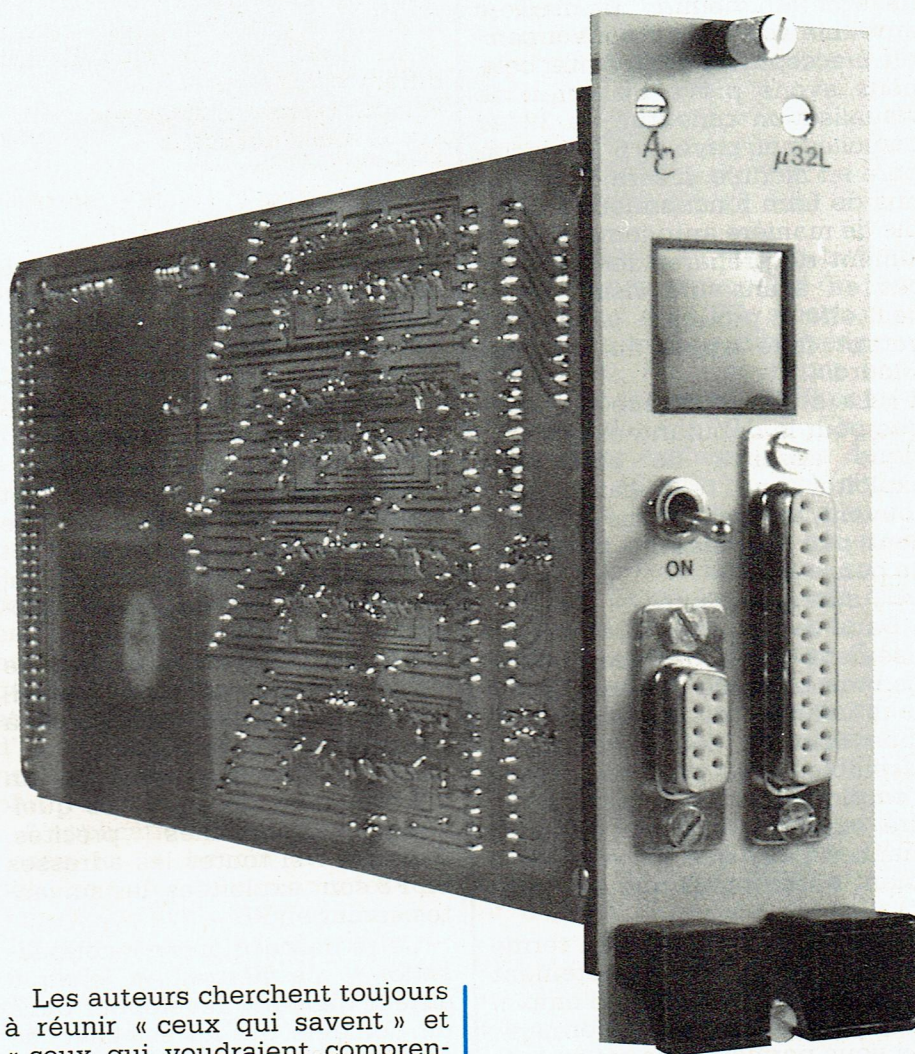


# Interface 32 sorties pour CPC : AC 32L

**S**i les numéros précédents ont permis de mettre en place une structure autorisant à votre ordinateur un dialogue sérieux avec « l'extérieur », la réalisation que nous vous proposons ici entre dans le vif du sujet : 32 sorties pilotables en « tout ou rien ». Quoi de plus commun ? diront les spécialistes. Rien, il est vrai, sinon que nos jeunes amis approcheront en douceur l'usage du port d'extensions de leur CPC et que contrairement aux vieilles habitudes, le « hard » comportera un « écran de contrôle » intégré et sera aimablement assisté de softs très sympathiques et performants permettant au réalisateur le plus « réfractaire à la micro » de commander aisément ses 32 sorties ! C'est à ce prix que parfois se paye l'envie d'aller plus loin.



Les auteurs cherchent toujours à réunir « ceux qui savent » et « ceux qui voudraient comprendre ». La tâche est rude, mais passionnante. Une CLE ouvrant peut-être la porte du succès, demanderait d'allier vulgarisation et nouveauté. C'est ce qu'ils vont tenter de vous proposer ici.

Pour que le débutant se sente à l'aise, il fallait commencer par des « sorties en tout ou rien », c'est-à-dire la mise en route ou l'arrêt d'éléments extérieurs (des lampes par exemple). Ensuite, il fallait qu'il soit possible de « créer quelque chose d'intéressant », car allumer ou éteindre une lampe (même si c'est la pre-

mière fois) reste d'un intérêt limité. Aussi avons nous choisi d'en commander 32, disposées en une matrice de 8 x 4 permettant ainsi de « créer » des lettres, des chiffres, des poursuites, etc.

Pourquoi pas des clips lumineux ? Exemple : un compte à rebours qui, arrivé au zéro, enchaînerait le morceau suivant ou une magnifique explosion (que le pavé lumineux pourrait d'ailleurs illustrer), etc., etc.



Mais si les interfaces de sorties ont souvent été proposées dans toutes les revues spécialisées, il y avait toujours deux lacunes profondes qui excluaient le débutant :

1°) Pour faire « tourner » la carte, il fallait construire impérativement autant de commandes que de récepteurs. Ainsi, être quasiment obligé de préparer par exemple 32 commandes 220 V reliées à autant de lampes avant de voir fonctionner de manière agréable son travail nous a semblé inacceptable. Pour éviter cela, nous avons prévu un écran de visualisation constitué de 32 led disposées en pavé de 8 x 4, incorporé au module. Il sera donc permis de faire fonctionner l'ensemble de manière autonome, et également de préparer des séquences en ayant une vision réelle des effets produits, sans pour autant faire appel au matériel « lourd ».

2°) La seconde source de panique pour le débutant, est la classique absence de soft pour exploiter une telle carte. Très souvent, les réalisations se terminent par un message laconique du genre :

OUT adresse, donnée.

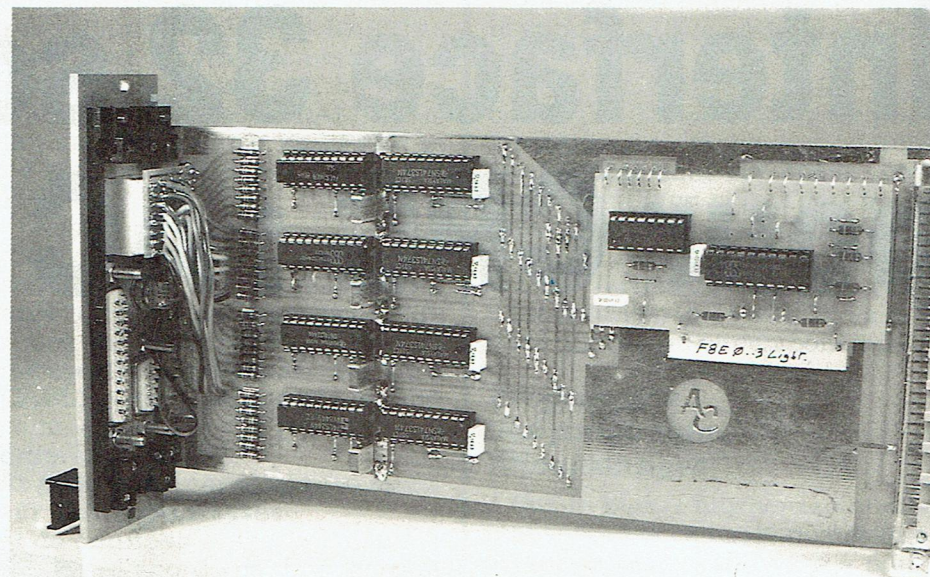
Dans ces conditions, on admet aisément que le débutant baisse les bras, et laisse tomber ! Si l'on nous avait appris à conduire en nous donnant déjà la voiture à assembler puis pour tout mode d'emploi : débraye, vitesse, accélérateur, embraye, il y aurait de fortes chances pour que les routes ne soient encombrées que de PROS (T)...

Donc vous trouverez au terme de la construction proprement dite, un soft très simple d'emploi et particulièrement performant, qui vous permettra de créer, sauvegarder, faire dérouler au pas à pas, etc., des séquences issues de votre imagination que nous savons fertile.

Passons à l'ouvrage.

## Convention

Nous avons vu le mois dernier que les adresses disponibles à l'utilisateur (sur CPC), étaient réparties en quatre groupes commençant respectivement par F8, F9, FA, FB. Notre rack étant ouvert à tous les pôles d'intérêt



(applications lumineuses, mesure, audio, etc.), les auteurs proposent une convention de répartition des adresses :

F8 : APPLICATIONS lumineuses, commandes de trains miniatures pilotage de machines diverses, applications domestiques.

F9 : MESURE (acquisition et traitement de données en vue d'observation et d'étude) et COMMUNICATION (modem, équipements à l'usage des radio-amateurs, etc.)

FA : AUDIO commande de VCA, pilotage de faders motorisés, etc. et toutes applications à usage sonore ou musical.

FB : LIBRE disponible à un débordement dans un des quelconques domaines précités. Exemple : si toutes les adresses en F8 sont exploitées, les suivantes seront en FB.

Cette proposition de « normalisation » permettrait à chaque auteur désirant développer dans un domaine qui lui est cher, de proposer au lecteur une réalisation directement exploitable sans l'obliger à de laborieux réadressages. De plus, un soft adapté à une carte particulière ne risquerait pas d'entrer en conflit avec une autre et le lecteur passionné par tous les domaines à la fois (c'est notre cas) pourra posséder plus de modules que n'en contiendrait son rack, et charger ce dernier au gré de ses envies ou de ses besoins : en moins de dix minutes un rack peut passer de l'AUDIO à la MESURE, pour peu que la convention soit observée.

