

ÉLECTRONIQUE et MICRO-INFORMATIQUE

www.elektor.fr

ANALYSEUR LOGIQUE 20/40 MHz

à géométrie
variable



Dé électronique
pour JdR



Enceintes
actives

Relève du
SAA3049

Adaptateur
de niveau

La logique
configurable

Réception
VLF sur PC



M 01531 - 296 - F: 5,45 €



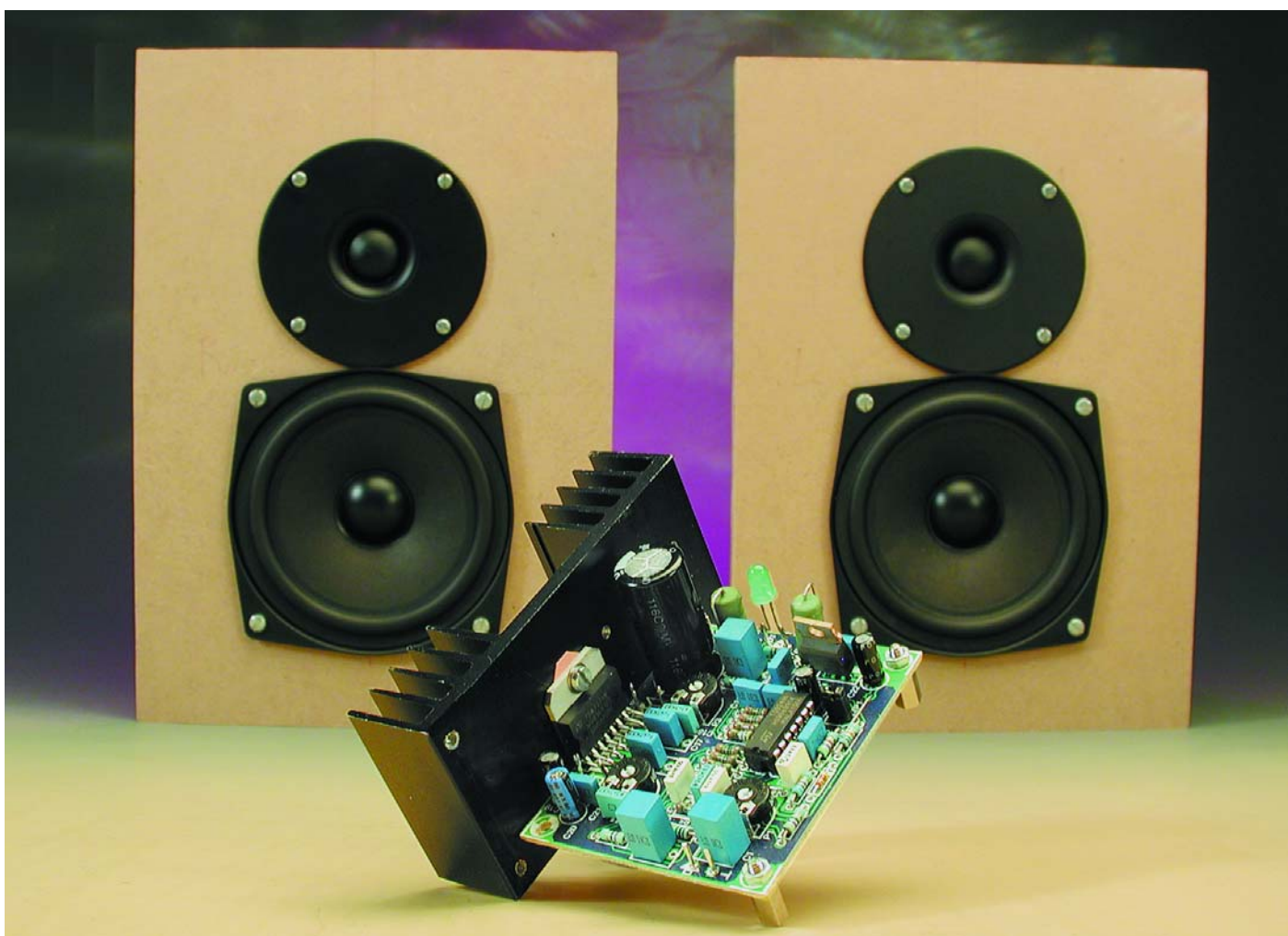
Enceintes actives

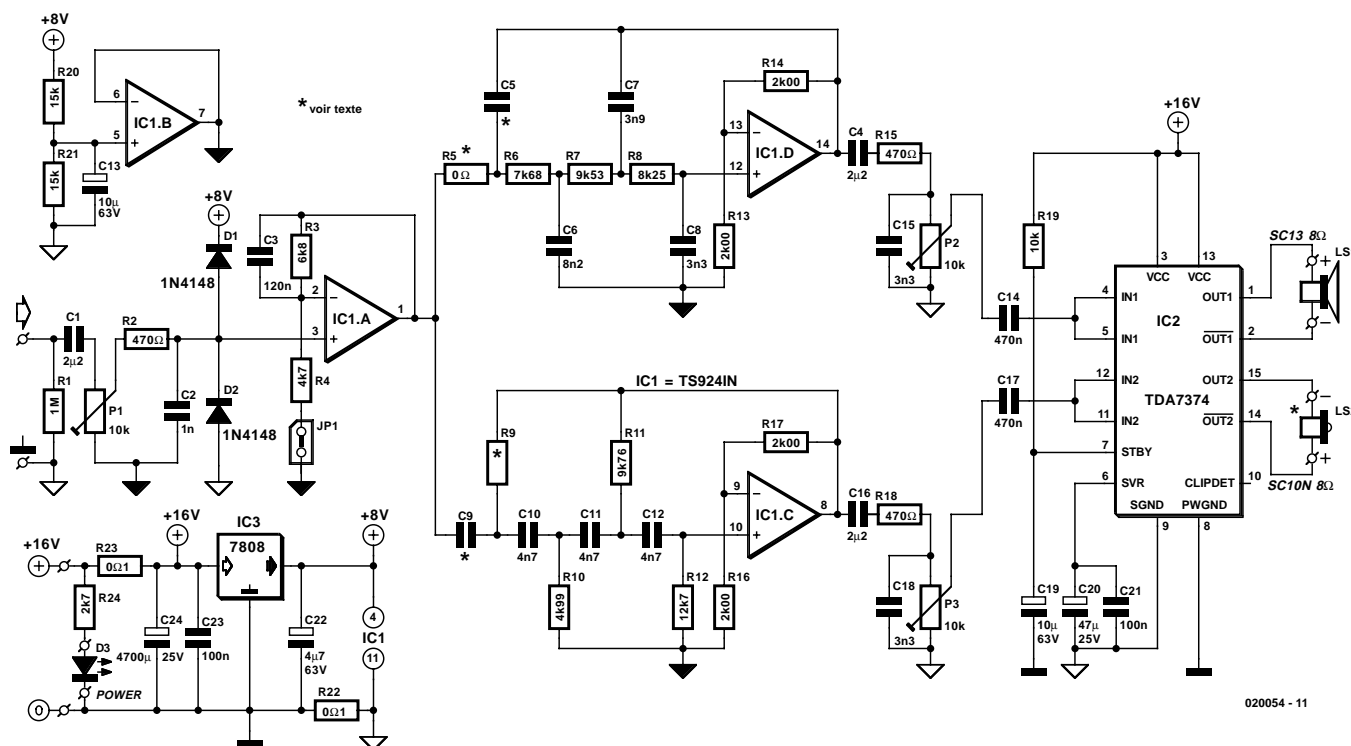
À usage multimédia

À la technique : Ton Giesberts

À la rédaction : Sjef van Rooij

Voici des enceintes actives à deux voies destinées, de prime abord, à voisiner avec un PC, mais elles vous raviront tout autant comme petite chaîne d'appoint, là où la perfection sonore n'est pas de rigueur. Et comme le filtre de séparation, vous pouvez le régler à votre goût, rien ne vous force à utiliser les haut-parleurs préconisés ni à respecter les dimensions de l'ébénisterie. Un caisson de grave actif est à l'étude et viendra sous peu les soutenir.





020054 - 11

Figure 1. Le schéma s'articule autour d'un amplificateur d'entrée, un filtre de séparation et un double amplificateur de puissance intégré.

La plupart de ces petites « boîtes à sons » pour PC nous ont tellement laissés sur notre faim, quand elles ne nous ont pas franchement dégoûtés, que nous avons décidé d'en concocter nous-mêmes. Nous voulions préserver la simplicité des moyens mis en œuvre autant que le portefeuille. On n'a pas besoin de la haute-fidélité de studio à côté de son PC, mais il est plus agréable d'écouter une reproduction correcte de ce qu'on a envie d'entendre, surtout s'il s'agit d'un CD ou d'un DVD.

C'est dans cette perspective que nous avons cherché tout d'abord une combinaison de haut-parleurs de grave et d'aigu compacte, d'un bon rapport qualité-prix. Il fallait en outre une protection magnétique, parce qu'à proximité immédiate de l'écran d'un moniteur ou d'un téléviseur, on doit veiller à ne pas déformer l'image reproduite. Finalement, pour notre prototype, nous avons déniché un tweeter à dôme de 25 mm et un médium-grave de 13 cm de la firme Visaton (le SC10N et le SC13), mais des marques comme Vifa ou Monacor, par exemple, proposent également d'excellents haut-parleurs blindés. C'est pourquoi nous insistons sur le fait que d'autres combinaisons

sont possibles, pour autant que l'on prenne en considération les recommandations du fabricant en définissant les dimensions du boîtier.

L'électronique nécessaire à « activer » les enceintes est particulièrement simple et générale. On y trouve un amplificateur opérationnel quadruple pour former un filtre de séparation à deux voies dont divers paramètres, la raideur des flancs, la courbe de réponse et la position des pôles, sont réglables dans certaines limites. Avec les haut-parleurs que nous avons utilisés, le filtre est un Butterworth du 3^{ème} ordre dont la fréquence de transition se situe à 4 kHz.

Pour l'amplificateur final, nous avons été séduits par un amplificateur intégré double en pont qui demande peu de composants externes et qui est capable de délivrer, sous une tension d'alimentation de 16 V, une puissance de 2 x 19 W sous 4 Ω ou 2 x 12 W sous 8 Ω. En regard des normes Hi-Fi, c'est peut-être un peu juste, mais avec des haut-parleurs de rendement moyen, on atteint une pression acoustique de quelque 100 dB, c'est déjà assez canon ! Que faut-il ajouter à l'ébauche de ce projet ? Précisément, une particularité attrayante de la méthode active

va nous permettre de nous affranchir d'un défaut de beaucoup de petites enceintes. La plupart d'entre elles, avec leurs quelques rares demi-litres de capacité, ont tendance à présenter une chute précoce de la courbe de réponse dans la partie basse. Nous avons appliqué ici une correction des graves, à commuter par cavalier, qui, de 1 000 Hz à 100 Hz relève la courbe jusqu'à un maximum de 6 dB. Notre prototype a manifestement tiré profit de cette correction et elle devrait apporter également un effet positif sur d'autres combinaisons de haut-parleurs similaires, avec un petit médium-grave.

L'électronique

Après un aussi long préambule, vous avez certainement l'envie d'aller parcourir le schéma. Il est reproduit à la **figure 1** et peut se ramener, en gros, à trois blocs.

Circuit d'entrée et alimentation

Explorons-le de gauche à droite, nous rencontrons une résistance de bouclage, un condensateur de couplage et le réglage de niveau P1 avant d'arriver au tampon d'entrée construit autour de IC1.A, lequel fait partie de l'amplificateur opérationnel quadruple TS924IN, un modèle dont la plage de sortie va d'un pôle d'alimentation à l'autre, mais surtout qui est capable de fournir un courant de sortie relativement grand : 80 mA.

La cellule R2/C2 forme un filtre passe-bas destiné à atténuer d'éventuels signaux parasites à haute fréquence. Le réseau C3/R3/R4 provoque le relèvement des basses fréquences (les graves) dont nous avons parlé, conditionné par la présence du cavalier JP1. Pour s'émanciper de l'alimentation symétrique traditionnelle, sans compromettre le traitement optimal du signal, IC1.B élabore une masse virtuelle. Le condensateur C13 découple le diviseur de tension R20/R21 qui positionne la masse virtuelle avec précision à la moitié de la tension d'alimentation. C'est ici que l'on apprécie l'ampleur du courant de sortie du TS924IN.

Si IC1 est doté de son propre stabilisateur de tension IC3, c'est pour s'assurer que l'étage d'entrée et le filtre soient bien protégés contre les parasites que la ligne d'alimentation pourrait véhiculer. Il faut, pour que ce régulateur de 6 V travaille convenablement, que la tension de l'alimentation du système actif s'élève au moins à 11 V. La résistance R22 a pour but de séparer le signal de masse et le zéro de l'alimentation si l'on se sert d'une source commune à deux ou plusieurs canaux. Mais on peut la remplacer par un pont de câblage pour une alimentation individuelle.

Filtre

Nous en arrivons au filtre de séparation. Le schéma indique clairement que l'étage d'entrée pilote deux sections de filtres, construits à l'aide des deux amplificateurs qu'il nous reste dans IC1. Le filtre passe-bas utilise IC1.D, tandis que IC1.C participe au filtre passe-haut. Au départ, il s'agit de configurations du 4^{ème} ordre, ce qui permet, sur la même platine, par simple suppression de composants, de façonner éventuellement des filtres plus simples.

À votre intention, nous en avons calculé un certain nombre de variantes, répertoriées dans le **tableau 1**. Vous y trouverez les valeurs des composants d'un Butterworth du 3^{ème} ordre et d'un Linkwitz-Riley d'ordre 4 pour des fréquences de pôle de 4 kHz, 2,5 kHz et 1 kHz. Pour notre prototype, équipé de haut-parleurs Visaton, c'est une combinaison de Butterworth d'ordre 3, centrés sur une fréquence de séparation de 4 kHz, qui nous a semblé la plus adéquate. Les valeurs de composants donnés dans le schéma se réfèrent donc à ce type de filtre. Les mesures, en configuration Linkwitz-Riley du 4^{ème} ordre et séparation à 4 kHz, nous ont fait apparaître des problèmes de rayonnement sous un certain angle, à cause d'un trop grand déphasage. La variante Butterworth du 3^{ème} ordre en souffrait manifestement

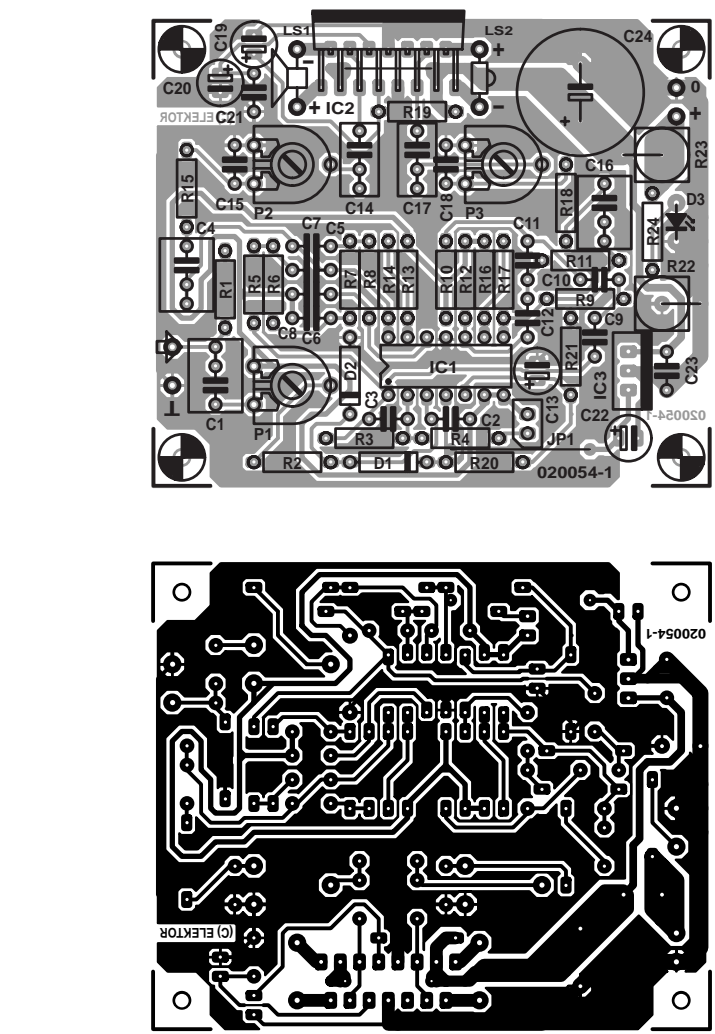


Figure 2. Difficile de s'imaginer que cette petite platine puisse accueillir, outre les filtres de séparation et le tampon d'entrée, deux amplificateurs de sortie de près de 20 W chacun !

moins. Nous donnons donc les valeurs pour le Linkwitz-Riley à l'intention de ceux qui souhaitent expérimenter. Il faut en outre faire attention à la polarité des haut-parleurs avec le filtre de Linkwitz-Riley, parce que les tweeters doivent être inversés. La platine renseigne la polarité pour un Butterworth d'ordre 3, dans lequel le tweeter est en opposition de phase par rapport au haut-parleur de graves ! Autre particularité à signaler : le point de raccordement du Butterworth se situe à -3 dB, alors que dans le Linkwitz-Riley, les deux courbes se croisent à -6 dB.

Amplificateur final

Les signaux de sortie des filtres vont, à travers les potentiomètres de réglage (P2 et P3) aux amplificateurs

de sortie. Les potentiomètres servent à compenser la différence de rendement entre haut-parleurs d'aigu et de grave. De très nombreux tweeters à dôme produisent, pour un même signal, un niveau de 3 dB supérieur à celui d'un petit médium/grave. Ceux que nous avons utilisés présentaient le même rendement, si bien que nous avons pu régler P2 et P3 au maximum. L'étage final utilise un double amplificateur intégré du type TDA7374B, normalement destiné à l'audio automobile, mais qui convient aussi bien à ce genre d'application. Cette puce ne nécessite que remarquablement peu de composants externes, pas de cellule de Boucherot, pas de condensateur électrolytique de sortie et sa protection contre l'excès de température ou le court-circuit est intégrée

Liste des composants

Résistances :

R1 = 1 M Ω
 R2,R15,R18 = 470 Ω
 R3 = 6k Ω
 R4 = 4k Ω
 R5 = 0 Ω
 R6 = 7k Ω
 R7 = 9k Ω
 R8 = 8k Ω
 R9 = ouvert *
 R10 = 4k Ω
 R11 = 9k Ω
 R12 = 12k Ω
 R13,R14,R16,R17 = 2k Ω
 R19 = 10 k Ω
 R20,R21 = 15 k Ω
 R22,R23 = 0 Ω /5W
 R24 = 2k Ω
 P1,P2,P3 = ajustable 10 k Ω

Condensateurs :

C1,C4,C16 = 2 μ F2 MKT au pas de 5/7,5 mm
 C2 = 1 nF au pas de 5 mm
 C3 = 120 nF au pas de 5 mm
 C5 = ouvert *
 C6 = 8nF2 au pas de 5 mm
 C7 = 3nF9 au pas de 5 mm
 C8,C15,C18 = 3nF3 au pas de 5 mm

C9 = pont de câblage *
 C10...C12 = 4nF7 au pas de 5 mm
 C13,C19 = 10 μ F/63 V radial
 C14,C17 = 470 nF
 C20 = 47 μ F/25 V radial
 C21,C23 = 100 nF au pas de 5 mm
 C22 = 4 μ F/63 V radial
 C24 = 4 700 μ F/25 V radial, au pas de 7,5 mm diamètre 17 mm max.

Semi-conducteurs :

D1,D2 = 1N4148
 D3 = LED à haut rendement verte
 IC1 = TS924IN (STMicroelectronic, entre autres chez Farnell)
 IC2 = TDA7374B (STMicroelectronic)
 IC3 = 7808

Divers :

JP1 = embase autosécable à 2 contacts + cavalier de court-circuit
 LS1 = SC13 8 Ω Visaton (Conrad)
 LS2 = SC10N 8 Ω Visaton (Conrad)
 Radiateur pour IC2: 3 K/W
 coffret: MDF 12 mm cf. figure 4
 matériau d'amortissement (BAF-wadding)

*) cf. texte et tableau I

bli de quelque 50 dB.

La platine

La **figure 2** reproduit la platine conçue pour l'électronique des enceintes actives.

Rien de particulier à signaler, le tracé est clair, les points de raccordement y sont clairement indiqués. Pour mieux se situer, mentionnons que les broches d'entrée se trouvent à gauche, en dessous, diagonalement à l'opposé des celles de l'alimentation, tandis qu'un peu plus bas, il y a la LED D3, l'indicateur de fonctionnement. Les bornes de raccordement du woofer et du tweeter (LS1 et LS2), vous les trouvez de part et d'autre de IC2.

Les ponts de câblage méritent un mot d'explication. Il y en a deux, l'un tout à côté de R20 et l'autre sous les broches de IC2. Ce dernier peut aussi se placer sous le circuit imprimé, mais dans un cas comme dans l'autre, il faut utiliser du fil isolé. IC2 a été placé volontairement en bordure de la platine, pour la facilité de lui attacher une plaque de refroidissement (3 K/W) ; n'oubliez pas d'intercaler un isolateur !

Une fois terminées la construction et la vérification de la platine, il est temps de songer à la caser. Différentes possibilités s'offrent à

d'origine. Des caractéristiques qui la propulsent en tête dans la recherche de compacité ultime. Le schéma en témoigne sans équivoque.

Nous avons déjà évoqué la puissance produite. Avec des haut-parleurs de 8 Ω , la charge reste limitée et un petit radiateur de 3 K/W suffit à assurer le refroidissement de IC2. La cellule R19 et C19 a pour tâche d'étouffer le transitoire à la mise en service, parce qu'il subsiste toujours une petite dérive en sortie de l'amplificateur. Les réseaux R15/C15 et R18/C18 limitent la bande passante vers les amplificateurs de sortie pour encore contribuer à les débarrasser des perturbations à haute fréquence. En principe, on aurait dû trouver ces réseaux derrière les potentiomètres, mais malheureusement, de là, ils influencent la bande passante de manière audible. Le condensateur C20 découple le diviseur de tension interne qui règle la polarisation de divers étages au milieu de la tension d'alimentation et détermine pour une bonne part l'atténuation de la tension de ronflement, puisque le 100 Hz y est affai-

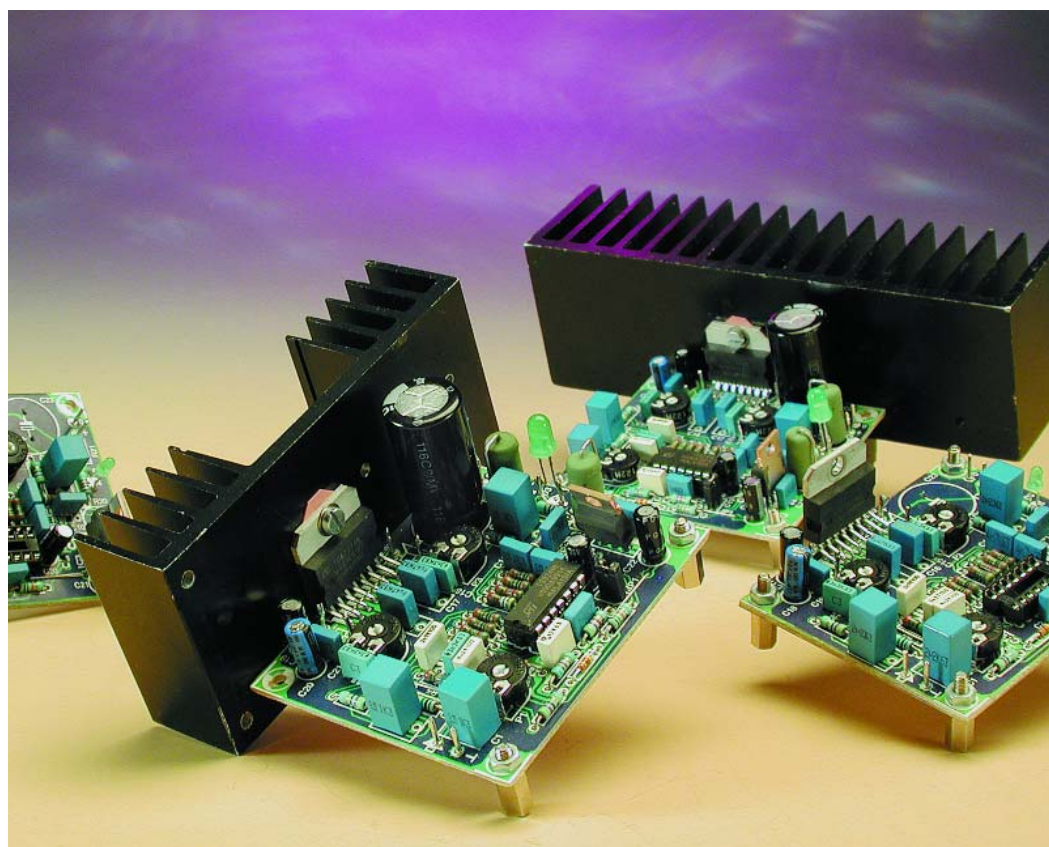


Figure 3. Avant d'arriver à la version définitive de la platine, il nous faut toujours réaliser quelques brouillons...

vous : dans l'enceinte, dans un boîtier séparé ou combinée au caisson de grave en préparation. Décider de l'héberger à part se défend, parce qu'un réglage de tonalité pour le système est aussi dans les cartons. Mais chacun peut s'en fabriquer un.

Pour l'alimenter, on peut partir du groupe classique transformateur, pont redresseur et condensateur électrolytique. Par canal, il faut un transformateur de 12 V/15 A et un condensateur de lissage de 4 700 µF/25 V, le double en stéréo. L'avantage, si l'on choisit une alimentation stabilisée, c'est de pouvoir monter de 16 V à 18 V, la tension maximale, et de gagner encore en puissance disponible. C'est ce à quoi travaillent les laboratoires d'Elektor en ce moment, donc vous avez peut-être intérêt à patienter encore un peu avant d'en attaquer la construction.

Menuiserie

Comme chacun sait, la capacité et les dimensions d'une enceinte dépendent des caractéristiques du haut-parleur de grave. Nous avons inséré le SC13 de Visaton dans un cofret fermé de 4 litres. Grande simplicité pour le travail de menuiserie, puisqu'il ne demande que l'assemblage de six panneaux. Peut-être un défi à relever pour qui ne s'y est jamais « frotté », mais une tâche loin d'être insurmontable, surtout quand le bois est mis à dimensions : il n'y a plus qu'à se munir de quelques bons serre-joints pour maintenir en place les pièces collées et l'affaire sera réglée.

	Butterworth 3 ^{ème} ordre			Linkwitz-Riley 4 ^{ème} ordre		
	1 kHz	2,5 kHz	4 kHz	1 kHz	2,5 kHz	4 kHz
R5	pont	pont	pont	6k34	6k34	6k98
R6	8k06	6k65	7k68	13k3	9k31	10k7
R7	8k25	8k45	9k53	6k49	7k68	8k25
R8	6k81	8k06	8k25	9k31	7k50	8k45
C5	ouvert	ouvert	ouvert	22 n	10 n	5n6
C6	33 n	15 n	8n2	39 n	18 n	10 n
C7	18 n	6n8	3n9	18 n	6n8	3n9
C8	15 n	5n6	3n3	8n2	3n9	2n2
R9	ouvert	ouvert	ouvert	7k50	6k49	7k15
R10	5k23	4k53	4k99	3k83	3k32	3k65
R11	10k2	8k87	9k76	11k0	9k53	10k5
R12	13k0	11k5	12k7	19k6	16k9	18k7
C9	pont	pont	pont	18 n	8n2	4n7
C10	18 n	8n2	4n7	18 n	8n2	4n7
C11	18 n	8n2	4n7	18 n	8n2	4n7
C12	18 n	8n2	4n7	18 n	8n2	4n7

Tableau 1. Les valeurs des composants de filtre à différentes fréquences. Pour un Butterworth d'ordre 3, C5 et R9 n'existent pas, quant à R5 et C9, ils sont remplacés par des ponts de câblage.

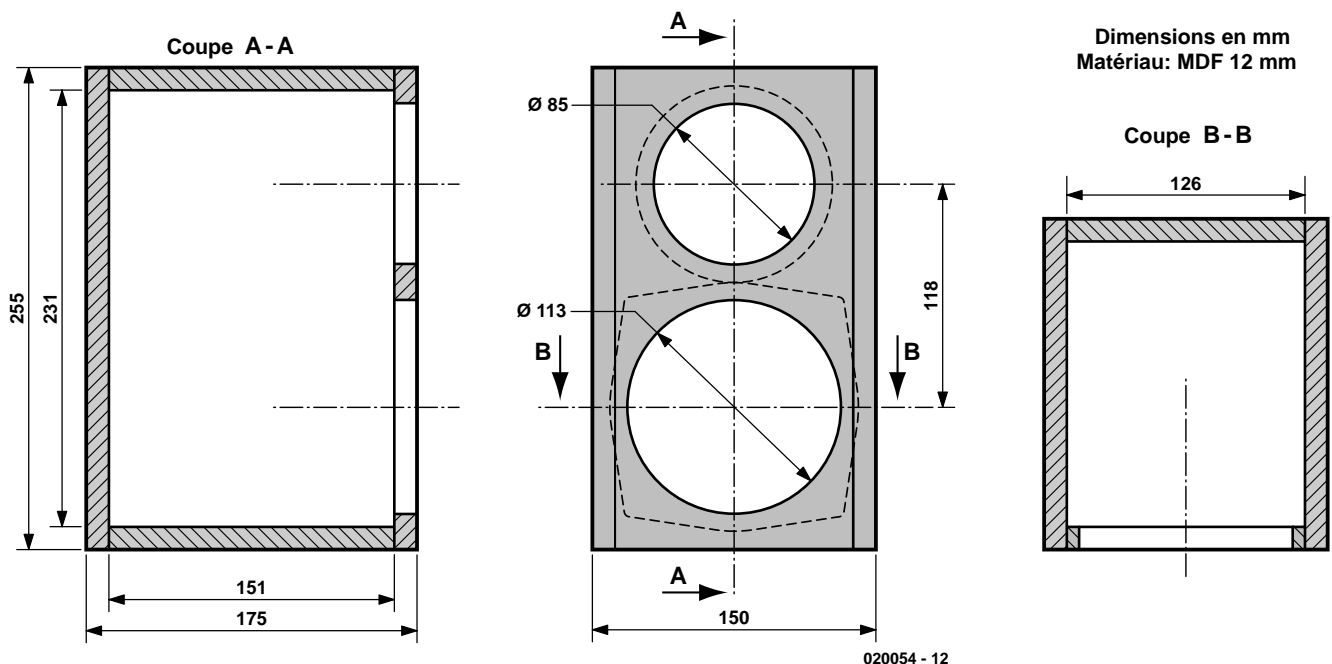


Figure 4. Plan de construction de l'enceinte avec toutes les cotes. Elles supposent l'emploi de panneaux de médium de 12 mm.

Aucune cloison intérieure, aucune ouïe à prévoir, la seule difficulté réside dans la découpe des trous pour les haut-parleurs en face avant. Eu égard aux petites dimensions, du bois de forte épaisseur n'est pas indispensable –mais ce ne serait évidemment pas un inconvénient– aussi avons-nous basé les cotes du plan d'assemblage, à la **figure 4**, sur un panneau de médium (MDF) de 12 mm. Pour l'amortissement acoustique nécessaire, il suffira de remplir le coffret d'ouate BAF, de la laine de polyester. Les bornes de raccordement traversent la face arrière. Est-il nécessaire de préciser que, si vous décidez d'implanter l'électronique

dans l'enceinte, il faudra que le radiateur couche dehors ? Pour éviter tout malentendu, soyez assuré que le dessin de la figure 4 n'est absolument pas intangible. Même la forme du boîtier est sujette à interprétation, pour autant que la capacité de 4 litres, environ, soit respectée. La photo du titre illustre d'ailleurs que nous n'avons pas hésité à nous écarter du rapport traditionnel de la « boîte à chaussures ». Poussés par un besoin d'originalité, nous avons, pour l'occasion, décidé d'expérimenter dans la structure triangulaire, le pentaèdre. L'avantage incontestable, c'est l'absence d'ondes stationnaires entre

les parois, mais le pépin vient de ce que trois des cinq panneaux doivent présenter des biseaux sous des angles peu habituels : 30 ° et 73,9 °. Seuls le plancher et la façade jouissent de la « normalité » des bords. Le fond est un triangle équilatéral et le dessus est en pente vers l'arrière d'un angle de 30 °. Qui est intéressé par la construction de cette « préparation » spéciale peut en télécharger, de notre site, le plan d'assemblage. On trouvera sur Internet un dessin coté additionnel de cet enceinte. La finition des coffrets est laissée au goût de chacun. À l'ancienne est une des options, mais on peut revêtir les parois de vinyle ou d'un autre matériau. La laque et la peinture au pistolet sont en vogue, moyennant l'aide éventuelle d'un carrossier.

(020054)

Spécifications

(sous 16 V d'alimentation)

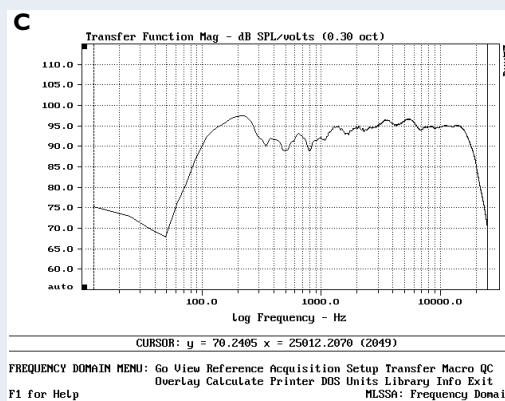
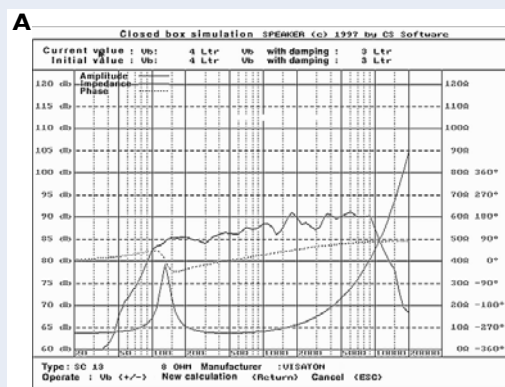
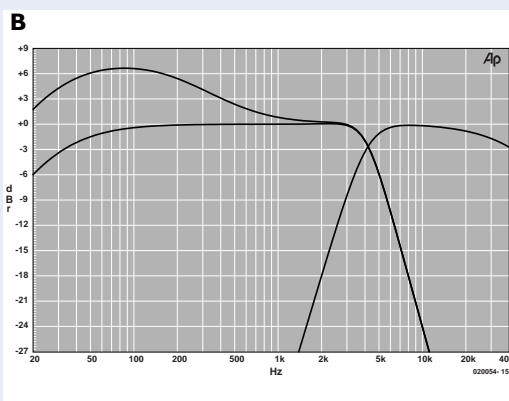
Impédance d'entrée	10 kΩ
Sensibilité (12 W/8 Ω, JPI ouvert, P1, P2, P3 max.)	270 mV
Distorsion harmonique totale + bruit (1 W/8 Ω, 1 kHz)	0,013 % (B = 80 kHz)
Bande passante woofer + ampli (P2 max. JPI ouvert)	32 Hz à 4 kHz
Bande passante woofer + ampli (P2 à moitié, JPI ouvert)	25 Hz à 4 kHz
Bande passante tweeter + amplificateur	4 kHz à 45 kHz
Puissance de sortie par ampli (DHT + Br = 0,5 %)	12 W (8 Ω)
	19 W (4 Ω)
Courant de repos (sans charge)	0,17 A
Bande passante amplificateur + enceinte (-3 dB)	100 Hz à 18 kHz

Outre cette liste de chiffres, nous vous avons encore réservé trois courbes. La première **(A)** retrace une simulation de la caractéristique de fréquence du woofer SC-13. Pas de forte pointe ni de creux, mais une assez forte pente vers les basses fréquences : à 100 Hz, l'amplitude a diminué de près de 8 dB par rapport à 2 kHz. Un rattrapage n'est donc pas un luxe superflu.

La deuxième courbe **(B)** trace la mesure de la caractéristique d'amplitude des filtres et de la correction des basses. On y remarque que le point de transition n'est pas tout à fait à -3 dB, ni exactement à 4 kHz. C'est qu'il y a la tolérance sur les condensateurs des filtres. Mais en pratique, ces écarts sont rigoureusement sans conséquence.

La courbe **C** montre enfin la mesure de la caractéristique d'amplitude de l'enceinte, raccordée à ses amplificateurs. On y voit clairement que la correction de fréquence aurait pu être un peu plus poussée.

Mais placée sur un bureau ou près d'un mur, les graves sont encore accentuées et la courbe se redresse légèrement. La bosse au voisinage de 200 Hz provient de la pièce dans laquelle les mesures ont été effectuées et dépend surtout du positionnement de l'enceinte.

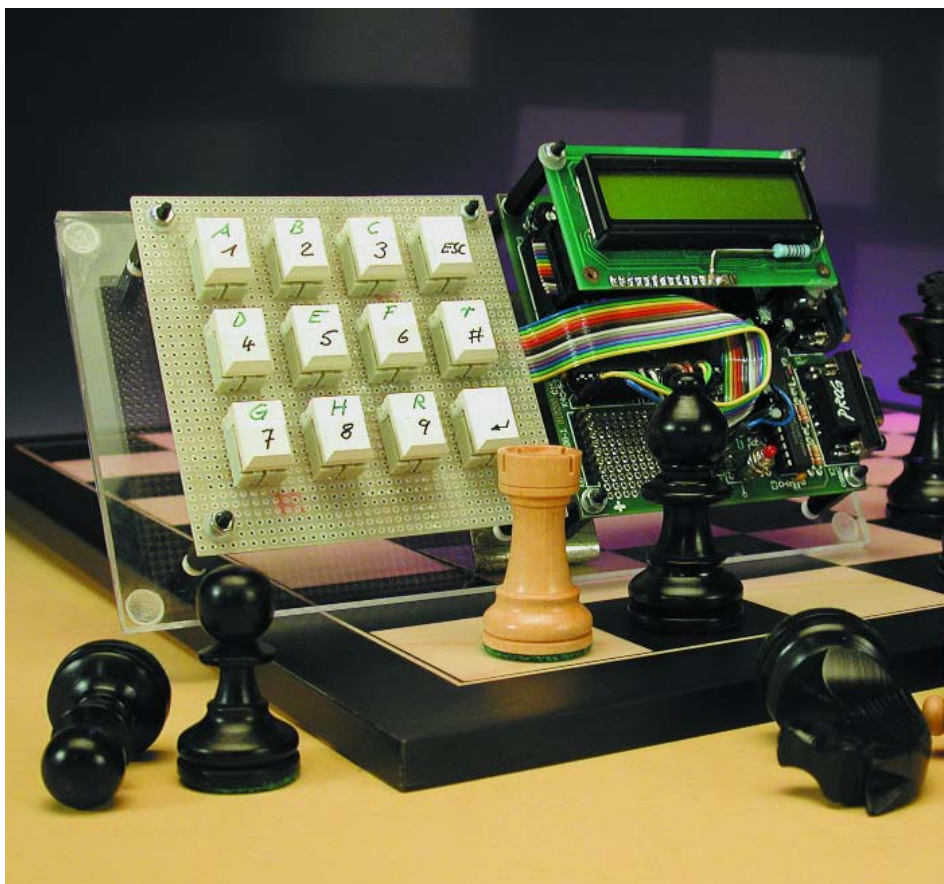


La carte à 89S8252 Flash en jeu d'échec

Deep Evelyn – le vainqueur de notre Concours « Flash »

Pedram Azad et Tilo Gockel

Si le sujet « Échecs sur gros ordinateurs » devait très bientôt ne plus être d'actualité (la Machine devient invincible), l'intérêt pour la programmation par ses propres soins d'un système à microcontrôleur pour le faire jouer aux échecs ne cesse de croître en raison du défi que représente l'application de règles plus strictes en raison de la faible taille du système adopté, la carte à 89S8252 Flash d'Elektor.



Les automates ou ordinateurs capables de jouer aux échecs ont toujours fasciné les hommes, en particulier depuis l'automate joueur d'échecs de Maelzel. Bon nombre de nos semblables se sont essayés au défi de réaliser quelque chose qui les dépasse, certains d'entre eux ayant pris à cette occasion des chemins fort intéressants. Ainsi, Alan Turing a implémenté, avant même que n'existent les premiers ordinateurs, un programme d'échecs à base de cartes perforées, faisant lui-même office de processeur pour leur traitement.

Depuis lors les choses ont beaucoup évolué au point que rares sont encore les cerveaux humains capables de battre l'Ordinateur, de sorte que l'on peut pratiquement affirmer que le sujet « Echecs sur ordinateur » a perdu tout intérêt [4, 13].

Par le biais du projet DEEP EVELYN nous avons tenté de rendre une certaine vigueur à l'intérêt de créer soi-même un programme de jeu d'échecs sur ordinateur en rendant

les règles encore plus sévères : programme dans 8 (12) Koctets de Flash, sur un microcontrôleur 8 bits doté d'une pile de 256 octets maximum et le tout programmé en C !

Le logiciel

Une fois née l'idée d'implémenter un programme de jeu d'échecs sur le microcontrôleur d'Atmel, il ne fut pas très difficile de découvrir sur Internet un noyau de programme en C chargé de générer les coups, d'évaluer les positions et de prendre en compte la récursivité (processus alpha/bêta) [1,2,3].

Des tests effectués à l'aide de différents compilateurs C (Rigel, Wickenhäuser) ne furent guère concluants, pire encore furent un raté complet, les compilateurs étant incapables de traiter la syntaxe (bien qu'elle ait été réduite à sa plus simple expression), se prirent les pieds dans le tapis de la récursivité voire créaient tout simplement un code de taille trop importante. Le compilateur OpenSource sdcc, dont Elektor a parlé dans un de ses derniers magazines (*small device c compiler*, [5,8]) parut enfin le mieux adapté. Il ne restait plus maintenant qu'à adapter le code-source à notre disposition au cahier des charges que nous avons établi. Voici les modifications les plus importantes à effectuer :

1. Se passer de bibliothèques standard
2. Réduction du code programme
3. Adaptation au modèle-mémoire de l'ATMEL
4. Optimisation de la vitesse

Point 1. : au cours de la première étape il a fallu supprimer toutes les fonctions `#include` (`stdio`, `stdlib`, `string`) et remplacer les instructions absentes de ce fait (`strcmp`, `abs`...) par des implémentations de fabrication maison. Ceci s'est traduit par la suppression d'un certain nombre d'instructions d'Entrées/Sorties et doter les positions concernées d'un commentaire en vu de faciliter, ultérieurement, l'insertion, à ces endroits, de nouvelles routines d'En-

trées/Sorties.

