

M. LEVREL F6DTA

INTERFACES POUR ORIC 1 ET ATMOS

ROBOTIQUE, E.A.O, AUTOMATISMES...



SORACOM informatique

**INTERFACES
POUR
ORIC 1
ET
ATMOS**

«La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part que «les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective» et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause, est illicite» (alinéa premier de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 du Code Pénal.»

© Editions SORACOM — 1984
ISBN 2-904032-19-3

M. LEVREL F6DTA

**INTERFACES
POUR
ORIC 1
ET
ATMOS**

ROBOTIQUE , E.A.O , AUTOMATISMES ...

SORACOM informatique

INTRODUCTION

Le dialogue ordinateur-écran, aussi enrichissant soit-il, disons "interactif", pour être branché, n'en exploite pas moins qu'une partie des très grandes possibilités d'un micro-ordinateur ORIC.

Par ailleurs, la dimension d'attrait qu'il peut avoir se trouvera décuplée si on lui adjoint, par les mérites d'interfaces adaptées, les possibilités de mouvoir des objets, de commander des automatismes complexes, de comparer des tensions, de prélever des températures...

Les applications deviennent rapidement innombrables, les centres d'intérêts en EAO (Enseignement Assisté par Ordinateur) ne se mesurent plus et l'on trouvera dans cet ouvrage beaucoup d'applications qui ont été conçues dans un but spécifiquement didactique afin de pouvoir être mises en œuvre directement par le jeu de modules enfichables.

L'apprentissage de la programmation en BASIC et l'accès au langage "machine" n'est plus considéré comme un en-soi fastidieux mais comme un puissant moyen pour accéder la CRÉATION.

Un pas très important vers la robotique pourra ainsi être franchi surtout auprès de jeunes (et de moins jeunes) fort attirés par tous les aspects de l'automatisme appliqué.

L'objectif de cet ouvrage sera ainsi atteint : permettre aux heureux possesseurs d'ORIC l'accès des connaissances de base afin d'utiliser au mieux les énormes possibilités de programmation de leur ordinateur.

Michel LEVREL

CHAPITRE I

LA COMMUNICATION AVEC L'EXTÉRIEUR : L'INTERFACE ENTRÉES/SORTIES

Voyons rapidement l'articulation de l'ORIC :
Il se compose essentiellement :

- d'un CPU 6502, le cerveau du système,
- d'une ROM, où se trouvent les programmes figés qui feront fonctionner le micro-processeur avec son langage propre : langage machine/basic,
- d'une RAM ou mémoire vive où l'utilisateur pourra créer librement ses propres programmes,
- d'un ULA permettant la gestion d'écran, un synthétiseur de sons complexes,

Plusieurs modes de dialogues sont autorisés : avec l'écran couleur, avec le clavier, avec un haut-parleur interne. De nombreux connecteurs permettent le fonctionnement indispensable avec une mémoire externe sous forme de bande magnétique ou de disquettes, une imprimante également ... !

Déjà bien pourvu à sa naissance, nous lui ajouterons la capacité de mesurer des tensions, recueillir des données d'une poignée de jeux ou de capteurs, de comparer des phénomènes lumineux ou électriques, de démarrer moteurs ou jeux de lumières ; tout cela par la flexibilité de la programmation.

Afin de véhiculer les signaux transitants de l'extérieur vers le micro-processeur ou les ordres issus de l'ordinateur pour exécution, il faut un aiguillage ordonné des cycles de lecture et d'écriture, une capacité de stockage des informations fugitives qui sont lancées par l'ORIC à des moments précis : l'interface va remplir tous ces rôles.

Nous allons décrire plusieurs types d'interfaces avec des tâches différentes suivant les besoins de l'utilisateur.

La première interface est particulièrement simple et ne nécessite que trois circuits intégrés au total mais pourra rendre déjà d'énormes services avec ses huit sorties programmables. Sur le plan du logiciel nous verrons qu'elle est remarquable puisqu'elle réagit au premier POKE venu !

