générer un code CRC (*Cyclic Redundancy Check* = contrôle de redondance cyclique) (octet d'état de canal 23), situation rendue sous la forme d'un pointillé dans le schéma synoptique. L'entrée sériele (broches 6 à 8) peut traiter 7 formats différents et des échantillons audio de 16 à 24 bits. La définition du format se fait par le biais des lignes M0 à M2; un coup d'oeil au schéma de la figure 3 montre qu'à nouveau il est fait appel, pour ce paramétrage, à un quadruple interrupteur DIP, S2. Les entrées sérieles de l'état de canal, C, de donnée utilisateur, U, et de validité, V, attaquent, au travers de quelques portes logiques, IC8 et IC9, l'EPROM IC7. Nous y reviendrons un peu plus loin.

L'examen de la figure 3 montre que la sortie symétrique TXP/TXN du CS8402A est convertie, par le biais du transformateur (à réaliser soi-même) Tr1, en 2 sorties S/PDIF standard, K3 et K4 (0,5 V_{cc} / 75 Ω). L'isolation galvanique introduite par le transformateur présente l'avantage additionnel d'éviter les boucles de masse. En outre, IC4 sert à créer une sortie optique.

Jongler avec les bits

Le cavalier JP1 permet de choisir le signal d'entrée à envoyer vers le

récepteur, coaxial ou optique. IC1 se charge du décodage, non seulement des données audio, mais aussi des bits U, C et V. La sortie VERF concerne le bit V (de validité) reçu, information ayant subi une fonction OU logique (OR) avec l'indicateur d'erreur interne ERF (*ERror Flag*), signal activé en cas de détection d'une erreur de parité, d'une violation de codage biphase ou d'un non-verrouillage de la PLL du récepteur. Un bit V activé indique qu'il y a un problème au niveau de l'échantillon audio concerné, auquel cas cette

donnée peut être, au niveau du convertisseur N/A (DAC) ou de l'enregistreur, soit éliminée soit interpolée (à partir des données adjacentes).

Ces 3 bits attaquent un certain nombre de portes, IC8 et IC9, ce qui permet de définir 4 états par bit. Il est partant possible, pour chacun des bits, au niveau de chaque trame (mais également de chaque sous-trame), de transmettre le bit concerné tel quel (inchangé), inversé, voire forcé à « 1 » ou à « 0 ». Une paire de bits de données de

l'EPROM, une porte OU et une porte EXOR permettent de créer les 4 combinaisons : D0/D1 pour le bit d'état de canal, D2/D3 pour le bit de canal utilisateur et D4/D5 pour le bit de validité.

Prenons, à titre d'exemple, les combinaisons de D0/D1 pour le bit C. Si la combinaison est 0/0 (inchangé), CI passe par la porte OU IC8.A et EXOR IC9.A, sans subir de modification. Dans le cas d'un 1/0 (forcé à « 1 ») : on aura un « 1 » aux sorties de 2 portes. 0/1 (inversé) : CI passe sans encombre par la porte OR mais est inversé par la porte EXOR. 1/1 (forcé à « 0 ») : on trouve un « 1 » en sortie de la porte OR, niveau inversé par la porte

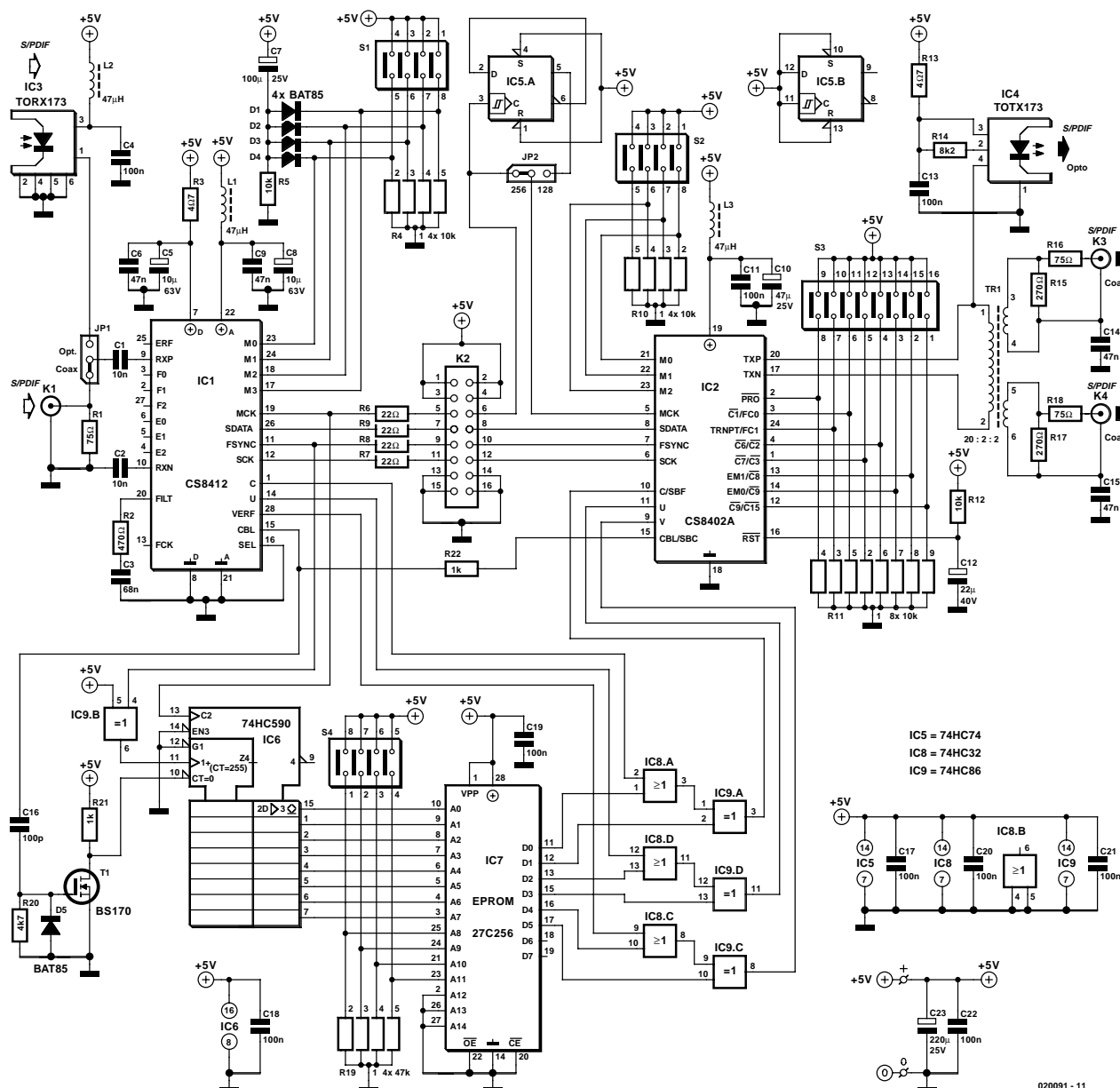


Figure 3. Nous avons essayé de proposer un concept aussi universel que possible. Les 4 interrupteurs DIP utilisés ici permettent de réaliser pratiquement chaque traitement imaginable.

Tableau 2. Contenu des champs du bloc d'état de canal pour le format « grand-public ».

EXOR à la sortie de laquelle on trouvera partant un « 0 ».

Les sorties des portes sont reliées aux entrées de même nom de l'émetteur. Il faut, pour pouvoir mettre ces bits à contribution, que l'émetteur soit paramétré en mode « Professionnel » (S3-8 ouvert, OFF) et en mode transparent (S3-7 fermé, ON). Le reste des contacts de l'interrupteur S3 ne remplit plus de fonction.

L'adressage de l'EPROM fait appel à un compteur binaire à 8 bits à registre de sortie, IC6, un 74HC590. L'horloge servant à cadencer le compteur est dérivée, après inversion, du signal de synchronisation de trame FSYNC (*Frame SYNChronisation*). Le fait d'utiliser, pour le cadencement du registre de sortie, le signal MCK (*Master Clock* = horloge-maître) fait que l'adresse de la trame suivante n'est active qu'après que l'émetteur ait déjà verrouillé la seconde sous-trame (droite), une partie de décalage dû aux durées de transfert de la porte EXOR, du compteur et de l'EPROM. La synchronisation du compteur pour chaque nouveau bloc d'état de compteur utilise la sortie CBL (*Channel Status Block*) du récepteur. On envoie une impulsion de remise à zéro à l'entrée CC (*Counter Clear*) du compteur, impulsion obtenue à l'aide du réseau différentiel R20/C16, du transistor T1 et de la résistance R21, de sorte que le redémarre à 0 (ce qui se passe après 192 trames). Sachant, qu'au niveau de l'émetteur, le signal CBL pourrait, en fonction du mode adopté, aussi fort bien être une sortie, cette ligne est dotée d'une résistance, R22, prise en série, de sorte qu'il n'y a pas de risque de ce côté.