La première conversion effectuée à l'aide de sdcc produisit un code-programme d'un peu plus de 16 Koctets. En tenant compte de la taille des routines d'Entrées/Sorties encore à implémenter il nous fallait réduire la taille de ce code d'au moins 9 Koctets. Pour éviter d'être mis « échec et mat » avant même d'avoir commencé, nous avons décidé de choisir une approche moins prétentieuse. Comme nous le mentionnons dans le paragraphe « Matériel » un peu plus loin, le AT89S53, compatible broche à broche avec le AT89S8252, possède une mémoire de programme en Flash un peu plus importante, 12 au lieu de 8 Koctets. Ce choix nous offrait la possibilité de n'avoir, dans le pire des cas, à réduire la taille de notre code-programme de 5 Koctets seulement.

Point 2. : les étapes de transformation du code-source en vue de réduire la taille du code-programme sont les suivantes :

- Extraction de constantes
- Mise en facteur de codes similaires Remplacer, partout où cela était possible, des types de donnée int par byte (char)
- L'extraction de constantes est, relativement, simple. Si, à l'intérieur d'une fonction, on a calcul répété d'une valeur donnée, il suffit de la calculer une première fois et de lui associer une variable. On fait ensuite appel à cette variable au lieu de recalculer la valeur. Cette transformation se traduit à la fois par une réduction de la taille du code et par une augmentation de l'efficacité de son efficacité. IL faut cependant signaler ici qu'il est aisé d'introduire des erreurs en cas d'extraction de constantes qui en fait n'en sont pas. Il est partant recommandé de toujours vérifier que la transformation envisagée est licite et ensuite qu'elle n'a pas de conséquence désastreuse sur le déroulement du programme.
- La mise en facteur de codes similaires est déjà sensiblement plus complexe. Si l'on fait abstraction de la situation triviale où 2 morceaux de code-source sont parfaitement identiques, cette étape implique intuition et créativité. Si 2

Extrait de code I

```
//ancien
void foo()
{
    char i;
    for (i = 0; i < 8; i++)
    {
        // gemeinsam ...
    }

    for (i = 0; i < 4; i++)
    {
        // gemeinsam ...
        // extra ...
    }
}

//nouveau
void help(char n)
{
    char i;
    for (i = 0; i < n; i++)
    {
        // gemeinsam ...
        if (n == 4) { // extra ... }
    }
}

void foo()
{
    help(8);
    help(4);
}
```

morceaux de code-source sont similaires, il est souvent possible, par l'adjonction de variables et de fonctions if, de les combiner en une seule fonction (cf. extrait de code 1). Cette transformation permet, si tant est qu'elle soit utilisée à bon escient, une réduction sensible du code-programme, mais introduit cependant un certain ralentissement de l'exécution. Le double appel de la fonction, y compris l'exécution implicite d'opérations push&pop pour l'adresse de saut retour, la sauvegarde d'éventuels paramètres de fonction et la sécurisation du contexte coûte à lui seul déjà un certain temps. À cela s'ajoute l'ajout de code additionnel servant à effectuer la différenciation, telles qu'instructions if.

- La substitution du type de donnée char à la place de int apporte, si tant est qu'il soit licite, une amélioration importante à différents points de vue : taille de programme, utilisation de la mémoire et temps d'exécution. Le sdcc interprète char comme un type de donnée 8 bits et int comme étant un 16 bits. Comme le AT89S53 possède une unité

Extrait de code 2

```
//ancien...

int value;
int foo(char a)
{
    char local;
    if (a == 5)
        return a;
    // code1 ...
    value = foo(a + 1);
    // code2 ...
    return value;
}

//nouveau...

xdata int value;
// pour contenu actuel de la
pile
xdata char a;
xdata int local;
xdata char ret;
// correspond au registre de
valeur de saut de retour
xdata char retval;
int foo()
{
begin:
    // Chercher paramètre,
adresse de saut de retour
    a = stack[stackp].a;
    ret = stack[stackp].ret;
    if (a == 5)
    {
        retval = a;
        if (ret)
            goto ret;
        return retval;
    }
    // code1 ...
    // Actualiser le contenu de
la pile (sauvegarder le
contexte)
    stack[stackp].a = a;
    stack[stackp].local = local;
    // push (Paramètre, Adresse
de saut de retour)
    stack[++stackp].a = a + 1;
    stack[stackp].ret = 1;
    goto begin;
ret:
    // pop (retransférer le
contexte)
    a = stack[--stackp].a;
    local = stack[stackp].local;
    ret = stack[stackp].ret;
    // Chercher valeur de retour
    value = retval;
    // code2 ...
    retval = value;
    if (ret)
        goto ret;
    return retval;
}
```

à-dire surtout également les matrices du plan de jeu, les coups, etc. dans la mémoire externe de 32 Koctets. On ordonne à sdcc, par le biais du mot-clé « xdata », placé devant une déclaration, de placer une variable dans la mémoire externe.

Une fois que nous étions arrivés là, nous avons la possibilité de laisser l'ordinateur jouer aux échecs contre lui-même et de suivre, par le biais de l'interface sérielle (signalons au passage que le téléchargement se faisait à l'aide de l'outil AtmelISP [6]. Le premier coup, un sanglant « a8a9 » (!!) donna le signal d'une recherche d'erreur longue et délicate.

Après avoir ramené l'ensemble du programme de jeu d'échec à quelque 5 lignes qui ne faisaient rien de plus que de remplir, à l'aide d'une boucle for, une matrice pour la relire ensuite, il apparût à l'évidence qu'il se passait quelque chose d'anormal au niveau du compilateur sdcc. À bout de nerfs nous avons interdit, par le biais de paramètres introduits dans la ligne d'instruction de lancement du programme, toute optimisation induite par le compilateur, opération qui fut couronnée de succès. Le lancement du compilateur sans optimisation se fait par la ligne de commande suivante :

```
sdcc -I c:\sdcc\include -L
c:\sdcc\lib\large chess.c --model-
large --nolooptreverse --nogcse --
noinduction --noinvariant --nojt-
bound
```

En tout état de cause, le coup initial de notre joueur d'échecs devint un « a2a3 », mouvement qui présentait un minimum de similitude avec le « e2e4 » qui était le premier coup du même programme tournant sous Windows...

Bien que la mémoire interne de 256 octets ait largement du suffire pour y placer les variables locales restantes et la pile, il apparut que la quasi-totalité des variables non précédées de xdata étaient des sources potentielles de problèmes. Notre expérience aidant, nous n'en étions plus à nous poser de question quant à l'origine du problème et avons passé le stade de la recherche systématique. Nous avons opté pour la solution la plus radicale, contourner, sans exception, la mémoire interne.

logique arithmétique (ALU) de 8 bits, une multiplication de 2 variables de type int sera découpée en plusieurs multiplications et additions sur 8 bits.

En règle générale, il n'est pas possible d'effectuer une transformation de int vers char sans autre forme de procès en raison des pertes d'information importantes qu'elle implique (passage de 65 536 à 256). Il devient cependant possible, finalement, de remplacer toutes les apparitions du type de données int par char, exception faite de la fonction de traitement. Au début les #defines servaient à la fois à la distinction des diffé-

rentes pièces ainsi qu'à leur évaluation. Une décomposition de l'identificateur ID et de la valeur de chacune des pièces en #defines distincts a permis de réduire la matrice du plan de jeu au type de donnée char, de sorte que l'on pouvait, exception faite de la fonction d'évaluation, utiliser les opérations 8 bits bien plus efficaces.

Point 3. : comme le AT89S53 ne dispose que de 256 octets de mémoire interne il faut commencer par placer toutes les variables globales, c'est-

Mode d'emploi

Entrée d'un coup après y avoir été invité, **e2e4** par exemple : touche **[E/]**, touche **[2]**, touche **[E/]**, touche **[4]**

Correction suite, par exemple, à une entrée erronée, suppression d'une ligne complète : **[ESC]**

Confirmation ou envoi de l'information saisie respectivement : **RETURN** et **[<-]**

Petit roc : touche **[#]** (rebaptisé depuis **[r/]**)

Grand roc : touche **[I/]** (rebaptisé depuis **[R/]**)

Choix du coup par l'ordinateur (le contrôleur calcule le prochain coup) : **RETURN** ou **[<-]** sans entrée

Nouvelle partie : actionner le bouton-poussoir de Reset de la carte à 89S8252 Flash

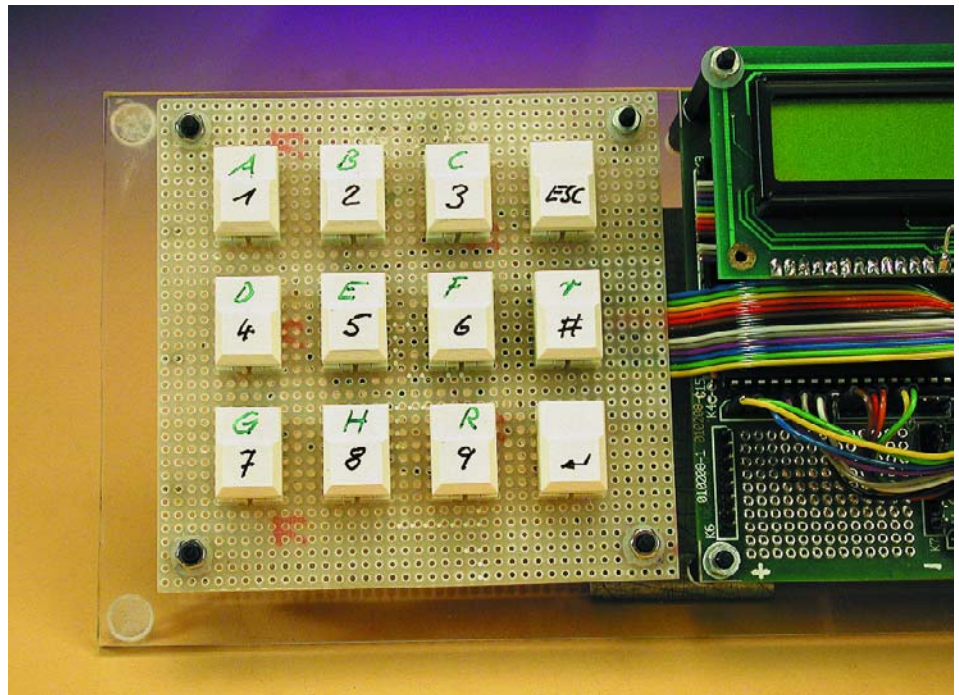
Voici le détail de ce qu'implique cette décision:

- Remplacement des paramètres de fonction par des variables globales
- Remplacement des variables locales par des variables globales
- Implémentation d'une pile de déroulement chronologique

La fonction alpha/bêta est la seule fonction récursive que comporte le programme d'échecs. Il faut, pour la dite fonction, placer sur la pile de déroulement chronologique tous les paramètres fonctionnels, les variables locales et son adresse de saut de retour dans le cas d'un appel récursif, pour ensuite, lors du saut de retour, en effectuer la lecture et descendre tous ces éléments de la pile. Ces différentes fonctions normalement effectuées à l'arrière plan par le langage de programmation doivent cette fois être programmés explicitement.

La pile prend la forme d'une matrice (array) Struct qui englobe toutes les variables évoquées plus haut. Comme, en C, les étiquettes de saut (*label*) ne sont pas des types et qu'il n'est partant pas possible de leur attribuer une variable, il a fallu réaliser l'adresse de saut de retour indirectement par le biais d'un identificateur (*ID*) et d'une interrogation if. Nous aurions sans doute pu envisager une solution qui aurait été plus efficace en faisant appel à l'assembleur en ligne (*Inline-Assembler*). L'extrait de code 2 illustre la transformation décrite.

Point 4. : dans le programme d'échecs, la différenciation de la couleur des pièces se fait par le signe de l'identificateur : les ID positifs concernent les pièces blanches, les ID négatifs les pièces noires. Dans la fonction d'évaluation nous avons pratiquement doublé l'efficacité en optant une évaluation différenciée pour l'algorithme de traitement d'une pièce blanche et d'une pièce noire sous la forme d'un embranchement double plutôt que d'opter pour de continuelles multiplications par une variable (1 ou -1 selon le cas) à l'intérieur d'un bloc unique. Cela a permis de supprimer tant de multiplications que le code de programme est devenu plus compact et ce en dépit de près d'un doublement du



nombre de lignes de programme. L'approche consistant à compiler le texte-source par le biais de `#defines` (`#ifdef WIN32 ... #endif`) et de `#defines` fictives (*dummy*) même sous Windows en tant que programme console s'est avérée très pratique. Il est possible ainsi de sauvegarder la fonction de base à tout moment voire de la déboguer confortablement sous Windows/Visual C. Le texte-source reste toujours utilisable sans le moindre problème sous Windows.

La possibilité de sortie par le biais de l'interface sériele s'est avérée d'un grand secours tout au long du développement. `sdcc` fournit un petit programme de démonstration (*hi.c*) dont il est possible d'extraire facilement les routines nécessaires. Une autre possibilité, plus confortable, est offerte par le programme en free-ware `PaulMon2`, mais nous n'avons malheureusement pas eu le temps, pendant le peu de temps (10 soirées au total qui ont duré...) dont nous disposons, de le tester.

Le matériel

C'est à dessein que nous nous sommes passés, avec ce projet, d'Entrées/Sorties les plus exotiques les unes que les autres, ayant porté l'aspect sur la programmation. Nous avons quand même effectué

quelques modifications sur le matériel existant, la carte à 89S8252 Flash décrite dans le n° 282 d'Elektor (décembre 2001) :

Le régulateur de tension fut remplacé par son équivalent faibles pertes (un `LT1086CT-5`, attention le brochage est différent : Masse/Out/In au lieu de In/Masse/Out; sinon `LM2940`, entre autres) et abaissé la tension d'entrée vu que le 7805 chauffait bien trop en raison de la chute élevée de près de 3 V lorsque le rétro-éclairage (avec résistance de limitation de 10 Ω) de l'affichage est en fonction.

L'organe d'entrée prend la forme d'un clavier constitué de 12 touches `Digitast` réalisé sur platine d'expérimentation à pastilles. L'un des contacts de chacune des touches est relié à la masse, ce qui explique qu'il faille 12 broches de port à ce clavier. Il n'était pas nécessaire ici d'économiser au niveau des broches de ports en optant pour une approche en matrice du clavier [9,11,12].

Il est facile de dériver du fichier de texte-source *userint.c* (fonction `AskPlayerMove()`) l'ordre d'attribution des broches de port. L'anti-rebond des touches prend la forme d'une simple temporisation de 50 ms, durée que l'on pourra modifier à loisir si besoin s'en fait sentir. La disposition des touches correspond à celle d'un clavier de téléphone, leur fonction et leur mode d'emploi est presque évident. Le code-source étant en allemand, ceux d'entre nos lecteurs qui ne maîtrisent pas la langue de Goethe auront inévitablement de gros problèmes de compréhension. Le programme n'effectue aucune vérification au niveau de la correction (légalité) des mou-

Sources :

- [1] www.trinimon.de/Chess
Une introduction intéressante à la programmation de jeu d'échecs avec exemples et téléchargements (allemand)
- [2] www.trinimon.de/Chess/ChessGame/Develop/chessCsrc.htm
Code-source en C pour mini-programme d'échecs (allemand)
- [3] www.artilect.co.uk/lego/default.asp?page=Chess
"The Chess Robot Project", Echecs sur Mindstorms de Lego, Téléchargement (anglais)
- [4] www.spiegel.de/spiegel/0,1518,216103,00.html
"Schach, ein Weg in die Dunkelheit", Wladimir Kramnik au sujet du duel contre Deep Fritz
- [5] <http://sdcc.sourceforge.net>
Page d'accueil du sdcc (small device c-compiler), documentation et téléchargement (anglais)
- [6] www.modul-bus.de/mbnews/mbnews02/isp.htm
AtmelISP-Flasher, documentation et téléchargement (allemand)
- [7] www.turbobit.com/mem51.html
Informations concernant la cartographie des dérivés du 8051, relatées au fichier .map du compilateur sdcc (anglais)
- [8] www.geocrawler.com/archives/3/3278/2000
Forum des utilisateurs du compilateur sdcc (anglais)
- [9] www.estpak.ee/~nq002a/liter/picapps.pdf
"Interfacing a Matrix Keypad" et autres applications des PIC (anglais)
- [10] Andreas Roth, Das Mikrocontroller-Applikations-Kochbuch, ITP-Verlag 1998
- [11] Andreas Roth, Das Microcontroller Kochbuch MCS51, ITP-Verlag 2002
- [12] Jürgen Maier-Wolf, 8051 Microcontroller erfolgreich anwenden, Franzis-Verlag 1996
- [13] Lars Bremer, Taktikmonster - Wie Schachprogramme denken, c't, le magazine pour la technologie micro, Magazine 21/2002, S. 216
- [14] www.pedram-azad.de
URL de téléchargement de la version la plus récente et de documentation pour l'ordinateur joueur d'échecs (allemand en construction)
- [15] www.farnell.com
Distributeur (du 89S53 d'Atmel par exemple)

vements, si partant vous voulez tricher (contre vous-même), libre à vous.

L'affichage à 2 lignes de 16 caractères a été connecté au système de la manière proposée par B. Kainka dans le n 288 d'Elektor (juin 2002, page 30 et suivantes), c'est-à-dire au bus. Il est accessible, dans la syntaxe du compilateur C utilisé (sdcc) par instruction xdata (accès à la RAM externe [5,7]). Prenons un exemple :

```
volatile xdata unsigned char at 0x8000 cmd_write;
```

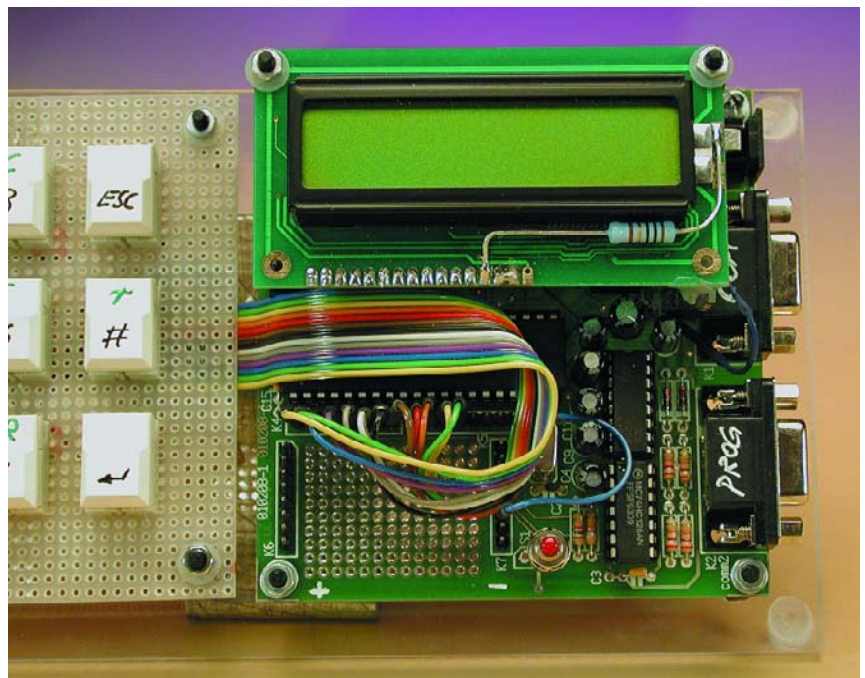
Pour le moment, le processeur implanté sur la carte est un Atmel AT89S53-24PC (source : entre autres, Farnell [15]). Ce composant est compatible broche à broche avec le AT89S8252 mais dispose, comme nous le disions plus haut, non pas 8 mais 12 Koctets de mémoire de programme Flash. Il ne comporte pas, en contrepartie, les 2 Koctets d'EEPROM de données. Il est fort probable, si l'on simplifie l'interface utilisateur (en se passant par exemple d'Entrée/Sortie par le biais de l'interface série) que l'on puisse stocker le programme dans un AT89S8252. Il est possible de connaître, sous CSEG [7], quelle est la quantité de mémoire nécessaire au code compilé grâce au fichier .map généré par le compilateur.

Résultats et perspectives

Dans sa version actuelle, EVELYN est un programme de jeu d'échecs capable de battre un débutant mais qui ne peut pas prétendre se mesurer à des systèmes professionnels ou à des programmes tournant sur PC. Ce n'était d'ailleurs pas l'objectif de ce projet, son but ayant plutôt été de prouver qu'il était possible de transposer un programme C d'une certaine complexité vers un microcontrôleur. Dans ce but nous avons mentionné les outils, liens et quelques astuces de programmation correspondants qui serviront peut-être à l'avenir à l'un ou l'autre lecteur intéressé. Si cet article a excité votre curiosité et que vous vouliez rendre EVELYN plus puissant, nous vous renvoyons à [13,1,2] et à la littérature professionnelle correspondante. En cas d'utilisation d'un AT89S53 il reste encore un peu de mémoire de programme disponible sinon le chemin caillouteux d'une augmentation de la vitesse ou d'un peaufinage du code passe inévitablement par la programmation en assembleur en ligne (*inline assembler*).

Ah oui, le nom de l'ordinateur jouant aux échecs fut inspiré par un message de compilation ô combien suggestif émis par sdcc : « *Warning, conditional flow changed by optimizer 'chess.c(386)': so said EVELYN the modified DOG* »....

(031001)



La pratique des réseaux neuraux (I)

1^{ère} partie : introduction aux réseaux neuraux

Chris McLeod et Grant Maxwell

Les réseaux neuraux artificiels, en abrégé ANN (*Artificial Neural Networks*), sont une forme célèbre de l'intelligence artificielle (IA). Ils se basent sur les modes opératoires des cellules nerveuses et de nombreux chercheurs estiment qu'ils sont notre meilleur espoir d'introduire une véritable intelligence dans une machine. Si vous êtes un amateur de la série télévisuelle « Star Trek », vous savez que M. Data est supposé disposer d'un cerveau « neuronique », de même que le robot de *Terminator*.

Bien que ces merveilles technologiques soient le *nec plus ultra* de la recherche en informatique, ils sont aussi à la portée de l'amateur enthousiaste. Le but de ces articles est de présenter ce sujet fascinant d'une manière pratique, ce qui vous permettra d'expérimenter vos propres réseaux neuraux.

Mais qu'est-ce exactement qu'un réseau neural ?

Un réseau neural est un réseau de processeurs interconnectés. Chaque processeur individuel n'est capable de réaliser qu'une tâche mathématique très simple, mais leur interconnexion fournit des résultats complexes. Le schéma de la **figure 1** présente un réseau neural type.

Pour comprendre le fonctionnement d'un réseau neural, examinons l'objet qui l'inspire : le cerveau.

Tout est dans votre tête

Ce qu'on nomme communément les « cellules nerveuses » sont baptisées « neurones » par les biologistes. Ce sont de minuscules cellules vivantes, qui composent notre cer-

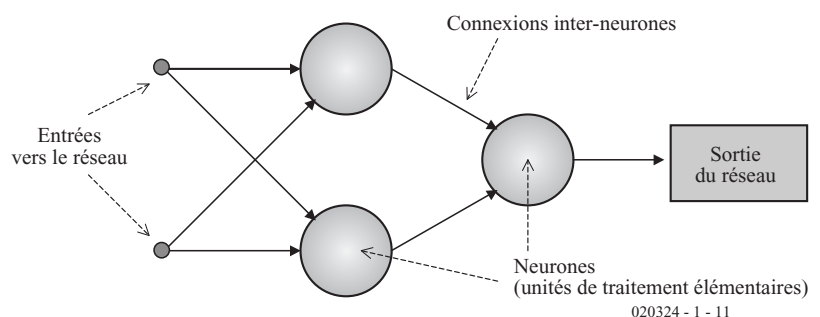


Figure 1. Un réseau neural type.

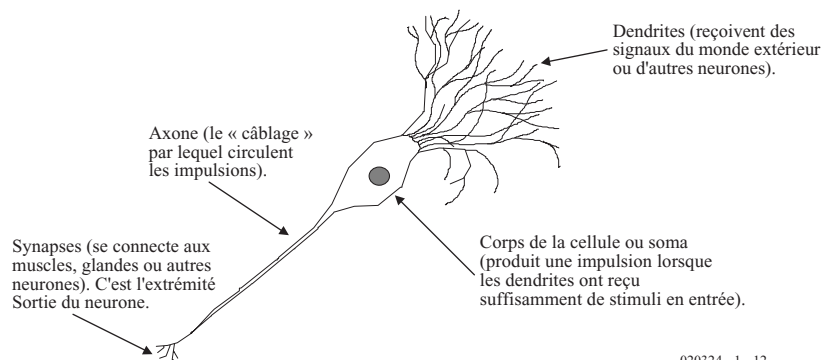


Figure 2. Un neurone biologique.

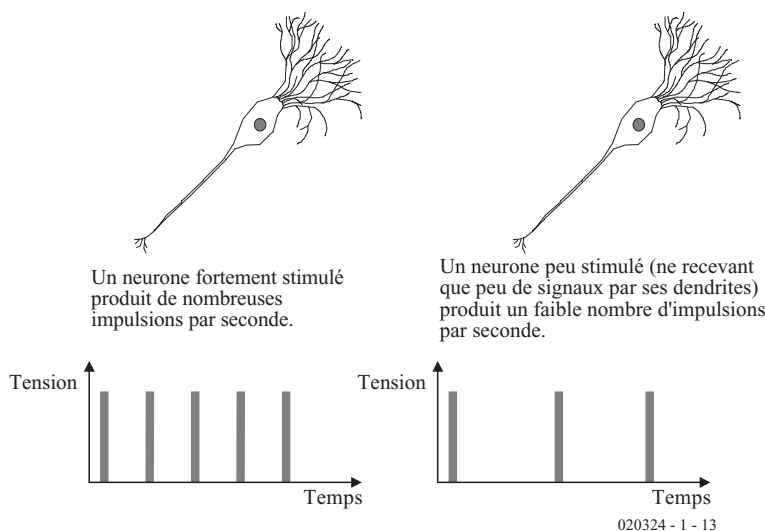


Figure 3. Comment un neurone traite l'information.

veau et notre système nerveux. Tout individu en comporte environ une centaine de milliards. Chacune a évolué pour devenir un processeur électronique. Si vous observez un neurone sous le microscope, il apparaît tel que sur la **figure 2**.

Les dendrites reçoivent les signaux d'autres neurones ou du monde extérieur : ce sont les « récepteurs » des neurones. Lorsque les dendrites reçoivent un stimulus suffisant, le corps de la cellule engendre une impulsion électrique qui se propage le long d'un axone vers les dendrites d'autres neurones ou vers les muscles afin de les stimuler.

Au fur et à mesure qu'un neurone est stimulé à travers ses dendrites, il produit de plus en plus d'impulsions par seconde, comme le montre la **figure 3**. L'information dans le système nerveux est ainsi modulée en fréquence.

Certains neurones, appelés neurones sensoriels, collectent l'information

du monde extérieur. Les bâtonnets et les cônes de vos yeux sont des neurones sensoriels. Les stimulus qu'ils reçoivent sont transmis au cerveau par les axones des neurones sensoriels. Ils sont alors traités par d'autres neurones, appelés les interneurones, qui ne sont connectés qu'à d'autres neurones. Le résultat de tout ceci est transmis à un ensemble final, appelé neurones moteurs, dont les axones sont connectés aux muscles et aux glandes.

C'est ainsi que nous pouvons saisir l'information en provenance du monde, la traiter et y réagir. Ces processus sont responsables de tout, des simples réflexes à la conscience elle-même.

La naissance du réseau neural artificiel

En 1943, deux chercheurs, Warren McCulloch et Walter Pitts, publièrent une communication qui marqua le

début de la science des réseaux neuraux artificiels. Elle proposait un modèle simple du neurone, base des réseaux artificiels (sous une forme améliorée) actuels. Le concept originel est présenté en **figure 4**.

Ce schéma montre quatre entrées baptisées i_1 à i_4 . Elles sont semblables aux dendrites de la version biologique, et sont connectées au monde extérieur ou à d'autres neurones. Ces entrées sont « pondérées » par quatre facteurs de pondération dits « poids », w_1 à w_4 . Ces poids caractérisent la force de la connexion qui relie chaque entrée au corps de la cellule dans le neurone réel. Dans la réalité, bien sûr, il peut y avoir n'importe quel nombre d'entrées.

Le système travaille ainsi : d'abord, chaque valeur d'entrée est pondérée en la multipliant par son facteur de pondération et additionnée aux autres valeurs d'entrée pondérées.

$$\text{Somme} = (i_1 \times w_1) + (i_2 \times w_2) + (i_3 \times w_3) + (i_4 \times w_4)$$

Ensuite, si la somme (qui, au sens biologique, correspond au stimulus global transmis au neurone) est supérieur à un seuil prédéterminé, le neurone produit une sortie « 1 » ; sinon une sortie « 0 ». Par exemple, en fixant le seuil à « 0,5 » :

```
If Somme > 0,5 then Sortie = 1
If Somme < 0,5 then Sortie = 0
```

Ceci ressemble un peu au neurone réel qui envoie une impulsion lorsque le stimulus est suffisant.

Essayons un petit exemple pour s'assurer que jusqu'ici tout est clair. Supposons que nous ayons un neurone avec des entrées et des facteurs de pondération tels que ceux présentés en **figure 5**, quelles sorties provoqueront-ils ? Calculons la sortie du neurone sur la courbe, avec un seuil de 0,5 :

$$\text{Somme} = (0,1 \times 0,5) + (0,5 \times 0,2) + (0,3 \times 0,1) = 0,18$$

Comme 0,18 est inférieur au seuil, Sortie = 0.

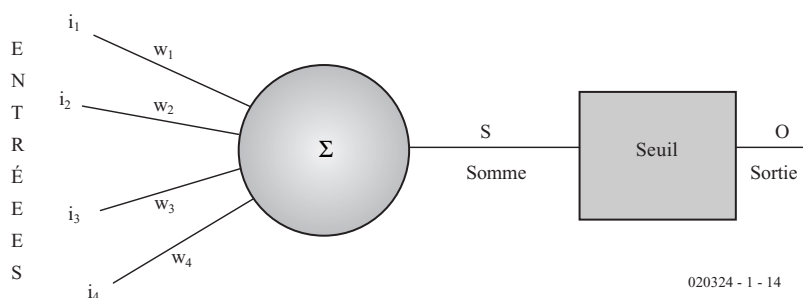


Figure 4. Le neurone artificiel de base.

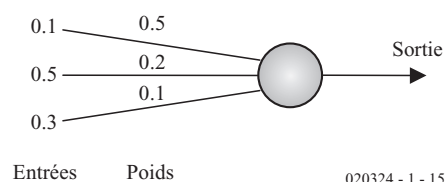


Figure 5. Un exemple de traitement.

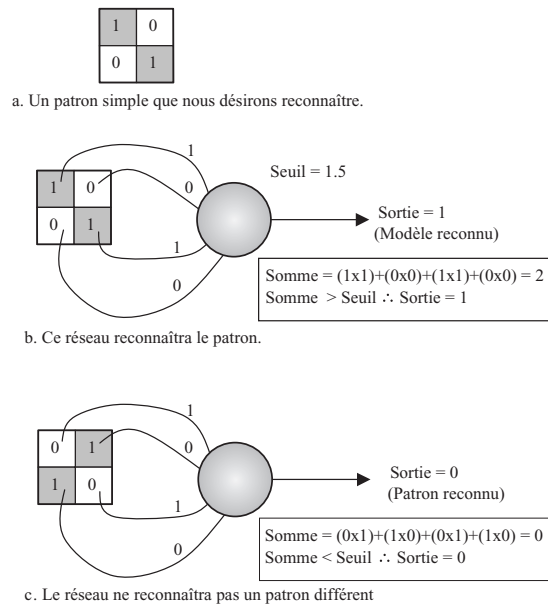
Alors, à quoi ça sert ?

Nous avons le modèle d'un neurone, mais à quoi cela sert-il ? Eh bien, considérons une forme visuelle comme celle de la **figure 6a**. Et supposons que nous voulions un dispositif qui puisse la reconnaître. Nous pouvons donner aux pixels sombres la valeur « 1 » et la valeur « 0 » aux pixels clairs. Si nous montons notre neurone comme indiqué en **figure 6b**, il produira une sortie « 1 » chaque fois qu'il « verra » cette forme. Vous pouvez probablement constater que, même si la forme n'est pas parfaite –si les pixels sombres ne sont pas tout à fait des uns et les pixels clairs pas tout à fait des zéros–, le neurone reconnaîtra malgré tout la forme (on dit que le neurone est tolérant au bruit).

Mais si nous produisons une forme totalement différente, comme celle présentée en **figure 6c**, le neurone ne la reconnaîtra pas. La reconnaissance des formes, même lorsqu'elles ne sont pas parfaites, est un des attributs de base de l'intelligence. Si vous êtes un animal, il faut que vous puissiez reconnaître un semblable, une proie ou un prédateur pour survivre. Avec un peu de réflexion, vous êtes capable de comprendre l'importance de la reconnaissance des formes. On pourrait dire que les réseaux neuraux sont les meilleurs outils pour y parvenir. Ils peuvent aussi reconnaître des formes n'importe quand, si celles-ci sont numérisées et introduites dans le réseau.

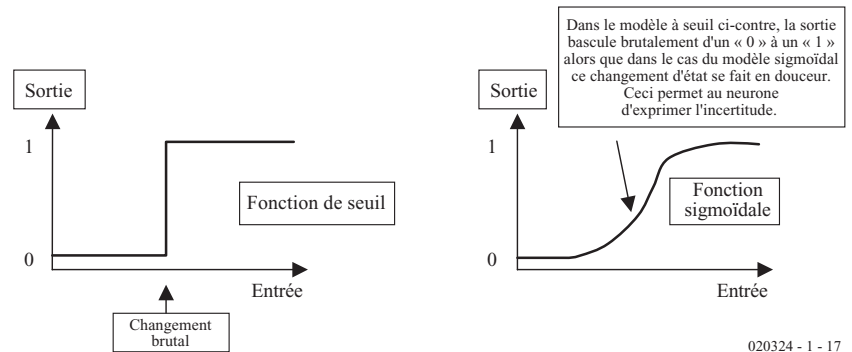
A vous d'apprendre !

Vous pouvez voir d'après les exemples cités plus haut que la fixation des facteurs de pondération est une des clés du fonctionnement du réseau. Comme ce sont ces poids qui déterminent quelles formes sont reconnues, la fixation des poids est la tâche principale du programme de réseau neural. Dans l'exemple ci-dessus, ces poids ont été fixés par le bon sens, ce qui est possible pour une tâche très simple comme celle-ci. Mais pour des tâches plus complexes comme la reconnaissance de centaines de formes par un réseau de neurones, ce n'est plus le cas. Nous voulons aussi que ces poids soient fixés automatiquement, hors de l'interférence d'un opérateur humain ; après tout, nous apprenons tous par nous-même sans aide extérieure. Arrive l'algorithme « d'éducation » ou « d'apprentissage ». C'est la partie du réseau neural qui fixe les poids permettant de reconnaître automatiquement une forme. Pour donner un exemple très simple de son fonctionnement, prenez un algorithme pour enseigner au neurone déjà décrit de reconnaître sa forme :



020324 - 1 - 16

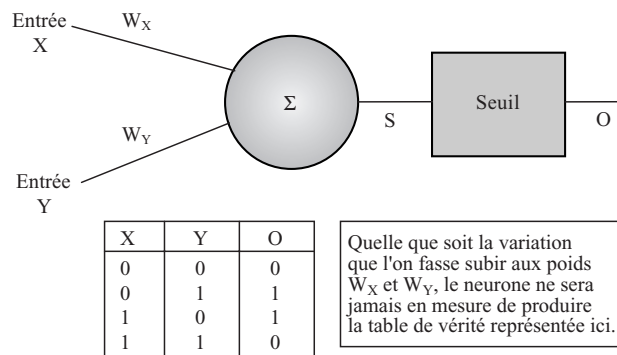
Figure 6. L'usage du neurone pour la reconnaissance de formes.



020324 - 1 - 17

Figure 7. Le neurone binaire et le sigmoïdal.

1. *If Sortie est bonne, ne rien faire.*
 2. *If Sortie est haute alors qu'elle devrait être basse, diminuer les poids attachés aux entrées hautes.*
 3. *If Sortie est basse alors qu'elle devrait être haute, augmenter les poids attachés aux grandes entrées.*
- Les poids sont généralement initialisés avec des chiffres pris au hasard.



020324 - 1 - 18

Figure 8. Un simple neurone ne peut pas simuler une porte XOR.

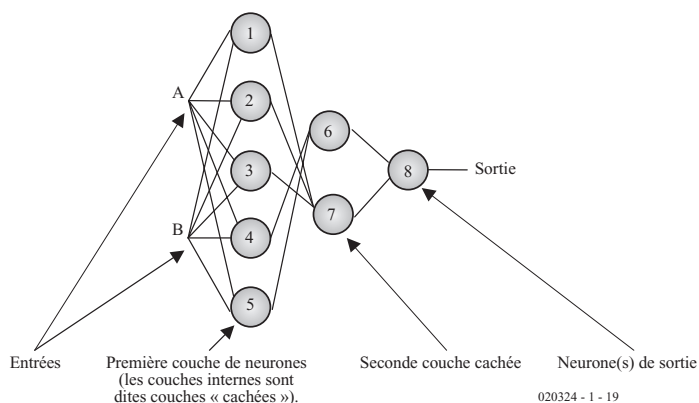


Figure 9. Un réseau neural multicouche.

Les autres articles de cette série présenteront les diverses méthodes d'éducation du réseau.

Deux autres points valent la peine d'être mentionnés. En premier, le fait de fixer à votre manière les poids vous permet de faire fonctionner le réseau comme (presque) n'importe quel type de porte logique. Aussi ne considérez pas les réseaux neuraux comme un simple outil de reconnaissance – ils sont plus souples que cela. En réalité, ils s'approchent plus d'un circuit à logique combinatoire apprenant. En second, plutôt que d'utiliser une pure sortie binaire comme dans l'exemple cité, on utilise habituellement pour la sortie de nos neurones une fonction mathématique connue sous le nom de fonction « sigmoïdale ». C'est :

$$Sortie = \frac{1}{1 + e^{-Somme}}$$

Ainsi, dans l'exemple de la figure 5, la sortie du neurone obtenue par la

fonction sigmoïdienne serait :

$$Sortie = \frac{1}{1 + e^{-0,18}} = 0,5449$$

En réalité, cette fonction produit toujours une sortie comprise entre zéro et un ; ainsi la sortie ressemble-t-elle au seuil binaire. Alors pourquoi l'utiliser ? Eh bien, elle montre ses avantages lors de la reconnaissance de deux formes très semblables. Elle permet au neurone d'exprimer son incertitude en produisant une sortie intermédiaire entre zéro et un. Le réseau dit : « Je ne suis pas tout à fait sûr ». La figure 7 présente le concept.

Réseaux de neurones

Les réseaux de neurones ont pris de l'ampleur dans les années qui ont suivi la proposition de leur modèle par McCulloch et Pitts. Mais en 1969 parut un ouvrage attaquant bien des concepts sous-jacents.

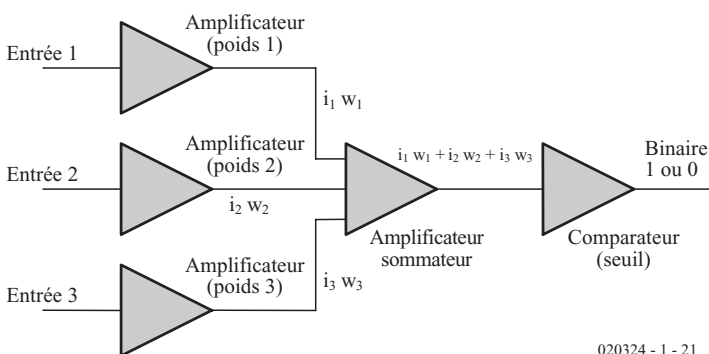


Figure 11. Un montage électronique d'un neurone.

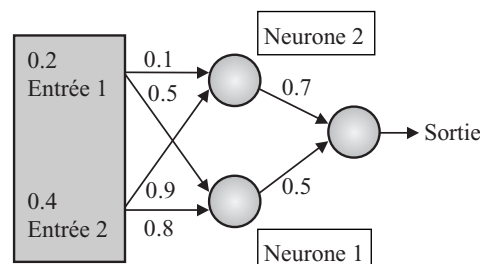


Figure 10. Un exemple d'un réseau neural.

Ce livre était *Perceptrons*, par Marvin Minsky et Seymour Papert. Il mettait en évidence un vice dans le neurone de base. Le problème était celui-ci : quelles que soient les valeurs que vous attribuez aux poids d'un neurone à double entrée, il ne sera jamais capable de simuler une porte OU exclusif (XOR) comme dans la figure 8.

Cet argument, en même temps que d'autres développés dans l'ouvrage, limita le développement des réseaux neuraux pendant de nombreuses années. La plupart des chercheurs abandonnèrent le champ d'étude, estimant que si un neurone ne pouvait remplir cette simple tâche, la recherche était vaine. En fait, ce qu'il fallait pour résoudre le problème et d'autres similaires, n'était pas un simple neurone, mais un réseau de neurones. D'ailleurs, un mathématicien russe nommé Kolmogorov avait déjà démontré qu'un réseau de neurones comportant trois niveaux, cf. la figure 9, pouvait résoudre n'importe quel problème de ce genre.

Malgré la révélation de cette solution, il fallut attendre 1982/83 pour que la recherche sur les réseaux neuraux reprenne. A ce moment, l'algorithme de rétro-propagation (qui sera décrit dans la deuxième partie) fut largement répandu, la puissance informatique à bon marché disponible pour tous et John Hopfield publia sa fameuse communication (dont nous parlerons dans la troisième partie).

Les neurones dans le réseau fonctionnent exactement de la même manière que ceux déjà présentés. Donnons un exemple pour le prouver avec des neurones sigmoïdiens, cf. figure 10.

Calculez la sortie de ce réseau en supposant une fonction de compression sigmoïde :

Entrée vers le neurone 1 = (0,2 x 0,5) + (0,4 x 0,8) = 0,42.

$$\text{Sortie} = \frac{1}{1 + e^{-0,42}} = 0,603$$

Entrée vers le neurone 2 = (0,2 x 0,1) + (0,4 x 0,9) = 0,38.

$$\text{Sortie} = \frac{1}{1 + e^{-0,38}} = 0,594$$

Entrée vers le neurone terminal = (0,594 x 0,7) + (0,603 x 0,5) = 0,717.

$$\text{Sortie} = \frac{1}{1 + e^{-0,717}} = 0,672$$

Faites le vôtre !