La seconde interface est beaucoup plus complète et permet un dialogue entrées-sorties total sur 16 lignes, c'est formidable ... Elle nécessitera quelques connaissances supplémentaires. Mais n'est-ce pas l'intérêt de la démarche informatique ?

UNE INTERFACE TRÈS SIMPLE 8 SORTIES PROGRAMMABLES

Nous débuterons par une réalisation extrêmement simple afin de faire dialoguer notre ordinateur ORIC avec le monde extérieur.

Pour cela nous prendrons les informations issues du BUS d'adresses et de données situé à l'arrière de l'ORIC. Un seul connecteur à 34 broches suffira et nous n'en prendrons qu'une partie seulement. Les adresses de A0 à A7, la broche appelée I/O : Input/Output, I/O control qui permet de déconnecter provisoirement le VIA interne 6522 et les huit données D0 à D7.

Le processus de fonctionnement est le suivant. Le circuit 74LS373 est une interface constituée de huit latches qui vont emmagasiner à leurs sorties les données présentes à l'entrée SEULEMENT lorsque la broche 11 est à l'état haut et ignoreront tout des entrées, à l'état bas. Nous réalisons ainsi un phénomène de "mémorisation".

Si, en décodant les adresses adéquates sur le BUS nous réalisons un niveau haut sur 11 du 74LS373 (et un niveau bas sur I/O control), la mise en mémoire sera réalisée, et l'ordinateur pourra partir pour d'autres tâches sans influencer sur les valeurs emmagasinées en S0 à S7.

L'utilité de la broche I/O de l'ORIC est de ne pas avoir à décoder les adresses hautes (A8-A15) lorsqu'il s'agit de faire intervenir la page 0300-03FF.

Le bas de page 0300 est utilisé pour les besoins du VIA interne 6522. On n'aura donc pas recours à ces adresses. Par contre le haut de page est totalement libre 0310 à 03FF.

Pour notre utilisation nous avons tout simplement utilisé 03FF qui correspond aux huit lignes d'adresses A0-A7 à l'état haut.

Le schéma est donc d'un total dépouillement et n'offrira pas de grosses difficultés de réalisation, hormis un soin méticuleux dans la reconnaissance des fils issus du BUS à 34 broches. Nous allons y revenir un peu plus loin.

Un mot sur l'alimentation de la platine elle-même.

Les trois circuits intégrés : 74LS30, 74LS27 et 74LS373 sont sous 5 Volts et leur alimentation pourra être prise sur la ligne de l'ORIC (fils 33 et 34 du connecteur). Par contre, à moins de n'allumer que des diodes LED !, il ne pourra être question de "tirer" sur la ligne pour un montage commandé par l'interface. Le transformateur de l'ORIC n'y suffirait pas et par ailleurs les appels brusques de courant risquent fort de planter le programme de l'ordinateur par impulsions parasites.

IL sera donc sage d'effectuer une sérieuse source de 5 Volts auxiliaire avec possibilité également de pouvoir utiliser la tension de 12 Volts (avant régulation) pour actionner tous les divers modèles issus de votre imagination : moteur électrique, platine interface pour commande de synthétiseur de fréquence, jeux de lumières ... Un petit transformateur pouvant délivrer 10 à 12 Volts alternatif sous 2 Ampères devrait faire l'affaire. On se souviendra simplement de la gourmandise de certaines ampoules, de lampe de poche par exemple, qui fonctionnent sous 3 Volts et 200 mA minimum : huit ampoules en ligne, toutes allumées, consomment déjà 1,6 Ampères !

RÉALISATION PRATIQUE

Le circuit imprimé est de toute petite taille. Les circuits intégrés sont tous orientés dans le même sens, nous avons employé un double face pour ramener le plan de masse à la partie supérieure, vieille habitude des circuits VHF, mais cette précaution n'est absolument pas nécessaire. Il faudra simplement effectuer un strap pour ramener à la masse les broches 1 et 10 du 74LS373 au commun du circuit imprimé en simple face.