Les données en EPROM

Le facteur le plus important de cette réalisation est la définition des données à mettre en EPROM. Il nous faut pour cela connaître la composition du bloc d'état de canal (CSB). Le **tableau 2** vous en propose le contenu pour le format « grand-public ». Le **tableau 3** ajoute des informations succinctes concernant les différents bits. Le **tableau 4** montre pour finir la structure du bloc d'état de canal (CS) en aval de son traitement par l'électronique (mise au paramétrage par défaut), ce qui correspond au second tableau de l'EPROM. Dans ce dernier tableau, INP représente le bit d'ori-

Tableau 2

| Byte | Consumer format channel status fields | | | | | | | |
|------|---------------------------------------|---------------|-----------|----------|---------------------------------------|----|--------------------------|----|
| 0 | Pro/con = 0 | Non-audio = 0 | Copyright | Emphasis | | | Channel status mode = 00 | |
| | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | Category code | | | | | | | |
| | bit 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | Source number | | | | Channel number | | | |
| | bit 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 3 | Sampling frequency | | | | Clock accuracy | | | |
| | bit 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 4 | Word length | | | | (Future original sampling frequency?) | | | |
| | bit 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 5-23 | Reserved | | | | | | | |
| | bits 40-191 | | | | | | | |

Tableau 3

| Consumer format channel status field interpretations | | |
|--|--------------------------|---|
| Bits | label | interpretation |
| 0 | pro/con | 0: consumer; 1: professional format |
| 1 | non-audio | 0: suitable for conversion to analog audio using linear PCM 1: not suitable |
| 2 | copyright | 0: asserted; 1: not asserted |
| 3-5 | emphasis | 000: Emphasis not indicated 100: emphasis—CD-type |
| 6-7 | channel status mode | 00: mode zero; other values reserved |
| 8-15 | category code | The category code depends on the equipment type. For most codes the MSB of the category code carries information about generation status of the material (refer to SCMS), and is called the L-bit. The following are common codes: <div> <div>LSB</div> <div>MSB</div> <div>CD player</div> <div>DAT player</div> <div>DCC player</div> <div>Mini disc</div> </div> <div> <div>1000</div> <div>0000</div> <div>1100</div> <div>000L</div> <div>1100</div> <div>001L</div> <div>1001</div> <div>001L</div> </div> |
| 16-19 | source number | (bit 16 is LSB) |
| 20-23 | channel number | (bit 20 is LSB) |
| 24-27 | sampling frequency | 0000: 44.1 kHz 0100: 48 kHz 1100: 32 kHz |
| 28-29 | clock accuracy | 10: Level I, ±50 ppm 00: Level II, ±1000 ppm 01: Level III, variable pitch shifted |
| 30-31 | reserved | |
| 32 | word length (field size) | 0: Maximum length 20 bits 1: Maximum length 24 bits |
| 33-35 | word length | <div> <div>000:</div> <div>101:</div> <div>001:</div> <div>010:</div> <div>011:</div> <div>100:</div> </div> <div> <div>if bit 32 = 1</div> <div>not indicated</div> <div>24 bits</div> <div>23 bits</div> <div>22 bits</div> <div>21 bits</div> <div>20 bits</div> </div> <div> <div>if bit 32 = 0</div> <div>not indicated</div> <div>20 bits</div> <div>19 bits</div> <div>18 bits</div> <div>17 bits</div> <div>16 bits</div> </div> |
| 36-39 | reserved | |
| 40-191 | reserved | |

Tableau 3. Signification succincte des différents bits pour expliciter les informations du tableau 2.

Tableau 4

Le contenu du bloc d'état de canal après son passage par le Processeur Audio Numérique (avec le paramétrage par défaut).

| | LSB | | | | | | MSB | |
|-------------|-----|-----|-----|-----|-----|-----|-----|---|
| Octet 0 | 0 | 0 | 1 | INP | INP | INP | 0 | 0 |
| Octet 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Octet 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Octet 3 | INP | INP | INP | INP | 0 | 0 | 0 | 0 |
| Octet 4 | INP | INP | INP | INP | 0 | 0 | 0 | 0 |
| Oct. 5 à 23 | 0 | 0 | 0 | 0 | 0 | 0 | | |

griller dans l'EPROM du **tableau 5** (MSB/LSB : U => 00, 1 => 01, 0 => 11).

Le tableau 5 se trouve dans l'EPROM en tant que second tableau (S4-1 fermé, ON). Le premier tableau (S4-1/2/3/4 ouverts, OFF) permet le passage sans la moindre modification de toutes les données, cas dans lequel tous les bits de donnée de l'EPROM se trouvent à « 0 ». Nous avons, à titre d'extra, ajouté le 3^{ème} tableau (S4-1/3/4 ouverts, S4-2 fermé) qui force le bit de validité en permanence à « 1 ». Nous n'avons pas testé

Tableau 5

Conversion des données du tableau 4 en information binaire à mettre dans l'EPROM.

| | | | | | | | | |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Octet 0 | 00000011 | 00000011 | 00000001 | 00000000 | 00000000 | 00000000 | 00000011 | 00000011 |
| Octet 1 | 00000001 | 00000001 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 |
| Octet 2 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 |
| Octet 3 | 00000000 | 00000000 | 00000000 | 00000000 | 00000011 | 00000011 | 00000011 | 00000011 |
| Octet 4 | 00000000 | 00000000 | 00000000 | 00000000 | 00000011 | 00000011 | 00000011 | 00000011 |
| Octet 5 à 23 | 00000011 | 00000011 | 00010011 | 00000011 | 00000011 | 00000011 | 00000011 | 00000011 |

(Tous les bits C à compter du quartet (*nibble*) de poids fort (MSB) à partir de l'octet 4 sont mis à « 0 »).

gine du signal d'entrée (INPUT). Nous avons opté, en ce qui concerne le code de catégorie, le format DAT sachant que, par le passé, c'est celui qui nous a posé le moins de problèmes. Le reste des informations

est suffisamment explicite.

Si nous traduisons les bits individuels du tableau 4 en combinaisons de bits à mettre dans l'EPROM, cela nous donne, en association avec le choix des bits U et V, les données à

d'enregistreur qui décrochât dans ces conditions (le reste du contenu de ce tableau est identique au second tableau de l'EPROM). Ceux d'entre nos lecteurs qui en auraient les moyens peuvent expérimenter avec les 13 tableaux restants.

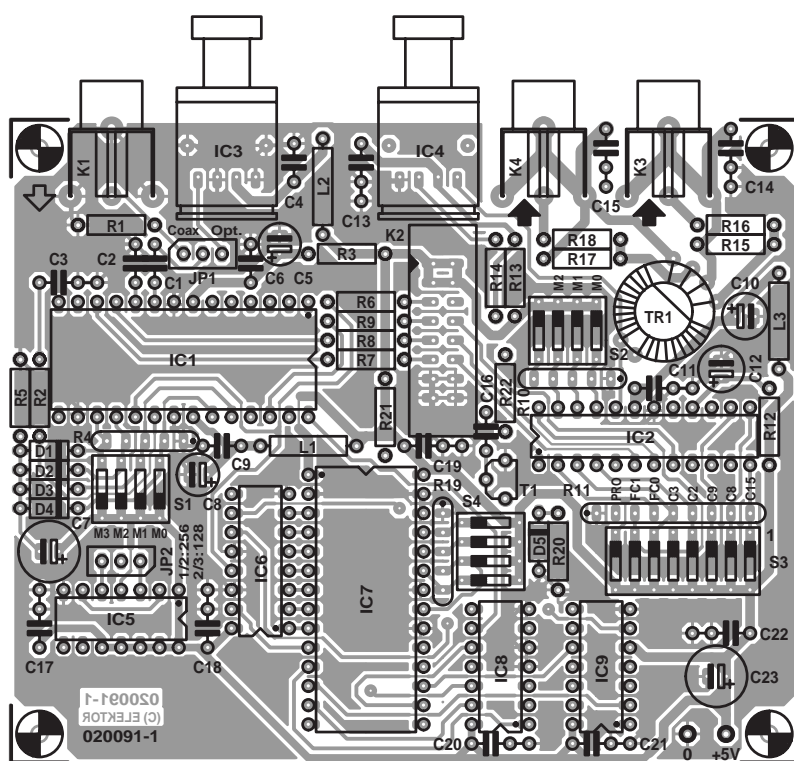
Option additionnelle

Comme nous le montre la figure 3, la transmission des données audio du récepteur vers l'émetteur transite par le biais de l'embase K2. Ceci explique qu'il faille ponter, à l'aide d'un cavalier, certaines paires de broches pour assurer le transfert de 4 des signaux (cf. tableau 1).