Bien évidemment nous respecterons scrupuleusement cette règle, et nous sommes certains que nos confrères l'adopteront également pour vous être agréable. Nous avons fait en sorte d'ailleurs de mettre au point un décodage d'adresse(s) particulièrement souple et universel, qui autorisera une standardisation aisée : si un auteur propose une application en conflit avec votre rack, il suffira de rendre les adresses du soft (s'il existe) conformes à notre convention, et de câbler « UCARTE » (Universelle CARTE) en rapport. Mais les auteurs qui publient dans RADIO-PLANS vous diront certainement comment procéder.

ATTENTION : les deux compères qui co-signent ces pages n'ont pas la prétention d'établir des lois universelles, pas plus que de les IMPOSER à qui que ce soit ! La proposition qu'ils font ici est issue d'une constatation simple : pour débiter, le DÉBUTANT à besoin de bases solides et rassurantes afin de se polariser en toute sécurité sur les nouvelles connaissances à « absorber ». Mais quoi de plus facile que lui tirer le tapis sous les pieds, et quoi de plus difficile que de lui bien tenir l'échelle !!

## Synoptique

La figure 1 présente le synoptique de notre réalisation. Avant de l'analyser ensemble, il peut être bon de rappeler que nous avons le mois dernier construit un pré-décodeur-buffer,



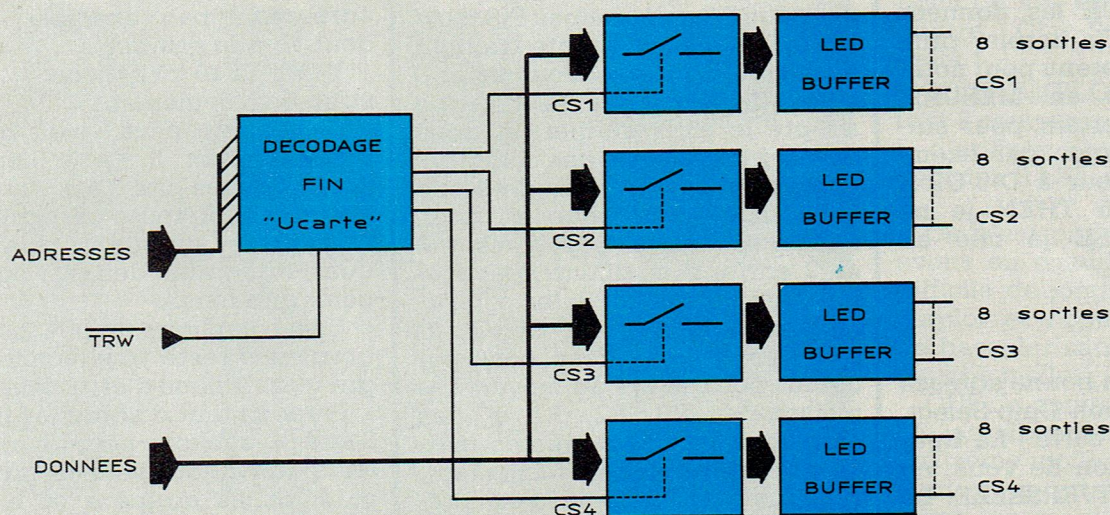


Figure 1

lequel sélectionnait les groupes d'adresses F8... FB, et fabriquait des signaux composites TRW (barre) et TRR (barre), au moyen du prédécodage, de IORQ (barre) et soit de WR (barre) soit de RD (barre). Ces signaux occupent actuellement les lignes CS1 et CS2 de la carte mère qui en offre quatre (CS3 et CS4 sont actuellement libres). Ces lignes « CS », sont porteuses de Composites Signaux, qu'il ne faudra pas confondre avec les « CS » des cartes, qui seront les « Chip Select » ou sélection de « pavé », mais plus précisément encore : adresse sélectionnée parmi le groupe d'adresse décodé par la carte.

Prenons un exemple : l'adresse de RADIO-PLANS est « 2 à 12, rue de BELLEVUE ». En fait, si vous frappez à la porte du « 2 à 12, rue de BELLEVUE », vous avez accès au GROUPE VENTIL-LARD, lequel porte en son sein RADIO-PLANS, mais aussi ELECTRONIQUE-APPLICATIONS, MICRO SYSTEMES, AUDIO TECH, etc. Dans le hall, une hôtesse vous conduira à la rédaction que vous souhaitez contacter dès que vous lui en aurez dit le nom... et si c'est RADIO-PLANS, ce sera par exemple le BUREAU n° 1 (Chip Select 1).

L'ADRESSE + le Chip Select vont donc vous conduire à la rédaction de RADIO-PLANS.

En informatique, il faut indiquer si l'on va PARLER ou ECOUTER. En fait se serait plutôt écrire (WRITE) ou lire (READ). Les signaux composites fabriqués le

mois dernier sont là pour ça, et suivant que l'on utilisera l'un ou l'autre en conjugaison avec l'adresse et éventuellement le Chip Select, on sera en mesure de transmettre une DONNEE ou d'en recevoir.

Si nous disons « éventuellement », c'est que le Chip Select peut être directement le décodage d'une adresse précise : Si RADIO-PLANS est situé dans un complexe important (2 à 12), il est fort possible que Vous, vous habitez au 16, rue des Saules, et qu'il n'y ait que Vous à cette adresse. Dans ce cas, le décodage d'adresse et la sélection de pavé ne font qu'UN. Vous voyez, c'est plutôt simple.

L'application lumineuse proposée ici a besoin de quatre adresses afin de commander les quatre colonnes de 8 lampes, chaque lampe correspondant à un bit 0 ou à 1 de la donnée envoyée, qui en comporte 8. Et  $4 \times 8 = 32$ . CQFD !

En respectant la convention, les quatre adresses commenceront par F8, et comme toutes les adresses sont actuellement libres nous n'avons pas cherché midi à quatorze heures : ce seront F8E0, F8E1, F8E2 et F8E3.

Notre maquette étant à ECRITURE seule, nous combinerons donc la sélection d'adresses avec TRW (barre).

Ah, il est possible que le débutant soit encore troublé par l'organisation globale du système et particulièrement par les Chip-Select. Résumons pour lui le principe retenu :

« DECODER », en plus de bufférer les lignes principales du port d'extension du CPC, sélectionne la zone d'adresses autorisées par AMSTRAD. Nous pourrions imaginer en disant que le BUS nous a introduit dans la VILLE. Chaque carte étant exclusivement destinée à une adresse précise (située bien entendu dans cette ville), on l'équipera de sa propre boussole afin qu'elle l'atteigne sans erreur possible. Cette formule est simple à comprendre mais peu performante, d'où l'idée des Chip-Select. En effet, dans notre cas il serait tout à fait possible de construire quatre cartes adressées respectivement en F8E0 à F8E1, F8E2, F8E3, mais le rack serait vite plein ! Comme ces adresses se suivent, nous adressons UNE carte aux adresses F8E0 à F8E3 et c'est à l'intérieur de celle-ci que se fait la distribution. Le Chip-Select est donc un « reconnaissseur » d'adresse précise parmi un petit groupe déjà trié.

La figure 1 doit désormais être claire : après un décodage fin effectué par UCARTE (complété de TWR (barre)), une distribution des données sera conduite par CS1 à CS4 qui transmettront à bon escient ces dernières vers les buffers de led correspondants, en activant les latches adéquats. Ça ressemble à s'y méprendre au bon vieux principe du multiplexage : pour commander les sorties, on balaye successivement les 4 adresses, CS1 à CS4 seraient alors sélectionnés par une horloge infernale.



Un soft optimisé permettrait de ne réafficher QUE les données qui ont bougé. Ce dernier principe n'a pas été retenu pour notre application mais en MESURE, l'intérêt serait certain pour surveiller une tolérance par exemple : IF X supérieur à D-5 OR X inférieur à D + 5 THEN je ne change rien ELSE je crie au secours...

## Ucarte

**P**our arriver à la bonne adresse et activer le bon Chip-Select, il faut une bonne carte ! La figure 2 a la prétention de vous, en proposer une UNIVERSELLE. Le tableau qui accompagne le schéma rappelle les bits d'adresses qu'il nous faut traiter : ils sont au nombre de 7, et se nomment A0 à A4, et A8 + A9.

Qui dit UNIVERSEL dit complexe... C'est parfois vrai, mais nous n'avons que 7 éléments à trier, on devrait donc s'en tirer sans trop de dommages.

Deux circuits intégrés font le travail, pour peu qu'on leur donne un petit coup de main au bon moment. Voyons tout d'abord IC1.

C'est un fidèle : ce comparateur déjà exploité le mois dernier est d'abord aux lignes 4, 8 et 9

qui à elles seules seront capables de réduire aux domaines F(8/9/A/B) (E/F) x si on présente les bonnes égalités à IC1. Pour cela, il suffit de respecter le principe adopté le mois dernier, et donc de fixer les entrées des comparateurs de telle sorte que la sortie P = Q soit active pour la combinaison souhaitée. Les lignes 4, 8 et 9 entrant sur IC1 respectivement en P5, P4 et P3, les comparaisons se feront avec Q5, Q4 et Q3. Ces trois entrées sont par défaut portées à 1 au moyen des résistances R3, R2, R1 et des emplacements de straps permettent de les passer à 0 facilement. Prenons pour exemple le cas présents : F8Ex. Ceci s'écrit en binaire 1111 1000 1110 xxxx. Si l'on remplace les bits déjà traités dans DECODER par la lettre Y, cela donne YYYY YY00 YY00 xxxx. Signalons à ceux qui seraient perdus, qu'en binaire les 16 lignes d'adresse ne sont pas écrites de 1 à 16, mais de 15 à 0, c'est-à-dire poids FORT à gauche, poids FAIBLE à droite, le zéro étant inclus dans les 16 bits.

Donc on observe que pour décoder F8Ex, il nous faut reconnaître les bits 4, 8 et 9 à 0, ce qui sera obtenu en portant les entrées de comparaisons de IC1 également à 0. De ce fait, il faudra implanter les straps g, h et i.

Amusez-vous à traiter d'autres adresses : par exemple FAFx dont voici la solution :

YYYY YY10 YYY1 xxxx, d'où strap en h seulement.