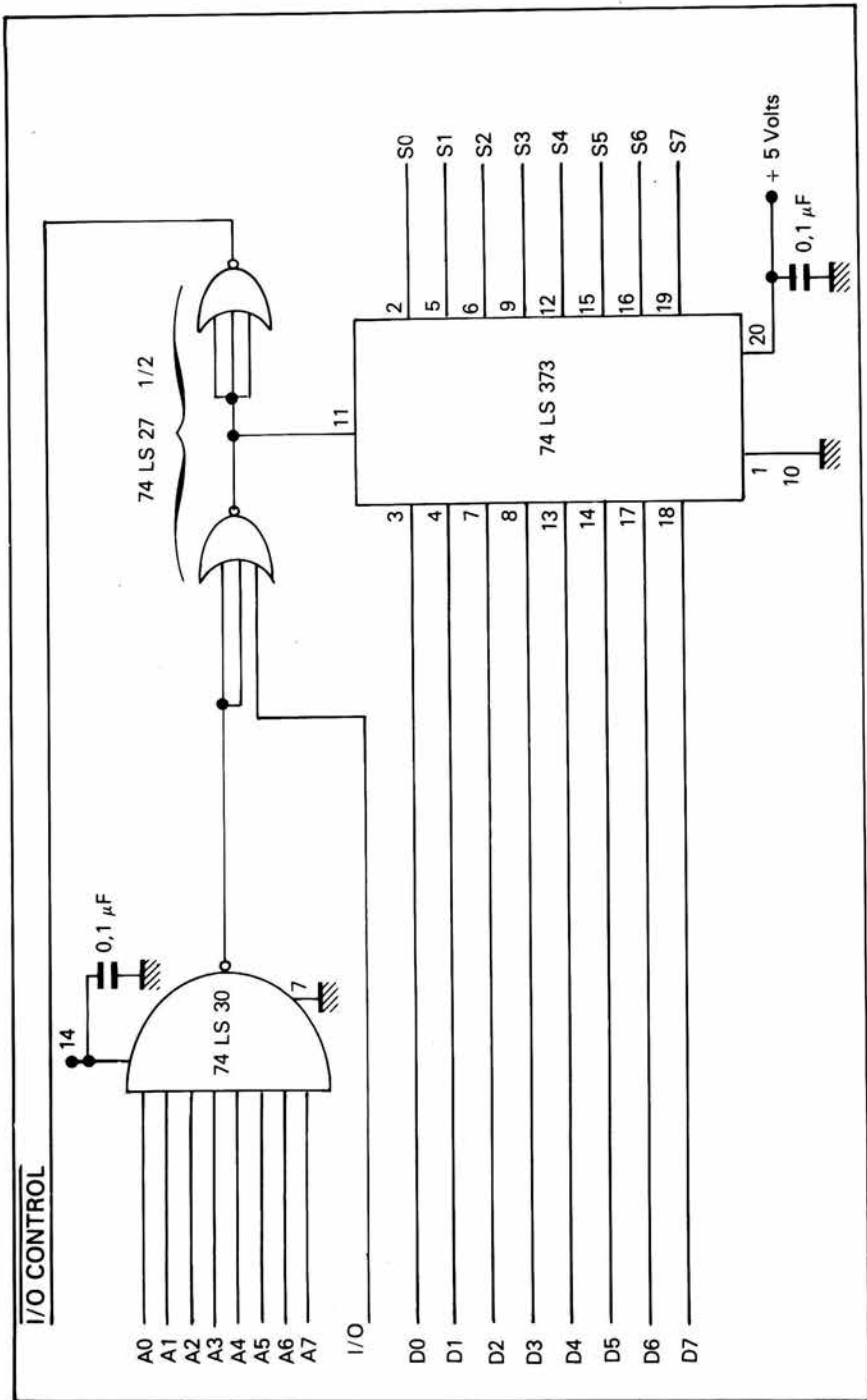
Les découplages d'alimentation sont au nombre de deux. La valeur utilisée pourra varier entre 10 nF et 1 uF sans problèmes.