La présence de cette embase offre des possibilités de traitement additionnel des données audio, à l'aide d'un PNS (Processeur Numérique de Signal = DSP pour *Digital Signal Processor*). Le format de données dépend bien entendu du paramétrage choisi au niveau du récepteur (par défaut format 2, compatible I²S).

À vos fers !

Il est temps maintenant de passer au « Grand-Oeuvre », la réalisation. Nous avons dessiné une platine double face à trous métallisés pour réduire au strict indispensable le nombre de ponts de câblage nécessaire (le risque d'oubli est en relation directe avec leur nombre). La **figure 4** vous en propose le dessin des pistes et la sérigraphie de l'implantation des composants. Comme vous



Liste des composants

Résistances :

R1,R16,R18 = 75 Ω
 R2 = 470 Ω
 R3,R13 = 4k Ω
 R4,R10 = réseau de 4 résistances de 10 k Ω
 R5,R12 = 10 k Ω
 R6...R9 = 22 Ω
 R11 = réseau de 4 résistances de 10 k Ω
 R14 = 8k Ω
 R15,R17 = 270 Ω
 R19 = réseau de 4 résistances de 47 k Ω
 R20 = 4k Ω
 R21,R22 = 1 k Ω

Condensateurs :

C1,C2 = 10 nF céramique au pas de 5 mm
 C3 = 68 nF
 C4 = 100 nF céramique au pas de 5 mm
 C11,C13,C17 à C22 = 100 nF céramique
 C5,C8 = 10 μ F/63 V radial
 C6,C9 = 47 nF céramique au pas de 5 mm
 C7 = 100 μ F/25 V radial
 C10 = 47 μ F/25 V radial
 C12 = 22 μ F/40 V radial
 C14,C15 = 47 nF
 C16 = 100 pF
 C23 = 220 μ F/25 V radial

Selfs :

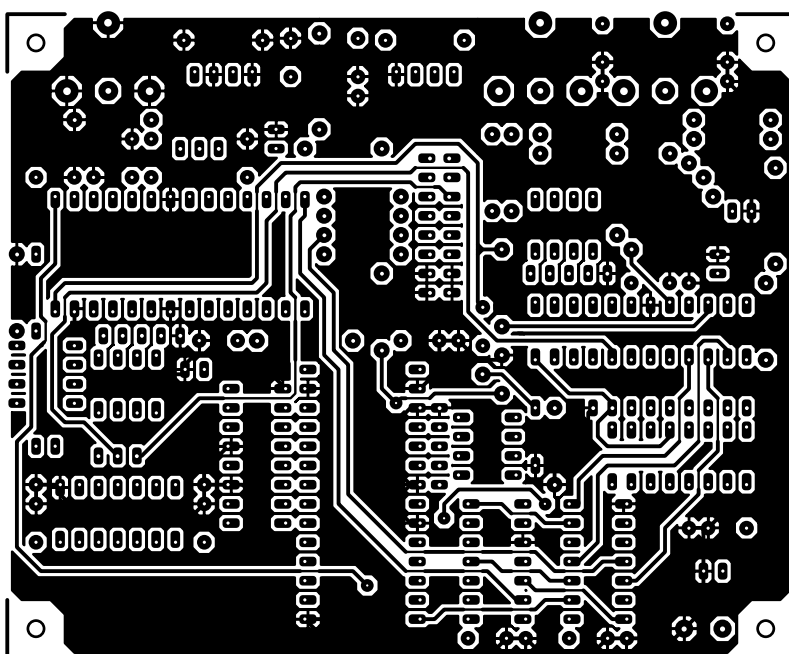
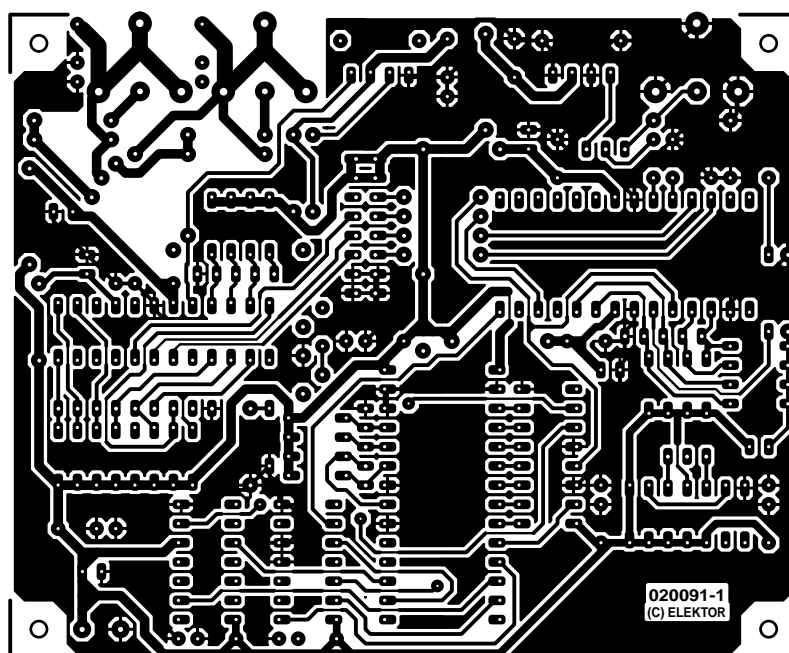
L1 à L3 = 47 μ H

Semi-conducteurs :

D1 à D5 = BAT85
 T1 = BS170
 IC1 = CS8412-CP (Crystal/Cirrus Logic, RS-components)
 IC2 = CS8402A-CP (Crystal/Cirrus Logic, RS-components)
 IC3 = TORX173 (Toshiba)
 IC4 = TOTX173 (Toshiba)
 IC5 = 74HC74
 IC6 = 74HC590
 IC7 = EPROM 27C256 programmée (EPS020091-21)
 IC8 = 74HC32
 IC9 = 74HC86

Divers :

JP1,JP2 = embase autosécable mâle à 3 contacts + cavalier
 K1,K3,K4 = embase Cinch encartable telle que, par exemple, T-709G (Monacor)
 K2 = embase HE-10 mâle à 2 rangées de 8 contacts + 4 cavaliers
 S1,S2,S4 = quadruple interrupteur DIP
 S3 = octuple interrupteur DIP
 Tr1 = primaire : 20 spires, secondaire : 2 x 2 spires, le tout de fil de cuivre émaillé de 0,5 mm de diamètre, sur tore de ferrite TN13/7,5/5-3E25 (Ferroxcube)



ne manquez pas de le constater, la platine est, bien que compacte, très « lisible ». Nous avons, pour peu que nous en ayons eu la place, indiqué à même la platine, au niveau des interrupteurs DIP et des embases à cavalier, les fonctions des organes correspondants. Attention à S1 où le contact S1-4 correspond à M0. Les embases Cinch et les connecteurs optiques se trouvent sur le même bord de platine, ce qui ne peut que simplifier la mise de cette platine

quates.

La mise en place des composants ne devrait guère poser de problème vu l'absence de composants CMS ou autres sujets exotiques. Les circuits intégrés pourront prendre place dans un support. On notera l'absence de composants côté « pistes ».

Petit ouvrage « fait main », le bobinage du transformateur Tr1. Mais ici, il ne s'agit pas de broderie. Le noyau est un Ferroxcube (autrefois Philips) du type TN13/7,5/5-3E25. L'enroulement primaire se compose de

20 spires et les deux secondaires chacun de 2 spires de fil de cuivre émaillé de 0,5 mm de diamètre. Les spires du primaire doivent être régulièrement réparties sur le noyau, en laissant un peu de place au milieu pour y bobiner les deux secondaires, ce que la **figure 5** illustre avec précision.