Il ne reste plus à voir que la série de xxxx. Il s'agit bien évidemment des bits 0 à 3, auxquels une attention toute particulière a été portée afin de satisfaire la quasi intégralité des cas, pour ne point dire tous...

Ceci complique un peu la situation, mais reste très simple malgré tout. A nous d'être clairs !

Prenons tout d'abord les lignes 3 et 4, et oublions pour un instant leurs dérivations vers les straps c et d. Si les straps a et b sont implantés horizontalement, les lignes A3 et A4 arrivent gentiment sur P1 et P2 afin de se soumettre au comparateurs de IC1 comme pour les lignes précédentes. Suivant qu'elles devront être à 0 ou à 1 pour que l'adresse soit reconnue, il faudra correctement positionner les entrées des comparateurs correspondants soit Q1 et Q2, portées à 1 par défaut ou à 0 par les straps e et f.

Qu'en est-il dans ce cas des lignes A0 et A1 ? Comme les lignes A2 et A3 sont envoyées vers les comparateurs, les straps c et d sont « horizontaux », donc A2 et A3 filent vers les entrées A et B de IC2. Ce dernier est un décodeur de 3 entrées vers 8 sorties (74 HC138). Pour ceux qui maîtrisent mieux la vieille série TTL et qui se rappellent des décodeurs BCD-décimaux, disons que c'est un ABC-8... En fait ce circuit activera une sortie parmi 8, en fonction du code binaire de 3 bits présenté aux entrées ABC.

Nous lui avons confié la charge de produire les CHIP-SELECT, c'est-à-dire de décoder tout seul les bits de poids faible. Ne lui fournissant que 2 bits (A0, A1), il ne produira que 4 Chip-Select, ce qui suffira dans la majorité des cas.

MAIS... revenons à A2 et A3, et imaginons que les straps c et d soient montés « verticalement ». Dans ce cas, il faut observer plusieurs règles :

- 1) Si A2 et A3 partent vers IC2 afin d'être porteurs des informations de Chip-Select, elles ne sont plus reliées à IC1.
- 2) Par voie de conséquence, ce dernier doit être inhibé, c'est-à-dire que les entrées P1 et P2

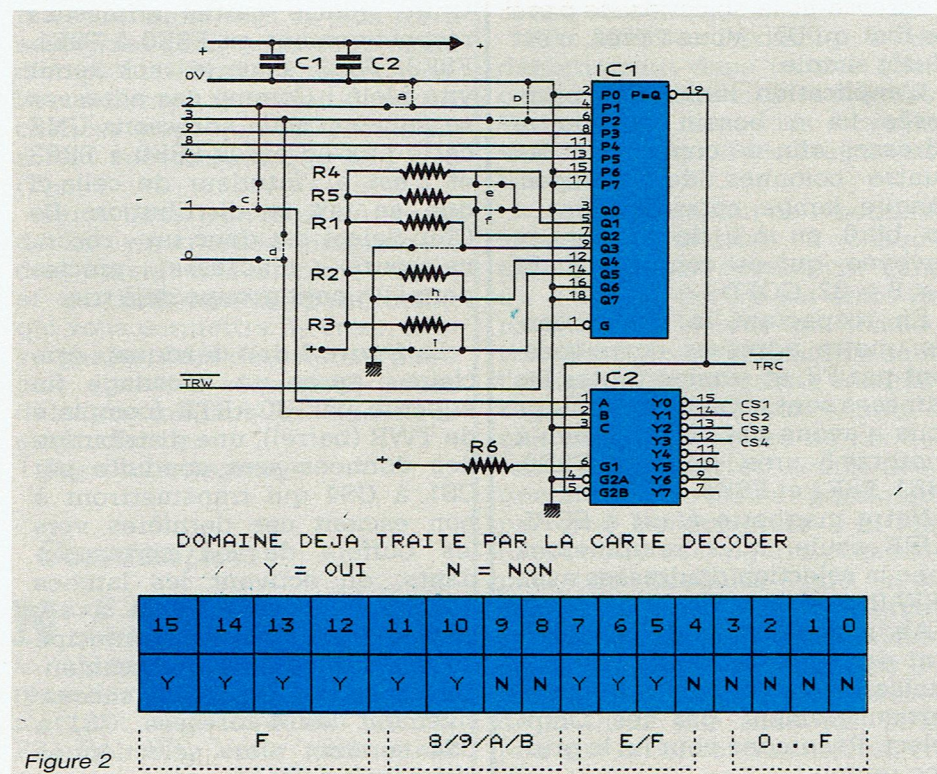
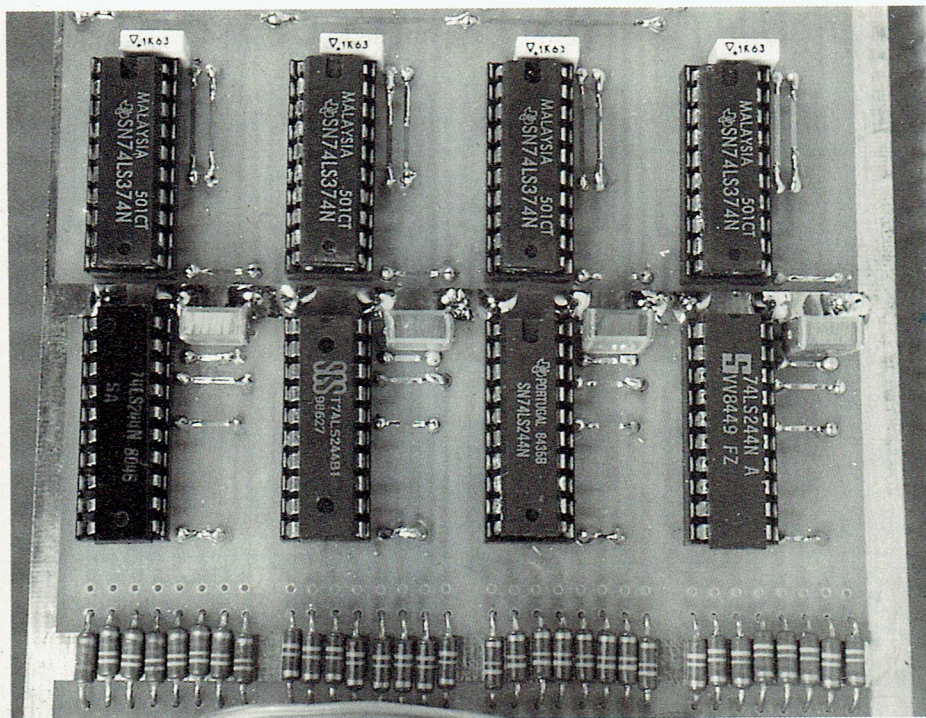


Figure 2





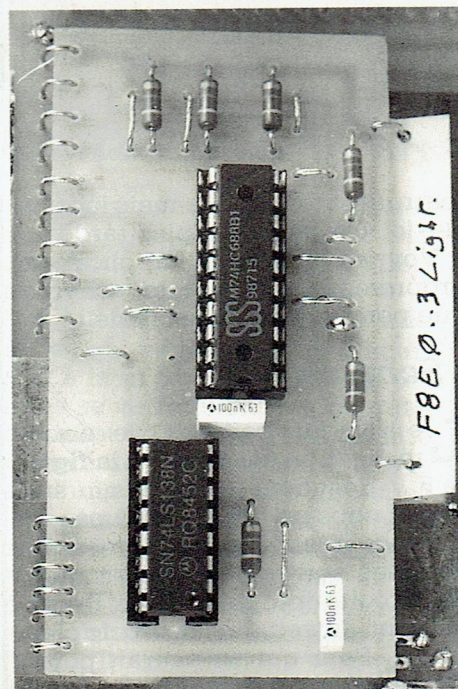
seront portées au même potentiel en Q1 et Q2, afin de ne pas perturber le travail fait par IC1 pour A4, A8 et A9. Pour des raisons pratiques et de sécurité, c'est le 0 qui a été choisi pour l'inhibitions : si les straps a et b lient P1 et P2 à la masse, il faudra mettre les straps e et f.

3) Puisque A2 et A3 vont vers IC1, cela voudra dire que A0 et A1 seront libres, ou tout du moins exploitées par ailleurs, indépendamment du décodage de carte (par exemple pour un ou des 8255 qui en traiteront eux mêmes les états afin d'en déduire les conséquences qui s'imposent...).

Cette situation étant un peu délicate pour le débutant, nous attendrons de rencontrer un cas précis y faisant appel pour la détailler. Les connaisseurs devraient toutefois constater l'universalité annoncée. Ceux qui cherchent et voudraient confronter leurs résultats avec le manuel du prof... trouveront un grand intérêt au programme UCARTE. Il leur suffira de rentrer l'adresse ou le groupe d'adresses sur leur CPC, afin de connaître le positionnement des straps de la carte UCARTE. On croit rêver !! Mais ne jouons pas de suite la facilité... et résolvons le problème posé, sans l'aide d'un soft.

Nous voulons traiter xxxx de sorte que 0000 à 0011 (F8E0 à F8E3) soient les seules adresses

reconnues, et créer quatre Chip-Select à l'intérieur de cette zone. Sur les quatre bits traités, le fait de tolérer de 0000 à 0011 montre que A2 et A3 restent à 0. Donc en reliant ces lignes à P1 et P2 au moyen des straps a et b et en forçant les comparateurs associés à 0 grâce à la mise en place de straps e et f, le problème est résolu. Les deux bits de poids fort de ce quartet de poids faible sont reconnus. Restent les deux bits A1 et A0 que les straps c et d relient à IC2 qui est prêt à tra-



duire en « décimal » les informations de ses entrées A et B. Les quatre Chip-Select sont produits !

**SOUS EMPLOIS :** IC1, comportant 8 comparateurs n'en voit que 5 d'utilisés et IC2, capable de décoder 3 bits n'en analyse que 2.

Ce n'est pas grave, mais nous avons eu chaud... Il est en effet difficile de jongler entre le sous-emploi et l'opulence, au risque de basculer sans transition de la gauche molle à la droite dure !