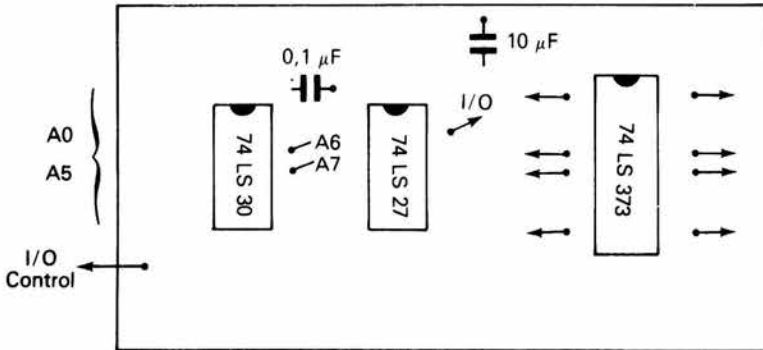
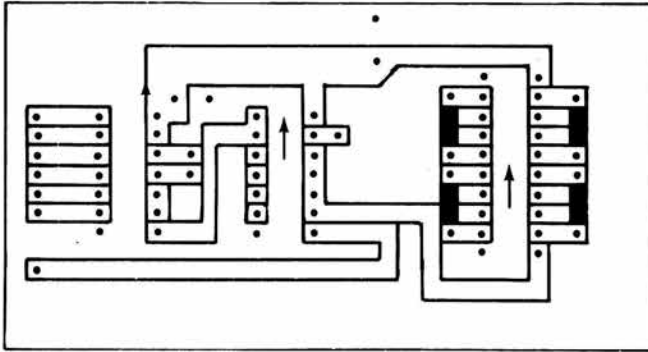
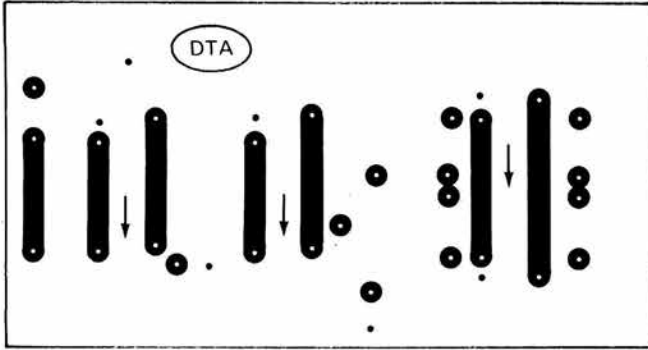
Veiller à ne pas intervertir sur le 74LS373 les broches d'entrées et de sorties : les entrées sont marquées d'une petite flèche, les sorties "sortent" par des trous sur le dessus du circuit imprimé.

L'ordre des fils d'adresse sur les entrées de la porte NAND 74LS30 n'a aucune importance. On pourra mettre A0 sur la broche 4 ou 7 sans inconvénient.