Il sera temps, une fois que l'on a terminé la mise en place des composants et que l'on aura contrôlé son travail en se référant à la sérigraphie et à la liste des composants, de brancher une source de tension fournissant 5 V entre les points identifiés par un « 0 » (masse) et « +5V » (pôle positif de l'alimentation). On pourra utiliser à cet effet toute alimentation si tant est qu'elle fournisse une tension correctement régulée et qu'elle soit en mesure de fournir un courant d'au moins 100 mA.

Une fois l'alimentation connectée, il est judicieux de vérifier au multimètre la présence aux bornes des condensateurs C5 et C10 de la tension requise, à savoir +5 V. Si ce test est concluant, il est fort probable que le reste de la réalisation soit OK. Si vous deviez constater des problèmes en cours d'utilisation il est plus que probable que ceux-ci soient dus à une erreur en cours de réalisation et qu'il faudra partant révéifier consciencieusement l'ensemble de la platine (court-circuit par une patte de composant restée trop longue, mauvaise soudure qui a pu lâcher, etc. la liste des sources potentielles de dysfonctionnement est extensible à volonté).

Nous n'avons pas indiqué, sur le schéma, de points de mesure ou de tensions de test vu qu'à l'exception de la ligne d'alimentation, le reste des signaux est de type numérique. Il est fort probable, pour peu que l'on ait, lors de la réalisation, travaillé avec le soin requis, que le montage fonctionne du premier coup, dès la mise sous tension.

(020091)

Bibliographie

- [1] **Vu-mètre numérique**,
Elektor n° 214, avril 1996,
page 18 et suivantes
- [2] **Convertisseur
de taux d'échantillonnage**
Elektor n° 220,
octobre 1996, page 20 et suivantes

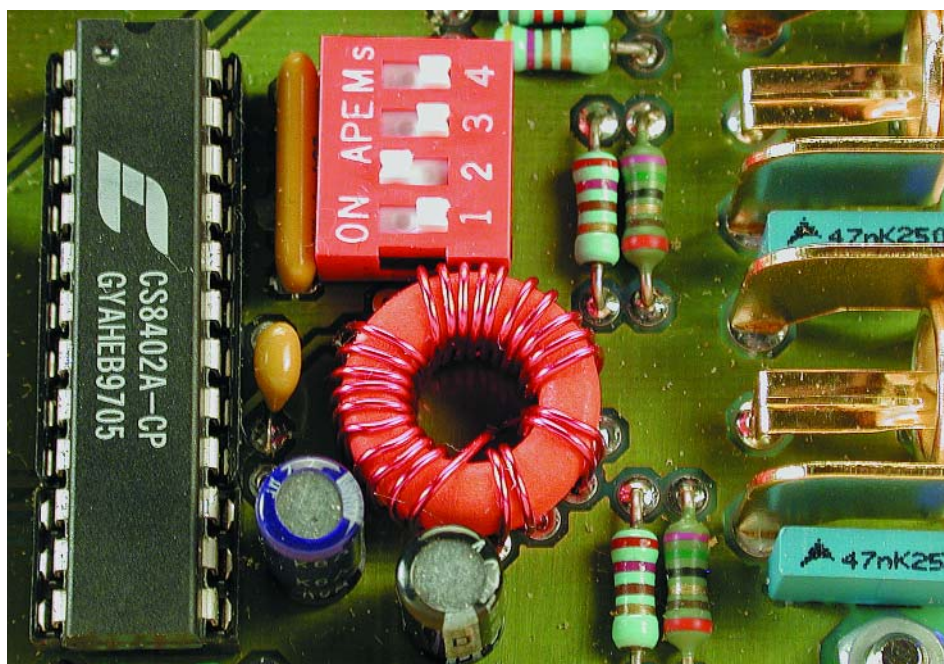


Figure 5. Le coeur du transformateur TrI est un tore de ferrite.

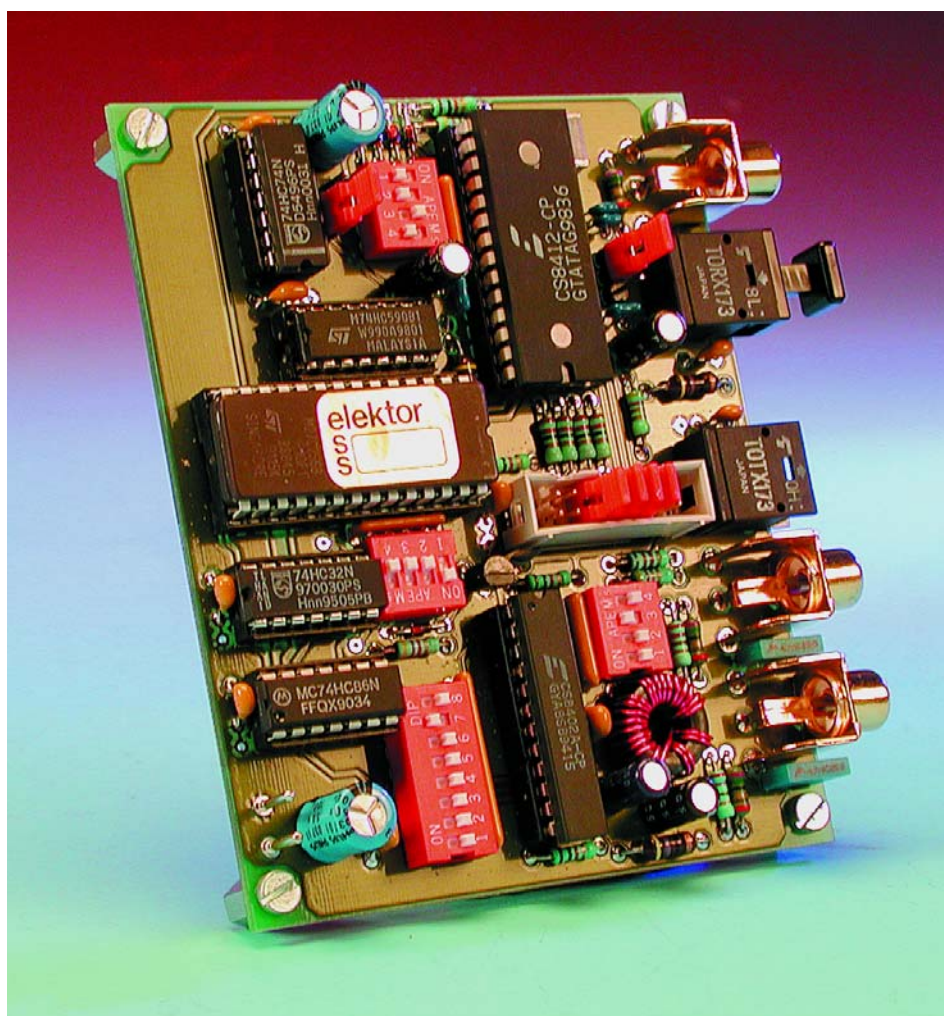


Figure 6. Si vous n'avez pas fait d'erreur, votre circuit imprimé terminé devrait être un frère-jumeau de notre prototype.

Autres capteurs d'image

Nouvelle définition du pixel

Reg Miles

regmiles@tesco.net

Bien que la grande majorité des capteurs d'image utilisés actuellement soit du type CCD (*Charge Coupled Device*), dans le monde des produits grand public en particulier et du type CMOS (*Complementary Metal Oxide Semiconductor*) pour les autres, certains autres types trouvent leur application dans nombre de produits en cours de développement voire déjà utilisés. Certains d'entre eux présentent des propriétés intéressantes qui pourraient les rendre très utilisables pour un marché plus large. Il existe en outre des variations sur le thème CCD en vue de l'adapter à des rôles spécifiques.

Il est cependant intéressant de noter que ces différentes variantes reposent toutes sur le même principe de fonctionnement. La zone sensible est subdivisée en pixels, emplacements où des photons entrants peuvent produire des paires électron-trou dans le silicium. On rassemble soit les trous, soit, plus couramment, les électrons dans un puits de charge, les autres s'évacuant par le biais du substrat. La charge du puits est déplacée, amplifiée et traitée, constituant, en combinaison avec les charges de tous les autres pixels, l'image finale.