Vous pouvez monter des réseaux neuraux électroniques, soit analogiques soit numériques. Un schéma synoptique d'un montage analogique est présenté en **figure 11**.

Trois amplificateurs simples fixent les poids ; ils doivent disposer de gains réglables électroniquement si le réseau doit apprendre en temps réel. La sortie est alors additionnée et le seuil est appliqué par un comparateur.

Ce type de circuit a été construit avec des composants discrets comme des amplificateurs opérationnels et comme parties d'un circuit intégré. L'inconvénient de cette méthode est qu'elle nécessite beaucoup de composants s'il y a beaucoup de neurones.

On peut fabriquer des circuits numériques exactement de la même façon. Les amplifica-

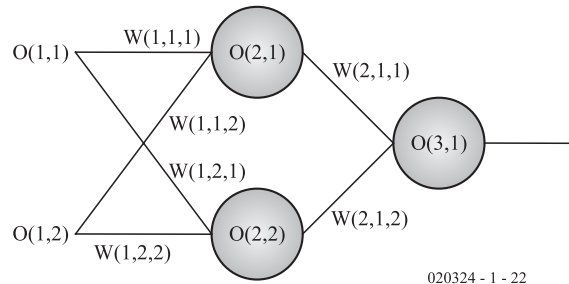


Figure 12. Stockage des poids dans un réseau typique.

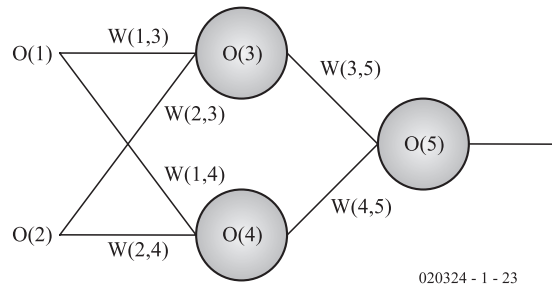


Figure 13. Une manière plus souple de programmer le réseau.

teurs fixant les poids sont remplacés par des multiplicateurs numériques, l'amplificateur additionneur par un mélangeur et le comparateur par un décodeur qui s'activera si la sortie binaire du mélangeur est supérieure à une valeur de seuil. Les réseaux neuraux numériques sont cependant dépen-

sésitent des multiplicateurs binaires qui sont en général des circuits gros et complexes. La manière la plus habituelle de monter un réseau neural est de le simuler par logiciel. Le programmeur peut changer rapidement les paramètres tels que le nombre de neurones et leurs connexions, sans coût.

Listage 1. Calcul de la sortie d'un réseau

```
FOR L = 1 TO number_of_layers
  FOR n = 1 TO number_of_neurons_in_biggest_layer
    FOR c = 1 TO number_weights_in_biggest_layer
      O(L+1, n) = O(L+1, n) + O(L, c) * W(L, n, c)
    NEXT c
    O(L+1, n) = 1/(1+exp(-1*O(L+1,n)))
  NEXT n
NEXT L
```

Listage 2. Programmation de sortie alternative

```
FOR t = 1 + number_of_inputs TO number of neurons
  FOR f = 1 TO number_of_neurons
    O(t) = O(t) + O(f) * W(f,t)
  NEXT f
  O(t) = 1/(1 + exp(-1 * O(t)))
NEXT t
```

teux recâblages, et signifie qu'un grand nombre de neurones peuvent être interconnectés facilement.

La méthode logicielle a beaucoup d'avantages pour l'expérimentateur amateur et il est relativement facile de programmer un réseau neural dans n'importe quel langage de haut niveau et de nombreux langages assembleurs (particulièrement ceux qui disposent de commandes de multiplication). Compte tenu de ces avantages, c'est cette méthode que nous retiendrons dans le reste de cette série.

Installation du logiciel

Comment programmer le réseau neural ? La façon la plus simple est d'entrer tous les paramètres nécessaires dans des tableaux. Les entrées, les sorties, le réseau peuvent être entrés dans un tableau et les poids dans un autre.

Par exemple, nous pouvons entrer tous les poids dans une matrice à trois dimensions :

```
Poids [numéro_de_couche,
numéro_de_neurone,
numéro_de_connexion]
```

Et les sorties (et les entrées) de chaque neurone dans une matrice à deux dimensions :

```
Sortie[numéro_de_couche,
numéro_de_neurone].
```

Le réseau engendré prend la forme présentée dans la **figure 12**.

Notez que $S(1,1)$ et $S(1,2)$ sont les entrées vers le réseau et $S(3,1)$ la sortie. On peut faire calculer la sortie d'un tel réseau par l'ordinateur avec le simple algorithme :

1. Initialiser tous les poids et les valeurs de sortie à zéro.
2. Donnez aux entrées, ici $S(1,1)$ et $S(1,2)$ les valeurs que vous voulez. La couche 1 est l'entrée dans le réseau.
3. Donnez aux poids utilisés dans le réseau les valeurs que vous voulez.
4. Calculez la sortie du réseau : cf. **listage 1**.

Il y a en fait bien des façons différentes de programmer ainsi le réseau. Une autre, peut-être plus souple, consiste à programmer les poids de cette manière :

```
poids
[le_poid_du_neurone_pro-
vient_de, le_poid_du_neu-
rone_se_dirige_vers]
```

et de juste numéroter chaque neurone séquentiellement :

```
sortie [numéro_de_neurone]
```

Ceci donne un réseau du type présenté en **figure 13**.

Dans ce cas de figure, $S(1)$ et $S(2)$

sont les entrées et $S(5)$ la sortie. L'algorithme pour calculer la sortie devient cette fois-ci :

1. Initialiser tous les poids et les sorties à zéro..
2. Donner aux entrées, dans ce cas $S(1)$ et $S(2)$, les valeurs que vous voulez.
3. Donner aux poids utilisés dans le réseau les valeurs que vous voulez.
4. Calculer la sortie du réseau, cf. **listage 2**.

Certains langages (comme C et C++) vous permettent d'allouer dynamiquement la taille des tableaux, ce qui peut être utile pour des réseaux complexes qui changent leur taille. D'autres comme Fortran et Matlab permettent la multiplication matricielle, ce qui rend la vie plus simple dès lors que les équations du réseau peuvent être décrites comme des opérations matricielles.

Les langages orientés objet vous permettent de définir les neurones et les réseaux comme des objets, qui introduiront les paramètres de vos neurones dans vos réseaux.

Dans les langages de bas niveaux, les poids et les sorties peuvent être stockées directement en mémoire, et des routines écrites pour aller les rechercher et les manipuler.

Le calcul de la sortie du réseau s'appelle la propagation. L'étape suivante consiste à modifier les poids de façon à ce que le réseau apprenne pour faire mieux à la prochaine propagation ; c'est ainsi qu'un réseau est éduqué et cela s'appelle la rétro-propagation.

Dans le deuxième article, nous examinerons à l'éducation du réseau, en commençant par l'algorithme d'apprentissage le plus populaire – l'algorithme de rétro-propagation.

(020324-1)

Bus DCI, Elektor n°293, novembre 2002, page 72 et suivantes

Sur le schéma du poste secondaire, 2 des sorties de IC1 sont reliées à la masse. Si cet état de faits n'a pas de conséquence sur le fonctionnement du montage on ne peut pas affirmer qu'il mérite un coup de chapeau. Cette situation se retrouve au niveau du circuit imprimé. La meilleure solution pour remédier à cela est de couper purement et simplement les pattes 2 et 12 de IC1 de la platine 020113-2, avant de mettre ce circuit intégré dans son support. Cela permet de ne pas avoir à modifier la platine. (020113)

Allô ! maman ? Ici bébé !, Elektor n°294, décembre 2002, page 50 et suivantes

Le concepteur de ce projet a perdu de vue qu'il est impératif de faire en sorte qu'il ne circule pas le moindre courant continu

dans le transformateur ligne, TR1. Si tel n'était pas le cas, cela pourrait avoir des conséquences néfastes pour le fonctionnement du montage. Il est possible, très simplement, de remédier à cette situation par l'adjonction d'un condensateur Cx de 1 µF/250 VCC en respect du schéma ci-contre.

Au niveau de la platine, la première étape consiste à interrompre, à l'aide d'un cutter bien affûté, les pistes allant aux bornes du secondaire de TR1. ON relie ensuite l'une des bornes de transformateur ainsi isolées au point nodal de R16/K1. Le condensateur additionnel Cx vient se placer entre l'autre borne du secondaire du transformateur et le point nodal R16/contact du relais. Il ne reste plus ensuite qu'à rétablir la liaison entre le contact-mère du relais (tout près du bord de la platine) et la ligne (que nous venons d'interrompre) allant vers

l'embase K1 (a et a'). Pour vous aider dans cette opération nous vous proposons ci-contre le schéma avant et après l'intervention et le dessin des pistes qui identifie bien les points concernés.

Dernière petite correction. Pour une fois, c'est la liste des composants et non pas le schéma qui a raison. La valeur correcte du condensateur C8 est bien de 470 pF.

(012016)

Regénérateur d'image DVD, Elektor n°289/290, juillet/août 2002, page 10 et suivantes

Suite à un petit problème de production, la sérigraphie de certaines des platines ne visualisent pas la position des ponts de câblage. Ceux-ci sont cependant indispensables au fonctionnement du montage, d'autant plus que certains doivent se « faufiler » autour et sous certains des composants. Il faudra, si vous avez en main une platine ne comportant pas la sérigraphie des ponts de câblage, vous reporter à la figure 5 de la page 15 pour effectuer ces implantations.

(010121)

Testeur de sonde lambda, Elektor n°255, septembre 1999, page 60 et suivantes

Il existe 2 discordances entre le schéma et la liste des composants ; comme cela est le cas dans la majorité des cas, c'est le schéma qui fait fois. Partant, la résistance R7 vaut 2kΩ74 et la résistance R8 n'existe pas.

Merci Mr Liboureau.

(990057)

Système de développement PICÉE, Elektor n°284, février 2002, page 8 et suivantes

Une petite erreur s'est glissée dans l'adresse du site de NTPicprog que ce soit dans la revue ou le site Internet. La bonne adresse est : <http://home.swipnet.se/~w-24528/NTPicprog/> (il y a un "-" entre le "w" et le "2"). Par contre, je suis un peu interrogatif vis à vis du PIC16F84 qui est mentionné comme devant être programmé EPS010062-41. Or je n'ai pas trouvé cette référence au niveau de la rubrique platines & logiciels, et l'article ne fait pas référence à une programmation particulière du PIC.

Cordialement, Merci Mr Masson, vous avez parfaitement raison sur ces 2 points, il n'existe effectivement pas de programmation spécifique du PIC...

(010062)

Pupitre d'éclairage, Elektor n°296, avril 2002, page 65 et suivantes

Comme le dit à raison le schéma, la dénomination des opto-coupleurs IC23 et IC24 est bien CNY74-4 et non pas CNY17-4, comme l'affirme à tort la liste des composants.

(000162)

Pico-API, Elektor n°293, novembre 2002, page 20 et suivantes

Dans les conditions les plus difficiles (*worst case*) il peut être nécessaire d'augmenter le courant à travers les LED des opto-coupleurs, ce que l'on obtient en faisant passer la valeur des résistances R19 à R25 des 10 kΩ d'origine à de l'ordre de 4kΩ7 environ. Si cette réduction ne devait pas suffire, on pourra adopter une valeur plus faible encore et aller jusqu'à 1 kΩ environ. Signalons que nous n'avons pas rencontré de problème avec la valeur de 10 kΩ d'origine.

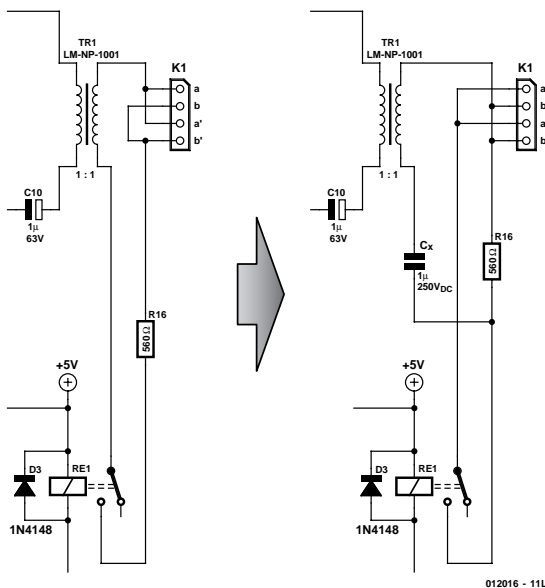
Pour le varistor donné pour R30 de la liste des composants on pourra utiliser le type SIOV-S124K30 (EPCOS) nomenclaturé B72214S0300K101.

(010059)

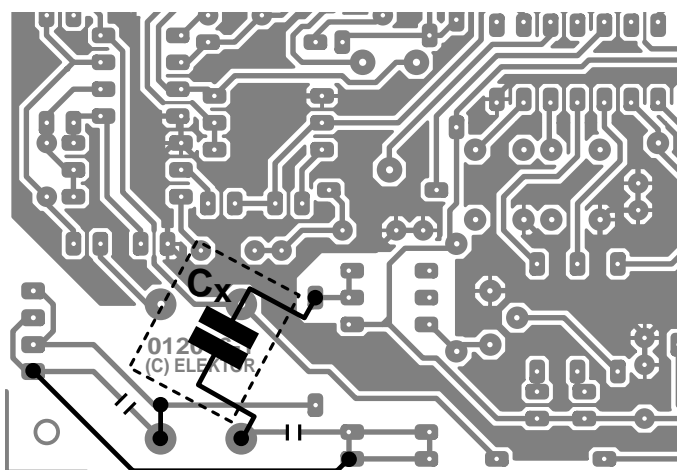
Récepteur IR pour le bus I²C, Elektor n°289/290, juillet/août 2002, page 102 et suivante

Le schéma de ce montage comporte une petite erreur au niveau de la sortie des lignes I²C. Il faut décaler légèrement les dénominations des sorties : SDA est la broche 15 de IC1, SCL la broche 14 de IC1 et INT va de la broche 13 de IC1 à la broche 19 de IC2. Depuis la publication de cet article il est apparu que le SAA3049 n'existait plus que sous la forme de surplus. Si par conséquent, vous vous décidez de réaliser ce montage, commencez par vous assurer que vous pouvez obtenir ce circuit intégré. Nous vous proposons dans le numéro de février 2003 un article intitulé « Successeur du SAA3049 ».

(024033)



012016 - 11L



Réception VLF sur PC

Une antenne à votre carte-son

La réception des OTL (Ondes Très Longues = VLF en anglais) nécessitait, dans le temps, un récepteur prévu à cet effet ou alors un convertisseur. De nos jours, il suffit tout simplement d'utiliser une carte-son pour PC dont il existe des dizaines de modèles et une bonne quantité (lire longueur) de câble.

Il est possible, à très peu de frais, de convertir tout PC (Pentium 100 avec W95 et mieux) doté d'une carte-son en un récepteur pour les fréquences inférieures à 24 kHz relativement performant. Le matériel nécessaire se résume à une antenne de caractéristiques adéquates, le logiciel prenant la forme d'un programme d'analyse de spectre disponible gratuitement sur Internet.

L'antenne

À l'image de ce qui est le cas pour la réception des Grandes Ondes (GO) et des Ondes Moyennes (OM), il est recommandé d'utiliser, pour la réception des Ondes Très Longues (OTL, on parle plus souvent des ondes déca-kilométriques), une antenne inductive. Cette antenne devrait être constituée d'une bobine ayant le diamètre le plus grand possible et/ou comporter le nombre de spires le plus élevé possible. Les caractéristiques électriques et mécaniques telles qu'inductance, résistance ohmique, nombre des spires etc., n'ont qu'une importance relative et peuvent varier sur une plage assez large. Ceci explique que la réalisation d'une telle antenne soit extrêmement simple. Tout ce dont on a besoin est un rouleau de câble isolé pour installations électriques qui sera connecté à l'entrée LINE ou MIC de la carte-son par le biais d'un câble bifilaire quelconque d'une longueur minimum de 2 mètres.

La distance entre la bobine et le PC (ainsi d'ailleurs qu'avec tout autre appareil électronique) doit être, si l'on veut garantir une réception optimale, d'au moins 2 mètres. La self de réception sera couchée. Si l'on se trouve dans un bâtiment construit en béton armé, on la placera de préférence à proximité d'une fenêtre. On pourra également envisager, pour obtenir



Figure 1. L'antenne de réception pour les OTL.

une sensibilité plus élevée, de monter plusieurs bobines en série; il est important alors, de s'assurer que le sens de bobinage des spires soit le même ceci pour éviter que les signaux induits ne s'éliminent l'un l'autre.

La **figure 1** montre une antenne de réception pour OTL constituée par la prise en série de 4 rouleaux de câble électrique, 2 montés 2 par 2 sur un axe commun. Chaque rouleau de câble possède un diamètre de 10,3 cm et comporte 309 spires. On pourra utiliser des agrafes (*paper-clip*) de bureau pour, lors de tests, court-circuiter l'un ou l'autre rouleau.

L'entrée de la carte-son

Une fois l'antenne connectée à l'entrée de la carte-son il faudra, sur le PC, ajuster l'entrée de signal, ce que

l'on fera par un clic souris droit sur le symbole de haut-parleur que l'on trouve sur la barre de tâches (normalement) au bas de l'écran à droite. On voit apparaître (dans le cas de Windows 98 ici) le panneau de pilotage de la reproduction dans lequel on cliquera sur le bouton Propriétés (*Properties*) après avoir activé l'onglet *Options*. Nous aimerions, à ce niveau, ajuster le volume lors de l'enregistrement (*Recording*). Dans la liste qui s'affiche alors, on coche l'entrée à laquelle est connectée l'antenne, avant d'appuyer sur OK pour confirmer le choix. Le potentiomètre de volume correspondant à cette entrée devra être, pour la réception OTL, être mis au volume maximum, le repère de Balance étant placé à mi-course. Rien n'interdit bien entendu d'envisager de modifier ce paramétrage ultérieurement lorsque

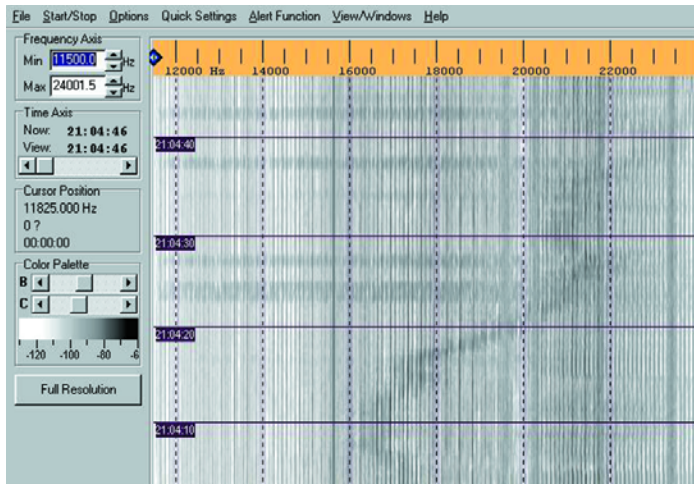


Figure 2. Fenêtre principale du programme SpecPlus.

l'on voudra, par exemple, procéder à l'analyse de spectre d'un autre signal.

Logiciel d'analyse

Il existe différents programmes d'analyse du spectre de signaux appliqués à un PC par le biais de sa carte-son (cf. www.vlf.it/harald/strangerec.htm). L'auteur utilise plus particulièrement **SpecPlus**, un programme très puissant qui permet la sauvegarde des signaux sous la forme de diagrammes fréquence-temps-intensité (connus sous la dénomination de spectrogrammes) automatiquement en fichiers au format .jpg qui permet d'économiser énormément de place. La **figure 2** vous propose la fenêtre principale de ce programme. Ce logi-

ciel présente l'avantage additionnel de tourner de façon très stable sous Windows 95, Windows 98 et Windows ME. Il ne sait rien du fonctionnement de ce programme sous NT ou des versions plus récentes de Windows (lire XP), ce qui est certain est qu'il ne tourne pas sous Windows 3.x.

On pourra télécharger SpecPlus qui est un freeware, à l'adresse : www.qsl.net/dl4yh/spectra1.html. Après avoir téléchargé le programme avec succès (à noter que ce programme évolue très rapidement et que les versions se suivent et ne se ressemblent pas, nous en étions à la version V1.9 b6 du 10 août 2002) et l'avoir décompressé avec WinZip dans un répertoire d'installation, le reste à installer le programme, opération qui, l'expérience nous l'a

appris, ne doit pas poser de problème.

Il faudra ensuite définir un certain nombre de paramètres, à commencer par la palette des couleurs. Le programme propose différentes possibilités de visualisation. Même si certaines combinaisons de couleurs semblent très attrayantes, il est recommandé d'opter pour une combinaison « signal noir sur fond blanc ». Cette solution permet une interprétation aisée du diagramme même si l'on ignore tout de la palette de couleurs choisie, sans même parler de la possibilité, si on le désire, de comprimer encore plus les fichiers.jpg obtenus sans risquer de perdre l'information ou de la rendre méconnaissable en cas de choix erroné des couleurs. Le paramétrage de la palette des couleurs se fait, dans le menu de Configuration (**figure 3**), par le point de menu *Options/Load Color Palette*, qui permet de charger une palette de couleurs adéquate.

Il faut ensuite paramétrer le taux d'échantillonnage (*Samplingrate*) par le biais du point de menu *Option/Audio Settings*. Le taux d'échantillonnage détermine la fréquence la plus élevée qu'il est possible de capter. Sa valeur répond à la formule suivante :

$$F = \text{taux d'échantillonnage} / (2 \times \text{diviseur de taux d'échantillonnage en entrée})$$

Pour la fréquence d'entrée maximale de 24 kHz (les cartes-son ordinaires ne peuvent pas traiter des signaux d'entrée de fréquence plus élevée) il faudra partant opter pour un taux d'échantillonnage de 48 kHz et un facteur de division (*Input Sample Rate Divisor*) de 1. On en arrive ensuite au paramétrage de la transformée rapide de Fourier (FFT = *Fast Fourier Transformation*). On choisit pour cela l'option *FFT Settings* sous l'onglet *Options*. Pour la réception de signaux OTL on mettra la sortie FFT (*FFT-Output*) sur Logarithmique (*Logarithmic*) et comme Type de FFT (*FFT-Typ*), cocher l'option « *Real Number FFT Starting at 0* ». Il reste à choisir le mode d'affichage. Pour cela, on fixera, sous l'onglet *Option/Display Spectrum display Settings*, la plage d'amplitude à visualiser pour la réception OTL, à une plage comprise entre -130 et -60 dB.

L'expérience nous a appris que c'était là la plage la plus intéressante pour la visualisation de signaux OTL, vu d'une part que l'on a peu de chances de rencontrer des signaux de niveau supérieur à cette limite, et que d'autre part le niveau de bruit des cartes-son se situe lui à un niveau compris entre -110 et -120 dB. Cette fenêtre permet de jouer sur d'autres paramètres concernant l'affichage (cf. la copie d'écran).

Il ne reste plus, pour en avoir terminé, à jouer,

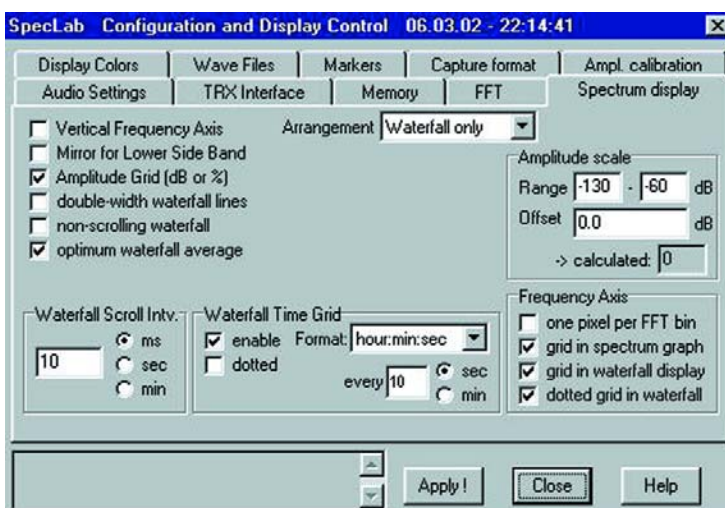


Figure 3. Écran de configuration de l'analyseur de spectre.

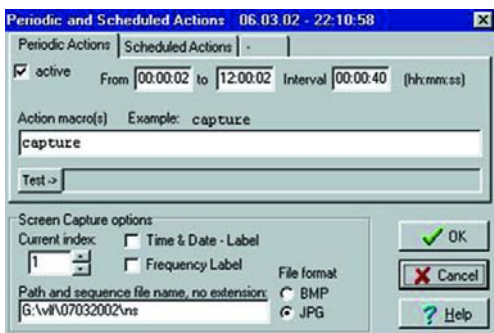


Figure 4. Automatisation de Speclab.

dans la fenêtre principale, sur les 2 réglottes de luminosité et de contraste, B (pour *Brightness*) et C (pour *Contrast*) de manière à obtenir la meilleure sensibilité. Pour ce faire on débranche l'antenne ou on la court-circuite et on joue sur la réglotte B jusqu'à ce qu'il ne reste plus tout juste qu'un arrière-plan blanc. On joue ensuite sur la réglotte C pour ajuster le contraste de l'affichage de manière à pouvoir voir de petites variations d'intensité. Pour un réglage fin, on rebranchera l'antenne. Comme ces 2 paramètres s'influencent l'un l'autre, il faudra, dans certains cas, reprendre les différentes étapes mentionnées ici jusqu'à ce que l'on soit satisfait du résultat à l'affichage. Le PC peut maintenant faire office de récepteur pour les OTL.

Les moniteurs (écrans) de PC sont des sources importantes de signaux parasites dans la plage des OTL. Pour cette raison, il est fortement recommandé de ne pas regarder

directement le spectrogramme sur l'écran, mais de commencer par le stocker sur le disque dur sous la forme d'un fichier graphique que l'on examinera ultérieurement. Ceci explique qu'un mode de fonctionnement automatique chronopiloté avec moniteur PC coupé soit le mode de fonctionnement le meilleur. L'expérience nous a appris que le meilleur créneau de réception OTL se situe entre minuit (00:00) et 6 heures du matin (06:00) pour la simple et bonne raison que la plupart des appareils électriques sources de parasites sont coupés et que le niveau des parasites dans cette plage de fréquences est le plus faible.

Il faudra, pour réaliser un fonctionnement automatique, choisir l'option *File/Periodic Actions* (cf. **figure 4**). Avec le paramétrage choisi ici, on a génération, toutes les 40 s, d'un fichier graphique de spectrogramme baptisé

ns + numéro d'ordre croissant + suffixe « .jpg »

entre 00:00:02 et 12:00:02, fichiers placés dans le répertoire g:\vfl07032002.

Les 2 premières lettres, ns, identifient l'axe de réception, nord-sud, ew pour East-West (Est-Ouest). Ceci est très important vu que l'antenne utilisée présente, comme toute antenne

inductive d'ailleurs, une caractéristique directionnelle indubitable. Il faudra s'assurer que le dit répertoire existe et que l'option active est bien cochée, vu que sinon il n'y aura pas de sauvegarde de fichier.

Signaux captés

On rencontre, dans le domaine de Ondes Très Longues, nombre de signaux d'origines diverses. Certains d'entre eux proviennent de stations d'émission monstrueuses et ont parcouru bien du chemin avant d'arriver au récepteur. D'autres ont pour origine le champ de votre propre maison et de son environnement proche. Le **tableau** ci-dessous donne une liste des stations les plus faciles à capter dans la plage de fréquences allant de 10 à 24 kHz. Dans ce domaine, on trouve principalement des stations par le biais desquels les États-Majors donnent des instructions à leurs sous-marins en plongée. En effet, les ondes électromagnétiques de cette plage de fréquences peuvent descendre, en fonction de la salinité et de la température des eaux dans lequel évolue le sous-marin, plusieurs dizaines de mètres sous la surface. On trouve également dans ce domaine quelques émetteurs de navigation et des services fournisseurs d'informations chrono-

Indicatif	Fréquence	Position	Spécificités
-	11,905 kHz	Russie (différentes localisations)	Navigation ALPHA
-	12,649 kHz	Russie (différentes localisations)	Navigation ALPHA
-	14,881 kHz	Russie (différentes localisations)	Navigation ALPHA
?	15,8 kHz	?	
GBR	16,0 kHz	Rugby (GrandeBretagne)	
JXN	16,4 kHz	Helgeland (Norvège)	
SAQ	17,2 kHz	Grimeton (Suède)	Actif à quelques occasions spéciales
?	17,8 kHz	?	
RDL/UPD/UFQE/UPP/UPD8	18,1 kHz	Russie (différentes localisations)	
HWU	18,3 kHz	Le Blanc (France)	Longues périodes d'inactivité
RKS	18,9 kHz	Russie	Périodes d'activité courtes et rares
GBZ	19,6 kHz	Criggon (GrandeBretagne)	Connait nombre de modes de fonctionnement
ICV	20,27 kHz	Tavolara (Italie)	
RJH63/RJH77/RJH99 et autres	20,5 kHz	Russie (différentes localisations)	Signaux horaires
ICV	20,76 kHz	Tavolara (Italie)	
HWU	20,9 kHz	Le Blanc (France)	
RDL	21,1 kHz	Russie (différentes localisations)	Périodes d'activité courtes et rares
HWU	21,75 kHz	Le Blanc (France)	
RJH63/RJH77/RJH99 et autres	23,0 kHz	Russie (différentes localisations)	Signaux horaires
DHO38	23,4 kHz	WestRhaderfehn (RFA)	
NAA	24,0 kHz	Cutler (USA)	

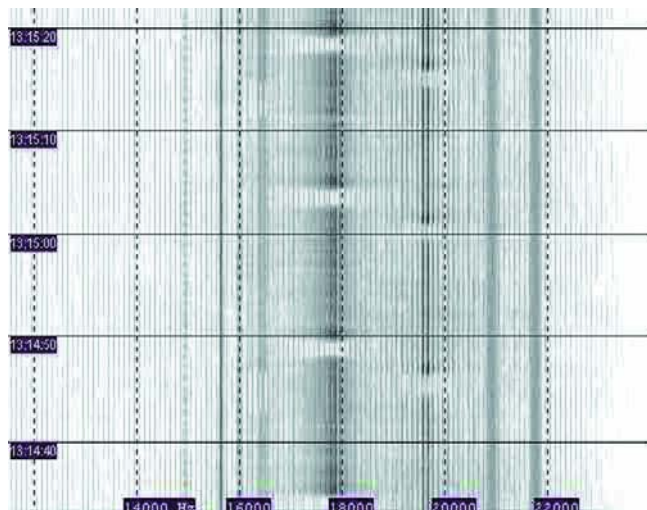


Figure 5. Exemple de spectre visualisant plusieurs signaux OTL et quelques parasites.

logiques. En ce qui concerne les applications civiles, cette plage de fréquences n'est pratiquement plus utilisée vu la taille gigantesque des antennes dont certaines requièrent une superficie supérieure à celle de la Principauté de Monaco et la faible largeur de la bande passante (quelques centaines de hertz au plus), et cela en dépit des excellentes conditions de propagation qu'elle connaît.

Comme les émissions des services militaires sont cryptées à l'aide de processus de codage très sophistiqués (qui d'entre nous se serait attendu à autre chose ?) il devrait être quasiment impossible de les décoder (ce qui est d'ailleurs interdit). Il ne reste partant, en règle générale, que la possibilité de déterminer les horaires de service (exception faite des systèmes de navigation ALPHA qui sont actifs en permanence), d'examiner l'intensité et la forme de signal des signaux et, le cas échéant, par goniométrie, de définir la direction de l'émetteur. On pourra, pour déterminer la direction de la source du signal, s'aider du classique principe de minimum de signal à l'aide d'une ferrite ou d'une antenne-cadre. Les signaux les plus spectaculaires du domaine OTL ont pour origine les émetteurs de différentes Marines. Ceux-ci travaillent en mode MSK crypté avec une largeur de bande passante de 100 à 200 Hz.

Sur les spectrogrammes, on reconnaît leurs signaux sous la forme de

lignes épaisses. La **figure 5** on reconnaît les signaux en salves (*burst*) dans le domaine OTL de JXN. GBZ et un émetteur nous identifié travaillant sur 17,8 kHz (NdIR : pourrait être Cutler, ME, une station de la marine US). On trouve en outre des signaux MSK de HWU sur 20,9 et 21,7 kHz ainsi que de GBR sur 16 kHz. La ligne située à 15,625 kHz est la fréquence de ligne horizontale d'un téléviseur situé à proximité, le signal sur 14,8 kHz étant produit par le PC.

On rencontre également d'autres types de modulation : on découvre ainsi sur les émetteurs russes à l'occasion des porteuses et des rayonnements en mode F1B, les émetteurs JXN sur 16,4 kHz et GBZ sur 19,6 kHz émettent de temps à autre des signaux en salve modulés en fréquence, les stations non identifiées trafiquant sur 15,8 et 17,8 kHz ne paraissant émettre que ce type de signaux. C'est GBZ qui a le plus grand nombre de modes de fonctionnement à son répertoire : outre MSK et signaux en salve modulés en fréquence, on découvre, occasionnellement, des trains d'impulsions qui évoquent des poissons ou des guirlandes !

Dans les états de l'ex-Union Soviétique on a émission, à des certains moments, de caractères de tampon horodateur sur plusieurs fréquences comprises entre 20 et 26 kHz. Ces émissions se sont selon un pattern relativement complexe mettant à

contribution plusieurs émetteurs dont les plus faciles à capter sont RJH69, RJH77 et RJH63. Sur les fréquences que peuvent capter les cartes-son pour PC classiques (20,5 et 23 kHz), on a émission (si l'on fait exception de RJH63) de porteuses permanentes uniquement.

Avec la mise à la retraite du système de navigation OMEGA, ALPHA, un système de navigation russe, est le seul qui soit encore opérationnel dans le domaine des OTL; il travaille sur les fréquences 11,905 kHz, 12,649 kHz et 14,881 kHz. Il émet des porteuses d'une durée de 3,6 s qui sont bien moins présentes sur les spectrogrammes que les autres émetteurs.

Parasites

On rencontre énormément de parasites dans le domaine des OTL : ils sont dus à différentes familles d'appareils électriques tels qu'alimentations à découpage, moniteurs de PC, téléviseurs, moteurs électriques, ampoules à décharge gazeuse et aussi PC. Le signal parasite le plus connu est la fréquence de balayage ligne horizontale d'un téléviseur sous tension, à savoir 15,625 kHz (25 Hz x 625 lignes).

Des parasites à fréquence variable générés par des moteurs électriques et véhiculés par le réseau secteur sont sans doute la raison des salves d'impulsions observées par l'auteur à des fréquences variant entre 17 et 18 kHz. Leur intensité également est variable et leur durée toujours de 20 s.

Même le PC utilisé pour la réception produit des parasites (comme nous le disions plus haut, le signal à 14,9 kHz visible sur les spectrogrammes reproduits a pour source le PC utilisé) qui lui sont spécifiques. Le meilleur moyen de les identifier est de découpler l'antenne. Tous les signaux qui continuent d'apparaître dans les spectrogrammes ont pour source le PC récepteur.

(020111)

NdIR :

il n'est pas exclu que lorsque vous lirez ces lignes, l'auteur de ce programme propose une version plus récente de son programme. Les options de base devraient être conservées.

Adresses Internet :

www.vlf.it et www.lwca.org

www.qsl.net/dl4yh/spectra.htm

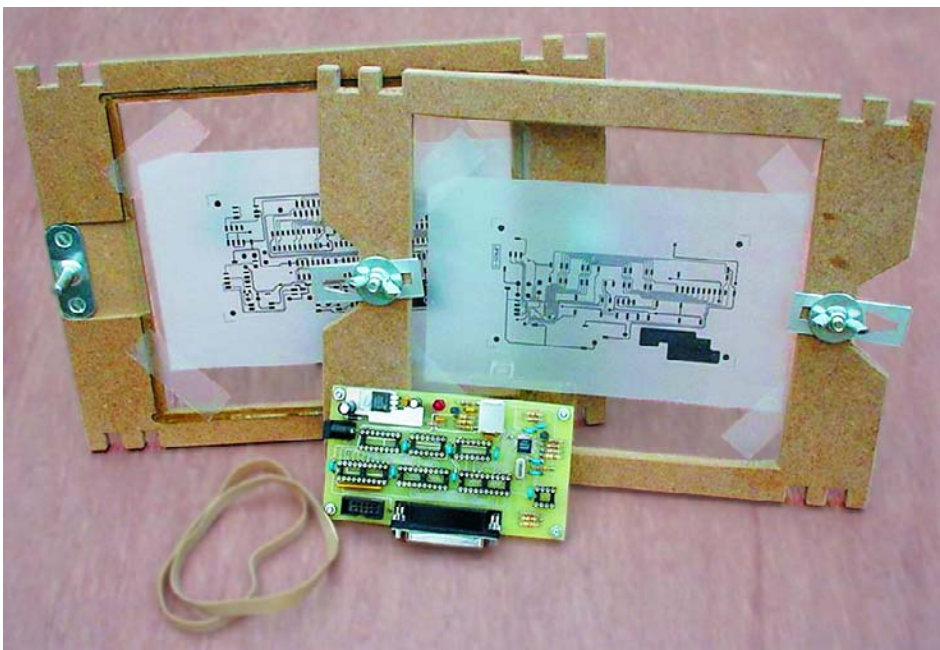
Platines double face

Faire ses gravures soi-même

Jan van de Kamer

jkamer@planet.nl

Il est possible de réaliser, à moindres frais et relativement facilement, un gabarit d'insolation qui permettra d'aligner avec précision les films requis par la fabrication d'un circuit imprimé double face.



Faire une platine soi-même ne devrait plus poser, avec les différents matériels proposés sur le marché, de problème insurmontable. Lorsqu'il s'agit d'un circuit imprimé simple face, l'insolation est une opération sans grands risques. Les choses changent du tout au tout lorsque l'on a affaire à une platine double face où il faut que les films soient parfaitement l'un en face de l'autre si l'on veut que, lors du perçage, les trous se trouvent bien au centre des îlots de cuivre utilisés pour la soudure.

La plupart des châssis d'insolation comportent des réticules de référence qui permettent de bien positionner les films. Dans la pra-

tique, il apparaît que cette technique ne donne pas la précision suffisante.

Le gabarit d'insolation

Il est possible, pour augmenter la précision, de réaliser soi-même un gabarit d'insolation qui permet d'aligner parfaitement les 2 films.

Le gabarit se compose de 2 cadres dotés d'une plaque de verre sur laquelle on pourra plaquer un film (cf. **figure 1**). Si l'on superpose les 2 cadres il est possible de décaler le cadre supérieur de manière à bien

aligner les films. La position du cadre supérieur ainsi déterminée est fixée à l'aide de d'une paire de dispositifs de fixation, les 2 cadres pouvant ensuite être séparés de manière à pouvoir intercaler la platine à insoler.

On pourra ensuite passer à l'étape de l'insolation, opération au cours de laquelle on pourra choisir entre une insolation simultanée des 2 côtés ou une insolation simple face, pour procéder à un retournement du gabarit à mi-temps de cette opération. Les dimensions données au gabarit d'insolation sont telles qu'il pourra aisément être placé sur un châssis d'insolation à UV dont on aura enlevé la plaque d'origine. On pourra bien entendu adapter les dimensions du gabarit en fonction du type de châssis d'insolation avec lequel on prévoit de travailler.

Des pièces et des morceaux

Les cadres sont réalisés en bois aggloméré de 8 mm d'épaisseur. Ce matériau est facile à travailler sans produire d'échardes. On découpera les 2 cadres à la scie égoïne. Il est préférable d'attendre, avant de découper les espaces destinés à recevoir le verre, d'avoir fraisé les rainures. Les zones grisées de la figure 1 devront être fraisées à 3 mm de profondeur à l'aide d'une fraise

descendante. Son diamètre n'a pas une importance capitale, 6 mm nous paraissant une bonne taille. Si vous n'avez pas de fraise à votre disposition, vous pourrez utiliser un ciseau à bois et un couteau bien aiguisés.

Une fois que l'on aura fraisé les zones représentées en gris on pourra scier les endroits où viendra prendre place le verre. On pourra également percer les différents orifices destinés aux boulons. Il restera à limer les angles et les bords en biseau pour donner à l'ensemble une bonne finition.

L'étape suivante est celle de la mise en place du verre. On pourra utiliser du verre à vitre ordinaire que l'on aura fait couper aux dimensions dans son magasin de bricolage préféré ou chez le peintre de ses amis. L'épaisseur, 4 mm, est elle un facteur important vu que le but de la manoeuvre est de faire en sorte que le verre dépasse le plan des cadres en contreplaqué. On assure ainsi un bon contact des films l'un contre l'autre lors du positionnement. Si le verre devait se trouver en retrait, ce seraient les cadres en bois qui seraient en contact l'un avec l'autre, laissant un interstice d'air entre les 2 films, ce qui rendrait impossible tout ajustage précis. Après avoir dégraissé le verre à l'aide d'un tampon imbibé d'alcool ou de benzine, on pourra le coller dans le cadre à l'aide d'un rien de colle à 2 composants.