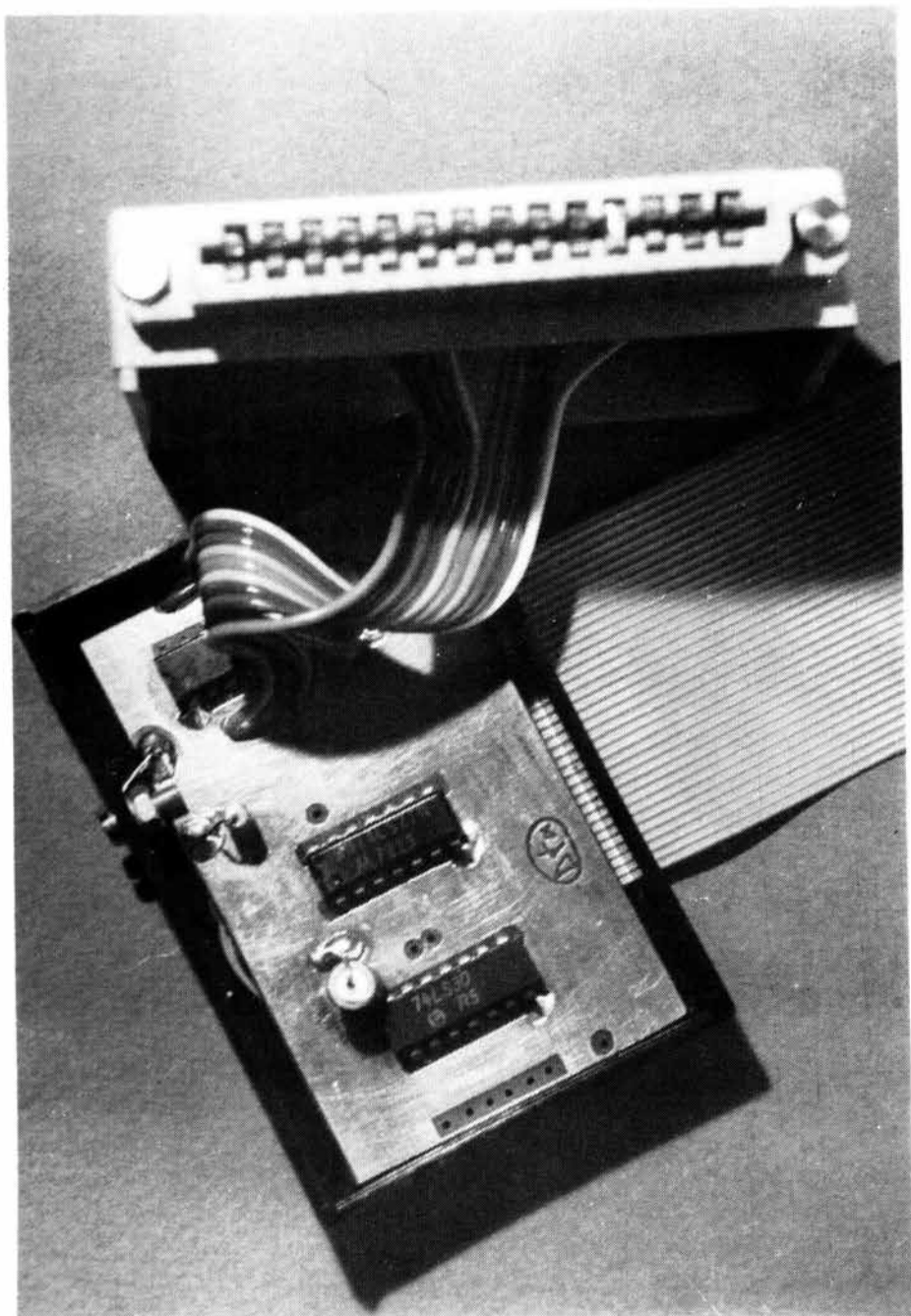
La seule difficulté dans un tel montage est le raccordement de la carte au connecteur 34 broches de l'ORIC. Il y faut un maximum de soin si l'on ne désire pas mettre à mal son ordinateur par court-circuits ou renvoi du 5 Volts sur le BUS.

Pour cela on achètera un connecteur tout monté à 34 broches du type femelle avec une nappe de fils déjà sertie. De nombreux revendeurs les procurent tout faits. Il suffit ensuite de suivre la numérotation des broches fournie dans le manuel de l'ORIC (et dans notre ouvrage). En regardant le connecteur de l'ordinateur de face, la broche 1 est en bas et à gauche, la broche 2, juste au-dessus, la broche 34 est à l'extrême droite en haut.





Implantation interface 8 sorties



On conservera une longueur de nappe d'une trentaine de centimètres, au maximum, afin de ne pas charger trop le BUS par capacités parasites. Les fils non nécessaires sont coupés avec une pince fine et l'isolant ramené sur le bout du conducteur interne par étirement à la main. On peut ainsi opérer un isolement efficace par quelques millimètres de plastique en bout. Le 9^e fil va à D2, le 11^e à A3, etc...

Le branchement de l'interface s'effectue lorsque l'ORIC n'est pas sous tension. On veillera à l'orientation correcte du connecteur 34 broches s'il ne comporte pas de détrompeur. Le branchement réalisé, on met l'ordinateur sous tension avec les modalités classiques de fonctionnement.

UTILISATION

Nous allons donner quelques exemples d'utilisation de cette interface (non limitatifs !) afin de familiariser le lecteur avec son maniement extrêmement simple.