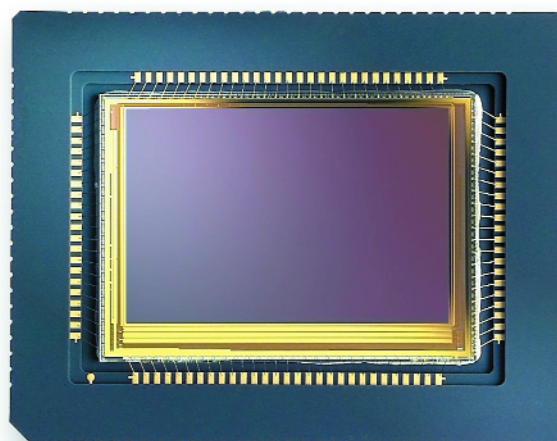
Capteur linéaire

L'une des variétés de capteurs à zone matricielle courants est le **capteur linéaire**, capteur constitué, comme l'indique son nom, d'une bande de pixels photosensibles.

Ce type de capteur est utilisé dans les appareils photo numériques et les scanners. Il travaille, comme l'illustre la **figure 1**, par exposition d'une bande à la fois. Sur chacune des positions, la charge est transférée (intégrée) dans les pixels correspondants, traduisant ainsi l'intensité de la lumière, sa luminosité. On procède ensuite à la lecture des charges, le capteur passant à la position suivante. Dans la pratique cela se traduit par un mouvement



FOVEON



continu. L'avantage de cette approche est qu'elle permet de couvrir une zone relativement importante avec un nombre de pixels sensiblement moins important que celui requis par l'approche capteur à zone matricielle; l'inconvénient majeur est que l'on peut uniquement l'utiliser avec des sujets

statiques. Si l'on veut une représentation monochrome il suffira au capteur d'effectuer une seule passe; si l'on a affaire à une représentation en couleur, il faudra effectuer 3 passes, par un filtre successivement rouge, vert et bleu (RVB, RGB en anglais pour *Red, Green et Blue*).

Il existe une alternative qui consiste à utiliser un **capteur tri-linéaire** comportant 3 bandes de pixels filtrés en RVB, de sorte que l'on n'aura qu'une passe à faire.

Les capteurs linéaires peuvent produire des images à résolution très élevée en raison de la zone relativement grande qu'ils couvrent. Il est également possible d'accroître la résolution des capteurs à recouvrement de zone par la mise en mouvement. On pourra obtenir ce résultat soit par un mouvement très décomposé (fractionnel) soit du capteur soit de l'image optique, mouvement correspondant à un pixel à chaque déplacement. Dans ces conditions un unique capteur doté de filtres sera déplacé par 3 fois de sorte que chaque pixel filtré en rouge, vert et bleu sera exposé séquentiellement à la même position, ce qui en fait se traduit par un triplement de la résolution. Il est possible d'augmenter encore la résolution par l'utilisation d'un capteur monochrome et que l'on réduit les mouvements de manière à ce que la zone réellement photosensible de chaque pixel soit exposée à une partie différente de l'image –remplissant ainsi les disparités (trous) présentées par la sensibilité. De nombreux capteurs possèdent des pixels qui ne sont pas uniformément sensibles à la lumière : ce rapport entre la sensibilité et l'insensibilité, connu sous le terme de **facteur de remplissage** (*fill factor*), peut varier entre 10 et 100%. Il faudra, si l'on a besoin d'une image couleur, répéter cette opération de déplacement par 3 fois.

Une autre variété de capteur linéaire est utilisée en reproduction d'image par scanning en ligne, normalement pour des processus industriels partant. Son fonctionnement est diamétralement opposé à celui de la paire capteur mobile + image statique –dans le cas présent c'est le capteur qui est immobile et l'image qui bouge (des objets se déplaçant sur une chaîne par exemple). On fait appel, pour obtenir la sensibilité la plus élevée possible, normalement à un processus appelé **Temporisation & Intégration** (TDI pour *Time Delay and Integration*). Cette technique a besoin d'une zone de capteur plus étendue que celle d'une matrice linéaire; on utilise des registres à

décalage pour l'intégration et le transport de charge dans le sens de mouvement de l'image.

Il va sans dire que ce processus doit se faire à grande vitesse. Pour une image en couleur, on a exposition séquentielle des sections à filtrage rouge vert et bleu.

Dans le cas d'une **matrice de surfaces** (*area array*), la vitesse élevée sera obtenue par la présence de sorties multiples qui reçoivent chacune la charge fournie par une faible partie de la surface du capteur, ce qui se traduit par des taux de balayage allant de 1 000 à quelque 5 000 trames (*frames*) par seconde, nombre qui dépend du nombre de pixels total. Pour peu que l'on travaille par petits morceaux, on parle alors de *windowing*, où l'on procède à la lecture d'une portion seulement de la surface totale, le taux de trames peut atteindre 50 000 voire plus encore. Ces techniques sont utilisées conjointement avec un diaphragme électronique qui gèle l'image en commençant par transférer la charge pour ne l'intégrer qu'ensuite pendant la très courte période nécessaire.

Bien que les capteurs à grande vitesse soient normalement du type CCD, l'Université de Stanford a développé un **DPS** (*Digital Pixel Sensor*) en technologie CMOS capable d'atteindre 10 000 trames par seconde. Ces performances sont dues à la présence d'un CAN (convertisseur analogique/numérique) à 8 bits dans chaque pixel, ce qui élimine tout risque de goulot d'étranglement analogique. En DSP, une taille de pixel acceptable requiert l'utilisation d'un processus de fabrication CMOS de 0,18 microns; ce processus connaît ses problèmes en raison de la diminution des tensions d'alimentation et de l'augmentation des courants de fuite. Les pixels de 9,4 microns de côté de la matrice de 352 x 288 pixels font appel à un **détecteur à photogrigle** (*photogate detector*, il s'agit en fait d'un condensateur MOS exposé à la lumière), plutôt que le détecteur à photodiode à jonction p-n plus classique. Chaque pixel incorpore 37 transistors, de sorte que l'on ne sera guère surpris d'apprendre que le taux de remplissage n'est que de 15%.

On utilise aussi le traitement de plan focal aux applications de capture requérant normalement un ordina-

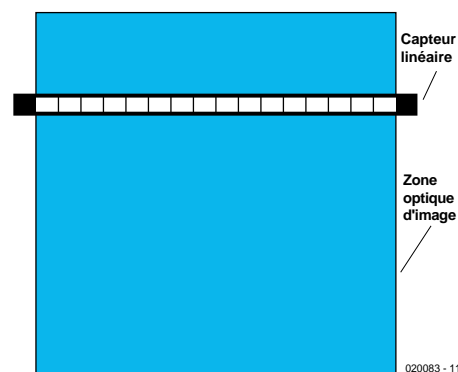


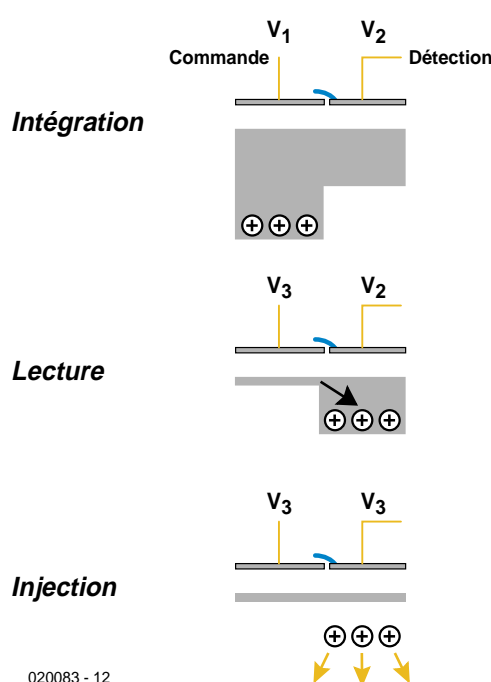
Figure 1. Principe du capteur optique linéaire.

teur externe pour un post-traitement des résultats de sortie. On pourra penser à la vision artificielle (*robotic vision*), à l'acquisition et suivi de cible (*target acquisition & tracking*), la navigation automobile et à bien d'autres applications qui reposent sur la détection de bords et autres prises en compte de mouvement. Comme, en essence, chaque pixel est en fait un minuscule ordinateur, le capteur devient un « gros » ordinateur parallèle, la nécessité d'un post-traitement s'en voyant considérablement réduit voire totalement éliminé. Le seul problème de ces capteurs « intelligents » est, nous le disions plus haut, un facteur de remplissage sensiblement plus faible.