OUF, le compromis est adoptable aimablement par tous, sauf l'extrême molle et l'extrême dure dont nous attendons impatiemment les maquettes sur-optimisées...

## UCARTE : réalisation

La figure 3 présente le circuit imprimé et l'implantation de UCARTE. La figure 4 apporte quelques précisions relatives au positionnement des straps, dont il faudra distinguer les fixes (imposés par le hard) et les autres, préposés variables. Sur la partie gauche on peut voir l'organisation physique des straps et à droite le câblage réel dans notre cas particulier.

Nous n'insisterons pas sur le soin tout particulier apporté à la réalisation de ce circuit, mais vous noterez toutefois que les combinaisons dangereuses ou impossibles sont exclues par le fait que, par exemple, si le strap c est horizontal, il n'est pas permis de positionner a vertical, à moins d'y mettre une mauvaise volonté évidente contre laquelle nous ne pouvons rien ! Idem pour le couple b, d. Quand nous disons « combinaisons dangereuses », c'est par exemple une mise à la masse d'une ligne d'adresse, et pour cela il faudrait placer des straps en biais, ce qui ne se fait pas.

Comme le montrent les photographies, UCARTE est placée parallèlement à la carte principale, et liée à celle-ci par des pattes de résistances. Deux pattes supplémentaires assurent la rigidité, mais il est indispensable de garantir une isolation parfaite avec la carte principale (en double face), au moyen d'une feuille de bristol sur laquelle vous pour-



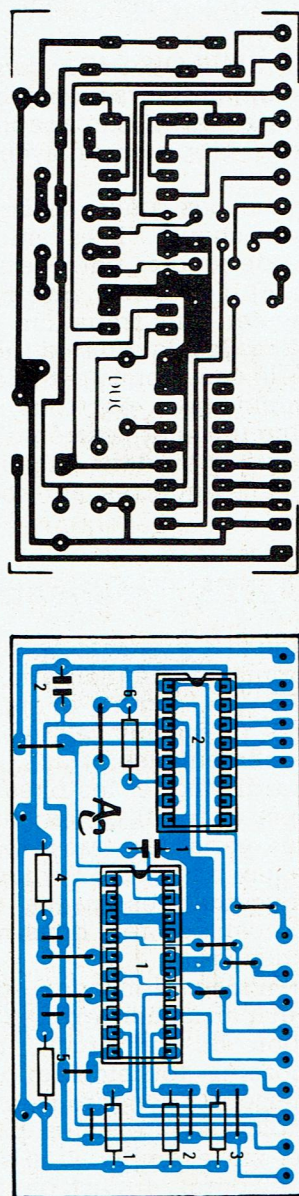


Figure 3

rez également porter la zone d'adresses décodées par la carte.

Cette disposition un peu particulière peut surprendre à première vue, mais le fait de rendre autonome la carte de décodage peut rendre de grands services pour vos conceptions personnelles. D'autre part, nous n'aurons à donner que l'implantation des straps pour les futures cartes d'applications, et nos confrères feront certainement de même, sans qu'ils soient obligés de s'imposer des adresses en conflit peut-être avec leurs équipements précédents.

Dernier point, la ligne TRC

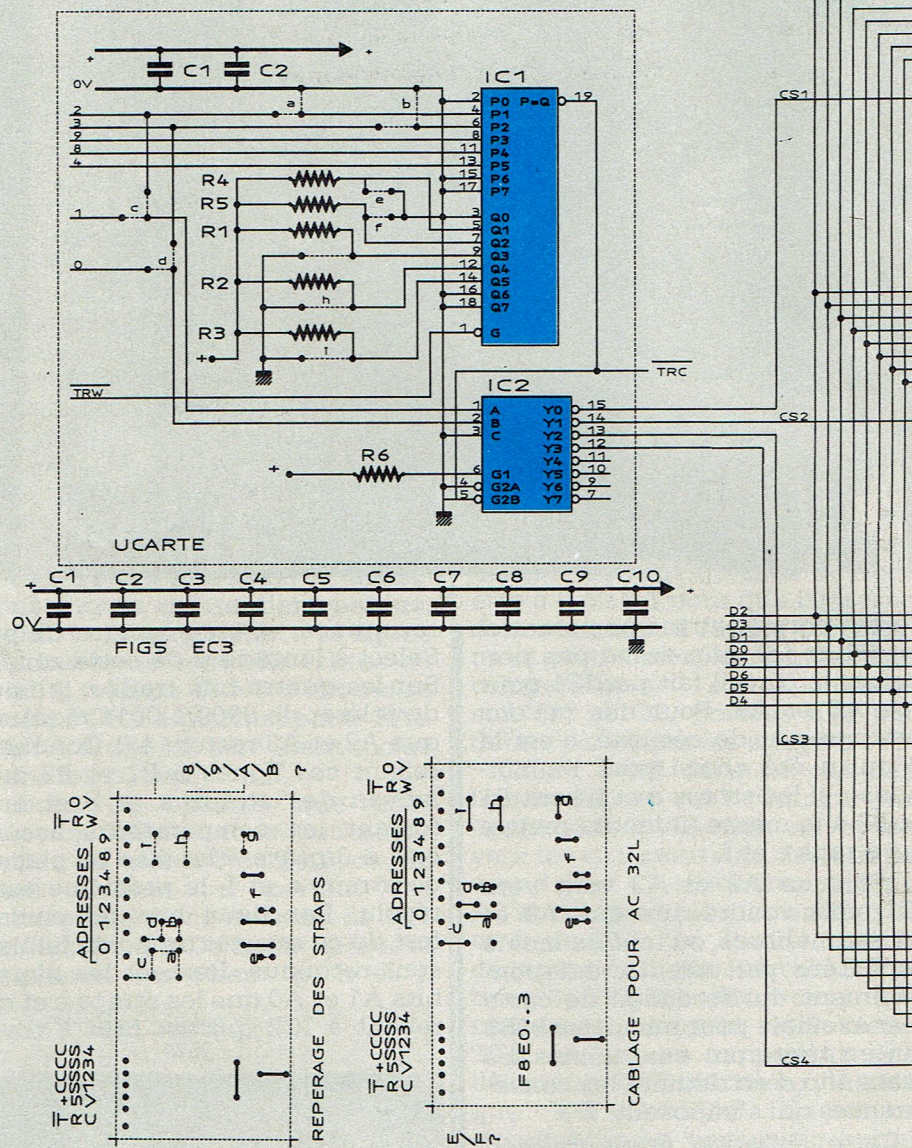


Figure 6

(barre) est disponible mais inutilisée ici. Ce sera pour plus tard !

Voyons maintenant l'application proprement dite, et le schéma que porte la carte principale.

## Schéma 32L

Le schéma complet de notre réalisation est donné à la figure 5, décodeur compris, sur lequel nous ne reviendrons pas.

On constate que les lignes de données sont distribuées à quatre circuits intégrés. IC3 à 6, qui ne sont autres que des octuples latches. Pour qui ne saurait point ce qu'est un latch, disons qu'il

s'agit d'une mémoire obéissant aux règles suivantes :

1 — elle conserve en sortie l'état présent à l'entrée, tant que son entrée de transfert n'est pas activée.

2 — son entrée peut donc prendre tous les états possibles, ne seront pris en compte que ceux présents simultanément avec une activation de l'entrée CLK.

On comprend donc aisément que ce genre d'outil est particulièrement adapté au multiplexage puisque les données doivent être distribuées en conformité avec le positionnement des récepteurs.



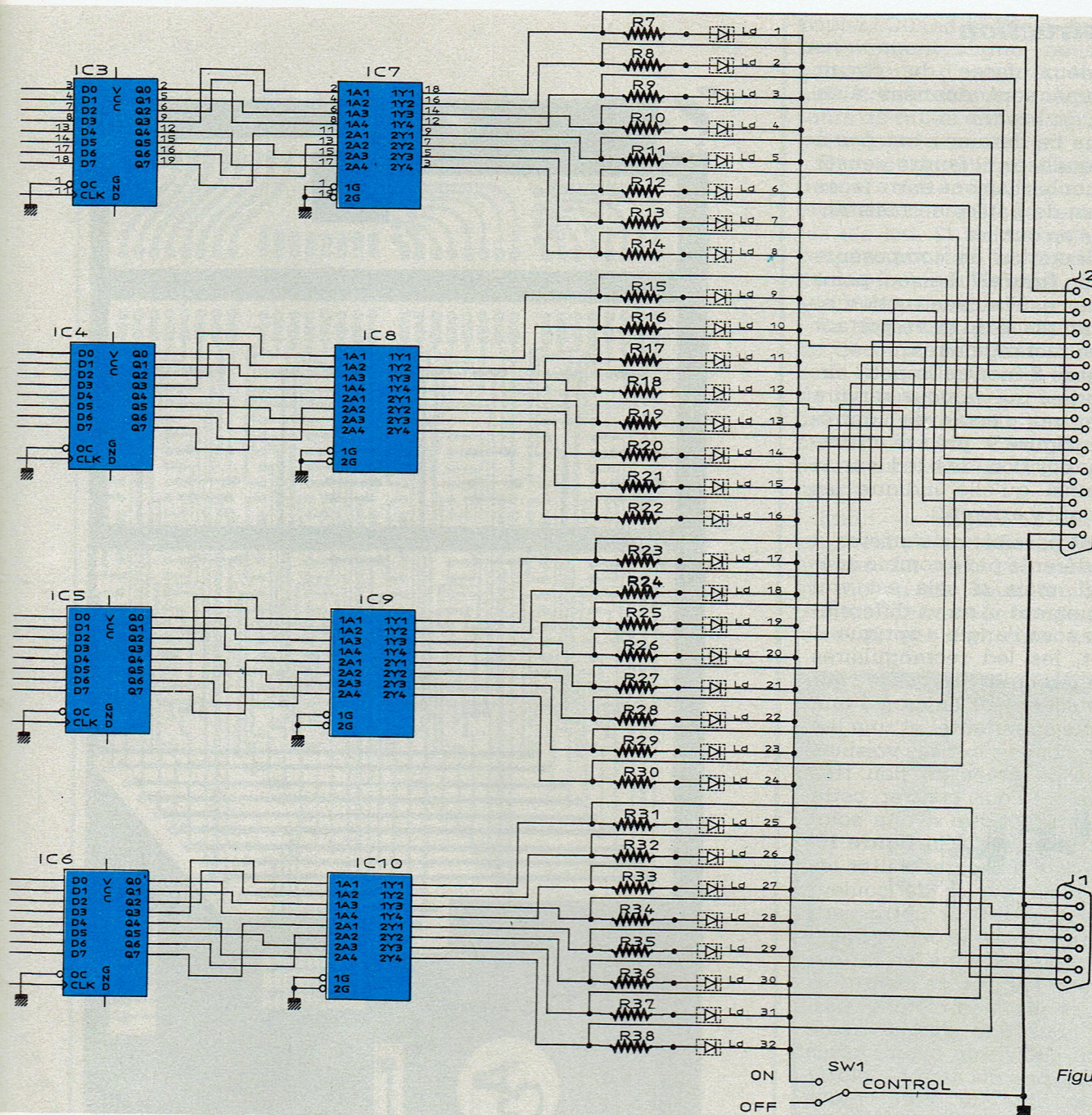


Figure 5

Pour commander les entrées CLK, nos CS 1 à 4 sont parfaits.