On utilisera la carte avec ampoules du chapitre suivant afin de bien visualiser les états logiques ainsi que le tableau donnant la correspondance décimale/hexadécimale/binaire.

Nous aurons d'ailleurs la correspondance directe des "1" et des "0" dans le registre à huit bits de cette façon :

Si l'on veut connaître les "0" (lampes éteintes) et les "1" (lampes allumées) pour 57 dec. , #39 nous ferons l'instruction POKE 1023,57 ce qui donnera la configuration des ampoules :

$$57 = \#39 = 00111001$$

soit en ampoules 0 0 1 1 1 0 0 1



1) L'instruction POKE :

Elle permet d'affecter une valeur entre 0 et 255 (0 et FF hexa.) à l'adresse que l'on a fixée.

POKE 1023,57 veut dire qu'à l'adresse 1023 (#03FF) nous affectons la valeur 57.

2) Le cheminement inverse est réalisé par l'instruction PEEK.

Si nous faisons PRINT PEEK (3145) nous donnerons l'ordre au micro-processeur de nous dire la valeur inscrite à l'adresse 3145. L'adresse se met entre parenthèses, sans oublier l'instruction PRINT.

PROGRAMMES D'APPLICATION

```
5 REM HUIT AMPOULES ALLUMÉES
10 POKE 1023,255
```

Une ampoule sur deux :

```
5 REM ALLUMAGE D'UNE AMPOULE/2
10 POKE 1023,85
20 REM 85=01010101
```

Chenillard

```
10 REM CHENILLARD SIMPLE
20 FOR N=0 TO 7
30 READ A
40 POKE 1023,A
45 WAIT50
50 NEXT
60 DATA 1,2,4,8,16,32,64,128
70 RESTORE
80 GOTO 20
```

```
10 REM CHENILLARD MONTANT/DESCENDANT
20 FOR N=0 TO 13
30 READ A
40 POKE 1023,A
45 WAIT50
50 NEXT
60 DATA 1,2,4,8,16,32,64,128,64,32,16,8,
4,2
70 RESTORE
80 GOTO 20
```

```
5 REM ALLUMAGE ALEATOIRE
10 A=INT(RND(1)*255)
20 POKE 1023,A
30 WAIT 25
40 GOTO 10
```

```
5 VISUALISATION DE "A" ALEATOIRE
10 A=INT(RND(1)*255)
20 POKE 1023,A
25 LPRINT A;
30 WAIT 25
40 GOTO 10
148 157 193 31 252 160 64 247 158 254 91
236 41 76 105 210 31 94
104 43 80 78 132 97 52 50 244 138 44 119
11 68 235 72 21 119 150
252 46 19 188 119 73 236 222 232 43 95 2
23 153 7 234 154 160 58
252 92 117 46 9 182 186 107 191 121
```

La suite de nombres après la ligne 40 est la visualisation du "A", variable aléatoire entre 2 et 254.

PROGRAMMATION D'UN SYNTHÉTISEUR DE FRÉQUENCES ENTRE 88 ET 108 MHZ

(Nous rentrerons dans plus de détails dans l'un des chapitres suivants).

Supposons que notre récepteur fonctionne avec une moyenne fréquence de 10,7 MHz et un battement infradyne, nous aurons pour la fréquence d'écoute de 98,600 MHz et un synthétiseur au pas de 100 kHz :

3 CLS

```
10 REM PROGRAMMATION DE SYNTHETISEUR DE
FREQUENCES
20 PRINT:PRINT:PAPER 3:INK 0
30 PRINT:PRINT:PRINT"      FREQUENCE DE R
ECEPTION (en KHz)"
40 INPUT A
50 REM Soustraction Moyenne Frequence
60 B=A-10700 :CLS:PRINT:PRINT:PRINT "LA
FREQUENCE DE RECEPTION EST
DE:"
65 PRINT:PRINT:PRINTA "KHz"
70 N=(B/100)-(512+256)
80 POKE 1023,N
90 GET A$
100 IF A$="M" THEN A=A+100:GOTO 60
110 IF A$="D" THEN A=A-100:GOTO 60
120 GOTO 90
```

L'instruction GET A\$ provoque un examen du clavier : si nous appuyons sur la lettre "M" = Montée, nous incrémenterons la fréquence d'écoute de 100 kHz.
Au contraire, un appui sur "D" = Descente, provoquera une décrémentation de 100 kHz.