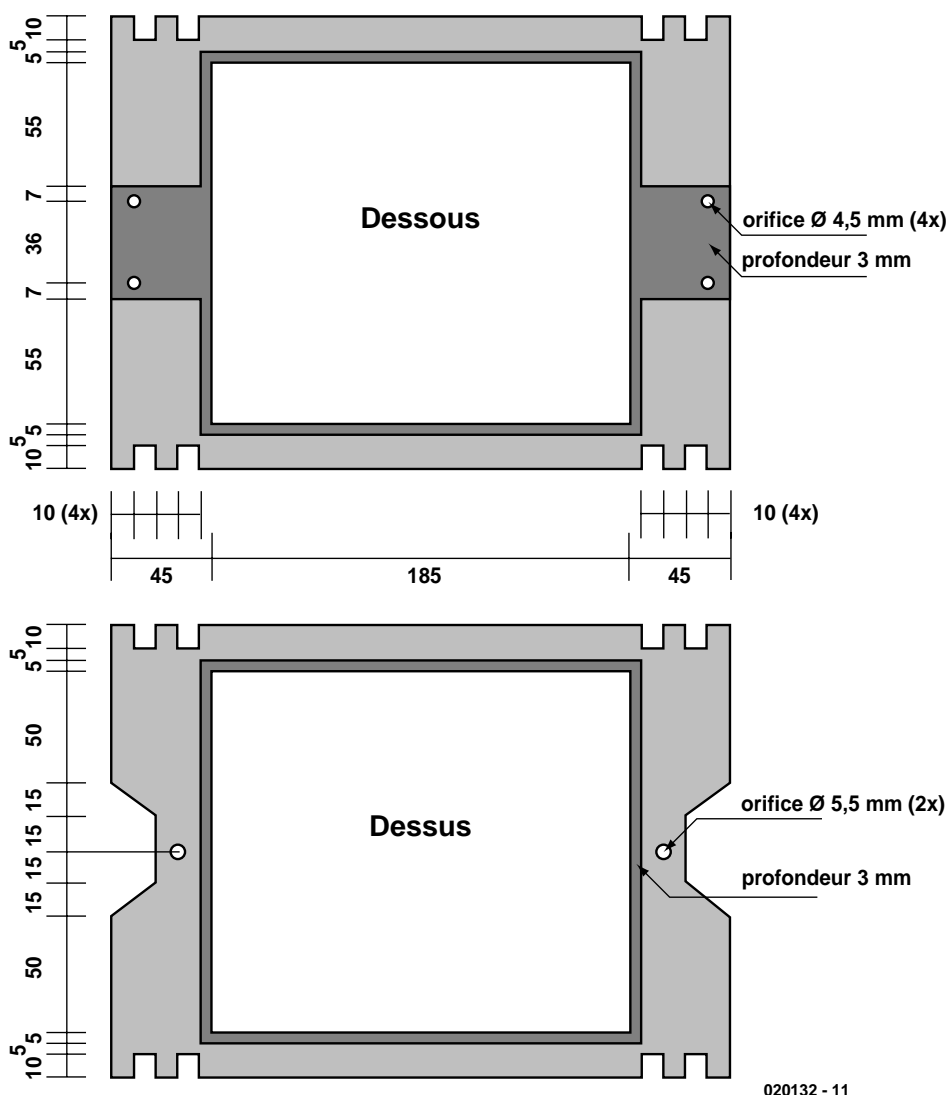


Figure 1. Dessin coté des 2 cadres.

L'alignement

La partie la plus importante de notre gabarit d'insolation est son mécanisme d'alignement. Il ne doit pas présenter le moindre jeu, être ajustable dans des limites déterminées et garder sa position lorsque les 2 cadres ont été serrés l'un contre l'autre. Ce cahier des charges paraît plus difficile à respecter que cela n'est en fait le cas dans la pratique. La broche d'alignement pourra prendre la forme d'un dispositif d'ouverture de fenêtre. Ce dont nous avons besoin est une broche droite de 6 à 8 mm de diamètre soudé sur un support métallique. La partie supérieure de cette broche doit se terminer légèrement en biseau vu que cela permet un positionnement plus facile du cadre supérieur après

le processus de fixation.

Il vous faudra faire vous-même la seconde partie du mécanisme d'alignement à l'aide d'un morceau d'aluminium plat. La **figure 2** en donne les cotes. L'orifice trapézoïdal de gauche

viendra se glisser sur la broche d'alignement. On aurait pu faire un orifice parfaitement rond, mais cela aurait impliqué qu'il ait exactement le diamètre de la broche pour éviter tout jeu. Ceci est pratiquement impossible à réaliser, raison pour laquelle nous avons opté pour cette

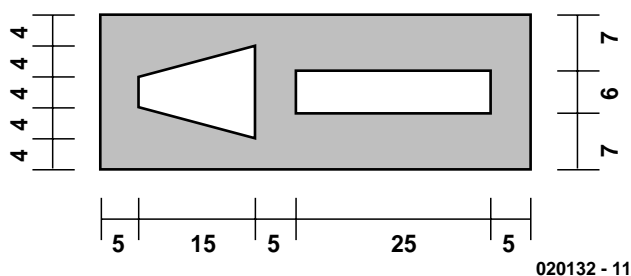


Figure 2. La réglette du mécanisme d'alignement.

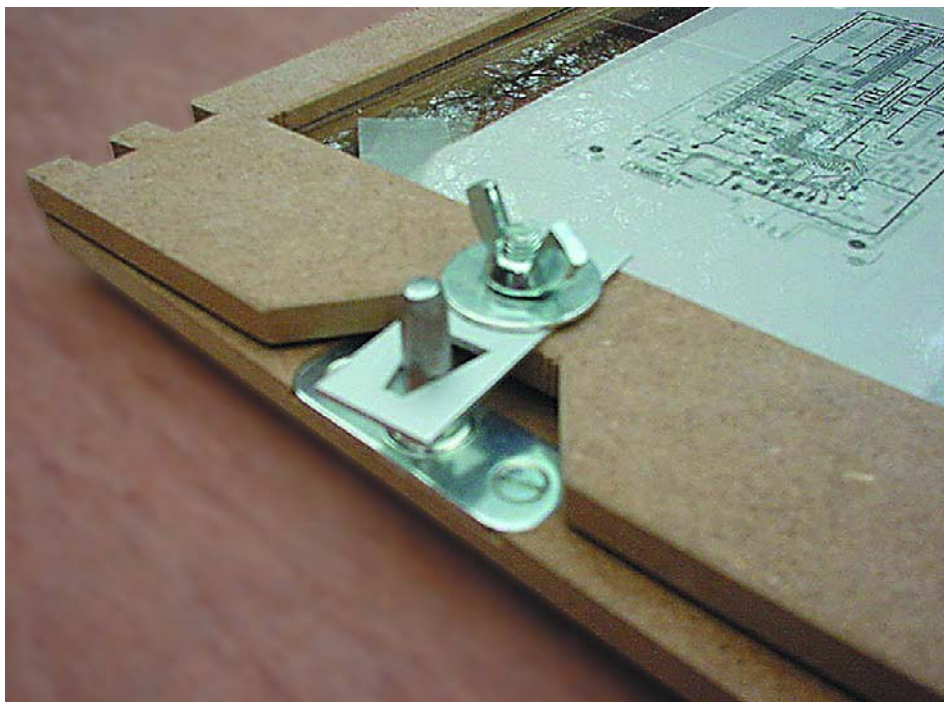


Figure 3. Vue rapprochée de l'ensemble broche, languette et écrou à ailettes.

forme en V. Les côtés en biseau permettent de corriger les tolérances dans les axes X et Y.

Il est possible de faire glisser dans l'encoche horizontale à droite la réglette de la figure 2 le long du boulon monté dans le cadre supérieur. Il vous faudra réaliser 2 exemplaires de cette réglette. Il est facile de découper les encoches dans la plaquette d'aluminium à l'aide d'une scie à chantourner. Il vous faudra à nouveau limer avec soin les bords en biseau pour éliminer toutes les ébarbures.

La construction

Vous disposez maintenant de toutes les pièces requises pour effectuer le montage du cadre

Commencez par fixer les 2 broches d'alignement sur le cadre inférieur, à l'aide, de préférence, de vis M4 biseautées (de 10 mm de long environ) et d'écrous borgnes. Le cadre supérieur sera doté de 2 boulons M5 de blocage de 20 mm de long; un coup de marteau bien placé en assurera la fixation dans l'orifice. Si l'on ne veut pas risquer de briser le verre on pourra, à l'aide d'une petite lime carrée, donner une forme carrée à l'orifice.

Sur chacun des boulons on glisse la plaquette d'aluminium sur laquelle on pose une grosse rondelle et un écrou à ailettes de manière à pouvoir la resserrer et la desserrer facilement. La **figure 3** montre une vue rapprochée du mécanisme d'alignement de notre prototype. Votre gabarit d'insolation est prêt à remplir la fonction pour laquelle il a été conçu.

Les films

Le film destiné à la couche de cuivre du côté « composants » pourra être fabriqué selon le même processus que celui destiné à la réalisation du côté « soudures ». Il faudra cependant que ce film soit imprimé en miroir. Il est en outre recommandé de supprimer les îlots non utilisés. Dans le cas de supports de circuits intégrés en particulier il est difficile de savoir quelles sont les broches à souder une fois que le support a été enfiché dans la platine. Si l'on a éliminé tous les îlots excédentaires, il faudra partant souder tous les îlots restants une fois l'implantation des composants terminée. Certains programmes de dessin de platine connaissent cette option. Layo1 par exemple offre une possibilité d'exportation du projet sous la forme de différentes couches où l'on pourra supprimer manuellement les îlots inutiles.

Mise en oeuvre du gabarit d'insolation

Une fois les 2 films imprimés on pourra les découper à l'aide d'un cutter acéré. Veillez à garder une marge suffisante tout autour de la taille finale du circuit imprimé. Les coins des films sont collés sur la

plaque de verre à l'aide de morceaux de ruban adhésif (scotch). Il faut ensuite éviter que l'un des angles de la platine ne se trouve sur l'adhésif, ce qui se traduirait par l'existence d'un film d'air entre le film et la couche photosensible, ce qui pourrait se traduire par une surinsolation des pistes à ce niveau.

Une fois les films collés sur la plaque de verre, on desserre les écrous à ailettes de manière à permettre le mouvement des réglettes en aluminium. On dispose les 2 cadres l'un sur l'autre en faisant glisser les orifices en V des réglettes sur les broches. Il ne reste plus qu'à superposer les 2 films parfaitement l'un sur l'autre en faisant glisser le cadre supérieur. Une fois que l'on a trouvé le positionnement correct il faut glisser les réglettes jusqu'à ce qu'elles soient bien calées sur les broches. Bien serrer les écrous à ailettes et le processus d'alignement est terminé. Dans la chambre noire la platine est débarrassée de ses 2 films de protection. On peut ensuite séparer le cadre supérieur du cadre inférieur en veillant à ce que les réglettes ne bougent pas d'un poil ! La platine est positionnée sur le film du dessous (ne pas la placer sur les morceaux de ruban adhésif); il reste à replacer le cadre supérieur qui sera automatiquement mis dans la position correcte.

Les films sont pressés avec force vers les 2 côtés de la platine par la mise en place d'une paire d'élastiques de bonne épaisseur; c'est à cet effet que servent les encoches rectangulaires en créneau que comporte les 2 cadres du gabarit.

En guise de conclusion

Après insolation, développement, gravure et perçage nous avons réalisé une belle platine où les orifices de trouent parfaitement, des 2 côtés, au milieu des îlots. On pourra réaliser la métallisation des orifices en soudant aux emplacements stratégiques les composants sur les 2 faces de la platine.

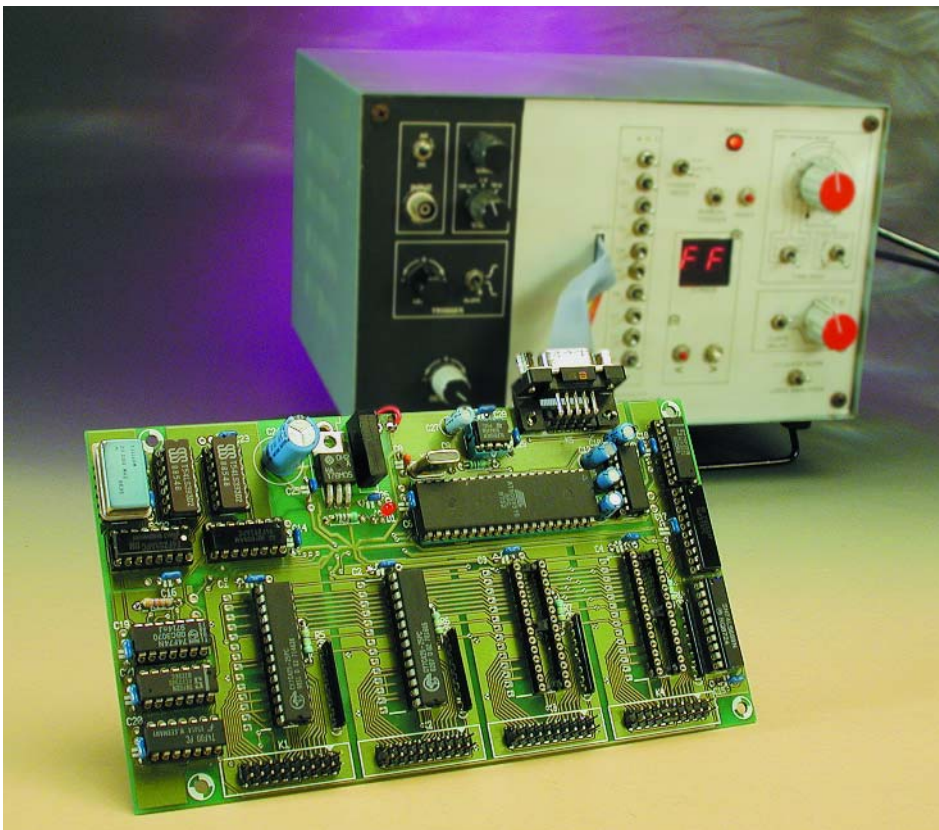
(20132)

Analyseur logique 20/40 MHz à géométrie variable

Quand simplicité rime avec efficacité

Projet : Benoît Bouchez

En électronique analogique, un oscilloscope suffit pour résoudre la majorité des problèmes qui peuvent survenir durant la conception et le dépannage de circuits.



Cependant, dans le cas de cartes équipées de microprocesseurs, DSP et autres composants numériques, l'oscilloscope devient quasiment inutilisable, avec sa bande passante trop limitée pour les signaux très rapides dont sont coutumiers ces composants, et surtout l'impossibilité de surveiller plus de deux signaux simultanément (quatre pour les chanceux qui peuvent accéder à des oscilloscopes 4 voies). Qui plus est, l'oscilloscope ne permet de surveiller que des signaux récurrents, ce qui n'est pas vraiment le cas des signaux couramment rencontrés autour de nos chers processeurs.

Si l'oscilloscope (y compris les modèles à mémoire numérique) s'est assez largement démocratisé ces dernières années, on ne peut pas en dire autant de l'analyseur logique, qui reste un outil coûteux, surtout pour les amateurs (et même pour les entreprises...).

NdIR. : Les lecteurs des premières années d'Elektor n'auront pas manqué de reconnaître, à l'arrière plan de cette photo, notre premier analyseur logique, qui date de 1981.

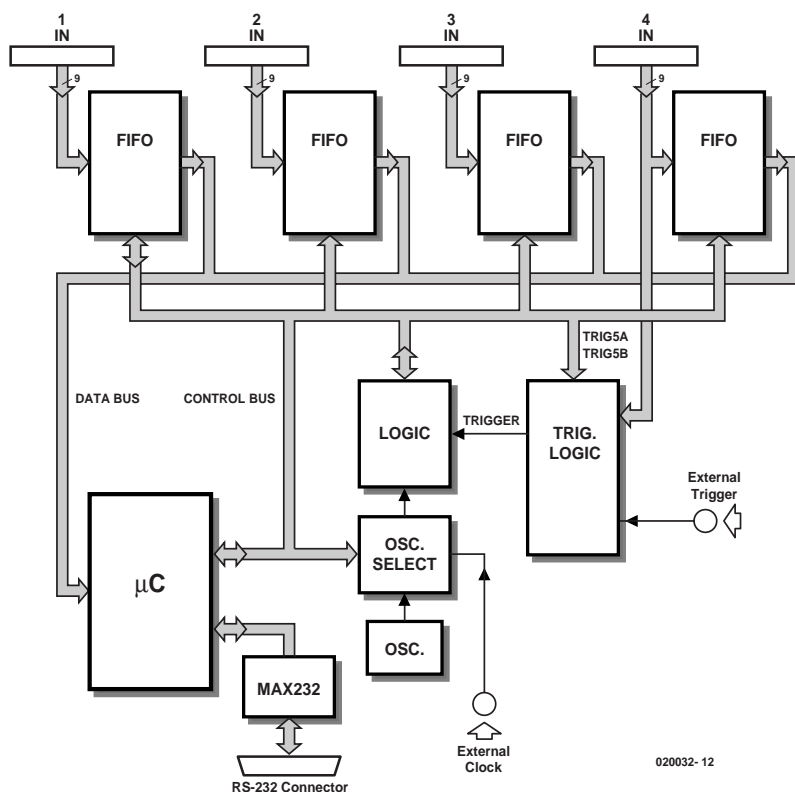


Figure 1. La structure d'un analyseur logique fait appel à une circuiterie logique complexe que les compteurs intégrés dans les FIFO permettent de simplifier considérablement.

Analyseur logique, mais encore...

Il n'est pas dans nos intentions de douter des connaissances techniques de nos lecteurs, mais il nous semble bon de rappeler la fonction des analyseurs logiques, ne serait-ce que pour en confirmer l'utilité à côté de son cousin l'oscilloscope.

Un analyseur logique n'est autre qu'un enregistreur de signaux numériques (donc ne pouvant valoir que « 1 » ou « 0 »), sur un certain nombre de voies simultanées, généralement par groupes de 8 (9 dans le cas présent). Contrairement aux oscilloscopes, les analyseurs logiques n'enregistrent qu'un seul cycle (équivalent du mode mono-coup ou *one-shot* sur les oscilloscopes), après déclenchement.

Dans un analyseur logique, la vitesse d'enregistrement (équivalente à la vitesse de balayage horizontal de l'oscilloscope) est définie soit par une horloge interne, soit par une horloge externe. À chaque front montant (voire descendant) de cette horloge, un nouvel échantillon numé-

rique est enregistré simultanément sur les n voies.

Autre point spécifique aux analyseurs logiques, l'enregistrement peut être déclenché soit manuellement, soit sur un événement spécifique, réglable par avance. On peut ainsi forcer l'appareil à attendre une certaine condition, généralement formée par un motif binaire qui doit être présent sur certaines entrées. Sur un oscilloscope, ce réglage se limite à détecter une certaine tension en entrée, variant dans une direction donnée (*trigger*).

Enfin, la majorité des analyseurs logiques est capable de travailler en *pre-trigger*, c'est-à-dire en acquisition bouclée. Lorsque le déclenchement est validé, cette acquisition se prolonge pour un certain nombre de cycles avant de s'arrêter. Grâce à cette astuce, il est ainsi possible de savoir ce qui s'est passé avant le déclenchement.

De par ces caractéristiques, l'analyseur logique est un outil indispensable à qui veut surveiller ce qui se passe sur les lignes d'adresses, de données (ou toute autre chose) de la

carte à microprocesseur et/ou à circuits logiques qu'il teste.

Un petit air de FIFO...

Vous l'aurez compris en lisant la description d'un analyseur logique dans les lignes qui précèdent, le cœur de ces appareils est formé de mémoires RAM, qui vont enregistrer sur leur bus de données les événements logiques présentés à l'entrée, pour les restituer par la suite à volonté.

Les lecteurs fidèles d'ELEKTOR se rappelleront peut-être de l'analyseur logique décrit au mois de Juin 1981 (ça ne nous rajeunit pas...), qui était probablement le premier appareil de ce genre mis à la portée des amateurs de l'époque. Il s'agissait d'un véritable monument, entièrement en circuits TTL LS...

En effet, le problème qui se pose durant la conception de l'unité mémoire d'un analyseur logique est qu'il faut non seulement générer les signaux de commande de mémorisation, mais surtout, il faut gérer l'incrémention des lignes d'adresse après chaque stockage, ce qui donne un circuit final pour le moins dense.

Dans les analyseurs « classiques », on fait appel pour cela à des compteurs ultrarapides, reliés à l'horloge interne ou externe. En entrelaçant plusieurs compteurs avec des bancs mémoires distincts, on peut même enregistrer des signaux logiques à des fréquences dépassant les limites (en termes de temps d'accès) des mémoires (c'est le secret des analyseurs logiques industriels –chut, ne le dites à personne...)

Dans notre cas, la conception de cette logique d'adressage se voulait simple (pas d'entrelacement). Hors, la réalisation des compteurs en logique discrète nécessite un nombre assez élevé de circuits, non seulement pour les compteurs eux-mêmes, mais aussi pour les multiplexeurs d'adresses (il faut bien relier les mémoires par la suite).

Afin de nous simplifier la vie au maximum, nous avons donc fait appel à des circuits mémoire qui contiennent d'origine ces compteurs : les mémoires FIFO (acronyme de *First-In First-Out*)

Assez peu connus du grand public, ces composants ont une structure interne assez complexe, reprise en **figure 2**. Contrairement aux mémoires classiques, qui disposent d'un bus de données bidirectionnel, les FIFO utilisent deux bus distincts, l'un pour l'écriture, l'autre pour la lecture (on rentre d'un côté et on sort de l'autre).

On remarque aisément l'absence de bus d'adresses, d'un côté comme de l'autre, au moins à l'extérieur du composant. En fait, les adresses sont générées de façon interne par

deux compteurs indépendants, l'un pour l'écriture, l'autre pour la lecture. Après remise à zéro (*reset*) d'un FIFO, les deux compteurs sont initialisés à la valeur 0.

Lorsqu'on souhaite mémoriser une donnée dans le FIFO, il faut présenter celle-ci sur le bus d'écriture, puis passer la ligne \overline{WR} à 0. La donnée est alors enregistrée dans la cellule RAM pointée par le compteur d'écriture.

Lorsque le signal \overline{WR} remonte à l'état 1, le compteur est alors incrémenté, et ainsi de suite à chaque cycle d'enregistrement. Côté lecture, la logique de fonctionnement est exactement la même. Chaque fois que le signal \overline{RD} sera mis à 0, la donnée pointée par le compteur de lecture est présentée sur l'autre bus de données, en lecture seule. À la remontée de \overline{RD} , le compteur de lecture est incrémenté à son tour.

Bien entendu, ces opérations de lecture et d'écriture peuvent survenir à des vitesses totalement différentes on dit que les FIFO sont asynchrones.

Pour illustrer le fonctionnement de la mémoire, un petit dessin vaut mieux qu'un long discours. Sur la **figure 3**, on voit un FIFO après quatre cycles d'écriture. Les quatre premières cellules mémoires ont été remplies avec des données, dans l'ordre de leur arrivée (\$AA est arrivé en premier, suivi de \$55, puis de \$00, et enfin \$FF dans notre exemple).

Arrivé à ce point, le processeur situé côté lecture active son signal \overline{RD} : comme le compteur de lecture est à 0, le bus de données lecture renvoie alors la valeur \$AA, qui était la première entrée, puis le compteur est incrémenté automatiquement à la remontée du signal \overline{RD} (fin du cycle de lecture). Au cycle suivant, le compteur de lecture valant maintenant 1, le processeur recevra la valeur \$55, qui a été mémorisée en deuxième place, et ainsi de suite, le compteur s'incrémentant à chaque fois.

On constate donc que les données ressortent côté lecture dans le même ordre qu'elles étaient entrées côté écriture, d'où le nom de FIFO : *First-In First-Out* soit premier entré, premier sorti...

Imaginons maintenant que le FIFO reçoit un nouvel ordre d'écriture, alors que le processeur de lecture est en train de faire son travail de son côté. Le compteur d'écriture était arrivé à l'adresse 4. La donnée est donc enregistrée à cette adresse, sans que le processeur de lecture ne soit perturbé de quelque façon que ce soit.

Dans notre exemple, lorsque le processeur aura fait cinq cycles de lecture, le compteur de lecture sera arrivé à la même valeur que le compteur d'écriture. Il n'y a plus rien à lire, le FIFO est vide. Cet état de fait est alors signalé automatiquement par la ligne \overline{FE}

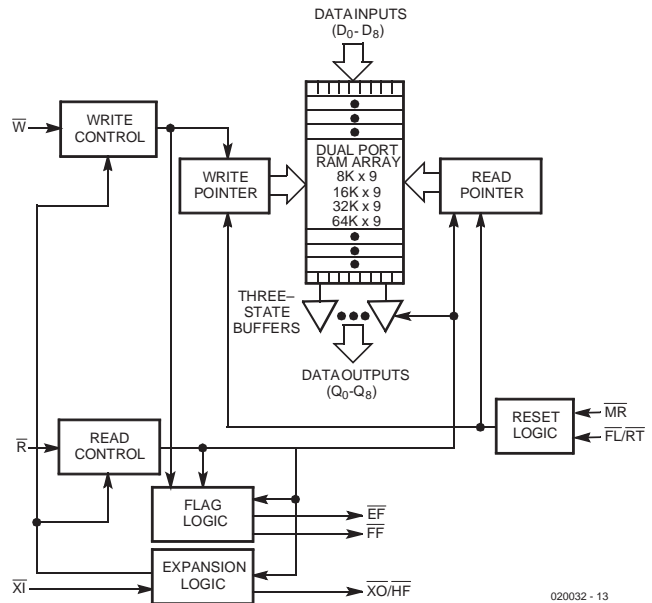


Figure 2. Structure interne d'une mémoire FIFO (Source Cypress).

(*FIFO Empty*), qui passe à 0. Il peut également se produire le phénomène inverse, à savoir que le compteur d'écriture rattrape le compteur de lecture, après rebouclage (la mémoire étant vidée moins vite qu'elle ne se remplit). Il n'y a

plus de place dans la mémoire : celle-ci est pleine. Dans ce cas, le compteur d'écriture se bloque automatiquement (jusqu'à ce que le compteur de lecture se remette à avancer, indiquant la libération de cellules mémoire). La ligne \overline{FF} (*FIFO*

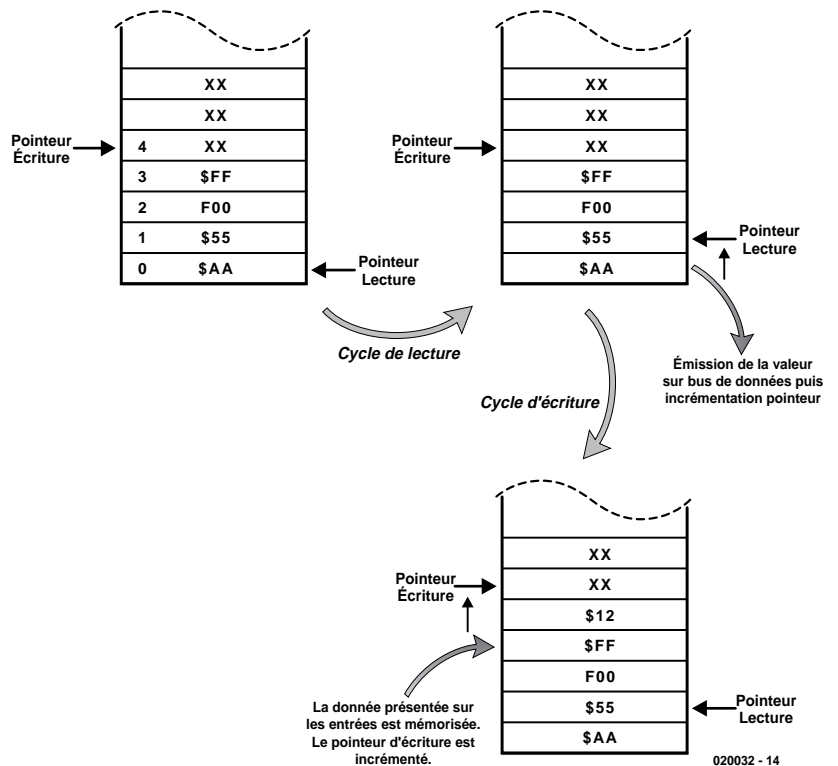


Figure 3. Principe de fonctionnement d'un FIFO asynchrone : les cycles de lecture et d'écriture peuvent être exécutés sans avoir à respecter un ordre précis, ni une horloge commune.

Full) est alors mise à 0 pour indiquer la saturation.

On le voit, les FIFO intègrent toute une logique qui simplifie considérablement leur utilisation. Ces mémoires sont en général utilisées comme tampon entre deux processeurs qui doivent s'échanger des données, à des vitesses différentes, et sans synchronisation.

Les compteurs rebouclant sur eux-mêmes chaque fois que la limite de mémoire est atteinte (ce qui ne signifie pas que la mémoire est pleine –voir ci-dessus), si on souhaite s'échanger des blocs de données de longueur variable, il est impossible de retrouver par lecture de ces compteurs le début ou la fin d'un tel bloc. C'est pour cette raison que les FIFO ne sont pas architecturés sur 8 bits, mais sur 9 bits.

Bien que ce ne soit pas une obligation, il est possible d'utiliser ce bit supplémentaire comme marqueur de début et de fin de bloc, permettant au processeur en lecture de « retrouver ses petits ».

Passons aux choses sérieuses

Après cette présentation des FIFO, nous allons maintenant nous attaquer au schéma de notre analyseur, qui vous paraîtra du coup plus simple à comprendre, tout étant quasiment intégré dans les FIFO.

Comme vous le voyez sur la **figure 4**, notre circuit est clairement architecturé autour de quatre de ces composants. Rien ne vous oblige par ailleurs à installer tous les FIFO. Notre analyseur peut fonctionner avec 1, 2, 3 ou 4 FIFO installés, sans aucune difficulté, d'où son caractère « à géométrie variable ».

Encore plus fort, sachez qu'il existe d'innombrables modèles de FIFO, dont la capacité va de 1 Kmot (quelqu'un connaît-il sinon le nom d'un « octet » de 9 bits ?) jusqu'à 16 Kmots (toujours de 9 bits !). Or, tous ces circuits ont exactement le même brochage (mais pas le même prix...). La géométrie variable de notre analyseur ne se limite donc pas seulement à la largeur (variable de 9 à 36 bits), mais s'applique aussi à la longueur. Voilà une bonne nouvelle pour nos comptes en banque d'électroniciens !

Enfin, comme nous le verrons dans la partie consacrée à la réalisation, il est possible de construire l'analyseur en version « allégée » à 20 MHz, ou en version 40 MHz.

Comme l'introduction vous l'a montré, les compteurs d'écriture sont intégrés dans les FIFO. Bien qu'ils n'apparaissent pas sur le schéma, il faut quand même les alimenter avec un signal d'horloge (ligne *FIFO WR*). Pour obtenir cette horloge d'écriture, on va utiliser un oscillateur programmable, construit autour d'un oscillateur à quartz (IC19) suivi d'une batterie de diviseurs binaires (IC8 et IC9).

La sélection de la fréquence d'acquisition se fait grâce à des multiplexeurs (IC10 et IC12), commandés par le processeur local (lignes *CLK Sx*). Cas particulier, la dernière entrée d'IC10 n'est pas reliée aux diviseurs, mais à une entrée externe (*EXT CLOCK*), permettant d'utiliser une fréquence d'acquisition autre que celles disponibles sur la carte.

Le signal d'horloge arrive aux FIFO après validation par IC6.A. À chaque front descendant de *SYSCLK*, les FIFO mémoriseront donc la donnée présente sur leur bus à cet instant, donnée qui vient en fait du « monde extérieur », par l'intermédiaire des connecteurs d'entrée K1 à K4. Nous reviendrons plus loin sur les diverses particularités de ces entrées.

Nous savons que l'horloge était conditionnée par IC6.A, et plus précisément par la ligne *ACQ ACT* sur le schéma. Pour valider cette ligne, il faut basculer le verrou (*latch*) formé par IC7.A et IC7.B. Ce basculement ne peut s'obtenir que si la sortie d'IC5.B passe à 1. Pour y arriver, il y a deux possibilités : soit forcer cette sortie par activation de la ligne *START* (cette fonctionnalité sera utilisée pour le déclenchement manuel), soit mettre à 1 la ligne *GO SYNC*. Dans ce cas, IC5.B ne basculera que si la ligne *TRIG* s'active (cette fonctionnalité sera utilisée pour le déclenchement automatique).

La bascule IC5.B peut par ailleurs être remise à 0 (ce qui n'entraîne pas la retombée du verrou autour d'IC7) par le processeur, pour interdire le redéclenchement.

Tant que le verrou IC7 est armé, le signal d'horloge est validé, et les

FIFO sont en acquisition. Il arrivera toutefois un moment où ces derniers seront remplis (le processeur local ne relit pas les FIFO durant l'acquisition). Comme nous l'avons vu juste avant, la ligne *FF* s'active alors. Cette ligne, après inversion par IC6.C, valide la bascule IC5.A, qui basculera alors au prochain front montant de l'horloge. Sa sortie passant à 1, elle bloque alors le verrou autour d'IC7 par la ligne *ACQ DONE* (qui sert à prévenir le processeur que le cycle d'acquisition est terminé). La ligne *ACQ ACT* retombant alors, les signaux d'horloge ne peuvent plus parvenir aux FIFO, qui stoppent alors toute acquisition (de toute façon, quand un FIFO est plein, il refuse d'enregistrer toute nouvelle donnée). Le processeur local a alors tout le loisir de relire tranquillement le contenu des FIFO, qui représente les signaux extérieurs enregistrés.

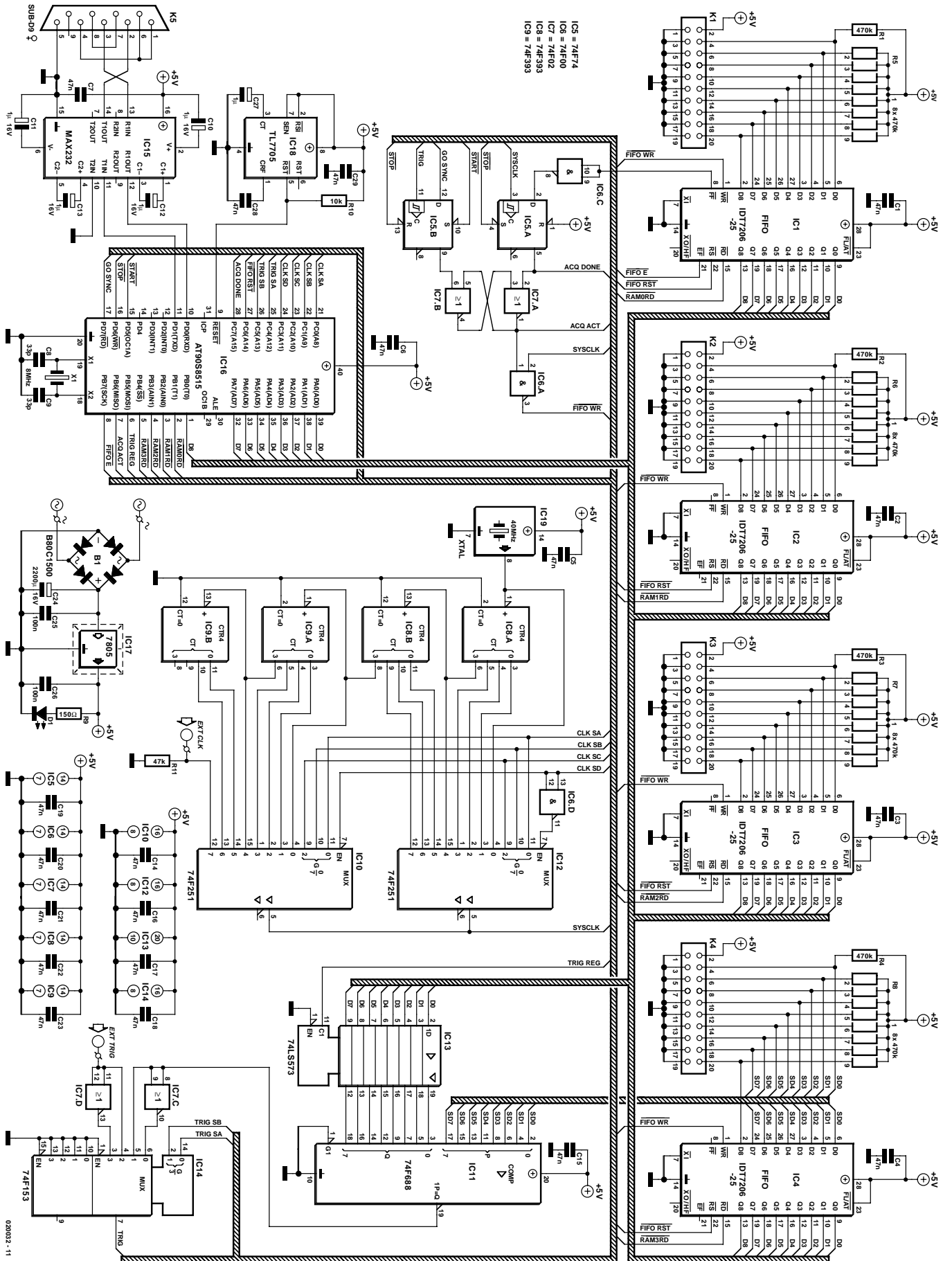
Pour cette opération de relecture, le processeur local, que nous étudierons plus loin, ne dispose que d'un bus 8 bits, alors que la largeur d'acquisition peut aller jusqu'à 36 bits. Le processeur va simplement sélectionner les mémoires l'une après l'autre, par les lignes *RAM0RD*, *RAM1RD*, *RAM2RD* et *RAM3RD*. Le 9^{ème} bit, utilisé ici comme bit de donnée supplémentaire, est traité par une ligne d'E/S dédiée du processeur.

Dernier point à étudier : la synchronisation (le *trigger* de nos amis anglophones). Nous avons vu que nous avons prévu deux possibilités : soit le trigger manuel (on force l'analyseur à démarrer immédiatement), soit le trigger automatique. Dans ce dernier cas, il faut appliquer à l'entrée 11 d'IC5 un front montant (moyennant la validation par le processeur de la broche 12).

Ce front montant peut provenir de diverses sources, sélectionnées par le multiplexeur IC14. La première source est tout simplement l'entrée *EXT TRIG* prise en direct. Cette entrée permet donc un déclenchement de l'analyseur sur front montant externe. L'entrée *EXT TRIG* étant reliée à une autre entrée de IC14 par une porte montée en inverseur, IC7.D, il est donc possible de déclencher l'analyseur sur front descendant externe.

Autre possibilité : le déclenchement interne. Dans ce mode, il faut que l'analyseur détecte une condition logique, formée par la présence d'un octet d'une valeur prédéterminée sur un certain nombre d'entrées. Dans les analyseurs logiques haut de gamme, il est possible de sélectionner le nombre d'entrées à contrôler et leur état respectif. Notre analyseur se voulant simple, le nombre d'entrées est fixé à 8, sans masquage possible.

Le déclenchement interne est obtenu par comparaison des 8 premières entrées sur K4 (qui servent aussi d'entrées d'analyse), avec



IC5 = 74F74
 IC6 = 74F00
 IC7 = 74F02
 IC8 = 74F393
 IC9 = 74F393

Figure 4. L'électronique de notre analyseur logique.

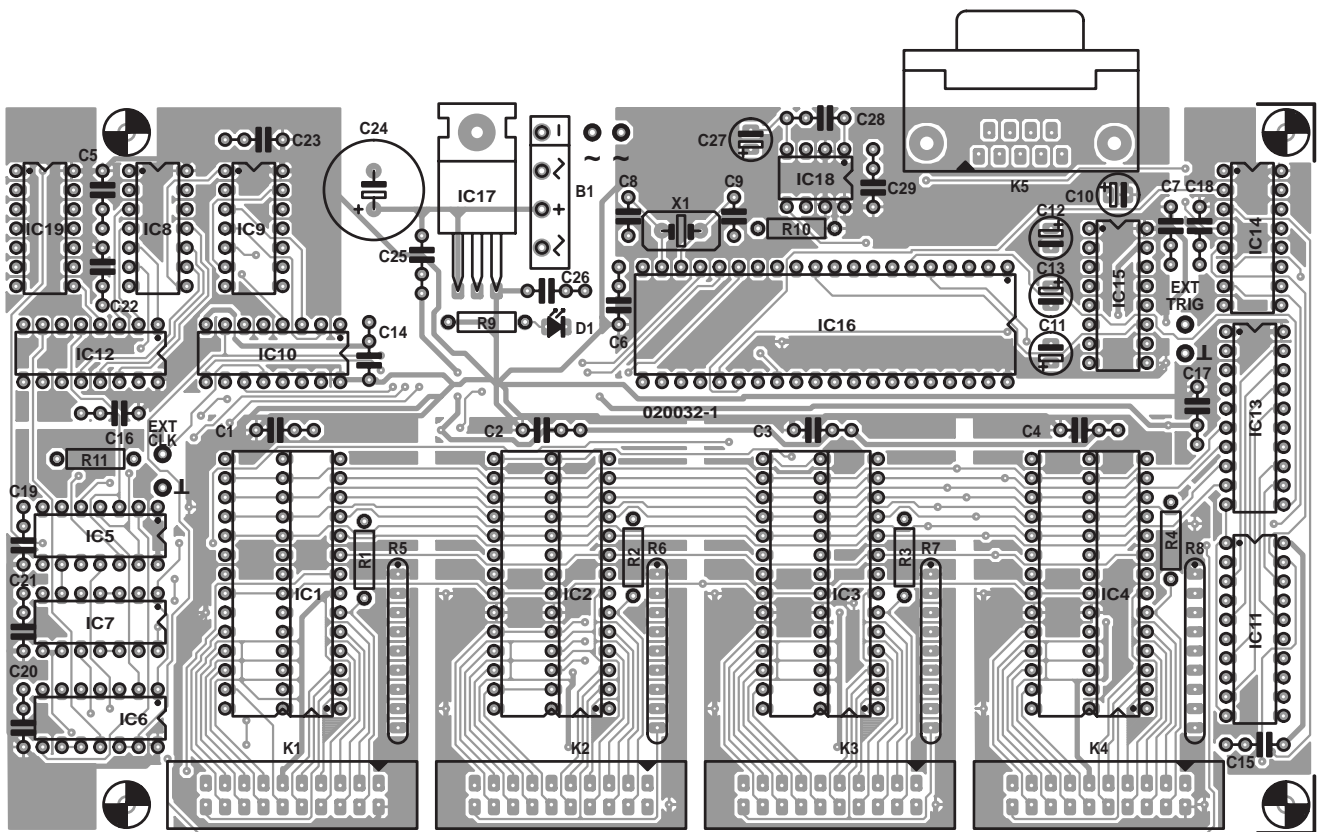


Figure 5a. Sérigraphie de l'implantation des composants de la platine de l'analyseur logique.