Ainsi une donnée prévue pour une des quatre adresses F8E0 à F8E3, sera transférée sur les sorties Q0 à Q7 de l'IC sélectionné.

Le problème est résolu. Les quatre circuits suivants (IC7 à IC10) ne sont que des buffers non inverseurs destinés à servir de drivers et de tampons pour les led et les sorties sur sub-d.

SW1 permettra de couper le pavé de led, gagant ainsi 150 mA quand les récepteurs seront en place et que le soft aura été testé.

Il faut noter encore que les buf-

fers non-inverseurs (74LS244) pourront être remplacés par des inverseurs (74LS240). En effet, dans l'état actuel, la mise à 1 d'un bit de donnée allume une led, ce qui semble logique. Pourtant, à la mise en route, tous les latches se verrouillent à 1, ce qui a pour effet de mettre à 1 la totalité du pavé de visualisation. Dans le cas d'une application lumineuse c'est plutôt un bien, car on peut constater le bon fonctionnement de la totalité des lampes avant de faire tourner des softs, mais dans d'autres cas cela peut être gênant. Deux solutions sont possibles :

1 — comme la séquence d'allumage est la suivante : rack, CPC, commandes de puissance, il est possible de lancer le soft qui effectuera une initialisation à 0 du pavé, avant de mettre en route les commandes de puissance.

2 — monter des buffers inverseurs, et demander au soft de transcrire les données et d'inverser tous les bits.

Vous ferez comme bon vous semblera, car les programmes d'exploitation du mois prochain vous donnerons la possibilité de transcrire les données dans vos délire artistiques.



## Construction

**L**es deux faces du circuit imprimé sont données à la figure 6.

Comme ce dernier n'est pas à trous métallisés, il faudra penser à effectuer les liaisons entre faces au moyen de pattes de résistances, sans en oublier...

L'implantation de composants est visible figure 7. Le seul point particulier est la construction et la mise en place du pavé portant les 32 led rectangulaires.

La figure 8 montre le petit circuit imprimé qui va nous servir à fabriquer cet « écran de contrôle ». La figure 9 précise clairement la position des led sur la carte, ainsi qu'elle indique les signaux qui y arrivent.

Il serait possible de s'arrêter là et considérer le pavé comme opérationnel, mais si cela est vrai électriquement, il en va différemment vu sous l'angle « optique ». En effet, les led rectangulaires sont légèrement dépolies sur tous les côtés (sur environ 1 mm à la partie supérieure) et une led allumée « bave » sur ses voisines au point de créer un flou très disgracieux. Pour contrer cette « capillarité optique », une solution est proposée à la figure 10. Elle consiste à séparer toutes les LED par du papier de couleur noire ou bleue. Pour notre part, nous avons utilisé le papier d'emballage qui enveloppe les cartouches d'une marque de cigarettes que nous ne nommerons pas. Autant vous dire quand même qu'il faut choisir un papier assez fort (mais pas du carton), plutôt glacé et que le bleu foncé convient parfaitement. Il faudra découper des bandes de 5 mm de large environ, et une de 20 cm de long si possible. La figure illustre le procédé qui consiste à passer dans un premier temps une longue bande entre chaque rangée de led. Ceci fait, il reste à ouvrir les trois passages transversaux au moyen d'un cutter, et d'engager trois petites bandes de

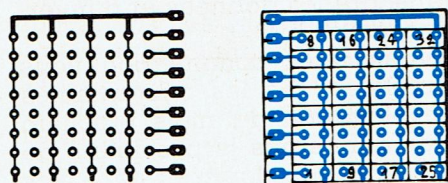
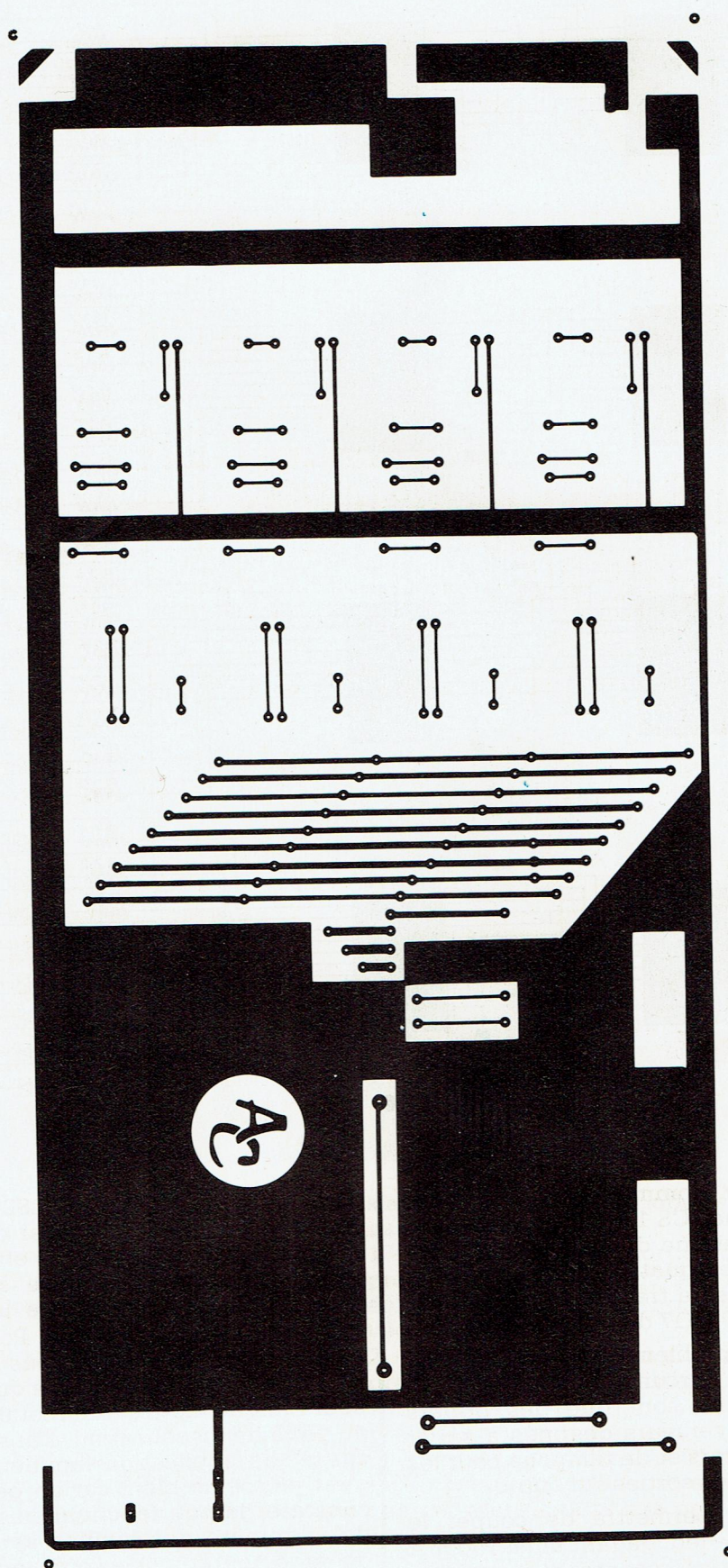