Cette petite interface pourra gérer des automatismes complexes avec 256 possibilités.
Les temporisations se font avec l'instruction WAIT.

WAIT 1 = 10 ms ou 1/100^e de seconde.
WAIT 100 = 1000 ms ou une seconde, etc...

Les routines utilitaires s'appellent par des GOSUB suivis de RETURN.

```
5 REM RELAIS 1,2,5 "TRAVAIL"  
10 POKE 1023,19  
20 WAIT 350  
40 REM RELAIS 4,7 "TRAVAIL"  
50 POKE 1023,72  
60 WAIT 55  
70 REM TOUS LES RELAIS "REPOS"  
80 POKE 1023,0  
90 WAIT 800  
100 REM RETOUR EN DEBUT DE CYCLE  
110 GOTO 20
```


CHAPITRE II

UNE INTERFACE A 16 LIGNES PROGRAMMABLES EN ENTRÉES-SORTIES

Elle sera essentiellement composée d'un PIA (peripheral interface adapter) d'origine Motorola, le 6821 qui pourra nous fournir simultanément 16 lignes de communication avec l'extérieur.

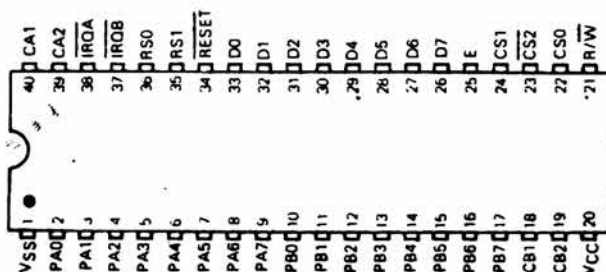
La particularité de ce genre de composant est de pouvoir, à la demande, transformer ces 16 lignes soit en **sorties** (pour donner des ordres), soit en **entrées** (pour en capter), et ceci avec tous les panachages permis : 15 entrées/1 sortie, 8 entrées/8 sorties, 2 entrées/14 sorties ... par le biais de la programmation qu'on lui fera. Les possibilités sont donc multiples et permettent de se plier à toutes les exigences.

CONSTITUTION PHYSIQUE DU 6821

On utilisera la version P = plastique, la plus économique ! et indifféremment le 6821 P (1 MHz), le 68A21 (1,5 MHz) et le 68B21 (2 MHz) : la vitesse n'étant pas prise en compte dans notre application avec l'horloge du CPU à 1 MHz.

C'est un circuit à 40 broches.