Liste des composants

Résistances :

R1 à R4 = 470 kΩ
 R5 à R8 = réseau de 8 résistances de 470 kΩ
 R9 = 1kΩ
 R10 = 10 kΩ
 R11 = 47 kΩ

Condensateurs :

C1 à C7, C14 à C23, C28,
 C29 = 47 nF
 C8, C9 = 22 pF
 C10..C13 = 1 μF/16 V radial
 C24 = 2 200 μF/16 V radial
 C25, C26 = 100 nF
 C27 = 1 μF/16 V radial

Semi-conducteurs :

B1 = B80C1500 (~ ~ + ~)
 D1 = LED à faible courant rouge
 IC1 à IC4 = IDT7206 (-15, etc. cf. texte) ou CY7C425-15PC

IC5 = 74F74
 IC6 = 74F00
 IC7 = 74F02
 IC8, IC9 = 74F393
 IC10, IC12 = 74F251
 IC11 = 74F688
 IC13 = 74LS573
 IC14 = 74F153
 IC15 = MAX232
 IC16 = AT90S8515-8PC (programmé EPS 020032-41)
 IC17 = 7805
 IC18 = TL7705-ACP
 IC19 = bloc oscillateur 40 ou 20 MHz en boîtier DIL14 (cf. texte)

Divers :

K1 à K4 = au choix, embase ou connecteur HE-10 à 2 rangées de 10 contacts (en équerre le cas échéant)
 K5 = embase Sub-D femelle encartable en équerre
 X1 = quartz 8 MHz

l'apparition d'un code binaire sur K4 identique au code mémorisé entraîne le déclenchement de l'analyseur. Si le déclenchement sélectionné est « interne inverse » (broche 6 de IC14), c'est la *disparition* du code binaire sur K4 (toujours par rapport au code mémorisé sur IC13) qui déclenche l'analyseur.

À noter qu'il est possible de contourner la limitation imposée par l'absence de masque de configuration sur le *trigger*. Les entrées sur K4 (comme sur les autres connecteurs d'ailleurs) sont rappelées à 1 par un réseau de résistances. Pour ignorer certaines entrées, il suffit donc de laisser les entrées correspondantes en l'air, et de programmer des « 1 » dans l'octet de configuration du *trigger*.

Dernière partie, et non moins importante : le processeur local de contrôle. Ce processeur est chargé de générer les différents signaux de commande en fonction des ordres qui seront transmis depuis le micro-ordinateur chargé de visualiser les signaux. Nous avons choisi ici un microcontrôleur RISC de chez Atmel, un AT90S8515, cadencé à 8 MHz et qui convient très largement à notre application. Pour ceux qui arriveraient à s'en procurer, il est également possible d'utiliser un AT90S4414 (version avec moins de mémoire, moins coûteuse, mais plus rare).

une valeur donnée par le processeur local et mémorisée par IC13. Lorsque le code présent sur les entrées est le même que celui programmé dans IC13, la sortie de IC11 passe à l'état

1. Cette sortie peut être sélectionnée par IC14 en tant que source de déclenchement. Si le déclenchement interne sélectionné est direct (broche 5 de IC14),

Ce composant intègre absolument tout ce dont nous avons besoin pour piloter les différentes sections du circuit : 8 Koctets de mémoire Flash pour le programme, 512 octets de RAM, un port série asynchrone, un coeur ultra-rapide, etc. D'autres fonctionnalités, tels que les temporisateurs (*timers*) ou

l'EEPROM de sauvegarde en Flash elles aussi sont également disponibles, mais ne sont pas utilisées sur ce circuit.

Le microcontrôleur est accompagné d'un circuit de Reset spécialisé (IC18), un TL7705, qui garantit la

fourniture d'impulsions de réinitialisation parfaitement calibrées lorsque cela est nécessaire (mise en service du circuit).

IC15, un classique MAX232 (ou équivalent), est chargé de l'adaptation des niveaux de tension entre le

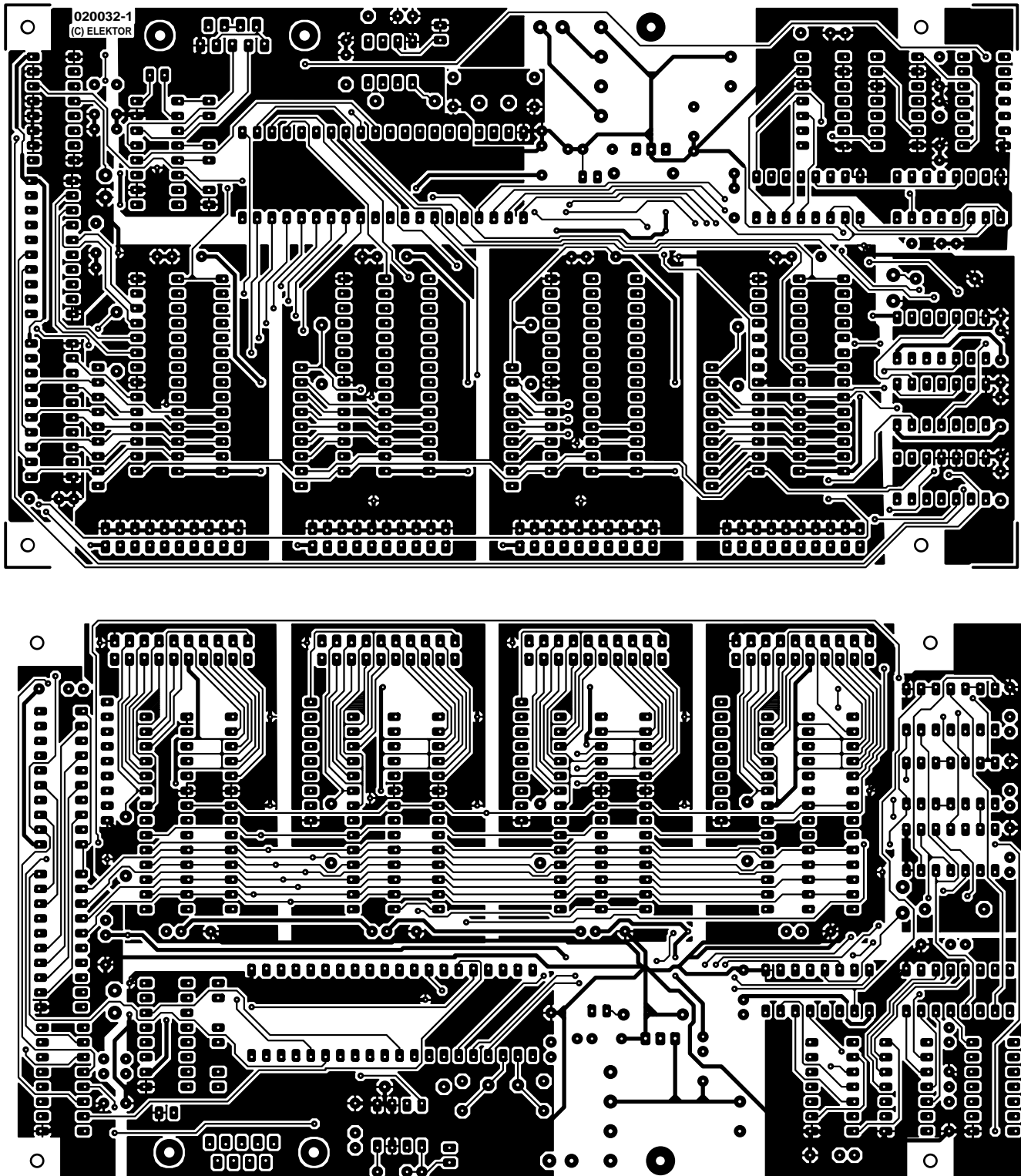


Figure 5b. Dessin des pistes de la platine de l'analyseur logique.

microcontrôleur (fonctionnant en +5 V) et les lignes RS-232 (fonctionnant en ± 12 V).

Enfin, n'oublions pas une partie très importante du montage : l'alimentation, construite autour d'IC17, un classique 7805, mais qui sera mis à rude épreuve, l'analyseur n'étant pas à proprement parler économe en courant (surtout en version 40 MHz).

Avant d'attaquer...

Et bien non, nous n'allons pas tout de suite commencer à souder, car il va falloir d'abord choisir la version que vous souhaitez construire. Comme nous en avons déjà succinctement parlé, notre analyseur est prévu pour fonctionner avec deux horloges différentes : 40 MHz ou 20 MHz.

La version 40 MHz est en mesure d'enregistrer 40 millions de mesures (échantillons) par seconde, ce qui est déjà bien rapide pour un montage de cette complexité, et s'avère très utile pour tester des montages à circuits rapides. Toutefois, si vous décidez de construire cette version, il vous faudra IMPÉRATIVEMENT respecter les consignes suivantes, faute de quoi votre analyseur ne fonctionnera pas

correctement à grande vitesse :

- les FIFO doivent être des versions ultra-rapides à 15 ns de temps d'accès (ou moins : on trouve des FIFO à 10 ns)
- les circuits liés à l'horloge (IC5, IC6, IC7, IC8, IC9, IC10, IC12) doivent être de la famille 74F. En aucun cas, il ne faut utiliser pour ces circuits des composants de la famille 74LS ou pire encore 74HCT.
- les circuits de déclenchement (IC11, IC13 et IC14) seront de préférence choisis dans la famille 74F, mais ce point est moins critique que pour le groupe de composants précédent. Des composants 74LS pourront faire l'affaire, sachant qu'ils induisent de légers retards de déclenchement, généralement peu gênants.

Si vous ne pouvez pas respecter les critères précédents (composants indisponibles ou trop chers à votre goût), rabattez-vous sur la version 20 MHz, qu'il est toujours possible de faire évoluer en 40 MHz par la suite. Si vous choisissez de construire la version 20 MHz, tous les circuits logiques pourront être de la famille 74LS (évités les 74HCT). Vous pourrez également vous

contenter de FIFO avec un temps d'accès de 50 ns (ou moins), plus faciles à trouver et surtout beaucoup moins chers.

Justement, puisque nous parlons des FIFO, outre le critère de vitesse, deux autres paramètres sont à prendre en compte : la capacité et le nombre des FIFO. Les FIFO existent en version 1 K*9, 2 K*9, 4 K*9, 8 K*9 et 16 K*9. Evidemment, plus leur capacité est élevée, plus leur coût l'est aussi (encore que...). En pratique, les versions 4 K sont amplement suffisantes pour la majorité des besoins. Mais rien ne vous empêche de prendre des modèles de plus grande capacité, voire de plus faible capacité...

Quant au nombre de FIFO à installer, à vous de voir combien de voies vous souhaitez acquérir simultanément (9, 18, 27 ou 36). C'est ici aussi la profondeur du porte-monnaie qui est à prendre en compte.

Outre le fait que l'analyseur s'accommode de tous ces paramètres, sachez que les FIFO existent chez de nombreux fabricants, avec des brochages compatibles entre eux. Le **tableau 1** donne quelques exemples de références faciles à trouver chez les principaux constructeurs.

Attention quand même lors de l'achat de ces composants, qui existent en différents boîtiers : PLCC32, DIP28 large et DIP28 étroit. Notre circuit imprimé est prévu pour accepter les deux versions DIP28, mais pas la version PLCC32.

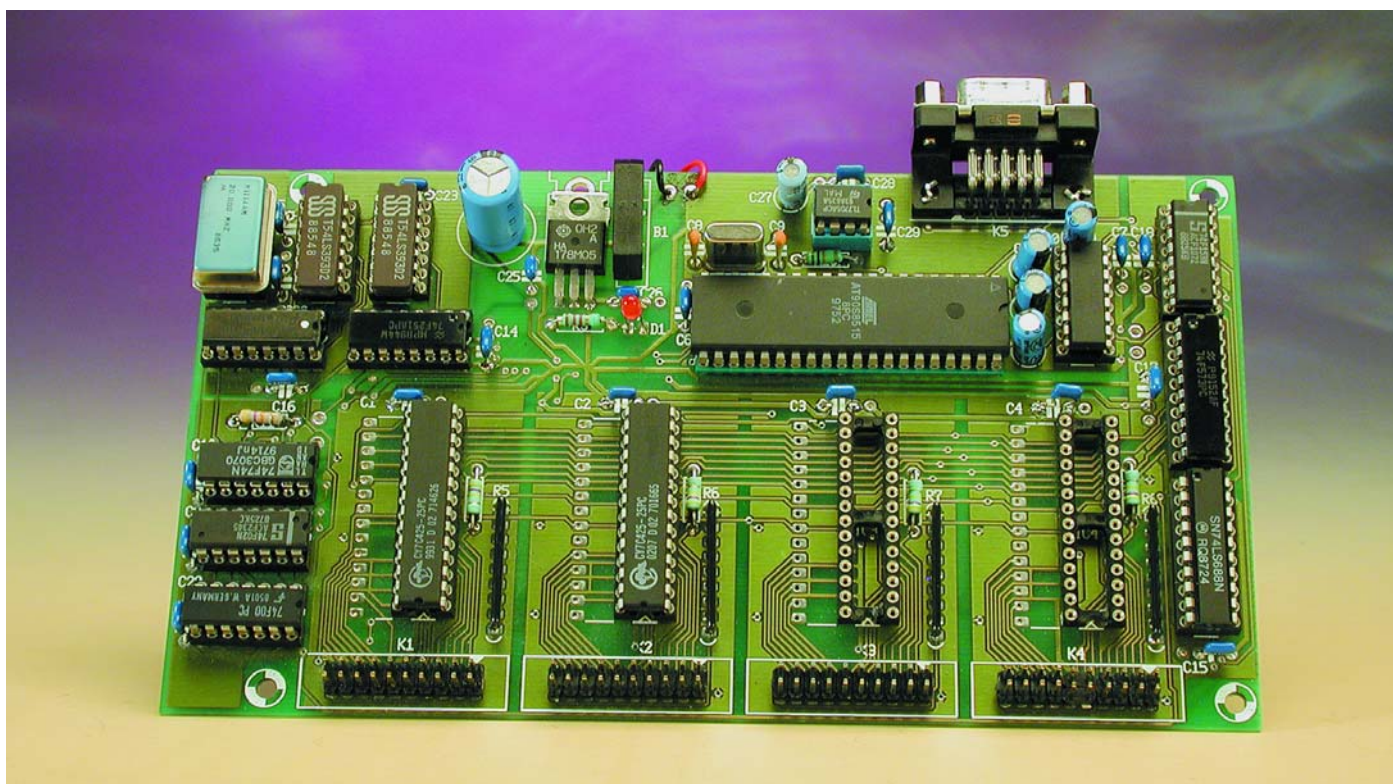


Figure 6. Exemple terminé de l'analyseur logique monté en version 2 canaux.

Dernier point : le microcontrôleur IC16 ; il pourra être acheté tout programmé aux adresses habituelles (cf. Pages Publitronic Service du mois prochain ou notre site Internet), ou être programmé par vos soins. Dans ce dernier cas, vous remarquerez que le circuit de l'analyseur n'intègre pas le programmeur ISP du processeur. Celui-ci devra donc être programmé sur un programmeur externe.

Passons à la pratique

Une fois le problème de l'approvisionnement des composants résolu (il est quand même plus simple qu'il n'y paraît), vous pourrez passer aux choses sérieuses : la construction.

Malgré le fait que les FIFO intègrent une grande partie de la logique nécessaire, le circuit reste assez touffu, comme le montre la sérigraphie du circuit imprimé présentée en **figure 6**.

Inutile de préciser qu'il est très fortement recommandé de monter les circuits logiques sur des supports et que ceux-ci devront être d'excellente qualité (contacts tulipe impératifs). Malgré tout, l'installation des composants reste plus longue que difficile et ne devrait pas poser de réels problèmes. Soulignons toutefois que le régulateur IC17 sera amené à dissiper pas mal de calories, surtout sur une version 40 MHz (les FIFO consomment à eux seuls 200 à 300 mA, les circuits logiques 74F pouvant demander 150 à 200 mA de leur côté, soit 500 mA dans le pire des cas). Prévoyez donc la place autour de ce composant pour installer un radiateur adéquat par la suite (ou montez le régulateur « debout »). Une solution simple consiste à insérer sous le régulateur (avant de le visser sur le circuit) une équerre métallique, dont la partie coudée à l'arrière sera montée sur un petit radiateur. Il nous faut également insister sur la nécessité de monter les condensateurs de découplage (et des bons...), pour éviter les mauvaises surprises avec les circuits à commutation rapide.

Une fois les composants passifs ainsi que ceux de l'alimentation installés, nous vous recommandons de vérifier la présence de la tension +5V sur les broches d'alimentation des supports, avant d'installer les circuits logiques.

Une fois tous les composants installés, on pourra passer à la dernière étape :

Le test du circuit

Bien que le circuit soit assez complexe (toute proportion gardée), la vérification de son bon fonctionnement est simple. Il suffit en effet de

Tableau I.

Quelques modèles de FIFO adaptés à la réalisation de l'analyseur. Les références complètes des circuits font apparaître la vitesse maximale de travail, point critique à vérifier.

Fabricant	Référence	Capacité
AMD	AM7202	1 K * 9
	AM7203	2 K * 9
	AM7204	4 K * 9
	AM7205	8 K * 9
IDT	IDT7203	2 K * 9
	IDT7204	4 K * 9
	IDT7205	8 K * 9
	IDT7206	16 K * 9
CYPRESS	CY7C425	1K * 9
	CY7C429	2K * 9
	CY7C433	4K * 9
	CY7C460	8K * 9
	CY7C462	16K * 9

disposer d'un PC sous WINDOWS, avec le logiciel HyperTerminal, livré en standard avec WINDOWS (toute autre machine avec un port série RS-232 et un logiciel d'émulation de terminal fera également l'affaire).

Lancez le logiciel d'émulation de terminal et configurez-le au paramétrage suivant : 8 bits, sans parité, 1 bit d'arrêt (*stop bit*), 38 400 bauds. Connectez l'analyseur au port série de l'ordinateur (avec le logiciel en train de tourner, ça va de soi...), et mettez l'analyseur sous tension.

Vous devriez voir s'afficher sur l'écran un message donnant la version du logiciel et quelques autres informations, ainsi qu'un signe d'invite « > ». Si le logiciel n'était pas encore lancé lorsque l'analyseur a été mis sous tension, entrez au clavier #1 pour réinitialiser le circuit : le message d'invite doit alors s'afficher. Si le message d'invite s'affiche bien, vous pouvez déjà être sûr que le processeur fonctionne et vous pouvez passer à la suite. Sinon, il faut vérifier avant tout la liaison série (et notamment le câble de liaison au PC), l'expérience prouvant que la majorité des problèmes surviennent à ce niveau. Vérifiez aussi le paramétrage du logiciel d'émulation de terminal. N'oubliez pas non plus que l'écho (renvoi des caractères reçus) est effectué par l'analyseur. Il faut donc désactiver l'écho local du logiciel. Si tout semble correct et que le cir-

cuit refuse toujours de communiquer, vérifiez le câblage de la zone située autour d'IC16, IC15 et IC18). Un circuit aussi simple (merci les microcontrôleurs !) doit fonctionner du premier coup.

Si, durant les premières utilisations, il semble que quelque chose aille de travers dans l'analyseur, il est très simple de tester ses différents sous-ensembles en envoyant les commandes adéquates depuis le PC. Par exemple, il est facile de vérifier le générateur d'horloge à l'aide de la commande #F (voir le prochain paragraphe), en mettant un oscilloscope sur la ligne SYSCLK.

Nous tenons toutefois à vous rassurer : les différents prototypes réalisés ont toujours fonctionné du premier coup (du moment que les règles sur le choix des circuits intégrés ont été respectées).

Mise en « boîte »

Hormis un éventuel interrupteur de mise sous tension, notre analyseur ne comporte aucune commande directe (tout se fait par liaison série). De même, les indications à destination de l'utilisateur se limitent à la LED de mise sous tension.

Notre analyseur se contentera donc d'un boîtier simplifié, comme un petit rack. Les connecteurs K1 à K4 ont été positionnés à l'avant du circuit, afin de permettre leur accès depuis une

Tableau 2.

Rappel succinct des commandes reconnues par l'analyseur.

#I :	Initialisation analyseur.
#G :	Lancement immédiat de l'enregistrement.
#C :	Arrêt immédiat de l'enregistrement.
#R :	Lecture de l'échantillon suivant dans les FIFO.
#P :	Répétition du dernier échantillon lu par la commande #R.
#Mxx :	Configuration du masque de déclenchement.
#Tx :	Sélection du mode de déclenchement automatique.
#Fx :	Sélection de la fréquence d'échantillonnage.
#S :	Lecture état de l'analyseur.
#A :	Armement du trigger.
#Z :	Effacement des FIFO.

face avant, moyennant l'utilisation de connecteurs soudés à 90°.

La face avant recevra également les entrées de synchronisation et d'horloge externes, par exemple sous la forme de connecteurs banane, mais ici aussi tout est affaire de goût.

En revanche, nous ne saurions trop vous recommander d'utiliser un boîtier métallique mis à la terre, pour limiter le rayonnement électromagnétique du circuit vers le monde extérieur, l'analyseur utilisant des signaux à fréquence élevée.

L'alimentation pourra se faire soit par l'intermédiaire d'un petit transformateur délivrant 9 V/8 VA environ sur son secondaire (avec un petit fusible sur le primaire, c'est mieux), soit par un petit bloc secteur externe. Si votre choix se porte sur un boîtier métallique contenant le transfo d'alimentation, la mise à la terre est impérative.

Pilotage de l'analyseur

Le pilotage de l'analyseur depuis un micro-ordinateur est extrêmement simple. Le principe de commande est du type maître-esclave : l'ordinateur envoie une commande et attend la réponse en retour. En aucun cas l'analyseur ne transmet d'informations si une commande n'est pas reçue précédemment. Noter que, même durant les phases d'acquisition, le processeur local

continue à répondre aux requêtes (la gestion des FIFO étant extérieure au processeur).

La liaison entre le PC (ou tout autre ordinateur) et l'analyseur se fait par une liaison série bidirectionnelle à 38 400 bauds, sur 8 bits de donnée, sans parité et avec un bit d'arrêt. Tout caractère reçu par l'analyseur est renvoyé en écho, ce qui permet de vérifier notamment que la communication n'est pas interrompue. Toute commande à destination de l'analyseur commence toujours par le symbole « # ». Ce symbole doit être suivi d'une lettre (majuscule ou minuscule), qui décrit la commande. Cette lettre peut être suivie par un ou plusieurs caractères ASCII hexadécimaux, utilisés en tant que paramètres. La valeur hexadécimale \$5A est ainsi représentée par deux caractères : « 5 » et « A ».

Les caractères de retour chariot (CR = code ASCII 13) et de passage à la ligne (LF = code ASCII 10) sont acceptés par l'analyseur, mais ne sont pas obligatoires.

Quand un ordre a été complètement transmis à l'analyseur, celui-ci peut répondre de trois façons :

« > » la commande a été exécutée correctement.

« ? » la commande n'a pas été reconnue (commande inexistante ou erreur dans les paramètres).

« ! » la commande a été reconnue, mais n'est pas acceptée dans l'état actuel de l'analyseur.

Le **tableau 2** récapitule succinctement les différentes commandes reconnues par l'analyseur.

Entrons dans leur détail car elles sont la substance de cette réalisation.

#I : Initialisation analyseur. Équivaut à la mise sous tension du circuit (FIFO vidés, acquisition stoppée). Cette commande permet également d'afficher le message d'accueil, contenant le numéro de version du logiciel.

#G : Lancement immédiat de l'enregistrement. Dès réception de cette commande, l'analyseur commence à enregistrer les données présentes sur ses entrées, à la cadence définie par l'instruction #F. Les FIFO sont vidés avant que l'acquisition ne commence. Cette commande est refusée si l'analyseur est déjà en acquisition.

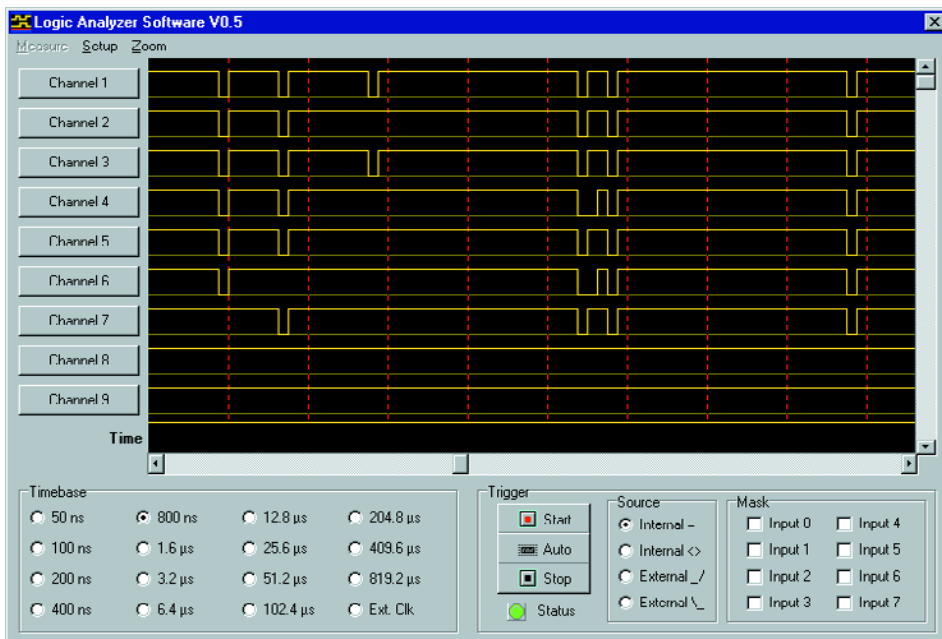
#C : Arrêt immédiat de l'enregistrement. À la réception de cette commande, l'analyseur stoppe immédiatement l'acquisition de ses entrées (même si l'analyseur n'était pas encore déclenché). Les éventuelles données enregistrées jusqu'à réception de la commande peuvent être consultées par la commande #R.

#R : Lecture de l'échantillon suivant dans les FIFO. En réponse à cette commande, l'analyseur renvoie une chaîne (terminée par CR/LF) représentant les 36 bits lus dans les FIFO. La chaîne est formée de la façon suivante : WWXXYYZZOV. WW est formé par les 8 bits du 1^{er} FIFO, XX par les 8 bits du second, YY par les 8 bits du 3^{ème} et ZZ par les 8 bits du dernier FIFO. Les 4 bits supplémentaires (bit 9 des quatre FIFO) sont regroupés dans le code V.

Cette commande est refusée si l'analyseur est en cours d'acquisition ou si les FIFO sont vides.

Lorsqu'un FIFO n'est pas installé, le processeur renvoie le code FF en lieu et place de ses données.

#P : Répétition du dernier échantillon lu par la commande #R. Si le logiciel de visualisation détecte une incohérence dans les données transmises en réponse à la commande #R (due par exemple à une erreur de transmission), la commande #P permet de faire répéter la trame autant de fois que nécessaire. La commande #P ne provoque pas de lecture dans les FIFO.



Si l'écriture d'un programme de commande pour l'analyseur vous rebute, l'auteur a développé un programme spécifique (à lui commander directement), disponible pour un prix modique. Une version de démonstration de ce programme est disponible sur le site d'Elektor (www.elektor.fr).

#Mxx : Configuration du masque de déclenchement. La commande #M permet de régler la valeur en sortie du registre de commande du comparateur. Pour valider ce dernier, il faut que la donnée présente sur les 8 entrées de K4 soit identique à la valeur donnée comme paramètre de la commande #M.

#Tx : Sélection du mode de déclenchement automatique. La valeur donnée en paramètre (de 0 à 3) permet de choisir la source de déclenchement :

- 0 : déclenchement de l'analyseur lorsque la valeur binaire sur K4 est égale à la valeur du masque de déclenchement (réglé par #M).
- 1 : déclenchement de l'analyseur lorsque la valeur binaire sur K4 diffère de la valeur du masque de déclenchement.
- 2 : déclenchement sur front montant sur entrée EXT TRIG.
- 3 : déclenchement sur front descendant sur entrée EXT TRIG.

#Fx : Sélection de la fréquence d'échantillonnage. Cette commande permet de choisir la valeur du facteur de division de l'horloge principale appliquée aux FIFO. La commande est refusée si une acquisition est déjà en cours. La fréquence réelle d'échantillonnage dépend de la fréquence de l'oscillateur IC19. Le tableau qui suit permet de connaître le temps entre chaque échantillon en fonc-

tion du code et de la fréquence de l'oscillateur.

	20 MHz	40 MHz
X	20 MHz	40 MHz
8	50 ns	25 ns
9	100 ns	50 ns
A	200 ns	100 ns
B	400 ns	200 ns
C	800 ns	400 ns
D	1,6 ms	800 ns
E	3,2 ms	1,6 ms
F	6,4 ms	3,2 ms
0	12,8 ms	6,4 ms
1	25,6 ms	12,8 ms
2	51,2 ms	25,6 ms
3	102,4 ms	51,2 ms
4	204,8 ms	102,4 ms
5	409,6 ms	204,8 ms
6	819,2 ms	409,6 ms
7	Horloge ext.	Horloge ext.

#S : Lecture de l'état de l'analyseur. L'analyseur répond à cette commande par un code XX<CR><LF>, représentant son état actuel. Dans la réponse, le code binaire XX (8 bits exprimés en ASCII) est formée de la façon suivante :

- Bit 0 : 1=Analyseur déclenché
- Bit 1 : 1=Acquisition terminée
- Bit 2 : 0=FIFO vides (1=FIFO non vides...)

À la mise sous tension, la valeur renvoyée par la commande #S est 00.

#A : Armement du trigger. Cette commande permet de déclencher l'analyseur lorsque la condition donnée par la commande #T est vérifiée.

#Z : Effacement des FIFO. Sans commentaires...

Le logiciel graphique de l'analyseur

Comme le paragraphe précédent l'a montré, la commande de l'analyseur est somme toute assez simple. Il est relativement aisé d'écrire un programme (peu importe le langage et l'environnement) capable de piloter l'analyseur et d'afficher sous forme graphique les données renvoyées.

Pour ceux qui n'ont pas trop envie de s'atteler à cette tâche, l'auteur a écrit un programme de gestion de l'analyseur ; nous pouvons vous proposer, sous la dénomination **EPS020032-11** (voir la copie d'écran dans cet article) une version de démonstration de ce logiciel disponible sur notre site Internet (www.elektor.fr) ou par le biais des adresses habituelles.

La version complète de ce logiciel est disponible sur la page Web de l'auteur, à l'adresse :

http://benoit.bouchez.free.fr/logic_analyzer

Toutes les informations relatives à l'achat de la version complète du logiciel de commande sont disponibles sur cette page.

Quelques remarques importantes

Tel qu'il est présenté ici, l'analyseur logique est apte à enregistrer sur ses entrées des signaux conformes à la norme TTL 5 V (y compris les CMOS alimentés en 5 V). Il est également possible d'utiliser l'analyseur sur des circuits 3,3 V, mais dans ce cas, **NE MONTEZ PAS** les résistances de rappel R1 à R8, faute de quoi vous risqueriez la détérioration du circuit testé (qui recevrait du 5 V sur des lignes 3,3 V par l'intermédiaire des résistances. Leur valeur élevée limite toutefois les risques réels pour le circuit testé). On pourrait être surpris du fait que les FIFO (alimentés sous

5 V) arrivent à interpréter correctement les signaux au standard 3,3 V. En fait, le standard TTL 3,3 V utilise les mêmes niveaux de tensions (supérieur à 2,4 V pour un « 1 », inférieur à 0,4 V pour un « 0 ») que le standard TTL 5 V, d'où cette compatibilité (au moins pour des lignes 3,3 V attaquant des entrées 5 V, l'inverse n'étant pas vrai).

Pour tester des circuits logiques fonctionnant sous 3 V ou 2,5 V (de plus en plus fréquents), des adaptateurs d'entrée sont nécessaires, car les seuils définissant les « 0 » et « 1 » logiques sont différents de ceux des circuits compatibles TTL (5 V ou 3,3 V). Signalons toutefois que nous avons réussi à tester des montages fonctionnant sous 3 V sans adaptateur.

De même, d'autres adaptateurs sont nécessaires pour tester des circuits CMOS alimentés sous des tensions dépassant 5 V (toute tension supérieure à 5 V appliquée à l'entrée des FIFO les détruit à coup sûr).

Ces deux types d'adaptateurs seront décrits dans un prochain article.

Nous devons également souligner un autre point important, généralement sous-estimé par les nouveaux utilisateurs d'analyseurs logiques : la (toute) relative fragilité des entrées de mesure. Ne perdez pas de vue que les entrées attaquent directement les FIFO. Si vous reliez une entrée à une ligne dont la tension dépasse 5 V, vous risquez la destruction à court terme des circuits. Même les analyseurs professionnels (pourtant beaucoup plus complexes

et chers) présentent le même genre de risque. Alors, direz-vous, pourquoi ne pas prévoir de protections sur les entrées ? Tout simplement parce que de telles protections altèrent les caractéristiques en commutation des circuits testés, pouvant résulter en un mauvais fonctionnement de ce circuit (un analyseur est censé aider à trouver les pannes, pas en ajouter...).

Dernier point important : pour relier l'analyseur au circuit à tester, utilisez des liaisons TRES COURTES, si possible en nappe (ce n'est pas pour rien que nous avons intégré des lignes de masse sur les connecteurs d'entrée...). Un fil de liaison trop long déforme considérablement les signaux, et peut même perturber le circuit testé, surtout pour des signaux rapides.

Il ne nous reste plus qu'à vous souhaiter bon courage pour cette réalisation.

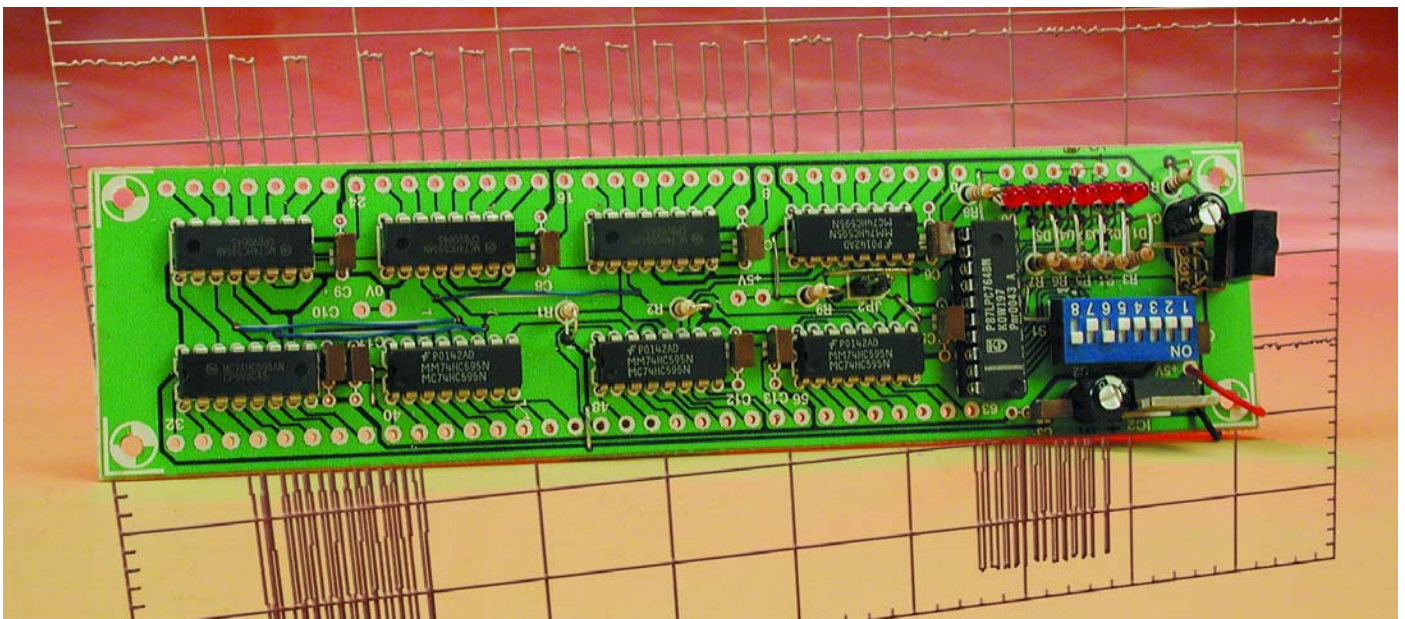
(020032)

Relève du SAA3049

Décodeur RC5 à microcontrôleur

Frank Wohlrabe

Le très recherché SAA3049 de Philips, le décodeur de télécommande RC5, on n'en trouve plus ! Aucun « forgeron du silicium » ne propose de remplaçant, pour la bonne raison que, dans l'électronique grand public, la fonction est reprise, en cumul, par le microcontrôleur en poste. Alors, faisons pareil !



Le nombre de puces destinées au décodage des messages infrarouges est singulièrement plus restreint que pour leur émission. Il est clair que si, côté récepteur, on a déjà un microcontrôleur sous la main, il est plus rentable de lui faire analyser les messages, à l'occasion. Non seulement on se passe de la puce de décryptage et de la logique qui l'entoure, mais on épargne de la place et des coûts de fabrication. Un microcontrôleur moderne s'acquitte de la tâche au moins aussi bien pour un prix de 0,30 €, en production de masse. D'ailleurs, le SAA3049, longtemps prisé pour le décodage des signaux de télécommande en RC5 ou RECS80, était lui-même un microcontrôleur,

auquel il fallait adjoindre un quartz de 4reil télécommandé devant s'occuper de traiter les informations décodées et en général, il fallait encore y ajouter d'autres composants, comme des bistables, pour réaliser les commutations. Il faut dire que le SAA3049 ne s'illustrait pas vraiment par une profusion de lignes de commande.

Philips l'a donc sorti du programme de fabrication et celui qui ne dispose pas de l'outil industriel ou des connaissances requises en programmation n'est pratiquement plus en mesure de construire un système de

télécommande par infrarouge.

Aussi, tant qu'à développer un remplaçant du SAA3049, profitons-en pour le débarrasser de quelques boulets qu'il traînait :

- supprimer le quartz à 4 MHz
- inclure les circuits commutation
- intégrer la mise à zéro
- augmenter les sorties.

L'émetteur sera la télécommande du téléviseur, du magnétoscope, de la chaîne stéréo ou de tout autre « machin » domestique. Si vous ne savez pas au juste selon quel format

Fleurons

- décodage des signaux RC5 et RECS80
- réglage de l'adresse à laquelle le CI doit réagir
- option d'adressage : si nécessaire, on peut masquer la fonction d'adressage, le CI réagit à tout coup
- possibilité de donner les commandes en binaire ou en décimal (réduit)
- fonction d'analyse du protocole transmis
- 6 sorties de commutation immédiatement disponibles
- les bornes de commutation peuvent fonctionner en quasi bidirectionnel
- interface sérielle pour le raccordement de 64 sorties
- signal de validation de commande disponible
- nul besoin de circuit oscillateur
- nul besoin de circuit de mise à zéro

Le récepteur-décodeur proposé ici collabore indifféremment avec, côté émetteur, les puces suivantes :

Code RC5 :	SAA3006, SAA3010 (Philips) HT6230 (Holtec) PT2211 (Princeton)
Code RECS80 :	SAA3004, SAA3007, SAA3008 (Philips) M3004, M3005, M3006 (ST microelectronics)

il transmet les messages, nous vous conseillons la construction de l'analyseur de code IR [1].

Microcontrôleur

Notre décodeur est basé sur un microcontrôleur 87LPC764, issu de la famille 80C51 et doté d'une mémoire morte de programme OTP (*one time programmable*) de 2 ou 4 Koctets. Le suffixe LPC provient de *Low power, low Price, low pin Count*, ce que l'on pourrait résumer en « peu de puis-

sance, petit prix et peu de pattes » ou P à la 6^e, pourquoi pas ? Avantage, à côté de la grande puissance des instructions, l'intégration des fonctions d'oscillateur RC et de mise à zéro. Cela épargne des broches, si bien que l'utilisateur dispose déjà de 18 E/S.

L'oscillateur interne scande la CPU à 6 MHz environ, le cycle dure donc 1 μ s, suffisamment rapide pour une lecture sûre des codes RC5. La conséquence de la tolérance sur le rythme, c'est que tous les temps qui

figurent dans le projet doivent pouvoir fluctuer de $\pm 20\%$. Un encadré vous résume les principales caractéristiques du contrôleur, mais vous en trouverez des informations plus détaillées sur le site Internet de Philips à l'adresse :

www.semiconductors.com/pip/P87LPC764.html ainsi que les feuillets de caractéristiques complètes via : www.semiconductors.com/acrobat/datasheets/87LPC764_10.pdf.

Logiciel

Lors de la conception, nous avons apporté un soin tout particulier à la sécurité du décodage. Confucius déjà disait que le logiciel est tout-puissant. Il suffit de s'assurer, par une procédure adéquate, que les commandes sont bien identifiées et que les messages intrus comme les parasites sont éliminés. Pour le code RC5, on peut, par exemple, exécuter les vérifications suivantes :

1. tester la largeur d'impulsion
2. analyser plusieurs fois les signaux
3. vérifier la complémentarité des deux moitiés de bit
4. vérifier que le code de la commande vaut moins de 64
5. vérifier si le message concorde avec l'adresse fournie.

La lecture des messages ne présente normalement pas de difficulté. Cependant, comme le microcontrôleur est scandé par un oscillateur RC, sa fréquence peut dériver de $\pm 20\%$. À cette imperfection s'ajoute l'altération de la largeur d'impulsion due au module photorécepteur. Regardons de quoi il s'agit à l'aide du module TSOP12xx de Vishay.

Le tracé de la **figure 1** montre la variation possible de largeur d'impulsion d'un signal de test en fonction de l'intensité du rayonnement, alors que le signal commute exactement toutes les 600 μ s.

Le logiciel doit être capable de s'accommoder de décalages considérables qui peuvent atteindre 100 μ s. Pas question de se fier, dès le début du message, au tempo du signal pour ensuite se contenter de sonder le niveau dans des intervalles fixes. Il ne faudrait pas attendre le milieu de la transmission pour que les repères soient complètement faussés. Plusieurs fois pendant la communication, le logiciel doit se resynchroniser sur les bits pour s'affranchir des tolérances.