Figure 8





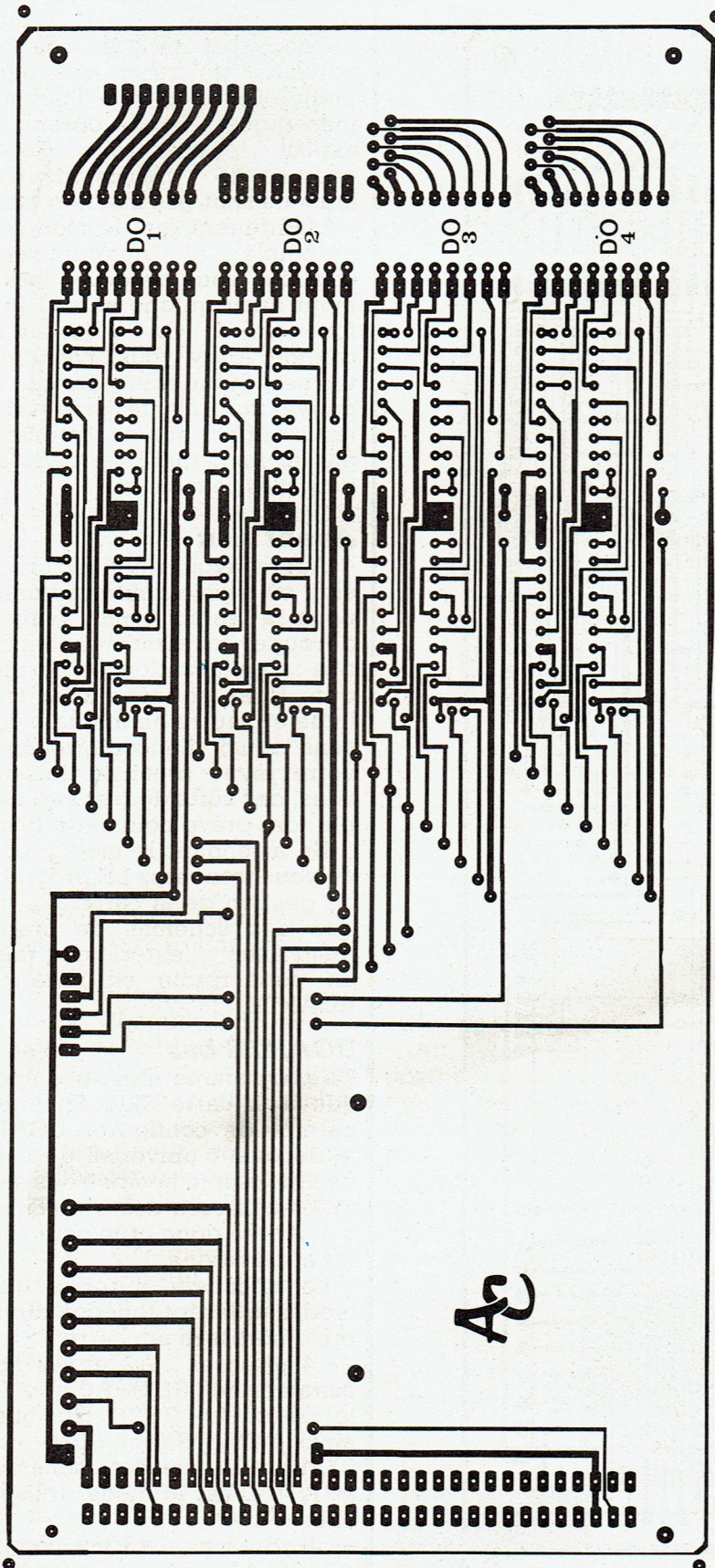


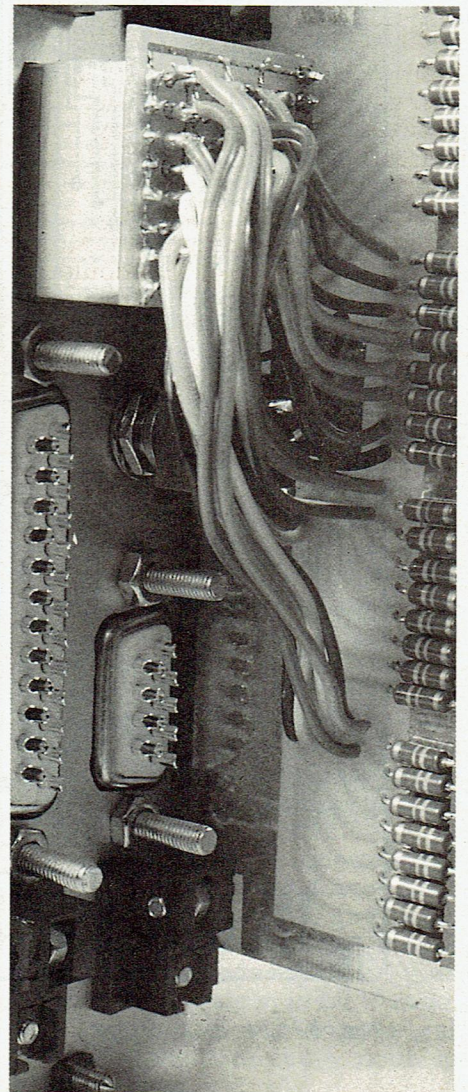
Figure 6

papier. Quand tout est en place, parfaitement aligné, nous vous conseillons d'entourer le pavé ainsi constitué avec une paire de tours de ruban adhésif souple, en le tendant fortement.

Bien évidemment, avant d'effectuer cette opération, il est vivement conseillé de tester toutes les led. Si les cathodes sont coupées courtes, les anodes garderont environ 1,5 mm, afin de servir de relais aux fils.

La **figure 11** propose un plan de découpe échelle 1, que nous vous conseillons de respecter scrupuleusement, car les marges de manœuvre sont particulièrement réduites. La **figure 12** donne une idée de la gravure de façade.

Enfin, la **figure 13** explique comment se fabrique un filtre bon marché. Le capuchon est réalisé au moyen de deux feuilles de gélatine de couleur (rouge ou violette si vos led sont rouges),





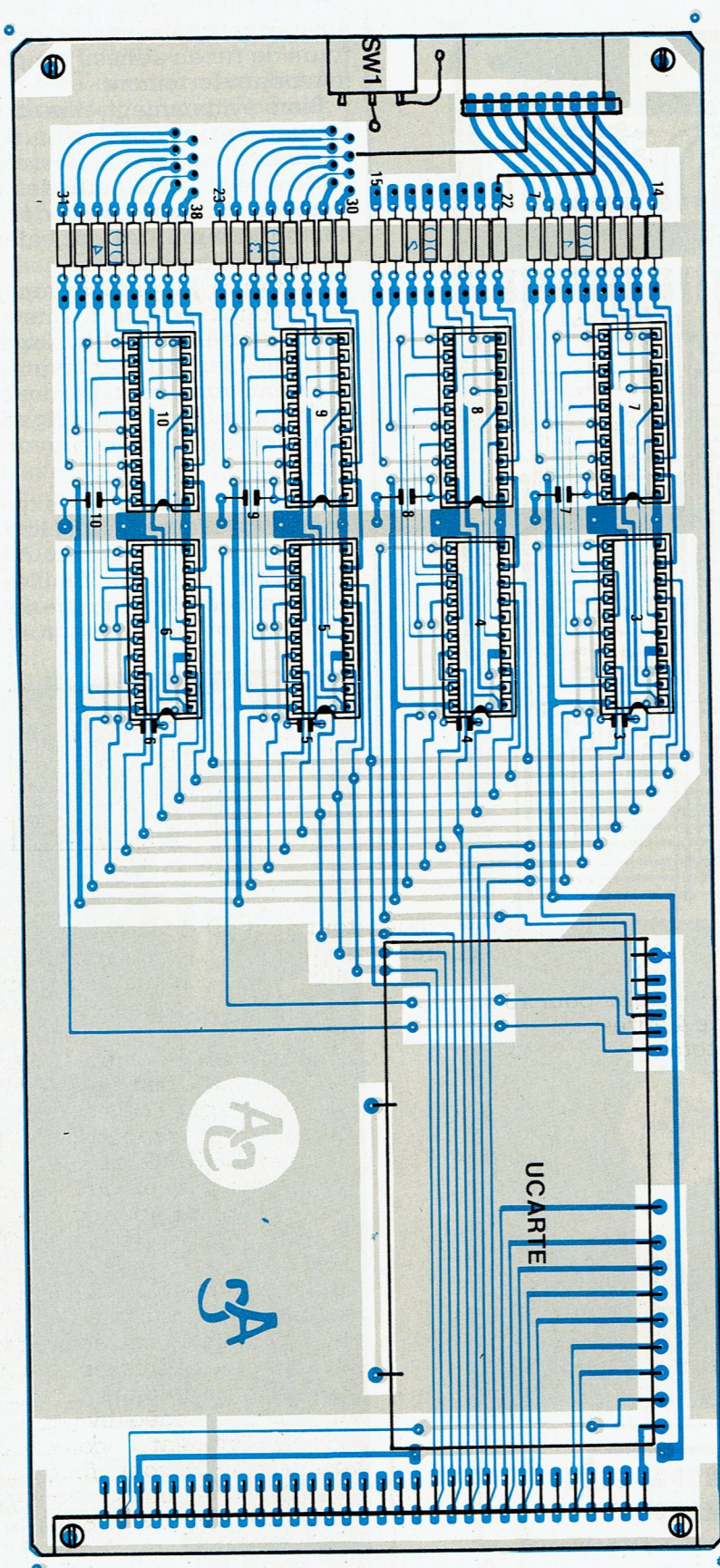


Figure 7

pliées en « U » et imbriquées. Une fois emboîté sur le pavé, ce capuchon est maintenu par un tour de ruban adhésif. Il suffira de découper dans la face avant adhésive un carré très légèrement plus petit que l'ouverture mécanique pour obtenir un aspect plutôt joli. Certains cadrans de « vieilles montres à led » ne sont pas mal non plus...

Maintenant que le module est assemblé et minutieusement contrôlé, vous ne devez logiquement avoir qu'une envie : le voir fonctionner. On y arrive ! Si comme Alary vous avez bricolé un petit simulateur passif, vous pouvez contrôler la carte et titiller les switches, sinon, il vous faut passer immédiatement aux softs proposés ce mois-ci.

## Avant tout

Après avoir réuni la commission en session extraordinaire, il nous a semblé plus sage de découper cet article en deux parties : la réalisation et l'exploitation. Nous pouvons ainsi traiter l'une et l'autre de manière égale, sans souci d'encombrement de votre revue préférée. Rassurez-vous, des softs de test et d'exemple sont prévus pour cette fois-ci.

En attendant le mois prochain où vous trouverez LE programme de gestion de la carte 32L, ainsi que des schémas de branchement avec « l'extérieur », faisons un petit retour en arrière sur UCARTE.

## UCARTE bas

Ce programme n'est pas spécifique à la carte 32L. En effet, il permet de configurer UCARTE, le décodeur universel de cartes, en affichant la position des straps et les adresses des Chip Select 1 à 4. Il sera donc utile pour toutes les applications.

Le décodeur autorise quatre modes, qui sont fonction du nombre d'adresses par carte :

- 4 adresses : A2 et A3 sur le comparateur (IC1), A0 et A1 sur le décodeur (IC2). On obtient ainsi 4 Chip Select qui sont 00, 01, 10 et 11 (combinaisons de A0 et A1). C'est le mode utilisé par 32L.