Progrès dans le monde du traitement par plan focal

Des chercheurs de CalTch ont conçu un système de **traitement par plan focal** dont la résolution est accrue par une vibration continue du capteur CMOS, ce qui se traduit par un remplissage des « trous » laissés par les parties non sensibles des pixels. Chaque pixel travaille indépendamment, captant l'information visuelle lors du balayage du plan à reproduire; chaque pixel est lu indépendamment après traitement au niveau du pixel.

Le fait de faire travailler les pixels indéniable l'un de l'autre élimine le bruit de pattern fixé de par l'absence de comparaison entre leurs sorties. On a imaginé 2 méthodes pour la génération des vibrations. Avec la première, on utilise un miroir orienté à 45° pour réfléchir l'image vers la lentille : le miroir est monté sur l'axe d'un moteur décalé à moins de 1 degré par rapport à la perpendiculaire de sorte que l'image « frétille » en suivant un mouvement circulaire régulier. Avec la seconde technique on obtient une vibration irrégulière de l'image par la suspension de la lentille à 4 ressorts : les vibrations sont induites soit naturellement, lors de la circulation sur un terrain



020083 - 12

Dynamiques des pixels CID

– Inversion d'une électrode ou plus

$$|V_1| > |V_2| > |V_3| = 0$$

- Changement de potentiel d'une électrode
- Détection de transfert de charge (signal) sur l'autre électrode. La charge reste confinée au pixel.
- La lecture (*readout*) n'est pas destructive si V₁ est réappliqué à l'électrode de commande (*drive*).
- On a injection lorsque les potentiels des 2 électrodes ont chuté (V₃ = 0). Les charges injectées se diffusent et se recombinent dans le collecteur sous-jacent.

Figure 2. Dynamiques du pixel CID (source : Thermo CIDTEC).

inégal etc., soit par le biais d'un composant piézo-électrique.

Le CID

Le **CID** (*Charge Injection Device*) est un type de capteur pouvant se targuer d'un facteur de remplissage élevé. À l'image du CCD, ce composant est en cours de développement depuis belle lurette : son concept a été inventé par des chercheurs travaillant pour la GEC, la fameuse General Electric Company; ceux-ci annoncèrent leur premier appareil photo à CID en 1972, la même année d'ailleurs que celle où fut annoncée le premier appareil photo à CCD des chercheurs de Bell Laboratories.

La compagnie la plus concernée par la technologie CID est Thermo CIDTEC, société née d'un « *management buyout* » de GEC en 1987. Le CID tient plus du CMOS que du CCD, l'adressage individuel des pixels de faisant par le biais d'électrodes de rangée et de colonne alors que dans le second cas le transfert des charges s'effectue en masse à l'aide de registres à décalage verticaux.

Une différence cependant, chaque pixel incorpore une paire de photo-grilles se chevauchant. Autre différence, on n'a pas transfert de charge depuis le pixel au cours de sa lecture de sorte que l'intégration de la charge pourra se poursuivre –approche très utilisée dans le cas d'une commande d'exposition variable (certains pixels ou groupes de pixels

définis pourront être lus plusieurs fois pour un suivi en temps réel de l'exposition au cours de durées d'intégration importantes).

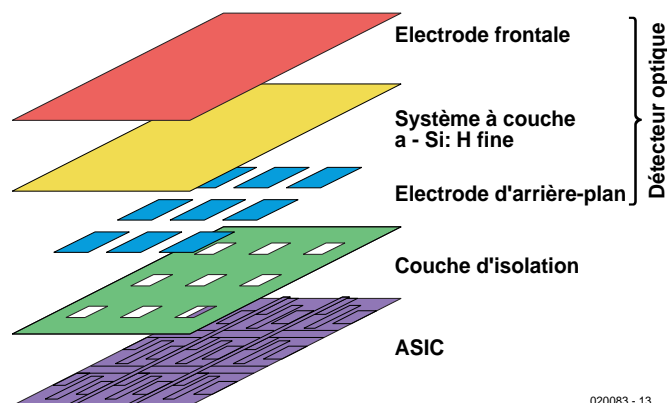
Lors de la sélection d'une photo-grille par l'électrode de commande (de rangée) sa charge est transférée vers l'autre photo-grille et l'électrode de détection (de colonne) lit un courant de déplacement proportionnel à la charge (cf. **figure 2**). Ce courant est ensuite amplifié, convertit en une tension appliquée en sortie. Le pixel ne sera vidé de sa charge que lorsque les 2 électrodes de com-

mande (*drive*) et de détection (*sense*) seront mises à la masse et que la charge sera « injectée » dans la couche épitaxiale sous-jacente. Ce processus pourra se faire à un taux de champ ou de trame normal mais aussi à toute autre vitesse.

Le CID peut se targuer d'un certain nombre d'avantages par rapport aux autres composants servant à la capture d'image. Le débordement, suite à un excédent de charge, d'un pixel vers les pixels adjacents qui se traduit par un bourgeonnement (*blooming*) est minimisé vu que l'excédent de charge est canalisé vers la couche épitaxiale et de par l'absence de registres à décalage il n'y a pas de route indirecte permettant un transfert des excédents de charge. La structure des pixels est contiguë. Les pixels peuvent contenir un nombre important d'électrons (cette quantité étant appelée la « capacité de puits »). Le composant possède en outre une réponse spectrale très large en raison de l'absence de registres à décalage absorbant la couleur et les couches très fines utilisées pour la fabrication des pixels. Côté inconvénients, on a production de bruit en raison de la connexion la capacité faible du pixel à celle relativement importante de la ligne de détection.

L'étape suivante : TFA

Une autre façon de réaliser une structure de pixels contigus consiste à opter pour une intégration verticale de la photodiode au-dessus des transistors du pixel, technique utili-



020083 - 13

Figure 3. Séquence de couche de base d'un TFA (Thick Film on ASIC) (source : Silicon Vision).

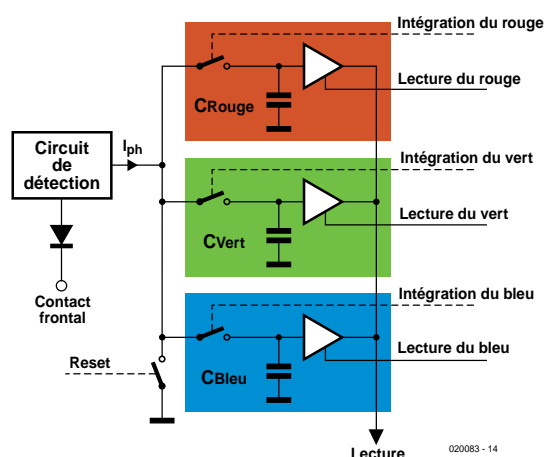


Figure 4. L'élément COSIMA fait appel à une application séquentielle des couleurs dans leurs circuits d'intégration respectifs (source : Silicon Vision).

sée par Silicon Vision pour leur technologie TFA (*Thin Film on ASIC*). Cette approche combine les excellentes propriétés optiques du silicium amorphe hydrogéné (a-Si:H) aux avantages du silicium cristallin (x-Si) des circuits intégrés et permet de réaliser des capteurs d'image intelligents. On dépose une couche de silicium amorphe très fine sur le silicium cristallin d'un ASIC (*Application Specific Integrated Circuit*) en utilisant un processus de dépôt baptisé PECVD (*Plasma Enhanced Chemical Vapour Deposition*). Le nombre de couches déposées peut aller de 4 à 7, leur épaisseur ne dépassant pas le micron.

La **figure 3** illustre la séquence de réalisation de la couche de base d'un TFA. La couche d'isolation présente une structure telle que l'on dispose des orifices de contact des pixels entre le détecteur et la circuiterie; la zone de pixels est définie par son électrode arrière qui prend la forme d'une couche métallique structurée. La couche supérieure est un oxyde transparent conducteur qui fait office d'électrode commune pour l'ensemble des pixels.