Le circuit

La relève du SAA3049, si l'on en croit la **figure 2**, semble, au premier abord, plutôt

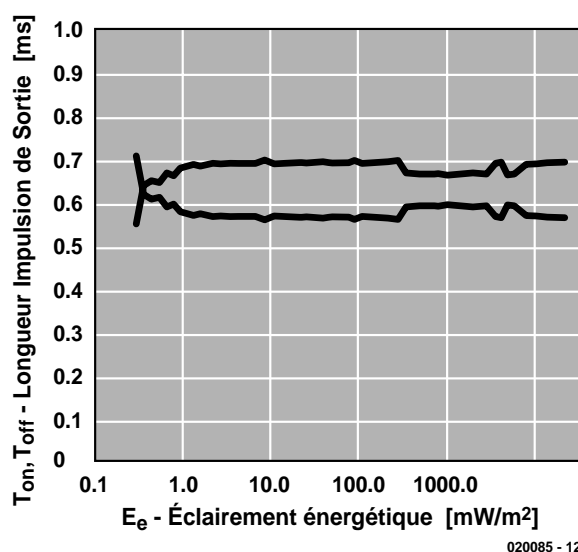
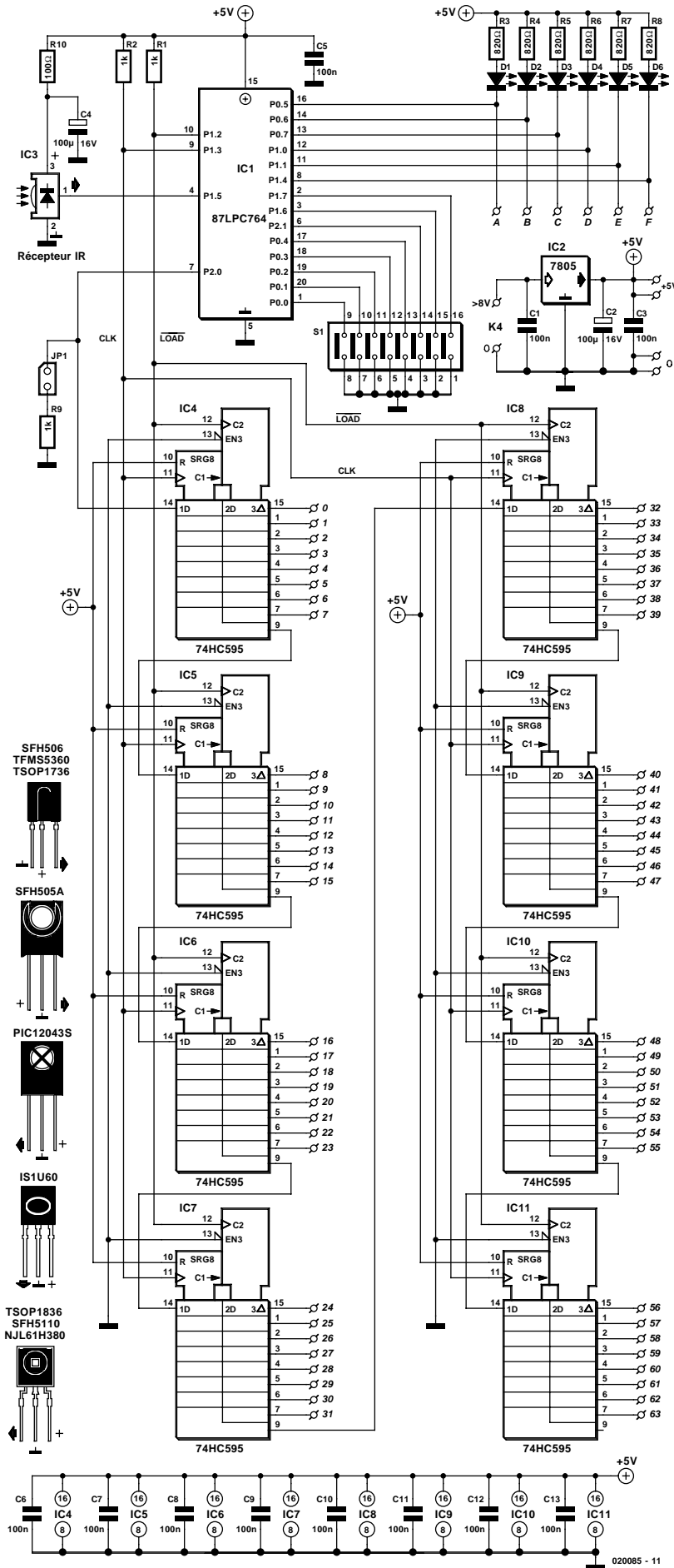


Figure 1. Altération de la largeur d'impulsion d'un signal expérimental en sortie du module photosensible.



imposante. Mais si l'on passe au recensement, on ne compte qu'un récepteur IR, le microcontrôleur et une rangée de convertisseurs de série en parallèle, qui permettront de fournir jusqu'à 64 sorties de commutation ou d'impulsions.

Récepteur infrarouge

La fréquence porteuse de l'émission infrarouge modulée à 36 kHz ou 38 kHz, c'est le module photosensible IC3 qui la capte. Plusieurs types de ces récepteurs sont envisageables, nous reviendrons sur leurs caractéristiques au moment de la construction.

IC3 est un récepteur miniature que l'on peut régler, selon le type, dans la plage de fréquence de 30 kHz et 56 kHz. La diode photosensible et l'amplificateur sont intégrés, aucun circuit externe n'est donc requis et le signal démodulé peut être traité directement par un microcontrôleur. Le fonctionnement est remarquablement efficace : même en environnement perturbé, il produit peu d'impulsions de sortie inopportunes.

Le code est acheminé directement (sous la forme inversée) au microcontrôleur IC1 pour le décodage. La cellule R10 et C4 constitue un filtre passe-bas de protection du CI récepteur contre les parasites sur la ligne d'alimentation. Elle est indispensable parce que ce CI réagit par une considérable perte de sensibilité en cas de pics de tension.

Inscription de l'adresse

L'adresse contenue dans le message à laquelle la puce doit réagir s'écrit aux bornes P0.0 à P0.4. Avec cinq interrupteurs DIP, on a donc le choix entre 32 adresses. Tout interrupteur DIP ouvert produit un 1 logique, puisque le microcontrôleur dispose de résistances de forçage haut intégrées. Si tous les interrupteurs sont fermés, toutes les broches sont à la masse, l'adresse est 00000. Le code RECS80 ne définit que sept adresses, on n'utilisera alors que les niveaux des bornes P0.0 à P0.2.

Figure 2. Le récepteur RC5 proprement dit se compose du module IR, du contrôleur et des interrupteurs DIP. Les registres à décalage sont en option.

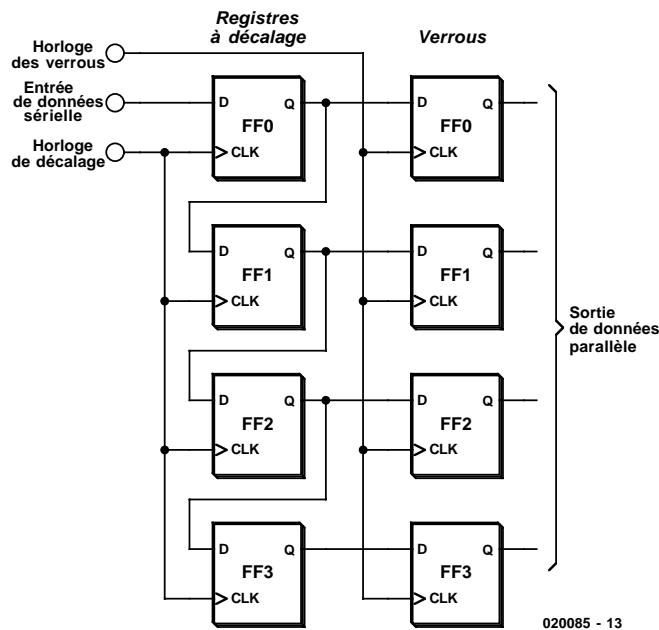


Figure 3. Le 74HC595 regroupe deux volées de huit bascules en cascade.

Validation de commande

La ligne de port P1.2 passe pendant environ 65 ms au niveau bas dès réception d'un message valide. Ce signal active le périphérique qui y est connecté, si par exemple l'information binaire de la commande va servir à déclencher un autre processus logique. Mais on peut aussi utiliser la fonction pour allumer une LED qui indiquera que le message a été reçu. La ligne fonctionne en drain ouvert, il faut donc lui brancher une résistance de forçage haut si l'on veut analyser le signal à l'oscilloscope.

Mise au point de la configuration

Il est possible de programmer à l'aide d'interrupteurs DIP, via deux broches du port, la conduite du processus.

P2.0 : choix du code

Le montage s'accommode, à l'instar du SAA3049, des codes RC5 et RECS80. Sur la puce d'origine, on communiquait le choix par l'intermédiaire d'une broche de code spéciale. Sur le remplaçant, qui ne dispose plus d'aucune ligne de contrôle

disponible, il faut une astuce logicielle pour conférer à la borne P2.0, normalement dévolue à la sortie série de données vers un autre matériel, une double fonction.

Au lancement du logiciel, la ligne est considérée comme une entrée dotée d'une résistance de forçage haut. Le logiciel lit le niveau, qui sera bas si un cavalier est en place, ce qui correspond au mode RECS80. En l'absence de cavalier, le niveau reste haut et le circuit comprend qu'il doit travailler en code RC5. Aussitôt après la lecture, la borne du port est convertie en sortie avec l'option push-pull, de manière à ce que R9 ne puisse plus exercer d'incidence sur les niveaux logiques des données.

Broche P1.7 : fonction de commutation ou de sortie binaire

Cette broche-ci possède également une résistance intégrée de forçage haut. Au niveau haut (laissée en l'air) la fonction de commutation est en vigueur et les lignes de sortie réagissent directement aux commandes 0 à 5 de la télécommande infrarouge. Voici les correspondances :

- commande 0 : broche P0.5 (A)
- commande 1 : broche P0.6 (B)
- commande 2 : broche P0.7 (C)
- commande 3 : broche P1.0 (D)
- commande 4 : broche P1.1 (E)
- commande 5 : broche P1.4 (F)

Presser une touche change le niveau logique de chaque broche. Avec un niveau bas sur P1.7, les informations binaires des commandes sont disponibles aux sorties A à F, comme sur le SAA3049.

Broche P2.1 : commutation entre lecture et pause

Une borne également pourvue d'une résistance de forçage intégrée. Au niveau haut (en l'air), la sortie est au repos, sinon, elle bascule d'état logique pour une seconde environ après une réception, puis revient à l'état de sortie (haute sans autre intervention). Ceci correspond à un monostable. Si l'on appuie sur une touche, la borne correspondante reste au potentiel bas aussi longtemps que la touche est enfoncée.

Applications

Parmi toutes les applications possibles, nous allons en examiner quatre.

Système de télécommande simple

Les lignes d'adresse P0.0 à P0.4 ainsi que les bornes 1.7 et 2.1 ne sont pas câblées, l'adresse de télécommande ne joue donc

En bref : le 87LPC764

- Tension d'alimentation 2.7 V...6.0 V
- Cycle d'instruction 300 ns...600 ns / 20 MHz
- RAM 128 octets
- ROM 2 ou 4 Ko (87LPC764)
- 2 temporisateurs
- 2 comparateurs analogiques à référence intégrée de 1.23 V
- UART en duplex
- Port de communication I²C
- 2 interruptions externes et huit par clavier
- Chien de garde à oscillateur indépendant
- Mise à zéro à l'allumage embarquée
- Réarmement après défaut de tension
- Oscillateur RC intégré sans composant externe
- Configuration de port programmable : quasi bidirectionnel - drain ouvert - push-pull - entrée seule
- Possibilité de programmation série
- Modes repos et basse puissance

aucun rôle. Chaque action sur une des touches 1 à 6 commute la sortie correspondante. Le signal CA sur la borne P1.2 s'active à la réception de chaque commande. Les sorties ne fournissent que 1 mA, mais peuvent absorber 20 mA au niveau bas.

Analyse d'adresse et de commande

Pour savoir sur quelle adresse RC5 une télécommande émet et pour reconnaître la différence entre la commande et l'appui sur une touche, on commence par relier à la masse l'entrée de codage P1.6. Aucune entrée d'adresse n'est en service, les informations aux lignes d'adresse et de commande sont présentes sous forme binaire. Au premier appel, on s'assure que les entrées d'adresse sont réellement en l'air. Ensuite, il n'est plus possible de le vérifier, puisqu'on ne pourra plus jamais lire que l'état des tampons de sortie et ceci en envoyant les informations d'adresse correspondantes. Mais on peut vérifier ces niveaux à l'oscilloscope, avec un testeur de potentiel ou à l'analyseur logique. À l'examen, toutes les sorties sont au repos, indépendamment des commutations aux autres entrées de code. Il ne faut en aucun cas, dans cette configuration, mettre à la masse une des entrées d'adresse P0.0 à P0.4, sous peine de risquer un court-circuit. Toutes ces broches sont actives à l'état bas. L'adresse 4 (00100B) et la commande 7 (000111B) entraînent les niveaux suivants :

P0.0 = 1	adresse
P0.1 = 1	
P0.2 = 0	
P0.3 = 1	
P0.4 = 1	
P0.5 = 0	commande
P0.6 = 0	
P0.7 = 0	
P1.0 = 1	
P1.1 = 1	
P1.4 = 1	

Ouvre-porte

Pour cette application, il faut activer brièvement, d'une action sur la touche, une fonction de commutation. Il nous faut une fonction de

monostable redéclenchable.

P1.7 reste en l'air, donc la fonction de commutation est active. La borne P2.1 est reliée à la masse, la fonction de saisie est en service, la borne de commande choisie s'active pour une seconde, environ. Prolonger la pression sur la touche allonge la durée de l'impulsion sur la borne correspondante. Le niveau de sortie actif est bas, il redevient haut après la période.

Télécommande à 64 canaux

Pour obtenir un maximum de sorties commutées avec un minimum de câblage, il faut, à l'extérieur du microcontrôleur, du matériel supplémentaire et à l'intérieur, un logiciel approprié. Notre choix s'est porté sur un convertisseur de série à parallèle bon marché et aisément disponible,

Liste des composants

Résistances :

R1,R2,R9 = 1 kW
R3...R8 = 820 W
R10 = 100 W

Condensateurs :

C1,C3,C5 = 100 nF
C2,C4 = 100 mF/16 V radial

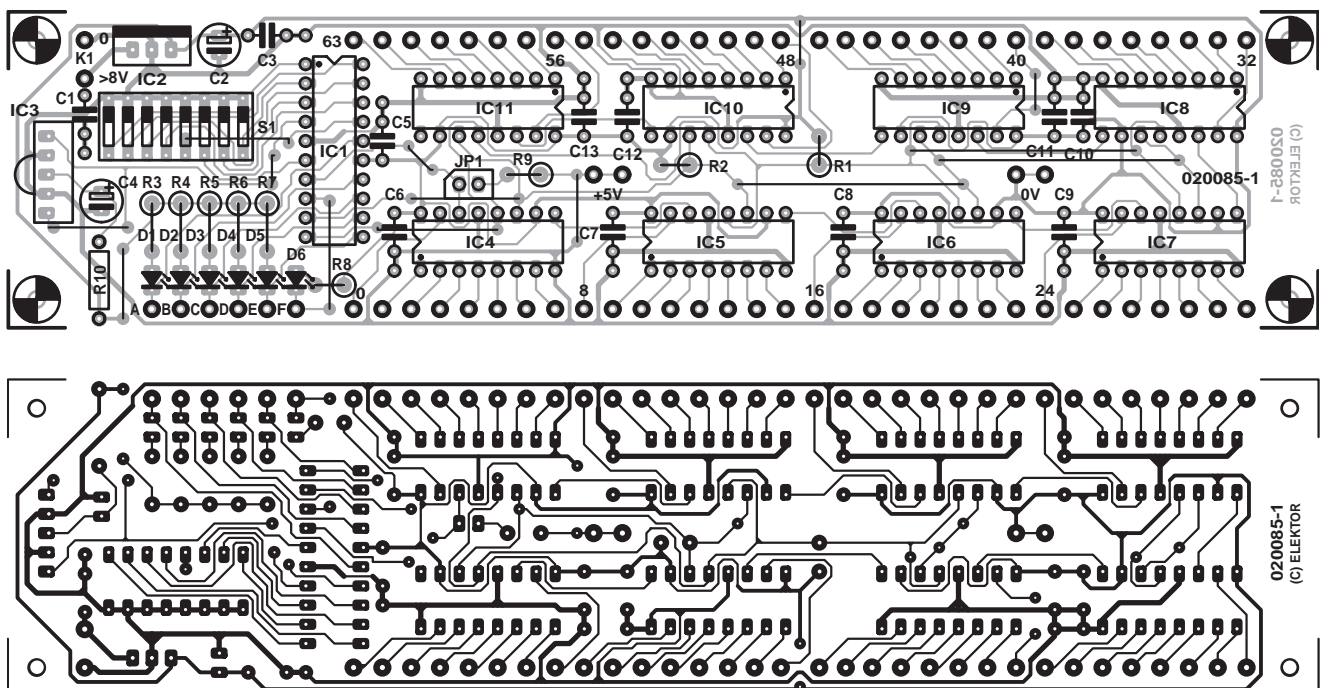
Semi-conducteurs :

D1 à D6 = LED rouge faible courant
IC1 = 87LPC764BN (programmé **EPS020085-41**)
IC2 = 7805
IC3 = module de réception IR tel que, entre autres, SFH5510, PIC2604SM, TFM5360
IC4 à IC11 = 74HC595

Divers :

JP1 = cavalier
S1 = octuple interrupteur DIP

Figure 4. Un dessin bien ordonné de la platine.

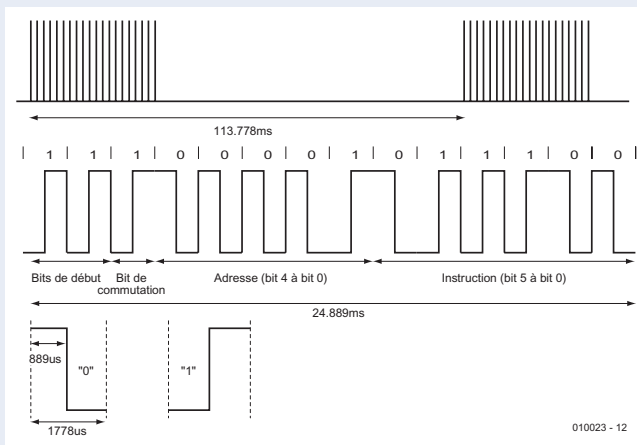


Normes

Le standard RC5, largement répandu en Europe, a été conçu par Philips. Il est capable de coder 2 048 messages, répartis en 32 groupes adressables dont chacun compte 64 commandes. Chaque appareil dispose, dans ce système, de sa propre adresse, de manière à ce que régler le volume de la télévision ne perturbe en rien celui de chaîne hi-fi, par exemple. Le code émis est un mot de 14 bits que l'on peut décoder comme suit :

- 2 bits de départ, destinés au réglage du CAG (contrôle automatique de gain) du récepteur IR.
- 1 bit de basculement qui dénote un changement de données dans la transmission
- 5 bits d'adresse système
- 6 bits de commande.

Le bit de basculement change de valeur à chaque nouvelle action sur une touche, de manière à faire la différence entre la pression continue et l'accès à une nouvelle touche. Suivent cinq bits d'adresse pour indiquer quel appareil va devoir réagir à la commande. Finalement, on transmet la commande proprement dite. En code RC5, les messages subissent une modulation biphasee, chaque bit se compose de la réunion de deux demi-bits transmis l'un après l'autre. La succession de niveaux B/H représente un bit « 1 », la suite H/B traduit un « 0 » logique. Comme la transmission de chaque bit prend 1,778 ms (à la fréquence théorique de 36 kHz !), un message complet dure à peu près 25 ms. Parmi beaucoup d'autres fabricants d'électronique grand public, Loeuwe, Philips, Grundig et Marantz utilisent le code RC5.



Spécifications du code RC5

Les codes RC5 des appareils

Adresse système	binaire	appareil
0	00000	TV1
1	00001	TV2
2	00010	télétexte
3	00011	extension pour TV1 et TV2
4	00100	lecteur DVD
5	00101	magnétoscope 1 (VCR1)
6	00110	magnétoscope 2 (VCR2)
7	00111	réservé

8	01000	SAT1
9	01001	extension VCR1 et VCR2
10	01010	SAT2
11	01011	réservé
12	01100	CD vidéo
13	01101	réservé
14	01110	CD photo
15	01111	réservé
16	10000	préampli audio 1
17	10001	récepteur radio
18	10010	enregistreur minicassette
19	10011	préampli audio 2
20	10100	CD
21	10101	chaîne hi-fi ou enregistreur
22	10110	récepteur satellite audio
23	10111	enregistreur DCC
24	11000	réservé
25	11001	réservé
26	11010	enregistreur CD
27...31	réservé

Quelques commandes du code RC5

Commande	signification
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
16	volume +
17	volume -
18	luminosité +
19	luminosité -
20	saturation couleur +
21	saturation couleur -

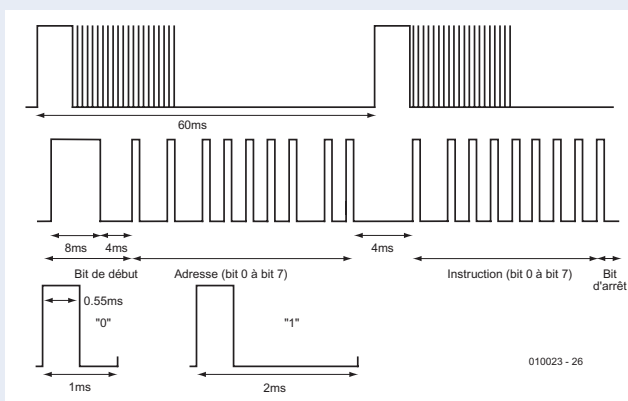
Code RECS80

Le code RECS80 de Philips utilise la modulation en position d'impulsion, caractérisée par le fait qu'une impulsion « haute » de largeur définie est suivie d'une pause de longueur variable, dont la durée définit l'état logique du bit transmis. Le code totalise 1 280 messages, répartis sur 20 systèmes secondaires dont chacun dispose de 64 commandes. Un système secondaire est un récepteur de télécommande, tel que téléviseur, magnétoscope, etc. Avec le bit de basculement et celui de départ, le mot comporte donc au maximum 12 bits.

À cause de contingences historiques, lorsque l'adresse est comprise entre 1 et 7, on ne transmet que 3 bits d'adresse, ce qui donne un mot de 11 bits, mais aussi pas mal de complications lors du décodage par logiciel. Les bits de basculement sont placés dans un compteur du CI émetteur dont le compte est incrémenté à chaque action sur une touche. Rien ne se produit quand la touche

est maintenue. Remarquons qu'on n'utilise deux bits de basculement que lors d'une transmission en mode Flash. Sur les circuits intégrés qui émettent l'infrarouge modulé, le premier bit de basculement est considéré comme bit de départ à largeur constante. Le protocole RECS80 code chaque bit par la différence de longueur des pauses, suivies d'impulsions lumineuses de largeur constante (140,8 μ s). Pour un bit « 1 », la distance entre les impulsions IR est de 7,60 ms, pour un bit « 0 », de 5,06 ms (en relation avec l'emploi d'un résonateur à 455 kHz dans l'émetteur). Bien que la durée de transmission d'un paquet de données puisse différer selon la commande, la période totale message et pause avant répétition reste identique à 121 ms. La fréquence de modulation est ici de 38 kHz.

Spécifications du code RECS80



le 74HC595.

Le convertisseur dont le schéma est reproduit à la figure 3 contient 16 bascules, en cascade sur deux lignes. Les informations sont passées en série à la première rangée de bascules. Le signal de décalage CLK transfère l'information en sortie des bascules, donc à l'entrée de la rangée suivante. Après le nombre voulu d'impulsions SCL, déterminé par l'utilisateur, c'est l'impulsion de mémorisation (LCL) qui transfère l'information, arrivée en mode sériel, pour se retrouver en parallèle aux sorties de la seconde cascade. Mais le 74HC595 dispose encore d'autres possibilités bien utiles.

MISE À ZÉRO

Un signal bas à cette broche (*reset*) remet la logique interne à l'état initial, sorties débranchées et toutes les bascules à zéro. Nous n'utilisons pas cette fonction parce qu'une validation de sortie (*output enable*) est disponible qui permet de débrancher toutes les sorties. Une relance du microcontrôleur remet également tous les registres à zéro.

VERROUILLAGE

Chaque ligne de sortie peut fournir et absorber 35 mA, de quoi piloter sans souci diodes électroluminescentes ou transistors de commutation, lesquels pourront à leur tour attaquer des relais de puissance. Le verrouillage (*latch*) s'obtient par un niveau bas sur la broche de validation de sortie et se maintient.

HORLOGE DE VERROUILLAGE

Un flanc montant sur cette ligne (*latch clock*) transfère les données du registre à décalage dans le tampon de sortie. Cette opération ne doit avoir lieu qu'après l'émission des 64 coups d'horloge qui positionnent les données.

Nul besoin d'installer plus de 74HC595 que nécessaire. Si par exemple on veut disposer

de 24 sorties tout au plus, on ne placera que trois puces. Le logiciel délivre, pour chaque commande, 64 coups d'horloge, de manière à envoyer à la bonne place les données dans les CI. Évidemment, les puces doivent se suivre sans intervalle.

La sortie d'horloge (P1.3) du microcontrôleur est configurée en drain ouvert et réclame donc une résistance de forçage haut. Les 64 bornes restent disponibles, chacune selon le codage en fonction d'entrée ou au repos.

Construction de la platine

Sur la **figure 4**, qui montre le dessin de la platine, le récepteur décodeur se trouve tout à gauche. Si vous l'utilisez pour construire une télécommande simple, comme décrite plus haut, vous pouvez sans hésiter scier la platine juste à la droite du contrôleur. On ne peut dès lors plus installer le cavalier JP1, le montage ne travaille qu'avec des signaux RC5. Les bornes du contrôleur P0.5 à P1.4 sont conduites directement aux picots à souder A à F, au-dessous de la rangée de LED.

Les registres à décalage en option sont disposés à gauche, les sorties 0 à 31 en dessous, 32 à 63 au-dessus, en bordure de carte. Comme la platine est à simple face, il y faut une série de ponts de câblage qui courent en partie sous les puces. Ce n'est pas la seule raison qui plaide pour l'usage de supports de circuits intégrés.

Le schéma mentionne une liste de récepteurs IR, dont les brochages ne

sont pas nécessairement compatibles. Nous avons conçu la platine pour permettre d'y installer n'importe quel type, à vous de le placer au bon endroit !

(020085)

Bibliographie

- Analyseur de code IR, n°280, octobre 2001, page 12 et suivantes [1]
- Protocoles de télécommande par IR, Elektor mars et avril 2001
- Télécommande pour modélisme par signaux IR RC5, Elektor avril 2001
- Télécommande RC5 (récepteur), juillet/août 2001 p. 48
- Décodeur RC5 intelligent, décembre 2000
- Émetteur/récepteur IR pour PC, Elektor décembre 2001 et janvier 2002
- Récepteur IR multi-code, Elektor mai 2002
- Émetteur RC5, Elektor juillet/août 2002

OPAx353

Amplificateur opérationnel
Rail-à-rail rapide

ELEKTOR
ÉLECTRONIQUE et MICRO-INFORMATIQUE
INFOCARTE 2/2003

Caractéristiques électriques ($V_S = 2,7$ à $5,5$ V, $T_A = 25$ °C, $R_L = 1$ k Ω sous $V_S/2$, $V_{OUT} = V_S$. Gras: spécifié pour $T_A = -40$ à $+85$ °C)

PARAMÈTRE	CONDITIONS	MIN	TYP	MAX	UNITÉ
Tension d'offset					
Tension d'offset d'entrée	$V_S = 5$ V		± 3	± 8	mV
$T_A = -40$ à $+85$ °C				± 10	mV
vs. Température	$T_A = -40...+85$ °C		± 5		$\mu\text{V}/^\circ\text{C}$
vs. Rapport de réjection de la tension d'alimentation	$V_S = 2,7$ à $5,5$ V, $V_{CH} = 0$ V		40	150	$\mu\text{V}/\text{V}$
$T_A = -40...+85$ °C	$V_S = 2,7$ à $5,5$ V, $V_{CH} = 0$ V			175	$\mu\text{V}/\text{V}$
Diaphonie (dual, quad)	DC		0,15		$\mu\text{V}/\text{V}$
Courant de polarisation d'entrée					
Courant de polarisation d'entrée			$\pm 0,5$	± 10	pA
Courant d'offset d'entrée			$\pm 0,5$	± 10	pA
Bruit					
Tension de bruit d'entrée	$f = 100$ Hz à 400 kHz		4		μV_{RMS}
Densité de tension de bruit d'entrée	$f = 10$ kHz		7		$\text{nV}/\sqrt{\text{Hz}}$
	$f = 100$ kHz		5		$\text{nV}/\sqrt{\text{Hz}}$
Stromrauschdichte	$f = 10$ kHz		4		$\text{fA}/\sqrt{\text{Hz}}$
Plage des tensions d'entrée					
Plage de tension en mode commun		-0,1		$V^+ + 0,1$	V
Rapport de réjection en mode commun	$-0,1$ V < V_{CH} < $V^+ - 2,4$ V	76	86		dB
	$V_S = 5$ V, $-0,1$ V < V_{CH} < $5,1$ V	60	74		dB
$T_A = -40...+85$ °C	$V_S = 5$ V, $-0,1$ V < V_{CH} < $5,1$ V	58			dB
Impédance d'entrée					
Différentielle			$10^{13} \parallel 2,5$		$\Omega \parallel \text{pF}$
En mode commun			$10^{13} \parallel 6,5$		$\Omega \parallel \text{pF}$
gain en boucle ouverte					
Gain en boucle ouverte					
$T_A = -40...+85$ °C	$R_L = 10$ k Ω , 50 mV < V_O < $V^+ - 50$ mV	100	122		dB
	$R_L = 10$ k Ω , 50 mV < V_O < $V^+ - 50$ mV	100			dB
	$R_L = 1$ k Ω , 200 mV < V_O < $V^+ - 200$ mV	100	120		dB
$T_A = -40...+85$ °C	$R_L = 1$ k Ω , 200 mV < V_O < $V^+ - 200$ mV	100			dB

OPAx353

Amplificateur opérationnel
Rail-à-rail rapide

ELEKTOR
ÉLECTRONIQUE et MICRO-INFORMATIQUE
INFOCARTE 2/2003

Fabricant :

Burr-Brown (Texas Instruments)
www-s.ti.com/sc/ds/opa2353.pdf

Caractéristiques :

- Entrée Rail-à-rail
- Sortie Rail-à-Rail (dans la limite de 10 mV)
- Bande passante : 44 MHz
- Taux de montée (slew rate) important : 22 V/ μs
- Bruit faible : 5 nV/ $\sqrt{\text{Hz}}$
- Taux de distorsion faible DHT+B : 0,000 6%
- Stabilité de suivi de tension

Domaines d'applications :

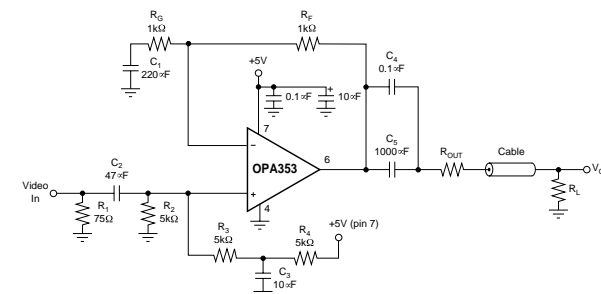
- Amplificateur pour téléphonie mobile
- Driver pour CAN
- Traitement de signaux vidéo
- Saisie de données
- Pilotage de processus
- Traitement de signaux audio
- Communication
- Filtres actifs
- Appareillage de test

Description :

La série d'amplificateurs opérationnels CMOS à caractéristiques rail-à-rail tant à l'entrée qu'à la sortie, celle des OPAx353, convient plus spécifiquement aux applications bon marché ne disposant que de peu de place. Les amplificateurs opérationnels ont

été optimisés pour travailler à une tension asymétrique faible et se distinguent par un bruit étonnamment faible de 5 nV/ $\sqrt{\text{Hz}}$. Leur bande passante de 44 MHz et leur taux de montée important de 22 V/ μs prédestinent ces Ampli Op à des applications telles que circuit de commande (driver) de CAN (Convertisseur Analogique/Numérique) et aux boucles de régulation pour la communication mobile. Leur capacité d'attaquer des charges de 75 Ω permet de les utiliser également pour le traitement de signaux vidéo. Il va sans dire que la série OPAx353 peut également être utilisée pour des applications audio ou générales. Tous les membres de la famille OPAx353 ont des paramètres identiques, quelle que soit la forme de boîtier (*single*, *dual*, *quad*).

Le OPAx353 requiert une tension d'alimentation unique de 2,5 V au minimum, sachant que la plage de mode commun de l'entrée peut se situer de 300 mV de part et d'autre de cette valeur. Les limites de tension de sortie se rapprochent à 10 mV près de la tension d'alimentation lorsque l'ampli op a à attaquer une charge de 10 k Ω . Dans le cas des versions double (*dual*) et quadruple (*quad*) les amplificateurs opérationnels sont totalement isolés les uns des autres de manière à obtenir la meilleure valeur de diaphonie (séparation des canaux) possible et éviter toute inter-influence. La plage de température des tous les OPAx353 va de -40 à $+85$ °C sachant que leur plage de température de service va elle des -55 à $+125$ °C.

Exemple d'application :

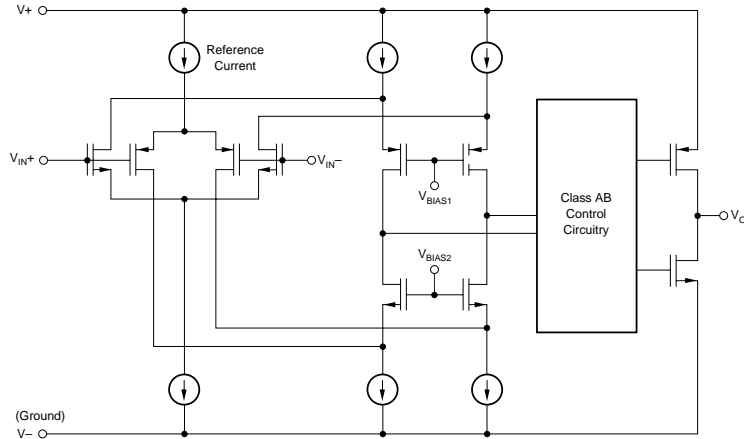
Driver de puissance vidéo à tension d'alimentation asymétrique

OPAx353

Amplificateur opérationnel
Rail-à-rail rapide



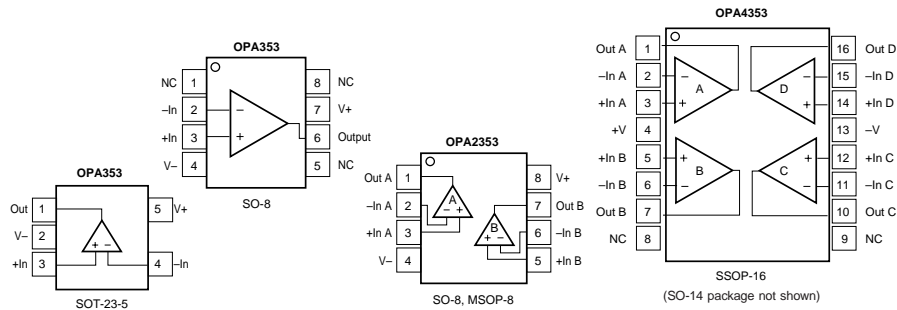
Structure interne



Valeurs maximales

Tension d'alimentation	5,5 V
Entrée de signal, tension	V-0,3 V à V+ + 0,3 V
Courant	10 mA
Court-circuit de la sortie	continu
Température de srvice	-55 à +125 °C
Température de la jonction	150 °C

Brochage et types de boîtier :



OPAx353

Amplificateur opérationnel
Rail-à-rail rapide



Caractéristiques électriques

($V_S = 2,7$ à $5,5$ V, $T_A = 25$ °C, $R_L = 1$ k Ω sous $V_S/2$, $V_{OUT} = V_S$. Gras spécifié pour $T_A = -40$ à $+85$ °C)

PARAMÈTRE	CONDITIONS	MIN	TYP	MAX	UNITÉ
Réponse en fréquence ($C_L = 100$ pF)					
Produit Gain/Bande passante	$G = 1$		44		MHz
Vitesse de montée	$G = 1$		22		V/ μ s
Temps d'établissement	0,1 %	$G = \pm 1, 2$ V/étage	0,22		μ s
	0,01 %	$G = \pm 1, 2$ V/étage	0,5		μ s
Temps de récupération après surcharge	$V_{IN} \cdot G = V_S$		0,1		μ s
Taux de distorsion harmonique total + bruit	$R_L = 600 \Omega, V_O = 2,5 V_{p-p}, G = 1, f = 1$ kHz		0,0006		%
Erreur de gain différentielle	$G = 2, R_L = 600 \Omega, V_O = 1,4$ V		0,17		%
Erreur de phase différentielle	$G = 2, R_L = 600 \Omega, V_O = 1,4$ V		0,17		Grad

Sortie

Écart tension de sortie par rapport à V+ /Masse	$R_L = 10$ kHz, $A_{OL} \geq 100$ dB	10	50	mV
$T_A = -40...+85$ °C	$R_L = 10$ kHz, $A_{OL} \geq 100$ dB		50	mV
	$R_L = 1$ kHz, $A_{OL} \geq 100$ dB	25	200	mV
$T_A = -40...+85$ °C	$R_L = 1$ kHz, $A_{OL} \geq 100$ dB		200	mV
	Courant de sortie		± 40	mA
Courant de court-circuit		± 80		mA

Tension d'alimentation

Plage des tensions d'alimentation	$T_A = -40...+85$ °C	2,7	5,5	V
Tension d'alimentation mnimale			2,5	V
Courant de repos par Ampli op		5,2	8	mA
$T_A = -40...+85$ °C			9	mA

Plage des températures

Spécification		-40	+85	°C
Service		-55	+125	°C
Résistance thermique	SOT-23-5		200	K/W
	MSOP-8/SO-8		150	K/W
	SSOP-16/SO-14		100	K/W

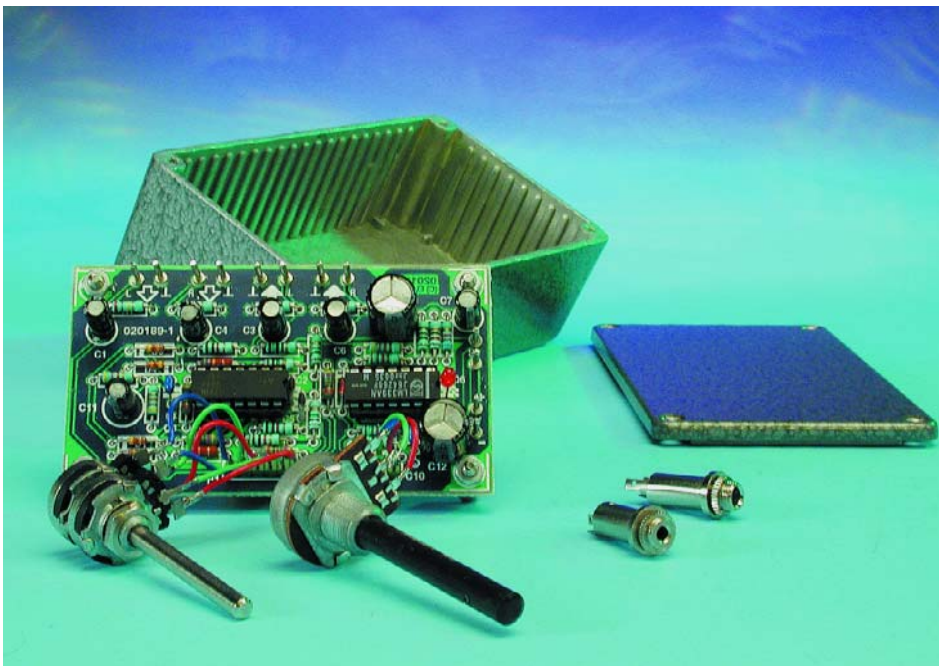
Adaptateur de niveau audio

Pour carte-son

Projet : Ton Giesberts

Texte : Sjef van Rooij

Ce montage de réglage simple permet d'ajuster très précisément le niveau d'une source de signal stéréo à la sensibilité de l'entrée d'une carte-son par exemple. Un indicateur à LED visualise une éventuelle surmodulation.



Dans nombre de domaines, la standardisation reste une illusion. Ceci est particulièrement vrai dans le cas des niveaux de signal en audio, situation à laquelle nous ne manquons pas d'être confronté un jour ou un autre. Dans le cas d'équipements audio tels que lecteurs de cassettes, tuners et autres lecteurs de CD/DVD on ne sait jamais auparavant quel est le niveau de sortie que l'on risque de rencontrer. On peut s'attendre à trouver n'importe quoi entre, disons 200 mV et 2 V. Si le côté où l'on veut brancher la source de

signal ne possède pas d'organe de réglage de niveau digne de ce nom, cette situation peut devenir franchement désagréable. La grande majorité des cartes-son pour PC par exemple ne comporte pas de réglage de niveau. Il faut noter en outre que, sur une telle carte, la tension maximale en entrée est, de par l'alimentation sous 5 V du set de puces, limitée, de sorte que l'entrée arrive en saturation dès que le

niveau de signal dépasse de l'ordre de 1 à 1,5 V.

Il est possible de contourner les inconvénients qui dérivent de telles sensibilité fixe et limitation de plage de modulation par la prise en amont de cette entrée « ligne » une électronique de régulation telle celle décrite dans le présent article. Les niveaux de signal de l'ordre de 2 V fournis par les lecteurs de CD/DVD moderne peuvent alors être ramenés à un maximum acceptable. De l'autre côté, il devient possible d'augmenter quelque peu le niveau de 200 mV fourni par les équipements plus anciens, ce qui ne peut qu'améliorer le rapport signal/bruit.

La plage de réglage de notre montage est de ± 20 dB, de sorte qu'il devrait permettre de faire face à toutes les situations nécessitant une adaptation possibles. Nous l'avons en outre doté d'un indicateur à LED pour éviter d'être surpris par une surmodulation, sachant que cette LED réagit immédiatement lorsque le signal de sortie dépasse une certaine valeur (ajustable).

La consommation de courant de notre montage n'étant que de quelques petits milliampères, on pourra, en principe, l'alimenter à l'aide d'une pile compacte de 9 V, ce qui ne fait qu'augmenter son universalité et faciliter sa mise en oeuvre.

L'électronique de réglage

Si nous nous étions contentés d'un atténuateur, nous aurions pu nous simplifier la vie vu que nous aurions pu nous contenter d'un simple diviseur de tension passif. Mais nous voulions également pouvoir amplifier sans mentionner le fait que nous tenions à réaliser un montage au comportement irréprochable dont les impédances d'entrée et de sortie ne seraient pas influencées par la position de potentiomètres.

Ceci explique notre choix d'un concept relativement mûri constitué, pour chacun des canaux, par un tampon d'entrée, un étage d'amplification à gain ajustable et d'un disposi-

tif de visualisation commun aux 2 voies. Il a en outre fallu créer, par le biais du diviseur de tension constitué par R27, R28 et C27, une masse virtuelle de manière à pouvoir alimenter les amplificateurs opérationnels asymétriquement à l'aide d'une pile de 9 V. On retrouve en **figure 1** le schéma du montage.

Nous reviendrons à l'indicateur et allons commencer par nous intéresser au tampon et à l'étage d'amplification.