- 8 adresses : A3 est comparé, A2 et A1 composent les Chip Select, A0 est libre. Toujours 4 CS mais comme on ne tient pas compte du bit de poids faible, il y



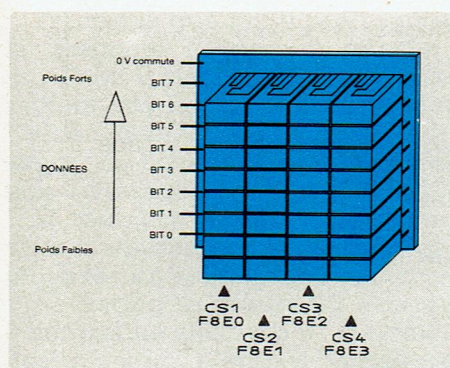


Figure 9

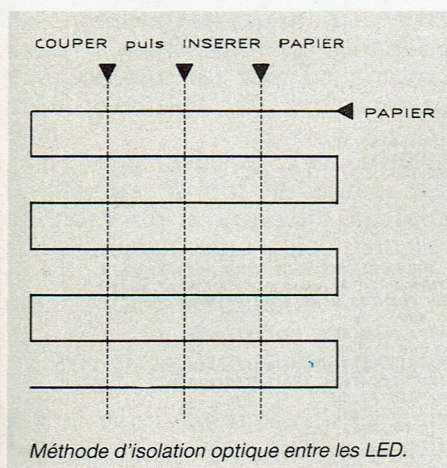


Figure 10

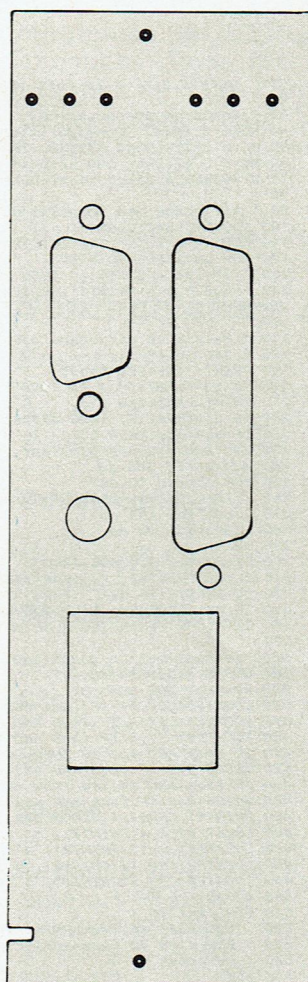


Figure 11

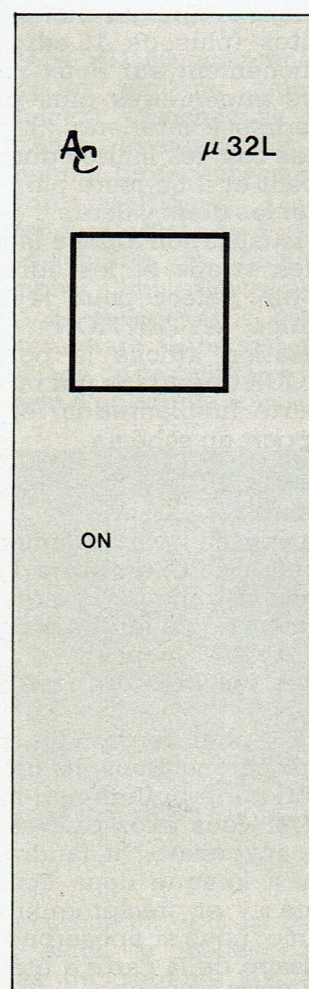


Figure 12

a maintenant deux adresses par Chip Select (paire et impaire).

— 16 adresses : A2 et A3 sont reliés au décodeur, A0 et A1 restent libres. Chaque Chip Select est alors validé pour quatre adresses consécutives puisque les bit 0 et 1 peuvent prendre n'importe quelle valeur (00, 01 et 11). Idéal pour commander quatre 8255 par exemple (ce circuit gère lui-même les lignes 0 et 1).

— Le cas où A3 est décodé et A2 comparé est interdit par le soft car il produirait un morcellement de l'adressage et un gaspillage des possibilités.

Cela nous amène donc à trois configurations (4, 8 ou 16 adresses par carte), à choisir dans 8 zones : F8Ex, F8Fx, F9Ex, F9Fx, FAEx, FAFx, FBEx, FBFx. En précisant pour la première de ces zones, on arrive à :

— F8E0 à F8E3, F8E4 à F8E7, F8E8 à F8EB ou F8EC à F8EF pour 4 adresses (4 CS de 1 adresse).

— F8E0 à F8E7 ou F8E8 à F8EF pour 8 adresses (4 CS de 2 adresses).

— F8E0 à F8EF pour 16 adresses (4 CS de 4 adresses).

Au total, nous obtenons 504 combinaisons de positionnement des straps ! (8 zones × 7 configurations par zone × 9 straps), et pour chacune d'elles, il faut calculer la ou les adresse(s) des quatre Chip Select. Vous comprenez sans doute maintenant l'utilité d'un programme qui fait le travail

tout seul : il nous est nécessaire en tant qu'auteurs, et il vous sera d'un grand secours pour configurer UCARTE comme vous le désirez.

La **figure 14** nous montre UCARTE BAS. Les lignes en italique-gras sont celles qui ont changé par rapport au programme CONFIG du mois dernier : vous pouvez donc charger celui-ci et ne retaper que les parties concernées (si le RENUM n'est pas passé par là !).

L'utilisateur de UCARTE est très proche de celle de CONFIG : on lui fournit en entrée la zone à décoder et les adresses effectivement comprises dans cette zone s'affichent en vidéo inverse.

« Effectivement comprises » signifie que la zone finale peut déborder des limites que vous aviez fixé : les bits 4, 8 et 9 sont toujours comparés à 1 ou à 0 (pas d'inhibition) mais si le bit 2 varie, par exemple, il est décodé par IC2 et donc « inhibé » pour le comparateur.

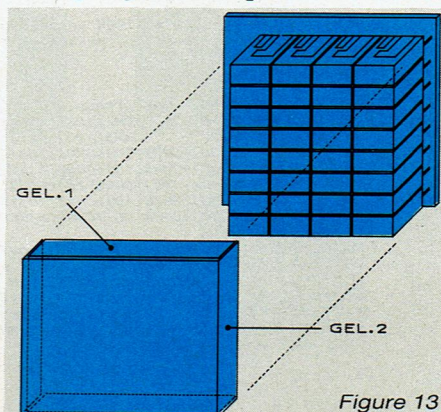


Figure 13



Certaines configurations sont interdites (plus de 16 adresses, chevauchement sur deux zones, adresse supérieures plus petite que l'adresse inférieure...), dans tous ces cas, les limites données s'effacent et il ne reste plus qu'à réentrer les deux valeurs.

La visualisation affiche la position des straps et les adresses des Chip Select pour la zone demandée. ATTENTION : Cette visualisation affiche la position PHYSIQUE des straps par rapport à la carte (implantation) et non par rapport au schéma.

## Les tests

### Important :

Commencez par allumer le rack, puis le CPC (cet ordre d'allumage devra toujours être respecté, de même qu'il faudra éteindre le CPC AVANT le rack).

Toutes les LED du pavé sont allumées.

NOTE : pour ce qui suit, nous nous placerons dans le cas de l'utilisation de buffers non-inverseurs : si vous avez préféré des buffers inverseurs, il faudra lire « éteint » lorsque nous écrirons « allumé », et réciproquement. De même, nous supposons que l'adressage de la carte a été respecté : F8E0 à F9E3.

Tapez l'instruction suivante : OUT &F8E0, 0. La première colonne à gauche s'éteint. Pour éteindre les colonnes suivantes, il suffit de remplacer successivement &F8E0 par &F8E1, &F8E2 et &F9E3.

Vous pouvez aussi remplacer la donnée (0 dans notre exemple) par une autre valeur, comprise entre 0 et 255.

Pour vous assurer que la valeur envoyée correspond bien à l'affichage, tapez le court listing de la figure 15, et vérifiez que vous obtenez un résultat analogue à celui qui est représenté sur la même figure.