Le TFA utilise un adressage de rangée et de colonne des pixels; cela se traduit par plusieurs possibilités d'approche de lecture, y compris, le cas échéant, un accès aléatoire. Il est même envisageable de connecter les pixels de façon à les faire communiquer l'un avec l'autre ceci en vue, par exemple, de procéder à une compression ou à une reconnaissance de

pattern. Un autre avantage de cette approche est une possibilité d'optimiser le détecteur et l'ASIC comme s'ils étaient des éléments séparés. Il faut noter en outre que le processus de fabrication est plus abordable que celui d'un capteur CMOS qui est lui moins onéreux à fabriquer qu'un CCD.

Silicon Vision a développé 3 types de capteur d'image basés sur la technologie TFA. Passons-les en revue.

COSIMA, LARS et HIRISE

Comme son acronyme peut donner à le penser, **COSIMA**, le *COlour Sensor to Integrated Memory Array* est une puce couleur; ce qui le différencie de ses homologues est que le traitement des composantes rouge, vert et bleue se fait au niveau-même de chaque pixel. La reconnaissance de couleur se fait séquentiellement et repose sur la longueur d'onde : la lumière bleue de longueur d'onde plus faible est absorbée à proximité de la surface du silicium, la lumière verte pénètre un peu plus profondément, le rouge allant encore un peu plus loin. Les profondeurs de pénétration sont successivement de l'ordre de 50, 200 et 500 à 1 000 nm (1 μm). La détection des 3 couleurs se fait par application de tensions de polarisation différentes. Au cours du processus d'intégration les 3 charges sont appliquées à leur propre circuit d'intégration de couleur et stockées au niveau de chaque pixel (cf. **figure 4**).

Lors de la lecture, les tensions de couleur intégrées sont appliquées simultanément à la ligne de sortie de colonne.

Le second type de capteur est le LARS, le *Lokal-AutoadaptiveR Sensor* (non il n'y a pas d'erreur d'orthographe, la compagnie étant allemande). Comme le suggère sa dénomination, dans le cas du LARS, les pixels s'adaptent au niveau d'illumination local; le résultat de ce processus est une plage de dynamique de 120 dB (la plage d'un CCD ou CMOS classique est de l'ordre de 80 dB). Ceci le rend utilisable pour les systèmes de vision industriel et automobile. Cela tient au fait qu'il subdivise la plage de dynamique totale en 2 signaux, qui ont chacun une plage de dynamique moins étendue. La tension d'intégration du pixel est comparée, à intervalles parfaitement définis par une horloge, à une tension de référence dont la valeur se situe légèrement en deçà de la moitié de la valeur de saturation; si la tension en question dépasse ce seuil, l'intégration est stoppée, si elle est restée en deçà, la durée d'intégration est doublée (la **figure 5a** illustre le chronodiagramme d'un pixel LARS). À chaque instant de comparaison on a augmentation d'un pas d'une rampe de tension, marche d'escalier marquant la durée de l'intégration.

Lors de l'arrêt de l'intégration on a échantillonnage de l'information ainsi créée, le résultat étant stocké dans le pixel sous forme d'information chronodatée (une sorte de tampon horodatée), ce qu'illustre le chronodiagramme de la **figure 5b**. Lorsque tous les pixels sont arrivés en fin d'intégration on a lecture des 2 tensions – l'information d'intensité intégrée et l'information chronodatée – de chacun des pixels. On a recreation du photo-courant d'origine par un traitement adéquat d'une combinaison de ces 2 signaux. L'obtention de la couleur pourra se faire par une exposition successive à travers des filtres rouge, vert et bleu. Il est également possible de désactiver la fonction d'auto-adaptation du LARS pour l'utiliser comme un capteur normal.

La 3^{ème} approche est **HIRISE**, un capteur à haute résolution. Il en existe 3 versions. Le HIRISE VGA possède une matrice de 640 x 480 pixels, de 7,4 μm^2 . Le HIRISE HDTV est un capteur au format 16:9 comportant une matrice de 1 920 x 1 040 pixels de 6,1 μm^2 . Le dernier, le HIRISE MEGA est une puce du type 3 mégapixels avec une matrice de 2 160 x 1 440 pixels de 5,4 μm^2 .

Les 3 versions utilisent un filtrage des couleurs et possèdent une possibilité de sélection entre 2 sensibilités de pixel, qui se traduiront soit par une haute qualité soit par une haute sensibilité.

Une autre compagnie, Foveon, a annoncé un

capteur d'image travaillant selon ce même principe de détection des 3 couleurs au cœur même de chaque pixel. Ce composant CMOS modifié a été utilisé dans l'appareil photographique numérique qui vient d'être présenté, le Sigma SD9. Il utilise le même principe, la détection des couleurs se faisant à différents niveaux dans le silicium. Foveon a également tiré avantage de cette approche par la mise en œuvre de ce qu'ils ont appelé le principe **VPS** (*Variable Pixel Size*) qui utilise tous les pixels pour une résolution maximale ou combine des pixels adjacents pour une sensibilité plus grande ou un enregistrement vidéo flexible (il s'agit en fait d'un capteur tant pour les images fixes que pour la vidéo) ; il ne s'agit pas en fait d'un concept révolutionnaire puisqu'il a été utilisé depuis de nombreuses années sous la dénomination de « *binning* », mais le besoin de conserver la relation de couleur correcte entre les pixels filtrés en a, jusqu'à présent, limité l'utilisation à des applications spécialisées. Le capteur F7 utilisé dans l'appareil photo Sigma mesure 20,7 x 13,8 mm et comporte un total de 2 304 x 1 536 pixels à un pas de 9,12 microns. Utilisé en vidéo il atteint de 2 images par seconde à résolution maximum à 25 images par seconde à 576 x 384 pixels. Un second capteur, le F10, devrait arriver sur le marché en cours d'année. Il fera 6,4 x 4,8 mm, aura un pas de pixel de 5 microns et comportera un total de 1 344 x 1 024 pixels. Il sera capable de 10 images par seconde à pleine résolution et 30 trames en résolution VGA. Contrairement au F7 il intégrera 3 convertisseurs A/N à 12 bits travaillant jusqu'à 20 MHz, les signaux rouge, vert et bleu étant multiplexés sur un unique bus de données de sortie 3 états à 10 bits, adjacent aux sorties analogiques RVB. Le capteur Foveon fait l'objet de la photo d'illustration en début d'article.

FillFactory et TI

Il est également possible d'améliorer la sensibilité des pixels CMOS actifs à transistors internes. FillFactory a développé ce qu'ils appellent un pixel à puit N à haut facteur de remplissage (*High Fill Factor N-Well Pixel*). Outre le facteur de remplissage il a également été tenu compte du rendement quantique (qui est le nombre d'électrons produits dans le pixel pour chacun des photons incidents, relaté à sa réponse spectrale). La perte de photons peut être due à une réflexion sur les diélectriques ou sur leur non-absorption; de plus il se peut que les électrons produits ne soient pas captés et intégrés ou que l'on ait une recombinaison électron-trou. Ce problème devient de plus en plus impor-

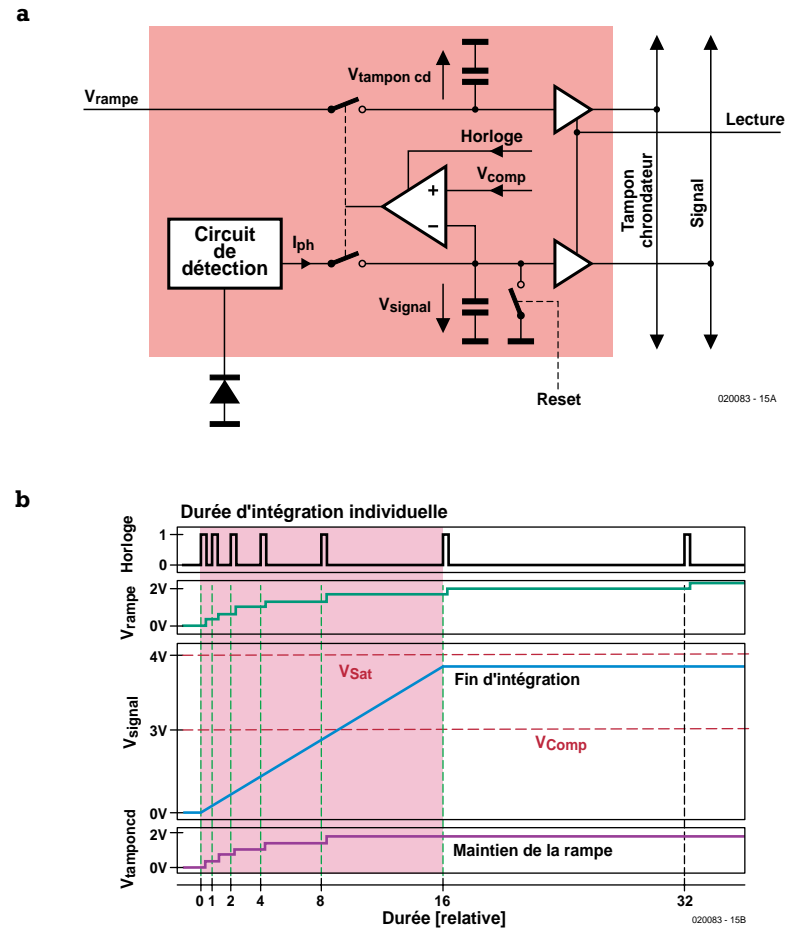


Figure 5. Pixel LARS (5a) et sa chronologie (5b) (source : Silicon Vision).