Les tampons d'entrée IC1.C et IC1.B ont pour fonction de constituer une charge constante pour la source de signal. Le dimensionnement choisi est tel que l'impédance d'entrée est fixée à la valeur standard de 47 kΩ. Les amplificateurs opérationnels

sont montés en suiveurs de tension, les résistances R3 et R12 les forçant à la moitié de la tension d'alimentation. L'alimentation du circuit étant asymétrique, l'entrée a été dotée d'un condensateur de couplage qui est chargé en permanence au travers des résistances R1 et R10 de manière à éviter les bruits parasites lors de la connexion de la source de signal. Le condensateur de couplage est suivi par un réseau de protection additionnel contre les crêtes de tension, réseau constitué de deux diodes et d'une résistance.

C'est à dessein que nous avons fait appel, au niveau des étages d'amplification à gain ajustable, IC1.A et IC1.D, à des amplificateurs inverseurs. En effet, par le choix d'une entrée inverseuse, le diviseur de tension de la masse virtuelle, R27/R28, se trouve moins chargé, ce

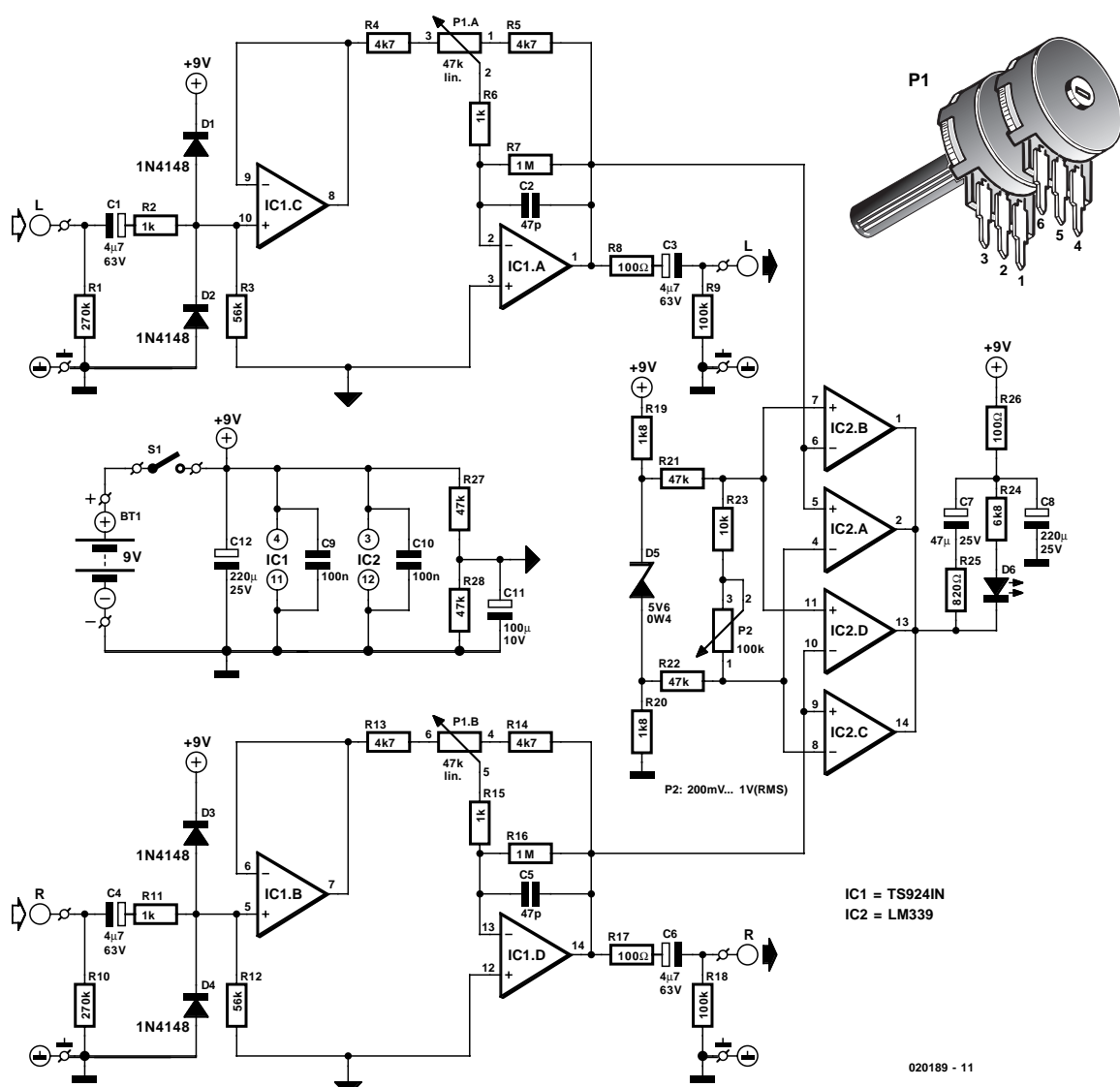
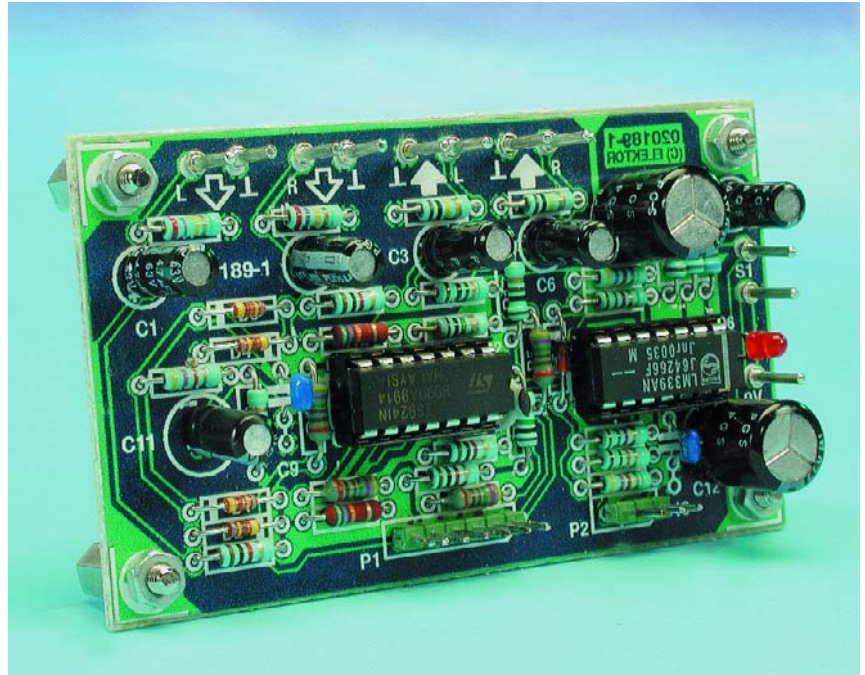


Figure 1. Grossièrement, notre adaptateur de niveau se compose de 2 sous-ensembles : une partie réglage et un indicateur de surmodulation.

qui lui permet de garder une impédance élevée et de réduire partant la consommation de courant.

Si nous nous limitons au canal gauche, nous pouvons déterminer que le gain maximal de IC1.A est égal à $(R5+P1)/R4$ et que le gain minimum répond à l'équation $R5/(P1+R4)$. En raison de petites pertes par-ci par-là la plage battue par l'excursion de réglage va d'un facteur 10, 5 à 0,09, ce qui nous donne effectivement les ± 20 dB évoqués plus haut.

La fonction de la résistance R6 (R15 pour l'autre canal) est d'éviter, en cas de câblage allongé de P1, que l'amplificateur ne devienne instable en raison de l'application d'une éventuelle capacité parasite à l'entrée de l'amplificateur opérationnel. S'il devait se faire que le curseur de P1 n'assure pas un bon contact en toute occasion, la résistance R7 (R16) garantit la présence permanente d'une réinjection vers l'amplificateur opéra-



Liste des composants

Résistances :

- R1, R10 = 270 k Ω
- R2, R6, R11, R15 = 1 k Ω
- R3, R12 = 56 k Ω
- R4, R5, R13, R14 = 4k Ω 7
- R7, R16 = 1 M Ω
- R8, R17, R26 = 100 Ω
- R9, R18 = 100 k Ω
- R19, R20 = 1k Ω 8
- R21, R22, R27, R28 = 47 k Ω
- R23 = 10 k Ω
- R24 = 6k Ω 8
- R25 = 820 Ω
- P1 = potentiomètre 47 k Ω stéréo linéaire
- P2 = potentiomètre 100 k Ω mono linéaire

Condensateurs :

- C1, C3, C4, C6 = 4 μ F/63 V radial
- C2, C5 = 47 pF
- C7 = 47 μ F/25 V radial
- C8, C12 = 220 μ F/25 V radial
- C9, C10 = 100 nF
- C11 = 100 μ F/10 V radial

Semi-conducteurs :

- D1 à D4 = 1N4148
- D5 = diode zener 5V6/0W4
- D6 = LED rouge à haut rendement
- IC1 = TS9241N (ST - chez Farnell)
- IC2 = LM339

Divers :

- S1 = interrupteur à bascule unipolaire pour montage châssis
- BT1 = pile compacte 9 V + connecteur à pression
- 2 embases-jack 3,5 mm stéréo pour montage châssis

tionnel et un forçage de la sortie à la moitié de la tension d'alimentation en toutes circonstances. Pour des raisons de stabilité, C2 (C5) limite, au gain maximal, la bande passante à 60 kHz.

Terminons ce passage en revue en

mentionnant 2 derniers composants : R8 (R17) garantit une stabilité à toute épreuve de l'amplificateur même en cas de charge capacitive de la sortie (en raison de la longueur des câbles de liaison par exemple). R9 (R18) assure le maintien en

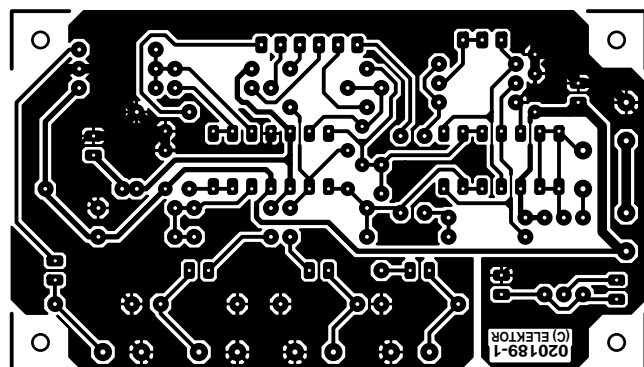
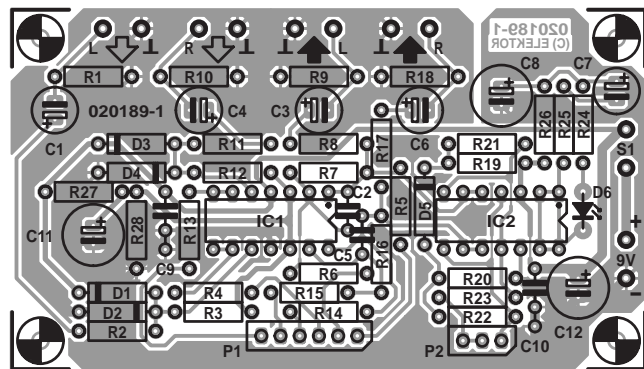


Figure 2. Les dimensions de la platine permettent de faire de l'adaptateur de niveau un montage extrêmement compact.

charge du condensateur de sortie C3 (C6), ceci pour éviter tout bruit gênant lors de la connexion de la charge en sortie.

Il nous paraît intéressant de signaler que si nous avons utilisé pour IC1 un amplificateur opérationnel rail-à-rail c'est à dessein que nous l'avons fait. Cette approche permet en effet d'accepter des niveaux d'entrée relativement élevés en dépit de la tension d'alimentation peu élevée adoptée : nominale, le maximum se situe à $3 V_{eff}$, lorsque la pile est pratiquement vide, un niveau de $2 V_{eff}$ ne pose pas de problème.

L'indicateur

Le coeur de l'indicateur de surmodulation est un quadruple comparateur, IC2. Pour chacun des canaux il sert à réaliser un comparateur à fenêtre chargé de comparer le signal de sortie de l'étage amplificateur à gain réglable à une référence ajustable elle aussi. Les sorties de IC2 sont du type à collecteur ouvert de sorte qu'il n'y a pas de contre-indication à les interconnecter. Si l'un des signaux appliqués aux comparateurs dépasse la tension de référence définie par le biais de P2, la LED D8 s'allume.

Il nous reste à évoquer certaines spécificités du montage. Par le biais de R19, D5 et R20, la référence est disposée de façon symétrique par rapport à la moitié de la tension d'alimentation de sorte que l'on ait un suivi tant de la composante posi-

tive de la tension de signal que de sa composante négative. Vu que le courant zener a été fixé à la valeur extrêmement faible de 1 mA, la tension qui chute aux bornes de D5 est de 5,3 V environ seulement. Le dimensionnement des résistances R21 à R23 et de P2 est tel que la tension de référence ajustable par le biais de P1 correspond à des tensions de signal comprises entre 200 mV et $1 V_{eff}$.

Un coup d'oeil au schéma nous permet de voir que l'on trouve, en parallèle sur la LED D6 (et sa résistance de limitation de courant R24) un condensateur, C7. Ce condensateur a pour fonction d'assurer une certaine temporisation de l'extinction de la LED de manière à mieux visualiser des crêtes de signal très brèves. R25 limite le courant de charge maximal de C7 à une valeur acceptable pour IC2. R26 et C8 servent, pour finir, au découplage de la tension d'alimentation de manière à ce que d'éventuels parasites que pourrait produire le changement d'état de la LED aient le moins d'influence possible sur les amplificateurs.

La réalisation

Nous avons dessiné une jolie platine bien compacte à l'intention de ce montage. La **figure 2** vous en propose le dessin des pistes et la sériographie de l'implantation des composants. Pour éviter que les sous-ensembles de réglage et de visualisation n'aient d'influence l'une sur l'autre, nous avons prévu à des-

sein 2 masses distinctes. L'implantation des composants est relativement serrée, raison de plus pour prendre son temps pour cette étape.

Les picots de connexion des entrées et des sorties ont été disposés tous du même côté de la platine, les 2 embases SIL destinées aux potentiomètres P1 et P2 se trouvant juste en face. C'est à dessein que nous n'avons pas monté ces organes de commande directement à même la platine, de manière à ce que chacun d'entre vous puisse décider lui-même quels types d'embases (mini-jack ou Cinch par exemple) il veut mettre en oeuvre. Cette approche offre en outre un choix plus grand au niveau du type de boîtier à utiliser.

Le câblage des embases d'entrée et de sortie vers la platine devra se faire à l'aide de câble blindé. On pourra, en ce qui concerne les potentiomètres, utiliser du fil de câblage souple classique, sachant qu'il est cependant recommandé, en vue d'obtenir la meilleure diaphonie (séparation des canaux) possible, de raccourcir autant que possible le câblage de P1. On utilisera de préférence, pour le dit potentiomètre, un exemplaire à boîtier métallique, partie que l'on reliera à la masse à l'aide d'un petit morceau de câble. Il faudra veiller, lors du câblage de P1 et P2, à ce que le niveau augmente lors d'une rotation vers la droite de ces derniers. La numérotation des broches données dans le schéma correspond à celle des embases SIL de la platine. Une fois que l'on aura mis le montage dans un boîtier on pourra doter les 2 potentiomètres d'une échelle qui donneront respectivement le gain et le niveau de l'indicateur.

On utilisera de préférence un boîtier métallique pour réduire le plus possible toute influence de signaux parasites extérieurs. Il faudra dans ce cas-là relier le métal du boîtier à la masse du montage. Nous avons utilisé un boîtier de type Box1590N1 de H.O.D., mais ce modèle ne permet pas de placer de pile. Si donc on prévoit d'alimenter le montage par pile il faudra opter pour le modèle au-dessus. Nous faut-il vraiment indiquer que le montage de la LED D6 sera fait de façon à ce qu'elle soit visible de l'extérieur.

Nous avons parlé, plus haut, de la consommation de courant du montage; elle est si faible, de l'ordre de 7 mA, ce qui permet d'envisager une alimentation par pile de 9 V. Si l'on envisage une utilisation (quasi-)permanente de ce montage il sera préférable d'utiliser un adaptateur secteur.

On utilisera de préférence un adaptateur secteur régulé sachant qu'il est important que la tension fournie hors-charge ne dépasse pas 12 V vu que c'est là la tension de service maximale de IC1.

Quelques valeurs de mesure

Impédance d'entrée		47 kΩ
Tension d'entrée max.	(THD = 0,1%)	3,2 V_{eff}
Tension de sortie max.	(THD = 0,1%)	3,1 V_{eff}
THD + N (B = 80 kHz)	200 mV en entrée, 20 mV en sortie	0,027 % (20 Hz à 20 kHz)
	200 mV en entrée, 200 mV en sortie	0,005 % (20 Hz à 20 kHz)
	200 mV en entrée, 2 V en sortie	0,002 % (20 Hz à 1 kHz)
THD + N (B = 80 kHz)		0,028 % (20 kHz)
	2 V en entrée, 200 mV en sortie	0,0026 % (20 Hz à 1 kHz)
	2 V en entrée, 2 V en sortie	0,007 % (20 kHz)
Diaphonie (séparation des canaux)		0,002 % (20 Hz à 1 kHz)
		0,01 % (20 kHz)
		> 66 dB (1 kHz)
Consommation de courant	LED éteinte	> 42 dB (20 kHz)
	LED allumée	6,7 mA
		7,8 mA

(020189)

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

NC7SZ57/58 circuits intégrés TinyLogic

Portes logiques configurables

Sjef van Rooij

L'aspect intéressant de ces circuits intégrés de la famille TinyLogic de la série UHS de Fairchild est qu'ils permettent de créer 5 configurations de portes différentes –à savoir à 2 entrées identiques ainsi qu'à une entrée inverseuse et à une entrée non-inverseuse.

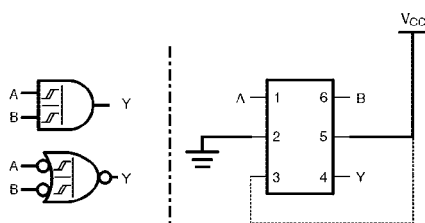


Figure 1. Porte AND à 2 entrées

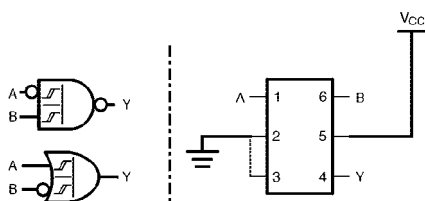


Figure 2. Porte NAND à 2 entrées à entrée A inversée

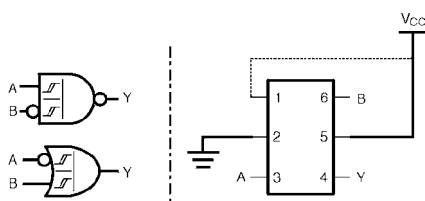


Figure 3. Porte NAND à 2 entrées à entrée B inversée

Figures 1 à 10. Les diverses possibilités de configuration.

La famille TinyLogic de Fairchild existe déjà depuis près d'un lustre. Elle comporte toute une série de fonctions logiques à une ou 2 portes présentées sous la forme de minuscules composants CMS. Leur mise en oeuvre permet, si on les compare aux boîtiers SO (*Small Outline*) de composants CMOS comparables, de gagner énormément de place lors de la réalisation d'une platine.

La caractéristique commune à tous les membres de la famille est leur vitesse élevée et leur faible consommation, ce qui explique qu'avec leur taille, ces composants soient souvent utilisés dans des appareils portables où l'on ne dispose que de très peu de place, tels que lecteurs de CD portables, téléphones GSM, assistants personnels (PDA) et autres ordinateurs portables.

La famille est subdivisée en plusieurs séries : les séries HS, HST, UHS et tout récemment ULP. Ces abréviations cachent respectivement les notions suivantes : *High Speed*, *High Speed TTL compatible*, *Ultra High Speed* et *Ultra Low Power*. Les exemplaires dont nous allons parler ici font partie de la série UHS; ils se caractérisent par une plage de ten-

sion d'alimentation allant de 1,65 à 5,5 V et leur vitesse de commutation extrêmement élevée. Si on les compare à de la logique HC classique la vitesse des types NC7SZ est, avec ses 2,8 ns sous 5 V typiques, 2 fois plus élevée. Autre caractéristique intéressante : ces circuits intégrés peuvent fournir jusqu'à 32 mA de courant, ce qui les met en mesure d'attaquer directement des LED, opto-coupleurs et autres relais.

Un circuit intégré, 5 fonctions

La spécificité la plus intéressante des NC7SZ57 et NC7SZ58 est sans aucun doute le fait qu'ils peuvent tous être configurés pour remplir 5 fonctions logiques différentes. Ces 2 circuits intégrés permettent de créer toutes les fonctions logiques à 2 entrées connues par une simple reconfiguration des broches de connexion.

Le **tableau 1** associé aux figures 1 à 10 correspondantes nous dit tout sur le principe de fonctionnement. Les **figure 1 à 5** concernent le NC7SZ57, les **figure 6 à 10** le NC7SZ58. Toutes les entrées sont dotées d'une hystérésis.

Tableau 1. Choix de fonctions

Fonction logique à 2 entrées	Composant à utiliser	Configuration de connexion
ET (AND) à 2 entrées	NC7SZ57	Figure 1
ET (AND) à 2 entrées à 1 entrée inverseuse	NC7SZ58	Figures 7, 8
ET (AND) à 2 entrées à 2 entrées inverseuses	NC7SZ57	Figure 4
NON-ET (NAND) à 2 entrées	NC7SZ58	Figure 6
NON-ET (NAND) à 2 entrées à 1 entrée inverseuse	NC7SZ57	Figures 2, 3
NON-ET (NAND) à 2 entrées à 2 entrées inverseuses	NC7SZ58	Figure 9
OU (OR) à 2 entrées	NC7SZ58	Figure 9
OU (OR) à 2 entrées à 1 entrée inverseuse	NC7SZ57	Figures 2, 3
OU (OR) à 2 entrées à 2 entrées inverseuses	NC7SZ58	Figure 6
NON-OU (NOR) à 2 entrées	NC7SZ57	Figure 4
NON-OU (NOR) à 2 entrées à 1 entrée inverseuse	NC7SZ58	Figures 7, 8
NON-OU (NOR) à 2 entrées à 2 entrées inverseuses	NC7SZ57	Figure 1
OU-eXclusif (XOR) à 2 entrées	NC7SZ58	Figure 10
NON-OU-eXclusif (XNOR) à 2 entrées	NC7SZ57	Figure 5

Confort d'utilisation

Les circuits intégrés sont dotés d'un boîtier CMS SC70 de 2 mm de long seulement. La **figure 11** en donne le brochage, un second tableau, donne l'état des broches.

Il est à noter que contrairement à ce à quoi nous sommes habitués, chaque porte de la famille TinyLogic possède son propre boîtier. Le boîtier ne comporte pas, comme d'habitude

sur la majorité des circuits intégrés, de repère identifiant la broche 1. C'est le marquage qui détermine la position de la broche 1. Lorsque cette information est lisible, la broche 1 se trouve en bas à gauche. Lors de la conception d'une platine et la réalisation d'un montage, cette approche une porte par circuit intégré présente un intérêt indéniable. Il devient possible de cette façon de placer chaque porte individuelle

Tableau 2. Spécifications

Valeurs maximales absolues

Caractéristiques électriques CC

Tension d'alimentation (V_{CC})	-0,5 V à +7,0 V
Tension d'entrée CC (V_{IN})	-0,5 V à +7,0 V
Tension de sortie CC (V_{OUT})	-0,5 V à +7,0 V
Courant de diode d'entrée CC (I_{IK}) @ $V_{IN} \leq 0,5$ V	-50 mA
Courant de diode de sortie CC (I_{OK}) @ $V_{IN} \leq -0,5$ V	-50 mA
Courant de sortie CC (source/drain) (I_{OUT})	± 50 mA
VCC CC ou courant de masse (I_{CC} / I_{GND})	± 50 mA
Plage des températures de stockage (T_{STG})	-65 °C à +150 °C
Température de jonction max sous polarisation (T_J)	150 °C
Température des connexions (T_L) (soudage, 10 s)	260 °C
Dissipation de puissance (P_D) @ +85 °C SC70-6	180 mW

Conditions de fonctionnement recommandées

Tension d'alimentation de service V_{CC}	1,65 V à 5,5 V
Tension d'alimentation de rétention de données (V_{CC})	1,5 V à 5,5 V
Tension d'entrée (V_{IN})	0 V à 5,5 V
Tension de sortie (V_{OUT})	0 V à V_{CC}
Température de service (T_A)	-40 °C à +85 °C
Résistance thermique (θ_{JA}) SC70-6	350 °C/W

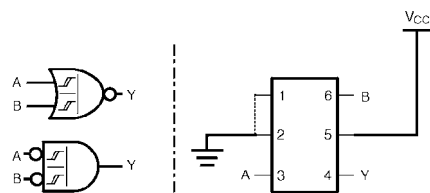


Figure 4. Porte NOR à 2 entrées

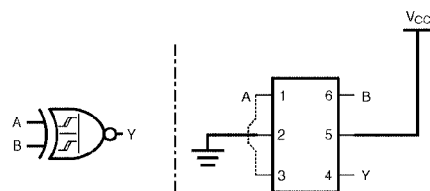


Figure 5. Porte EXNOR à 2 entrées

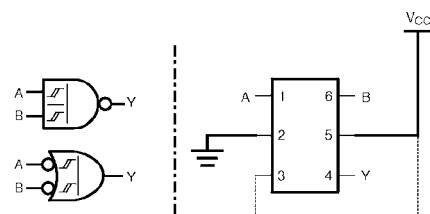


Figure 6. Porte NAND à 2 entrées

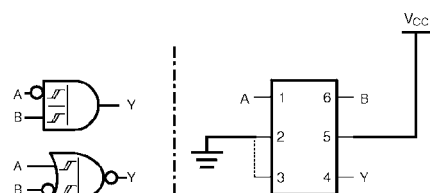


Figure 7. Porte AND à 2 entrées à entrée A inversée

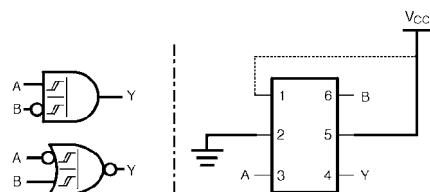


Figure 8. Porte AND à 2 entrées à entrée B inversée

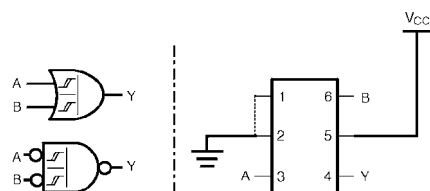


Figure 9. Porte OR à 2 entrées

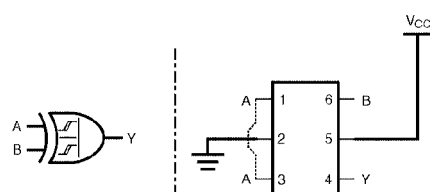


Figure 10. Porte EXOR à 2 entrées

directement sur le trajet du signal, sans qu'il ne soit nécessaire de « tirer des fils » vers un circuit intégré central à 4 portes. Cette approche permet de raccourcir les liaisons, ce qui ne peut avoir que des effets bénéfiques sur le fonctionnement de l'électronique. De plus, cela donne plus de liberté lors du dessin du circuit imprimé et évite que l'on ait, en cas de modifications imprévues, à recommencer le dessin des pistes (le *layout*) à zéro.

En conclusion

On peut donc affirmer que les portes TinyLogic constituent un développement très intéressant et une adjonction très utile à l'arsenal de circuits intégrés logiques disponibles actuellement. Compacts, rapides, petits consommateurs de courant et aisément configurables ils sont en fait étonnamment universels.

En complément de cette description succincte, le **tableau 2** récapitule les spécifications les plus importantes des NC7SZ57 et NC7SZ58.

Si ces composants étonnants ont attiré votre

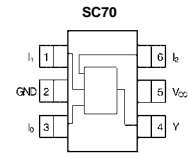
Pin Descriptions

Pin Name	Description
I_0, I_1, I_2	Data Inputs
Y	Output

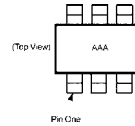
Function Table

Inputs			NC7SZ57				NC7SZ58					
I_2	I_1	I_0	Y	(I_0)	(I_2)	(I_1)	(I_2)	Y	(I_0)	(I_2)	(I_1)	(I_2)
L	L	L		H				L				
L	L	H		L				H				
L	H	L		H				L				
L	H	H		L				H				
H	L	L		L				H				
H	L	H		L				H				
H	H	L		H				L				
H	H	H		H				L				

H HIGH Logic Level L LOW Logic Level



(Top View)

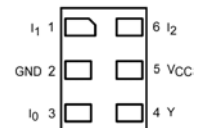


Pin One

AAA Product Code Top Mark - see ordering code

Note: Orientation of Top Mark determines Pin One location. Read the top product code mark left to right. Pin One is the lower left pin (see diagram).

Pad Assignment for MicroPak



(Top Thru View)

020353 - 12

Figure 11. Boîtier et brochages.

attention et que vous voulez en savoir davantage, nous vous conseillons de télécharger les fiches de caractéristiques consacrées aux

TinyLogic de Fairchild et disponibles à l'adresse :

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

(020353)

Dé électronique pour JdR

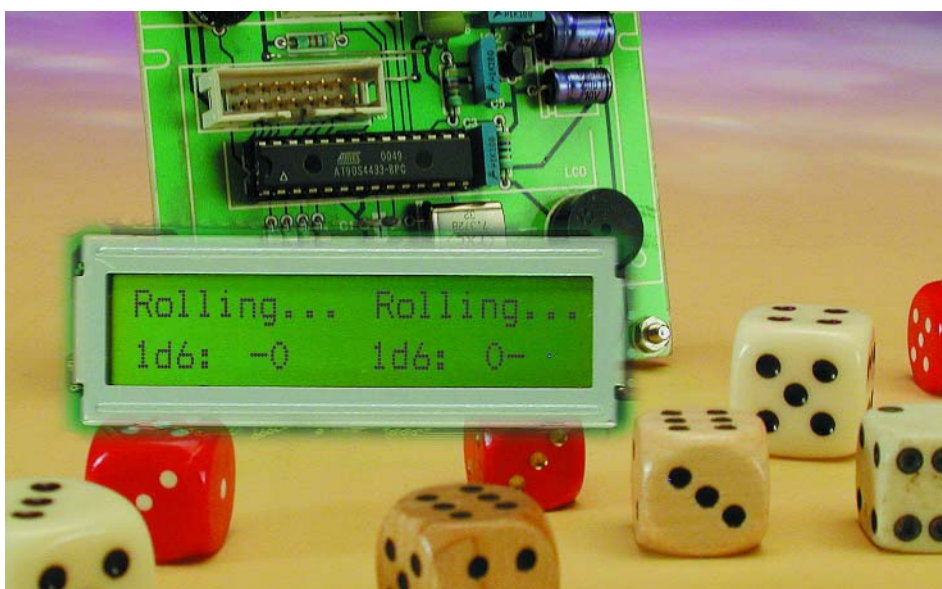
Entrez dans le monde des Jeux de Rôle

Projet : E. Eugeni

bitlab@tin.it

Les Jeux de Rôle (JdR = RPG pour *Role Playing Games* en anglais) sont régis par la succession d'événements et requièrent partant l'utilisation de différentes variantes de dés, y compris certains modèles que vous n'avez sans doute jamais vus voire dont vous n'avez jamais entendu parler.

La présente électronique à base de microprocesseur est en mesure de les simuler tous et ne manquera pas de faire les délices de tous les amateurs de JdR.



Si vous jetez un dé sur le plan d'une table vous vous attendez à le voir rouler une seconde ou deux avant de le voir afficher un chiffre compris entre 1 et 6. C'est le physique du dé, un cube classique, qui est la raison de cette plage de valeurs spécifique. L'utilisation de 2 dés voire plus permet d'augmenter cette

plage de valeurs tout en diminuant les chances d'obtenir un chiffre ou nombre donné. Une version électronique simulant une poignée de dés requiert une électronique ayant un bon niveau de comportement pseudo-aléatoire.

Les Jeux de Rôle

Comme ils sont commandés par des événements ou situations (*event-driven* disent les Anglais), les JdR requièrent un « dé » aux caractéristiques sortant du cadre des possibilités d'un dé à 6 faces classique; selon le cas, les simulations d'action peuvent aller du simple pile ou face (les joueurs ayant à choisir entre oui et non, vrai et faux, noir ou blanc, etc.), jusqu'à une évaluation pourcentuelle d'une probabilité de 1 à 100% à l'aide d'un « dé » à 100 faces « triangulaires » numérotées de 1 à 100. Entre ces limites extrêmes les JdR utilisent des dés à 8, 10, 20, voire même 3 faces. Il est évident que l'on aura des problèmes de réalisation physique d'un dé à 3 faces : il s'agit en fait d'un cube normal dont 2 des faces comportent les chiffres 1, 2 et 3. Tout cela vous semble inconnu. Vous n'avez jamais entendu parler de Jeux de Rôle ? Pourquoi ne pas faire un saut sur l'un des sites Internet suivants : www.ffjdr.org

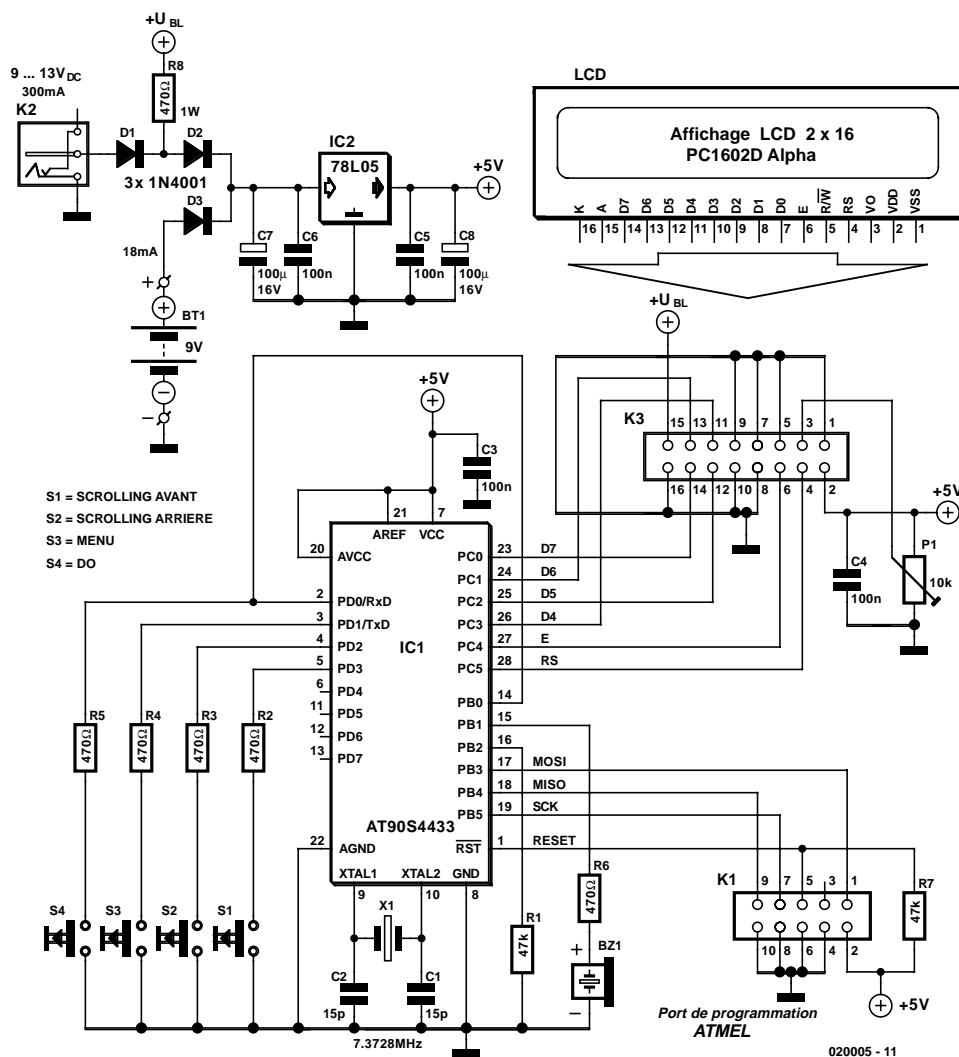


Figure 1. Schéma du dé pour JdR. Toute l'intelligence (et la mémoire) se trouve au niveau du microcontrôleur IC1.

Il suffira d'entrer les mots-clés « jeu de rôle » dans un moteur de recherche tel que Google (www.google.com) pour se trouver en présence de dizaines de liens fort intéressants.

Pour les anglophones, un tour sur les sites suivants :

www.webrpg.com

www.rpgghost.com

www.rpg.net s'impose.

La décision du modèle de dé à utiliser pour un JdR donné est du ressort du Maître de Jeu (le *Game Master*). Il/elle peut décider d'opter la totalité de la plage, mais également limiter les jets à 6 ou 10, le 100 pouvant prendre la forme d'un double 10 (premier jet pour les dizaines et second pour les unités). Cependant en gardant en mémoire la maxime latine melius abundare quam deficere (mieux en avoir trop que pas assez),

le joueur de JdR se promène souvent avec une bourse abritant une douzaine de dés voire plus, juste au cas où. Dans les lignes qui suivent, nous allons examiner le fonctionnement de l'équivalent 100% électronique d'une bourse de ce genre.

Description du schéma

L'électronique de notre dé pour JdR, représentée en **figure 1**, est une application classique d'un microcontrôleur Flash de la famille AT90 d'Atmel, entouré des organes d'entrée (les boutons-poussoirs S1 à S4), des « périphériques » de sortie (résonateur piézo BZ1 et un affichage LCD), le tout alimenté par une alimentation centrée sur IC2.

Si on vous présente ce schéma sans vous en décrire la fonction il pourrait tout aussi bien s'agir d'un lanceur de

roquettes que d'un sablier. À l'évidence, les fonctionnalités du circuit sont définies par le programme stocké dans la puce du AT90.

Il est prévu une possibilité de programmation en circuit (ICP = *In Circuit Programming*) du circuit intégré Flash d'Atmel par le biais de l'embase K1. Nous consacrerons un petit article, à paraître dans l'un des prochains numéros de ce magazine, au sujet en question, Atmel ISP (*In-System Programming*). L'alimentation de l'électronique pourra se faire par le biais d'une pile compacte de 9 V ou à l'aide d'un adaptateur secteur fournissant entre 9 et 13 volts continu sous 300 mA. En raison de sa consommation de courant importante, l'éclairage d'arrière-plan (BL) de l'affichage LCD (si tant est qu'il en ait un) n'est activé qu'en cas d'utilisation d'un adaptateur secteur. On pourra laisser la pile en place lors du branchement de l'adaptateur secteur. Il se peut qu'il vous faille redimensionner la résistance R8 en fonction du courant de BL requis par le type d'affichage LCD

utilisé. La consommation de l'affichage proprement dit est de l'ordre de 18 mA, valeur qui augmente fortement en cas d'activation de l'éclairage d'arrière-plan. Avec sa valeur du schéma, R8 définit un courant de 10 mA pour le circuit de BL. Il se peut qu'il vous faille opter pour une valeur plus réaliste (comprise entre 47 et 82 Ω) pour obtenir un niveau d'éclairage adéquat.

Le choix actuel d'affichage LCD est impressionnant : nombreux fabricants, composants neufs, de surplus voire d'occasion, les options sont diverses. Nous avons utilisé ici un exemplaire à 2 lignes de 16 caractères tout ce qu'il y a de plus courant.

Cependant, en dépit de l'offre importante, les fabricants ne sont pas arrivés à se mettre d'accord pour une standardisation du brochage du connecteur 16 points ce qui explique la présence dans le circuit du dé pour JdR d'une embase destinée à recevoir un connecteur relié à l'affichage LCD par le biais d'une courte longueur de câble en nappe. La broche 1 de l'embase sera reliée au contact 1 du connecteur présent sur l'affichage, la broche 2 au contact 2 etc.

La même remarque d'absence de standardisation s'applique aux orifices de fixation des

modules LCD, ce qui explique l'absence sur la platine de trous destinés à recevoir les entretoises servant à la fixation de l'affichage LCD sur la platine du dé pour JdR.

Les affichages LCD à rétro-éclairage n'ont pas tous leur dispositif d'éclairage (ampoule ou LED) connecté aux broches 15 et 16 comme le montre le schéma de la figure 1.

Dans certains cas, ces 2 connexions se trouvent ailleurs sur le module LCD, de sorte qu'il faudra, le cas échéant, rerouter les conducteurs y allant. Si vous avez le moindre doute, essayez de trouver sur Internet la fiche des caractéristiques techniques de votre affichage.

L'ajustable P1 permet de jouer sur le

contraste de l'affichage LCD de manière à permettre une visualisation sous différents angles. Si, après avoir appliqué la tension d'alimentation au circuit, vous ne voyez s'afficher que des petits rectangles noirs voire rien du tout, jouez sur P1 jusqu'à l'apparition de texte. Si, quelle que soit la position de P1, vous ne voyez rien s'afficher sur le module LCD, il y a un problème au niveau, soit du matériel, soit du logiciel.