Petite explication. La ligne 20 provoque une « raz » générale du pavé : on adresse successivement F8E0 (ad + 0), F8E1 (ad + 1), F8E2 (ad + 2) et F8E3 (ad + 3), et on y écrit la valeur 0. Le « CALL &BB18 » attend l'appui sur une touche quelconque avant de continuer. En 30, on envoie des données formant un damier et en 40, on dessine le damier inverse. On boucle ensuite sur le premier.

```

10 ' UCARTE.BAS * AC Soft 1989 *
20 '
30 ' Ecran de presentation
40 MODE 2:PRINT "Configuration carte";TAB(69);"AC Soft 1989"
50 MOVE 0,380:DRAW 639,380:DRAW 639,60:DRAW 0,60:DRAW 0,380
60 MOVE 0,62:DRAW 639,62:DRAW 639,0:DRAW 0,0:DRAW 0,62
70 WINDOW#0,2,22,23,24:WINDOW#1,4,79,3,21:WINDOW#2,50,79,23,24
80 '
90 ' Affichage des adresses autorisees
100 CLS#0:CLS#1:CLS#2
110 FOR i%=0 TO 3:LOCATE#1,i%*20+5,1:PRINT#1,"Zone"+STR$(i%+1):NEXT
120 FOR ad=&F8E0 TO &FBFF
130 IF (ad AND 255)=0 THEN ad=ad+E0
140 GOSUB 610:PRINT#1,HEX$(ad,4)
150 NEXT
160 '
170 ' Saisie et affichage des limites
180 CLS#0:CLS#2
190 PRINT "Adresse debut : F":PRINT#2,"ou RETURN pour quitter":GOSUB 660
200 IF q$=CHR$(13) THEN 1000 ELSE ad=VAL("&F"+ad$):PRINT:CLS#2:PRINT#2
210 PRINT "Adresse fin : F":PRINT#2,"ou RETURN si meme adresse":GOSUB 660
220 IF q$=CHR$(13) THEN adf=ad:PRINT ad$ ELSE adf=VAL("&F"+ad$)
230 IF adf<ad THEN GOTO 170
235 IF (adf AND &FFF0)<(&ad AND &FFF0) THEN 170
240 adf=&FFF0:ad$=ad$
250 FOR ad=adf TO adf
260 IF (ad AND 255)=0 THEN ad=ad+E0
270 adf=ad AND ad
280 ad$=ad$ OR ad
290 NEXT
300 c$=CHR$(24):GOSUB 780
310 CLS#2:PRINT#2,"(V)ualisation","(A)utre selection"
320 q$=INKEY$:IF q$="" THEN 320 ELSE q$=UPPER$(q$)
330 IF q$="A" THEN c$="":GOSUB 780:GOTO 170
340 IF q$="V" THEN 360 ELSE GOTO 320
350 '
360 ' Visualisation graphique
365 GOSUB 850:CLS#1:CLS#2
370 adx=adi XOR ads
380 LOCATE#1,28,14:IF (ads AND &200)=&200 THEN PRINT#1,pho$ ELSE PRINT#1,phf$
390 LOCATE#1,27,8:IF (ads AND &100)=&100 THEN PRINT#1,gho$ ELSE PRINT#1,ghf$
400 LOCATE#1,27,4:IF (ads AND &10)=&10 THEN PRINT#1,gho$ ELSE PRINT#1,ghf$
410 IF (adx AND &8)=&8 THEN GOSUB 980:GOTO 430
420 GOSUB 970:IF (ads AND &8)=&8 THEN PRINT#1,pvo$ ELSE PRINT#1,pvf$
430 IF (adx AND &4)=&4 THEN GOSUB 950:GOTO 450
440 GOSUB 940:IF (ads AND &4)=&4 THEN PRINT#1,pvo$ ELSE PRINT#1,pvf$
450 PRINT#1,CHR$(24):LOCATE#1,9,7:PRINT#1,"a":LOCATE#1,18,7:PRINT#1,"b"
460 LOCATE#1,2,4:PRINT#1,"c":LOCATE#1,12,5:PRINT#1,"d"
470 LOCATE#1,4,17:PRINT#1,"e":LOCATE#1,15,17:PRINT#1,"f"
480 LOCATE#1,26,14:PRINT#1,"g":LOCATE#1,25,8:PRINT#1,"h"
490 LOCATE#1,25,4:PRINT#1,"i":PRINT#1,CHR$(24)
500 ad=adi
510 FOR i%=1 TO 4
520 LOCATE#1,44,i%*4:PRINT#1,"CS"+RIGHT$(STR$(i%),1)+"=" ;
530 FOR cs=0 TO (ads-adi)\4:PRINT#1,HEX$(ad+cs);" ";:NEXT
540 ad=ad+cs
550 NEXT
560 LOCATE 17,1:PRINT HEX$(adi,4):LOCATE 17,2:PRINT HEX$(ads,4)
570 PRINT#2,"(A)utre selection","(Q)uitter"

580 q$=INKEY$:IF q$="" THEN 580 ELSE q$=UPPER$(q$)
590 IF q$="A" THEN 90 ELSE IF q$="Q" THEN 1000 ELSE GOTO 580
600 '
610 ' S/P de Calcul des coordonnees de l'adresse
620 x%=((ad AND 784)\256)*20+2+(((ad AND 784) MOD 256)/2)
630 y%=(ad AND 15)+3
640 LOCATE#1,x%,y%:RETURN
650 '
660 ' S/P de Saisie de l'adresse
670 q$=INKEY$:IF q$="" THEN 670 ELSE q$=UPPER$(q$):IF q$=CHR$(13) THEN RETURN
680 IF (q$)="8" AND q$<="B" THEN PRINT q$;ad$=q$:GOTO 700
690 PRINT CHR$(7);:GOTO 670
700 q$=INKEY$:IF q$="" THEN 700 ELSE q$=UPPER$(q$)
710 IF (q$)="E" AND q$<="F" THEN PRINT q$;ad$=ad$+q$:GOTO 730
720 PRINT CHR$(7);:GOTO 700
730 q$=INKEY$:IF q$="" THEN 730 ELSE q$=UPPER$(q$)
740 IF (q$)="0" AND q$<="F" THEN PRINT q$;ad$=ad$+q$:GOTO 760
750 PRINT CHR$(7);:GOTO 730
760 RETURN
770 '
780 ' S/P de selection/deselection des adresses
790 FOR ad=adi TO ads
800 IF (ad AND 255)=0 THEN ad=ad+E0
810 GOSUB 610:PRINT#1,c$;HEX$(ad,4);c$
820 NEXT
830 RETURN
840 '
850 ' S/P de dessin des straps
860 gho$=CHR$(230)+" "+CHR$(230):ghf$=CHR$(230)+CHR$(154)+CHR$(154)+CHR$(230)
870 pho$=CHR$(230)+" "+CHR$(230):phf$=CHR$(230)+CHR$(154)+CHR$(230)
880 gvo$=CHR$(230)+CHR$(10)+CHR$(10)+CHR$(10)+CHR$(8)+CHR$(230)
890 gvfo$=CHR$(230)+CHR$(10)+CHR$(8)+CHR$(149)+CHR$(10)+CHR$(8)
900 gvff$=gvfo$+CHR$(149)+CHR$(10)+CHR$(8)+CHR$(230)
910 pvo$=CHR$(230)+CHR$(10)+CHR$(10)+CHR$(8)+CHR$(230)
920 pvff$=CHR$(230)+CHR$(10)+CHR$(8)+CHR$(149)+CHR$(10)+CHR$(8)+CHR$(230)
930 RETURN
940 LOCATE#1,4,1:PRINT#1,gvff$+" "+gvff$+STRING$(4,8)+gho$:LOCATE#1,6,16:RETURN
950 LOCATE#1,4,1:PRINT#1,gvo$+CHR$(8)+ghf$+CHR$(8)+gvo$+STRING$(4,8)+ghf$
960 LOCATE#1,6,16:PRINT#1,pvff$:RETURN
970 LOCATE#1,14,3:PRINT#1,pvff$+" "+pvff$+STRING$(3,8)+pho$:LOCATE#1,17,16:RETURN
980 LOCATE#1,14,3:PRINT#1,pvo$+CHR$(8)+phf$+CHR$(8)+pvo$+STRING$(3,8)+phf$
990 LOCATE#1,17,16:PRINT#1,pvff$:RETURN
1000 MODE 2:END

```

Figure 14



Précisons que F8E0 représente la première colonne à gauche, que le bit 0 est en bas, et que des buffers inverseurs affichent en « vidéo inverse » (la figure 16 présente la séquence obtenue en non-inverseurs).

Si tout est correct, nous pouvons maintenant nous amuser un peu avec un exemple :

Le listing la **figure 16** est un « bargraphe aléatoire » : il simule des niveaux d'entrée variables sur nos quatre colonnes. Notez que sous son aspect ludique, il suffit de remplacer la génération du nombre aléatoire par une carte d'acquisition de données pour obtenir un vrai bargraphe, avec échelle et zéro programmables !

Si vous utilisez des buffers inverseurs, il faut remplacer les données de la ligne 40 par leur complément, qui se calcule comme suit : complément = (255 - donnée).

## Conclusion provisoire

Ces deux exemples, si ils sont spectaculaires, ne rendent pas vraiment compte des possibilités de l'application 32L. Imaginez sur l'écran de votre CPC, un cadre à l'image du pavé de LED, chacune de ces LED pouvant être allumée ou éteinte. Imaginez ensuite que l'on puisse programmer ainsi jusqu'à 256 pavés différents, et les envoyer sur la carte l'un après l'autre. C'est ce but

que nous nous sommes fixé avec le programme SEQUENCE du mois prochain. Pour les impatients rappelons que l'offre du mois dernier concernant la disquette « MICRO-RP » tient toujours.

A bientôt.

Alain CAPO & Jean ALARY

## Nomenclature

### Résistances

R<sub>1</sub> à R<sub>6</sub> : 4,7 kΩ  
R<sub>7</sub> à R<sub>38</sub> : 330 Ω

### Condensateurs

C<sub>1</sub> à C<sub>10</sub> : 0,1 μF

### Circuits intégrés

IC<sub>1</sub> : 74LS688  
IC<sub>2</sub> : 74LS138  
IC<sub>3</sub> à IC<sub>6</sub> : 74LS374  
IC<sub>7</sub> à IC<sub>10</sub> : 74LS244

### Divers

Supports 20 Br. : 9  
Support 16 Br. : 1  
J<sub>1</sub> : SUB-D 9 Br.  
J<sub>2</sub> : SUB-D 25 Br.  
LD<sub>1</sub> à LD<sub>32</sub> : U510PA  
SW<sub>1</sub> : Invers. mini  
Porte-cartes 8TE  
Cadre AR.  
Blindage 220  
416 12 AC M

### VÉRIFICATION « SOFT » 32L

```
10 ad=&F8E0
20 FOR i=0 TO 3:OUT ad+i,0:NEXT CALL &BB18
30 OUT ad+0,&55:OUT ad+1,&AA:OUT ad+2,&55:OUT ad+3,&AA:CALL &BB18
40 OUT ad+0,&AA:OUT ad+1,&55:OUT ad+2,&AA:OUT ad+3,&55:CALL &BB18
50 GOTO 30
```

RAZ generale



DAMIER 1



DAMIER 2

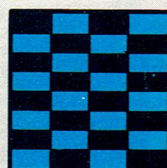


Figure 15

### BARGRAPHE

```
10 ad=&F8E0
20 FOR i=0 TO 3:OUT ad+i,0:NEXT
30 FOR i=0 TO 6:READ d(i):NEXT
40 DATA 1,3,7,15,31,63,255
50 RANDOMIZE TIME
60 FOR i=0 TO 3:OUT ad+i,d(INT(7 * RND))):NEXT
70 FOR tempo=1 TO 150:NEXT
80 GOTO 60
```

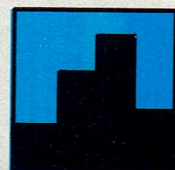


Figure 16