tant en raison de la réduction sans cesse croissante de la taille des pixels, diminution de taille destinée à accroître la résolution. L'approche adoptée par FillFactory est de permettre à la photodiode de détecter les électrons produits dans la couche épitaxiale située sous l'ensemble du pixel (y compris en dessous des zones insensibles), élec-

trons qui auraient, normalement, été perdus. Ces électrons sont véhiculés, le long de barrières électrostatiques qui les protègent de la circuiterie et du substrat du pixel actif, vers la jonction de la photodiode (cf. figure 6). Apparemment, la quasi-totalité des électrons fugitifs vont être canalisés le long de cette filière, ce qui se tra-

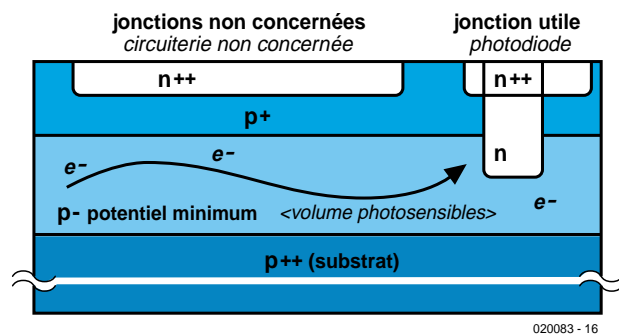


Figure 6. Méthode de population d'électrons destinée à augmenter le facteur de remplissage (source : Fillfactory).

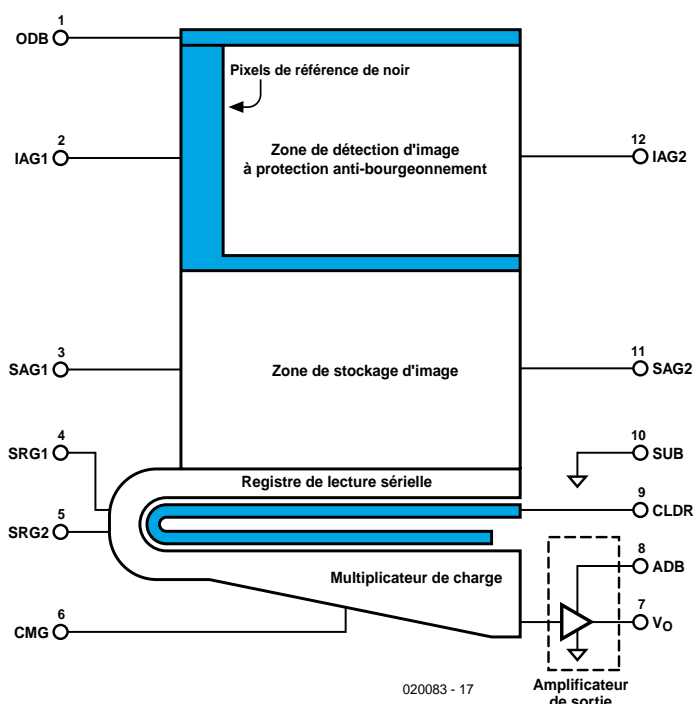


Figure 7. On pense que l'approche SPD (Single Photon Detection) ne devrait pas tarder à trouver son application dans les CCD grand public (source : Texas Instruments).

duit par un facteur de remplissage de 100%, sans nécessiter de modification de la structure du pixel. Le seul inconvénient est une diaphonie (*crosstalk*) inter-pixel plus importante sachant qu'il existe un risque de 16% de voir les électrons dériver dans un pixel adjacent (encore que l'on puisse contrer ce phénomène). Les CCD n'ont pas été en reste non plus en ce qui concerne l'augmentation de sensibilité. Texas Instruments a développé un capteur à transfert de trame à intensification d'image intégrale qu'ils ont baptisé SPD (non cela n'a rien à voir avec le parti politique de nos voisins d'outre-Rhin, mais signifie *Single Photon Detection*). Avec ce processus, on procède à une multiplication de la charge avant qu'elle ne soit transformée en tension de sortie. La multiplication de porteur de charge (CCM pour *Charge Carrier Multiplication*) est réalisée par création, entre 2 grilles adjacentes, d'une région à champ élevé et injection de la charge dans le dit champ. Lorsque les électrons traversent la région à champ élevé, ils gagnent de l'énergie et dès le dépassement d'un seuil donné, on a une ionisation

d'impact. Ceci se fait au niveau du registre à décalage horizontal lors de la lecture en sortie des charges. Le registre possède une longueur 2 fois plus importante que normalement (cf. **figure 7**) : la première moitié du registre, de structure classique, est interfacée avec la zone de stockage comme sur n'importe quel autre CCD à transfert de trame; la seconde moitié cependant intègre 400 étages CCM.

On a, à ce niveau, création, par ionisation d'impact, de nouvelles paires électron-trou; les nouveaux électrons sont recueillis dans des puits de potentiels où ils s'ajoutent à la charge d'origine, le substrat se voyant débarrassé de ses trous. On a répétition de ce processus dans chaque étage jusqu'à ce que la charge, devenue bien plus importante, atteigne l'amplificateur de sortie.

Le premier CCD SPD vient d'être proposé; il possède une plage de dynamique supérieure à 90 dB. D'après Texas Instruments, la technologie SPD peut être appliquée à toute architecture de capteur d'image connue actuellement en vue d'augmenter la sensibilité de tous les cap-

teurs d'image CCD standard, la multipliant par 50. Cette technologie a en outre l'avantage d'accroître l'étendue du spectre de sorte que sa plage spectrale s'étend alors de 200 nm (UV profond) à 1 100 nm (IR proche).

Vision robotisée ?

Il existe, pour finir, une catégorie de capteurs d'image qui dérive de l'oeil des vertébrés, le capteur focalisé. Ce type de capteur est utilisé de plus en plus souvent dans les automates dotés d'une faculté de vision, les robots par exemple. Plutôt que de répartir les pixels régulièrement sur l'ensemble de la surface, on a réalisé une zone centrale dotée de pixels plus compacts ce qui se traduit par une résolution plus élevée. Dans la pratique, la zone périphérique détecte un objet digne d'intérêt et instruit la zone centrale d'y jeter un coup d'oeil plus détaillé. Le champ de vision peut partant être très grand, mais comme seule la zone centrale fournit les détails, le nombre total de pixels nécessaires pourra être sensiblement réduit par rapport à une disposition conventionnelle, ce qui est également le cas du traitement par plan focal au cas où celui-ci serait utilisé. La plupart des approches sont du type circulaires, la disposition des pixels étant radiale, mais il existe également des versions carrées de ce type de capteur. La majorité d'entre eux sont réalisés en technologie CMOS : cette dernière a l'avantage non seulement de permettre une intégration aisée du traitement au niveau de la puce mais aussi de permettre la fabrication de prototypes à moindre coût. Cette dernière caractéristique semble déterminante vu que la majorité des concepts naissent dans des universités à des fins de recherches et de mises au point.

En conclusion

Les capteurs d'image à semi-conducteur (*Solid State*) ont fait des progrès extraordinaires depuis qu'ils ont vu le jour dans le premier appareil photographique en 1972. Les premiers progrès furent très lents; il existe, de nos jours, des alternatives intéressantes au CCD et au CMOS plus récent, la qualité ne cessant de croître alors que le coût ne cesse de diminuer.

Les capteurs « intelligents » venant de voir le jour remplissent des fonctions qui auraient requis, par le passé, des processus de traitement complexes hors-puce.

(020083-1)