La construction

La réalisation du dé pour JdR ne devrait guère poser de problème vu l'existence d'un circuit imprimé tout fait disponible auprès des adresses

Liste des composants

Résistances :

- R1, R7 = 47 kΩ
- R2-R6 = 470 Ω
- R8 = 470 Ω/1 W (cf. texte)
- P1 = ajustable 10 kΩ

Condensateurs :

- C1, C2 = 15 pF
- C3-C6 = 100 nF
- C7, C8 = 100 μF/16 V

Semi-conducteurs :

- D1 à D3 = 1N4001
- IC1 = AT90S4433-8PC, programmé EPS020005-41
- IC2 = 78L05

Divers :

- BT1 = pile 9 V
- BZ1 = résonateur CC 5 V
- K1 = embase à 2 rangées de 5 contacts
- K2 = embase-jack d'alimentation encartable
- K3 = embase à 2 rangées de 8 contacts
- S1-S4 = bouton-poussoir unipolaire à contact travail encartable
- X1 = quartz 7,372 8 MHz
- module LCD à 2 lignes de 16 caractères avec, de préférence un rétro-éclairage tel que, par exemple, PCI602D, LM052L ou LM016
- en option : adaptateur secteur : 9 à 13 V CC 300 mA

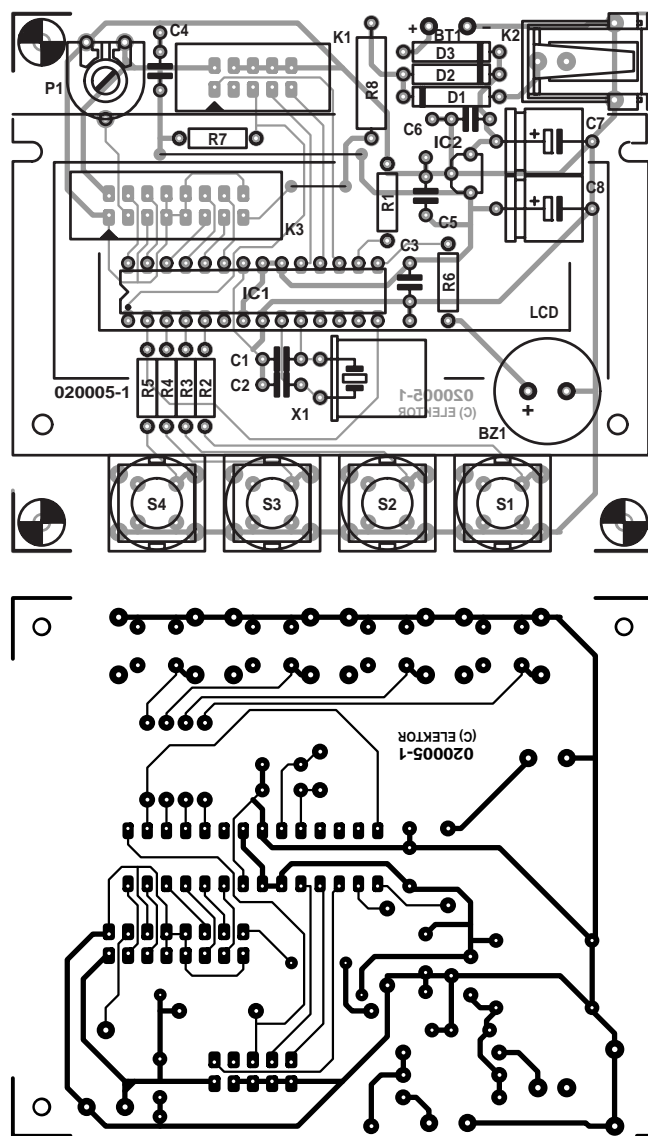


Figure 2. Dessin des pistes et sérigraphie de l'implantation des composants de la platine dessinée à l'intention de cette réalisation et disponible auprès des adresses habituelles.

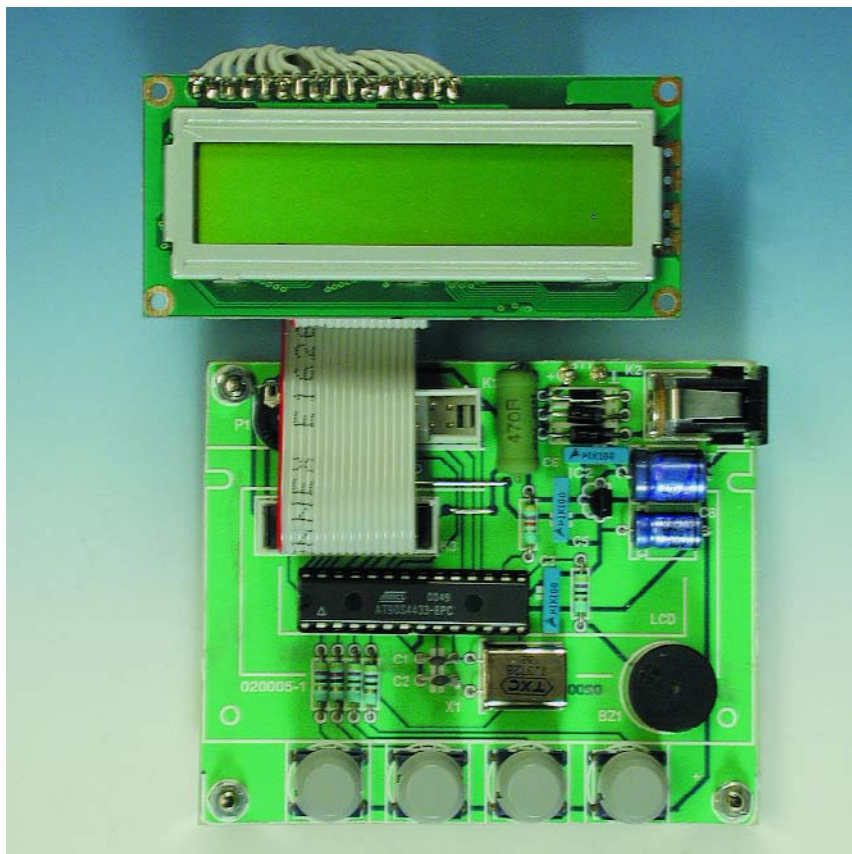


Figure 3. Exemple terminé de notre dé pour JdR (prototype).

habituelles. La figure 2 vous en propose le dessin des pistes et la sériographie de l'implantation des composants à côté de la liste des composants.

Nous n'avons pas de remarque particulière à faire pour cette étape de réalisation, à ceci près qu'il est recommandé d'utiliser un support pour le microcontrôleur et de vérifier les connexions électriques des boutons-poussoirs avant d'enficher ces boutons carrés dans les orifices prévus à leur intention. En d'autres termes, assurez-vous que la fermeture des contacts arrive bien aux îlots de soudure prévus. Le schéma donne la fonction de chacun des boutons-poussoirs S1 à S4.

Dernière remarque : il n'est pas nécessaire de connecter quoi que ce soit au port de programmation (K1) Atmel si vous utilisez un microcontrôleur programmé.

Mise en oeuvre pratique

La dénomination des 4 boutons sur le circuit imprimé est, de la gauche vers la droite :

- S4 DO (au travail !)
- S3 MENU
- S2 SCROLLING ARRIERE / '-' (moins)
- S1 SCROLLING AVANT / '+' (plus)

Lors de l'application de la tension d'alimentation vous verrez s'afficher, pendant quelques secondes, un message de bienvenue. Le programme passe ensuite en mode MENU et prend le paramétrage par défaut.

Le message de bienvenue est en fait :

```
RPG DICE
V 1.0
```

Il est accompagné par une triplette de bips courts produits par le résonateur. La page de MENU est en 2 lignes : la ligne du haut est statique et se limite à l'affichage du mot « MENU ». La ligne du bas affiche elle le mode de fonctionnement actuel. Il est possible de passer d'un mode à l'autre par une action sur les boutons SCROLLING ARRIERE et SCROLLING AVANT.

```
MENU
Roll Dice
```

```
MENU
Armor Class
```

```
MENU
Hit Points memo
```

Une fois que l'on a vu apparaître un mode donné sur la ligne du bas, il pourra être activé par une action sur le bouton DO. Pour revenir au mode MENU il suffit d'actionner le bouton MENU (S3). Notez au passage l'existence d'un raccourci utile : si vous en êtes au point de confirmer un choix vous pouvez actionner la touche MENU au lieu de la touche DO. En d'autres termes la touche MENU pourra servir tant pour entrer dans un mode donné que pour en sortir. Ce comportement « inusité » a été suggéré par un certain nombre de Joueurs de Rôle vu qu'ils préfèrent à avoir actionner à répétition la même touche plutôt qu'à avoir à changer de bouton.

Mode Jet de dé

À partir de la première page de MENU,

```
MENU
Roll Dice
```

vous pouvez appuyer ou sur DO ou sur MENU pour passer à la page de sélection de dé.

```
Select dice
1d4:
```

La page de choix de dé se compose d'un message statique « *Select dice* » et d'une ligne affichant le type de dé simulé actuellement. En respect des conventions régissant le standard RPG, les types de dés sont représentés sous forme compacte du type « 1dn » dénomination dans laquelle « 1d » est un préfixe et n représente le nombre de côtés du dé concerné.

La version actuelle du progiciel (*firmware*) permet de simuler le jet des dés suivants :

1d4 : dé ne comportant que 4 faces (en forme de pyramide à base triangulaire).

1d6 : le dé classique à 6 faces.

1d8 : un dé à 8 faces (2 pyramides à base carrée collées l'une contre l'autre par la base).

1d10 : un dé à 10 faces (2 pyramides à base pentagonale collées l'une contre l'autre par la base).

1d20 : un dé à 20 faces (ressemble à une

Puis-je m'y fier ?

La première question que l'on ne manque pas de se poser lors de la réalisation de l'équivalent électronique d'un objet aussi commun qu'un dé en bois est bien entendu : Quel est le degré d'« aléatoire » offert par une approche aussi déterministique qu'un microcontrôleur ? Si l'on veut coder dans un progiciel un minimum de caractère aléatoire (notre aléatoire) on fait inévitablement appel, en fait, à un algorithme pseudo-aléatoire dont le résultat final peut être deviné vu que l'on a affaire à des facteurs connus et prévisibles. L'approche utilisée par l'auteur est différente et pourrait être comparée à une cible pour fléchettes (darts) tournant rapidement sur elle-même pendant que vous essayez de lancer votre fléchette. Comment vous serait-il possible de deviner le point d'impact de la fléchette ? Vous voyez bien la cible, mais comme il vous est impossible de reconnaître ou d'anticiper la position exacte de chaque secteur numéroté, en raison de la vitesse de rotation élevée, vous vous trouvez dans l'impossibilité de viser quelque secteur donné que ce soit. De plus, si vous lancez plusieurs dizaines de fléchettes les yeux bandés même si vous le faites à intervalle régulier, il vous est impossible de prévoir à l'avance chacun des points d'impact. Le

fonctionnement interne du programme du dé pour JdR simule de très près la situation évoquée dans les lignes précédentes. De manière à simuler du mieux possible le comportement aléatoire d'un vrai dé de bois, le résultat de chaque « jet » de dé simulé n'est pas fourni par un algorithme pseudo-aléatoire mais par la simple capture d'un mot à 16 bits se trouvant dans un compteur dont le contenu est modifié des centaines de fois toutes les secondes. Ce type de compteur existe physiquement dans le microcontrôleur mais fonctionne sans intervention directe de l'unité de traitement (la CPU). Lors d'une action sur le bouton DO, le progiciel jette un coup d'oeil sur le contenu du compteur et en dérive la valeur affichée suite à une série d'opérations arithmétiques. Inutile de le signaler, la vraie variable « aléatoire » de ce jeu est l'utilisateur vu qu'il est impossible de deviner, intentionnellement ou non, le résultat de chaque jet à partir des résultats précédents ou en suivant le déroulement de l'exécution du programme. Les amateurs de programmes se réjouiront d'apprendre que le code-source du dé pour JdR est disponible au téléchargement gratuitement depuis le site Internet d'Elektor sous la dénomination **EPS020005-11**, mais qu'il est également disponible auprès des adresses habituelles.

petite balle de golf rabotée).

1d100 : également connu sous la dénomination « dé de pourcentage », ce dé est souvent remplacé par une paire de 1d10 que le joueur lance 2 fois pour obtenir respectivement les dizaines et les unités de pour-cent.

Comme d'habitude, l'utilisateur peut passer la liste de types de dés en revue par action sur les boutons de défilement (*scroll*). Une fois que le dé souhaité apparaît à l'affichage le joueur pourra le « faire rouler dans sa main » et le jeter en appuyant et en maintenant la pression sur le bouton DO; mais il peut également, s'il le désire, arrêter par une action sur le bouton MENU.

Suivons, pas à pas, une situation de jeu :

Étape 1 :

```
Select dice
1d6:
```

Étape 2 : Appuyer en la maintenant enfoncée, sur la touche DO pendant un bref instant. On voit apparaître un

```
Rolling... Rolling...
1d6: -0 1d6: 0-
```

Le patron « -0 » et « 0- » de la ligne supérieure apparaît plusieurs fois, confirmant visuellement le mouvement des dés. Simultanément, le résonateur produit une série de clics imitant un dé en train de rouler sur une table en bois.

Étape 3 : Vous relâchez enfin le bouton et

l'on voit apparaître le résultat à l'écran (4 dans le cas présent).

```
[DO] to roll
1d6: 04
```

À partir de là, il suffit, pour un nouveau jet de dé, d'appuyer sur le bouton DO et de le maintenir enfoncé, et de répéter la même séquence que lors de l'étape 1. Notez qu'il n'est pas possible maintenant de changer le type de dé vu que le message qui s'affiche est « [DO] to roll » et non pas « Select dice » comme précédemment. Pour changer de type de dé il suffira d'appuyer sur le bouton MEN pour revenir à la page de choix de dé.

Mode de classe d'Armement

Lorsque le circuit se trouve en mode de classe d'armement (*Armour Class*)

```
MENU
Armour Class
```

l'utilisateur peut actionner soit DO soit MENU pour accéder à la page suivante :

```
Input THACO
09
```

page où il/elle est invité(e) à choisir un nombre entre 01 et 20, connu sous la dénomination de valeur THACO (THAC zéro).

Il est possible d'incrémenter ou de décrémenter le nombre à 2 chiffres par action sur les boutons SCROLLING AVANT/« + » et SCROLLING ARRIERE/« - ». La version actuelle du progiciel ne possède pas de fonction d'autorépétition (*auto-repeat*) des touches de sorte qu'il faudra appuyer une fois pour chaque unité d'incrément ou de décrémentation. En tout état de cause le résonateur émet un court signal sonore lors de chaque changement d'un chiffre. Lorsque l'affichage visualise une valeur THACO convenable vous pouvez appuyer sur DO pour passer à la page suivante,

```
[DO] to hit!
Roll: AC:
```

ou MENU pour quitter et revenir en début du mode Armour Class.

A partir de là, le cycle se répète comme en mode Roll : actionner et maintenir DO, voir passer les patrons « -0 » / « 0- » à côté des mots « Roll » et « AC », et noter le bruit de roulement des dés émis par le résonateur.

```
[DO] to hit! [DO] to hit!
Roll:-0 AC:-0 Roll:0- AC:0-
```

Lorsque vous relâchez le bouton DO, vous obtenez 2 nombres donnant respectivement le nombre repré-

senté par jet de dé(s) (*Roll*) et celui correspondant au coup (*hit*) sur la classe d'armement ($AC = \textit{Armour Class}$) :

```
[DO] to hit!
Roll:13 AC:-04
```

Les nombres cités en exemple ci-dessus nous apprennent que le jet de dés est de 13, et vu que la valeur prédéfinie de THAC0 était de 9, tout combattant ayant une valeur AC égale à ou supérieure (en négatif) à $9 - 13 = -4$, est considéré comme ayant été touché ou battu. Remarque : le mode de AC est utilisé lorsque le scénario de votre JdR requiert un « calcul » de résultat de confrontation, qu'il s'agisse d'un simple corps à corps (un contre un), d'une structure de bataille plus complexe, voire même de Star Wars.

Mode Notation des Points

Ce mode (*Hit Points Memo*) est une fonction simple n'ayant que peu affaire avec la génération de nombres aléatoires ou le jet de dé(s). Il ne s'agit en fait de rien de plus qu'une alternative électronique au petit calepin de papier où l'on note les résultats, documents que l'on ne trouve jamais lorsque l'on en a besoin. Nous osons croire qu'il s'agit là d'une fonction utile.

Sachant que la majorité des jeux

impliquent une notation de score, sous une forme ou une autre, et que les JdR font eux appel à des points de coup (*Hit Points*) servant à mesurer la « vitalité » de chaque personnage, cette fonction de Hit Points Memo pourra servir à mémoriser en permanence votre score et de le stocker jusqu'à la session suivante. Comme d'habitude, en partant de la page MENU

```
MENU
Hit Points Memo
```

vous pouvez appuyer sur DO ou MENU pour passer à la page suivante,

```
Change HP value
100
```

puis, il vous faudra appuyer sur SCROLLING AVANT ou SCROLLING ARRIERE pour, respectivement, incrémenter ou décrémenter la valeur sur une plage allant de 001 à 250.

Comme nous l'avons vu dans le cas de la valeur THAC0, la version actuelle du progiciel ne supporte pas de réponse à auto-répétition des touches de sorte qu'il faudra appuyer une fois pour chaque incrément ou décrement.

Cependant, de manière à accélérer

le processus, on pourra, lorsqu'il faudra travailler sur le chiffre des dizaines ou des centaines, commencer sur le bouton DO, le maintenir enfoncé et appuyer ensuite sur la touche SCROLLING AVANT ou SCROLLING ARRIERE pour avoir une incrémentation ou une décrémentation par 10 d'un seul coup. Si le thème Hit Points vous intrigue, sachez que chaque joueur de JdR se voit attribuer, en début de jeu, un niveau de « vitalité » qui vaut dans la plupart des cas de 4 à 6 fois le niveau du personnage. Dès lors, si le joueur est touché, qu'il boit du poison, qu'il a un accident ou qu'il souffre quelque dommage physique ce soit, il/elle perd un ou plusieurs Hit Point(s) en fonction de l'importance de chaque événement négatif concerné.

Est-il bien nécessaire de le préciser, lorsque la valeur Hit Point est tombée à zéro, le joueur quitte la scène. Si le jeu que vous jouez est court vous pouvez démarrer avec 15 points et vous retrouver sur le carreau (0) au bout de 3 heures, mais si le jeu est long et qu'il dure des jours voire des mois, il vous faudra garder trace de l'historique de votre vitalité.

Le calepin HIT Points Memo pourra être utile pour remplacer un carnet voire une pile de feuillets de papier, vu que la mémoire non volatile du microcontrôleur conserve ses données pour des dizaines d'années et ne risque pas d'être effacée par une tasse de café renversée malencontreusement sur un feuillet stratégique. Nous vous souhaitons bien du plaisir !

(020005-1)

FAQ concernant les Armour Classes et Tutti Quanti

Une AC, c'est quoi ça ?

Une Armour Class est un nombre entier représentant la capacité pour un adversaire de faire face à notre attaque en termes et de vulnérabilité de son armure, de son être ou de quoi que ce soit d'autre et de capacité de s'esquiver pour éviter d'être atteint par un coup. Chaque objectif potentiel d'un Jeu de Rôle se doit de posséder sa propre valeur AC, située, en règle générale, dans une plage allant de -19 à 1, traduisant de façon évidente ses états de service en combat.

Une valeur THAC0, connais pas !

Une valeur THAC0, acronyme de Throw to Hit Armour Class zero, est également un nombre entier, mais au lieu de concerner un adversaire attaqué, il qualifie un joueur attaquant. Chaque joueur se doit d'avoir un THAC0, situé dans une plage allant de 1 à 20.

Un Throw to Hit, pas vu !

Un throw to hit n'est rien de plus qu'un unique jet de dé 1d20, dont le résultat est à soustraire de THAC0 pour donner un résultat valide du combat en cours. En d'autres termes, la combinaison d'une valeur THAC0 du joueur attaquant, une valeur AC du joueur

se défendant, le tout assaisonné du caractère aléatoire du dé, se traduit par un scénario JdR à la fois convainquant et fascinant.

Le joueur devra, pour tirer le maximum lorsqu'il se trouve en mode Armour Class, passer par les étapes suivantes :

1. Prédéfinir une valeur THAC0 adéquate (le circuit mémorise la dernière valeur prédéfinie).
2. Effectuer une action « Throw To Hit ».
3. Ne pas tenir compte du résultat du jet de dé et noter uniquement la valeur AC.
4. Marquer comme étant battu tout joueur ayant une valeur AC supérieure ou égale à la valeur notée.

Notez au passage que les scores de jet de dé dont il n'a pas été tenu compte au point 3 peuvent être utiles dans certaines situations de jeu spécifiques telles que ce que l'on appelle un effleuré (*fumble*), associé au score 1 et le coup critique (*critical hit*) associé lui à 20 points. Les lecteurs qui aimeraient en savoir plus au sujet du monde fascinant des JdR peuvent consulter des milliers de pages Web en entrant un mot-clé tel jeu de rôle, *fumble* (ils arriveront sur la page de la Guilde Les Maîtres du Temps (<http://pages.infinet.net/wezel/fumble.html>), ou pour ceux qui maîtrisent la langue de Shakespeare, RPG ou Role Playing Games (où l'on retrouve l'orthographe incorrecte de Roll) dans un moteur de recherche tel que Google (www.google.com).

C8051F30X Development Kit

Le kit de développement C8051F30X de Cygnal est destiné, comme le suggère sa dénomination, à se faire la main sur le X8951F30X de Cygnal, un processeur de la famille 8051 développé par cette société.



La première chose qui frappe lorsque l'on examine la platine, qui arrive dans un paquet fort bien pensé, est qu'elle diffère énormément de l'image illustrant le coffret. D'un composant devant comporter quelque 64 contacts, on passe à un circuit intégré CMS qui n'en a pas même une douzaine. Le coffret comporte tout ce dont on a besoin pour se lancer dans le développement d'applications à base de C8051F30X, à savoir :

- La carte-cible du C8051F30X baptisée C8051F300-TB avec une section de prototype à pastilles
- Un adaptateur sériel EC2 (convertisseur de protocole RS-232 vers carte-cible)
- Un câble sériel
- Une alimentation à fiche « continentale » (donc utilisable dans l'Hexagone)

- Un câble multibrin à 10 conducteurs
- Un CD-ROM qui comporte l'environnement de développement intégré (IDE = *Integrated Development Environment*), de la documentation technique aisément accessible par le biais de l'Explorateur. Pas moins de 255 Moctets d'occupés sur ce CD-ROM qui justifie bien son utilisation.
- Une feuille A3 qui illustre très fidèlement les différentes étapes à suivre avant de pouvoir essayer le premier programme de démonstration. C'est une approche simple, mais ô combien efficace dont pourraient s'inspirer nombre de fabricants de processeurs offrant des kits de développement.

Note : Il est impressionnant de voir comment une simple feuille A3 bien pensée permet de passer en revue les différentes étapes de mise en oeuvre d'un kit de développement aussi performant que celui-ci.

- Un manuel utilisateur de 4 pages. En dépit de sa minceur, ce mode d'emploi succinct illustre bien les différentes étapes à suivre pour un nouveau projet : installation du matériel, du logiciel, utilisation des outils de Keil, etc.

Un premier coup d'oeil à la carte d'évaluation réserve une surprise : mais où se trouve bien ce microcontrôleur 8051 ? L'habitude nous pousse à chercher un composant DIL à 40 broches... Ceci nous amène à parler de la famille C8051F30X. Cette famille comprend plusieurs membres, à savoir (pour le moment ?) les F300, F301, F302, F303, F304 et F305; il s'agit à chaque fois d'un microcontrôleur à noyau 8051 rapide à mémoire Flash, à 11 broches seulement, qui se différencie l'un de l'autre par leur capacité de mémoire et la présence ou non d'un convertisseur analogique/numérique (CAN). Ils se distinguent également par une tension d'alimentation relativement faible, de 3 V (2,7 mini, 3,6 V maxi). Sa consommation est étonnamment faible à fréquence d'horloge faible (11 µA à 32 kHz). Le C8051F300, puisque c'est de lui qu'il s'agit sur cette carte d'évaluation, possède 8 Koctets de mémoire Flash programmable en site en secteurs de 512 octets, 256 octets de mémoire de données en RAM. Son architecture d'instruction en pipeline permet l'exécution de 70% des instructions en 1 voire 2

cycles d'horloge.

La caractéristique la plus étonnante se manifeste lorsque l'on examine la fiche de caractéristiques du C8051F300 : si ce composant ne possède que 11 broches, 8 d'entre elles constituent le port P0. Il n'en reste plus que 3 : une pour l'alimentation, la seconde pour la masse et la troisième pour, selon le cas, le signal d'horloge ou de réinitialisation (*reset*). L'installation de l'environnement de développement intégré (IDE) est étonnamment simple. Il suffit d'activer le bouton d'installation en fin de phase d'*auto-run* du CD-ROM mis dans le lecteur.

Un petit tour d'horizon de la documentation fournie sur le CD-ROM est une approche préliminaire fortement recommandée. Près d'une trentaine de Notes d'applications font partie du menu. Les 415 pages du manuel du Macro-assembleur de Keil valent sans aucun doute le détour, de même d'ailleurs que les 393 pages consacrées au Compilateur Cx51.

Après avoir effectué la connexion du matériel en respectant les instructions de la documentation jointe, nous pouvons effectuer nos premiers pas. Tout se passe comme indiqué.

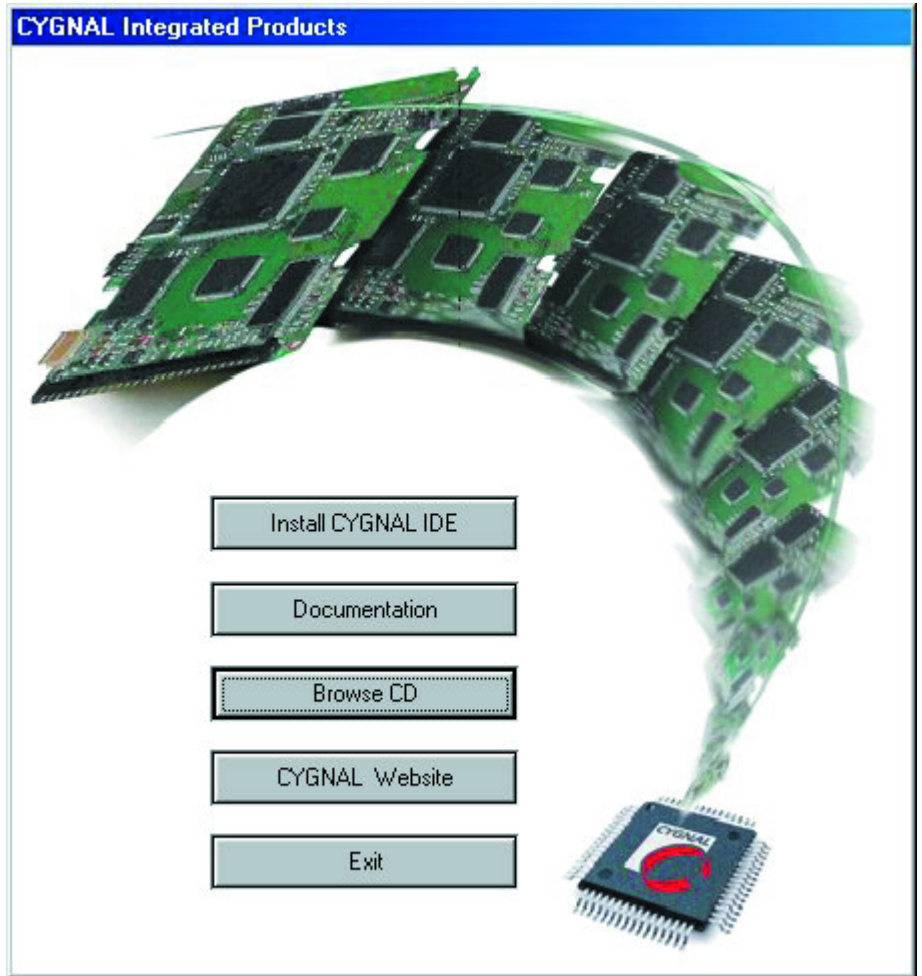
Il est important de s'assurer que la carte est bien reliée au port COM concerné, COM1 par défaut, sinon il faudra modifier le port en question dans le menu Options, vu que sinon on se voit, logiquement, confronté à toute une série de messages d'erreur. Il manque, nous semble-t-il, quand même un pas entre la *build* et l'exécution du programme de démo (point vert = Run), le transfert du programme vers le composant (DL pour *download*). On peut suivre l'évolution du processus. Le point vert (RUN) s'allume ensuite. Il ne reste plus qu'à lancer l'exécution du programme.

Hourra ! La LED verte clignote.

Signalons, en guise de conclusion, l'existence d'un site Internet bien fait à l'adresse :

www.cygnal.com.

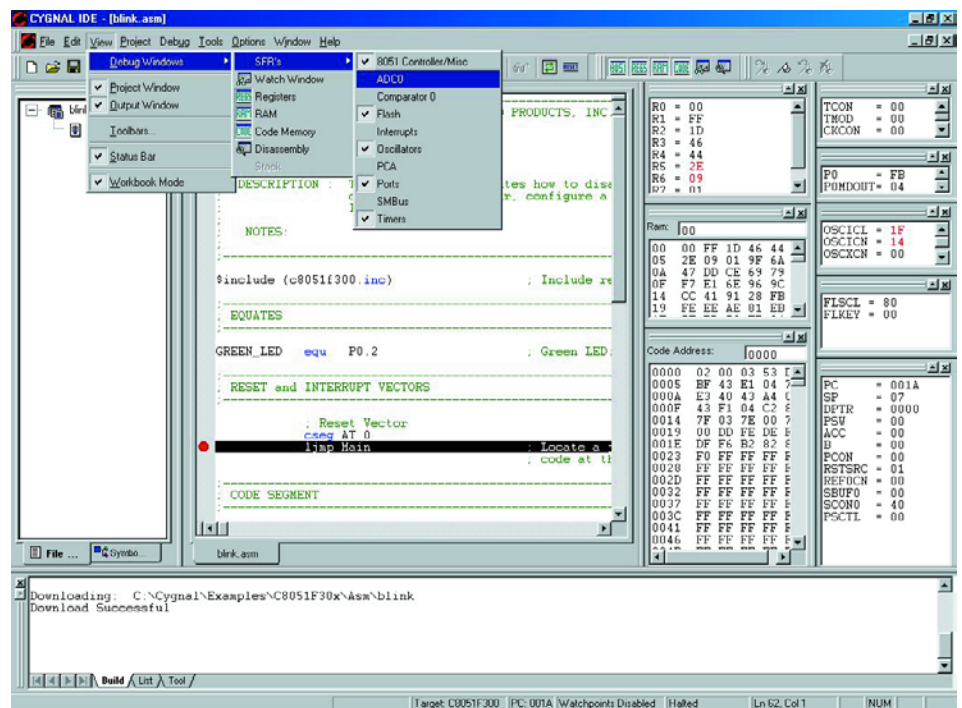
Notons que Cygnal propose différents composants à base du C8051F pour des applications spécifiques, détection de gaz, chargeur d'accus, balance, alimentation interruptible... Il n'existe pas, à notre connaissance, d'autre canal qu'Internet pour



acquérir ce kit au prix extrêmement abordable (99 US\$ + 20 US\$ de frais de port et d'emballage au minimum).

En résumé, un kit très bien conçu en dépit de son apparente simplicité.

(027165)



La vérité au sujet des accus

Les Li-Ion sont-ils vraiment aussi mauvais ?

Harry Baggen

Dans la société moderne, de plus en plus d'appareils peuvent se targuer d'être « sans fil ». Ceci n'est pas uniquement vrai dans le monde de la communication mais également dans celui de l'alimentation. Les ordinateurs portables et les téléphones GSM sont les exemples les plus frappants de cette évolution. Grâce aux accumulateurs puissants dont ils sont dotés, ils sont

en mesure de fonctionner pendant des durées (relativement) importantes sans être reliés au secteur. Mais qu'en est-il réellement de la qualité des différents types d'accus et quelle est leur durée de vie ?



Les familles d'appareils font de plus en plus appel, pour leur alimentation, à des piles (rechargeables le cas échéant auquel cas on parle d'accumulateurs). Tant à l'intérieur qu'à l'extérieur, nous mettons des piles dans notre thermostat de chauffage central, nos télécommandes en tout genre, nos lampes de poche, nos récepteurs de radio portatifs et autres baladeurs (qu'ils soient MP3 ou non).

Dans le cas des équipements à la consommation plus importante tels que les appareils photo numériques, les caméscopes et autres ordinateurs portables de courant nous faisons normalement appel à des accus. Nos téléphones GMS ne peuvent pas non plus se passer d'accu, vu qu'ils sont « allumés » en quasi-permanence et requièrent d'être rechargés à intervalle plus ou moins régulier.

Il existe différents types d'accus, nous n'avons pas la prétention de vous l'apprendre, à vous qui lisez régulièrement Elektor : Cd-Ni, NiMH, Li-Ion et Li-Polymère (sans parler de l'accu Pb (plomb) scellé rarement utilisé dans les appareils modernes. La variété au cadmium-nickel est la plus ancienne et convient fort bien aux applications requérant des courants élevés en raison de sa résistance interne faible. Ce type d'accumulateur utilise du cadmium, un métal lourd polluant, et devra, dans les années qui viennent, avoir totalement disparu de la scène européenne. Pour le remplacer on a développé l'accu NiMH (Nickel-Métal Hydrure) dont les qualités se sont bien améliorées au point de pouvoir prétendre être un parfait remplaçant des CdNi (encore que sa résistance interne soit encore quelque peu plus importante). Pour les domaines nécessitant une densité énergétique élevée (quantité d'énergie la plus importante possible dans le volume le plus faible possible), on a développé les accus Li-Ion (Li = lithium) qui présentent cependant l'inconvénient de requérir un chargeur spécifique. L'accu Li-Ion Polymère est un dérivé du précé-

dent et, dans sa forme flexible, est utilisé dans les AP (Assistant Personnel), ces petits « ordinateurs » de poing.

Chacun de ces types d'accu a ses avantages et ses inconvénients spécifiques. Le site **Digit-life.com** [1] donne un panorama complet de leurs caractéristiques.

D'après des tests effectués par des organisations de défense des consommateurs de différents pays, les accus Cd-Ni modernes ont des performances très proches de celles de leurs homologues NiMH. Les Cd-Ni ne semblent pratiquement plus souffrir du fameux effet de mémoire de sinistre mémoire. Il semblerait cependant que des tests sur Internet prétendent souvent le contraire. Il est difficile de savoir qui a vraiment raison. Ces caractéristiques peuvent être spécifiques à un type ou à une marque donnée, mais il semblerait que ces 2 types d'accu aient dans l'ensemble des caractéristiques similaires.

Le Li-Ion semble être l'accu idéal pour les systèmes mobiles : une capacité importante pour des dimensions compactes, auto-décharge faible et résistance interne faible. Les seules critiques relevées concernent leur (re)charge ! Grâce à la

mise en oeuvre de toutes sortes de combinaisons chimiques bien dosées, le lithium, dont la tendance à l'explosion est connue, reste sagement à l'intérieur de l'accu. Malheureusement cependant, le Li-Ion s'est fait ces derniers temps une mauvaise réputation dans un certain nombre de magazines d'Outre-Rhin consacrés à la micro-informatique. Des essais ont démontré que les accus Li-Ion sont sujets à un vieillissement naturel des produits chimiques qui les constituent. Le fait que l'accu soit utilisé ou non n'a pas

Choosing a Battery that Will Last Page 3 Netscape

PowerPulse.Net 19 November 2002

News | Features | Resources | Media | Power Communities

The Web's Leading Power Electronics Publication

Technical Paper

Choosing a Battery that Will Last Page 3

Is Lithium-Ion Truly Superior?

Today's battery research is heavily focused on lithium systems, and one could assume that all future applications will be lithium-based. This lithium type is especially apparent when attending battery seminars where battery manufacturers worldwide meet. The emphasis is on lithium-polymer; older chemistries such as NiCd and NiMH are hardly mentioned.

In many aspects, the lithium-ion battery is superior to nickel or lead-based batteries. There is one weak point with the lithium-ion that, for unknown reasons, is seldom mentioned by the battery manufacturer. It is aging. Capacity deterioration is noticeable after one year, whether the battery is in storage or use. Past two years, the battery frequently fails.

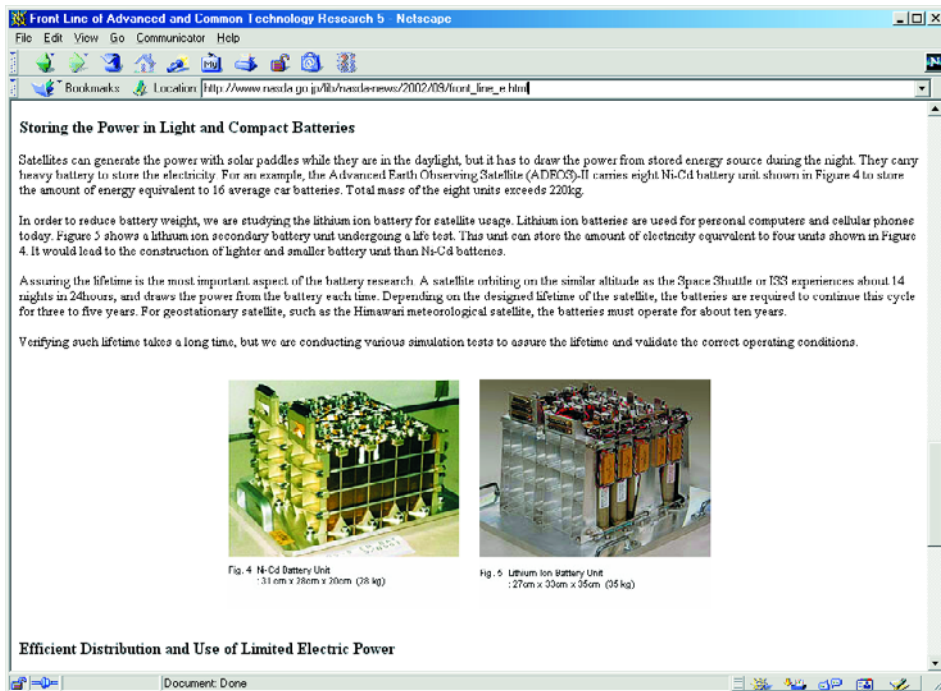
Although less of a concern in the fast-moving cellular phone and notebook market, storage at operational readiness is an important factor in defense applications. In Figure 6, we examine the capacity loss as a function of charge level and storage temperature.

	40% charge level [shipping and storage charge level]	100% charge level [user charge level]
0°C	98% after 1 year	94% after 1 year
+25°C	96% after 1 year	80% after 1 year
+40°C	85% after 1 year	65% after 1 year
+60°C	75% after 1 year	60% after 3 months

Document: Done

Adresses Internet

- [1] Digit-life.com:
www.digit-life.com/articles/mobileaccumulators/
- [2] PowerPulse.Net:
www.powerpulse.net/powerpulse/archive/aa_040201b3.stm
- [3] Cadex: www.cadex.com
- [4] Batteries in a portable world:
www.buchmann.ca/default.asp
- [5] Caractéristiques Li-ion:
www.buchmann.ca/Article9-page3.asp
- [6] LG:
http://battery.rnd.lgchem.co.kr/english/file/LG_Li-ion.pdf
- [7] NASDA:
www.nasda.go.jp/lib/nasda-news/2002/09/front_line_e.html



d'effet sur le déroulement de ce processus. D'après certaines sources, au bout de 2 ans, un accu Li-Ion ne possède plus que la moitié de sa capacité nominale ! L'industrie se console en se disant que, dans notre société de consommation, de toutes façons, nos ordinateurs portables et nos GSM ont de toutes façon une durée de vie limitée à quelques années seulement. Mais ce phénomène joue aussi un rôle majeur lors de l'achat d'un nouvel accu pour un ordinateur ou un GSM. Pour peu que le dit accu ait passé quelques années sur les étagères, il aura inévitablement perdu une partie de sa capacité. Il n'en reste pas moins que l'on vous fera payer le prix du neuf. La plupart des fabricants ne

soufflent mot sur les caractéristiques de leurs accus Li-Ion et n'indiquent pas de date de production (compréhensible) sur la date de production de leurs accus.

On trouve sur Internet différents sites soulevant ce problème tel que l'article « *Choosing a Battery that Will Last* » sur **PowerPulse.Net** [2]. La plupart des articles consacrés au sujet et disponibles sur Internet dérivent des expérimentations et des articles de Isidor Buchmann de la société canadienne **Cadex** [3]. Cette société produit, entre autres appareils, des analyseurs pour accus.

Buchmann lui-même dirige un site qu'il a baptisé « **Batteries in a portable world** [4]. On trouve en référence [5] une description des caractéristiques des accus Li-Ion. À la suite de ses expérimentations, Buchmann tire la conclusion que les cellules Li-Ion vieillissent même si elles restent inutilisées. Au bout de 2 ans, l'évolution du processus est telle que nombre d'accus rendent l'âme.

Les fabricants eux-mêmes ne soufflent pratiquement pas mot à ce sujet. Après quelques recherches nous avons trouvé une fiche de caractéristiques de **LG** [6] consacrée à leurs accus Li-Ion et comportant un graphique qui montre qu'un accu Li-Ion mis sur une étagère ne possède plus, après 6 mois, que 90% de sa capacité d'origine (attention, il ne s'agit pas ici d'auto-décharge !). Cette perte est établie à une température de stockage de 20 °C, la détérioration de capacité se faisant encore plus rapidement à une température plus élevée.

Malheureusement en dépit de toutes nos recherches, nous n'avons trouvé qu'un seul test sérieux évoquant ce phénomène. Les autres sites qui en parlent n'ont pas fait eux-mêmes de test. Nous avons l'impression que le vieillissement dépend de la composition chimique de l'accu et qu'il peut partant être différent d'un fabricant à l'autre.

De plus, la technologie de fabrication des accus ne cesse de s'affiner et il se peut fort bien que les modèles actuels soient moins sensibles à cet effet qu'il y a quelques années.

Il nous paraît intéressant de signaler que les organisations spatiales et les fabricants de véhicules électriques effectuent des recherches quant à l'utilisation d'accus Li-Ion de grande capacité dans leurs produits. S'il est un domaine où la durée de vie est un facteur important, c'est bien celui des satellites. Il faut qu'un accu tienne au moins 10 ans sans que sa qualité ne se détériore. Le site **NASDA** [7] en donne un bel exemple.

Après tout, se pourrait-il que l'accu Li-Ion soit moins mauvais que ne nous le donnent à penser certaines histoires ?

BATTERIES

IN A PORTABLE WORLD

a handbook on rechargeable batteries for non-engineers

Table of Contents

Battery FAQ

New Articles

About the Author

Order Book

Links

The battery has become our steady travel companion - it helps call a friend, it expands our workplace beyond four walls, and it supports critical missions for people in need. Performance specifications for batteries and chargers are often based on ideal conditions.

Manufacturers carry out battery tests on brand new equipment within protected environments. In Batteries in a Portable World, Mr. Buchmann observes the battery as used by consumers in everyday life.

The battery handbook addresses the strengths and limitations of the modern battery. It illustrates which battery chemistry is most appropriate for each application. You will learn about charger technology, battery maintenance to prolong battery life and methods to restore weak batteries with a battery analyzer.

Isidor Buchmann's latest edition of "Batteries in a Portable World" fills the definite need for practical information about rechargeable batteries.

This site is sponsored by Cadex Electronics Inc, the maker of battery analyzers and chargers. Visit the [CADEX site](http://www.cadex.com).

[Table of Contents](#) | [Battery FAQ](#) | [New Articles](#) | [About the Author](#) | [Links](#) | [Site Map](#) | [Contact Us](#)

Access foreign language articles here: [FR](#) [DE](#) [ES](#) [IT](#)