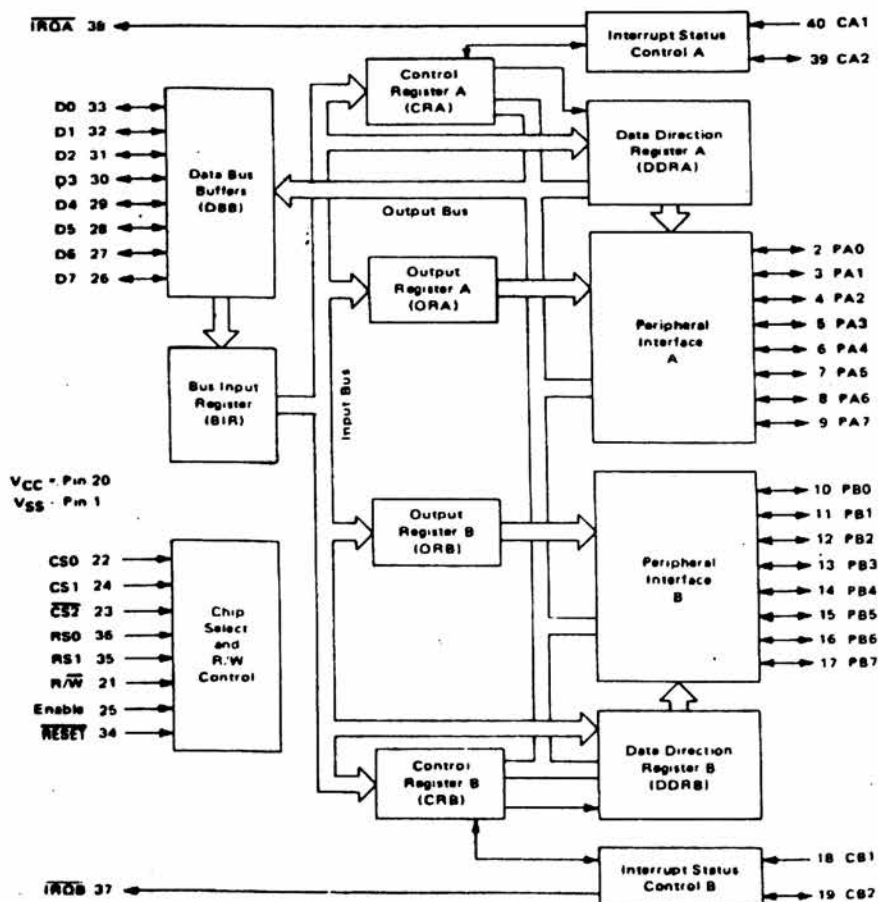
Toutes les sorties de PORTS sont du même côté et dans un ordre identique (contrairement au 6532, par exemple), ce qui est extrêmement pratique sur le plan du circuit imprimé. Les commandes principales s'exécutent sur CS0, CS1 et CS2.



Brochage du 6821

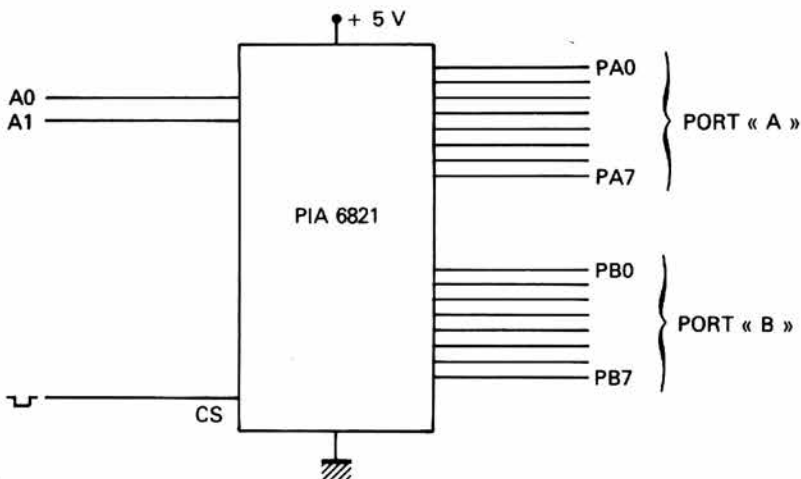
L'organigramme complet est donné ci-dessous et montre bien l'organisation de l'ensemble. Nous n'utiliserons pas, dans notre montage, les possibilités d'interruption mais elles peuvent être précieuses dans certains cas précis.

Document MOTOROLA



Quelques points de vocabulaire importants :

Les 16 lignes sont subdivisées en deux groupes de 8 lignes appelés : "PORT" ; nous aurons ainsi le Port "A" et le Port "B".



L'affectation se fait de 0 à 7, par convention (et non de 1 à 8 comme cela aurait pu l'être)...

Nous aurons donc respectivement :

- Pour le Port A :
PA 0, PA 1, PA 2, PA 3, PA 4, PA 5, PA 6, PA 7.
- de même pour le port B :
PB 0, PB 1, PB 2, PB 3, PB 4, PB 5, PB 6, PB 7.

De cette façon toutes les lignes pourront être reconnues facilement. Comment accéder à ces lignes, d'abord pour les assigner en entrées ou sorties, puis leur fournir ou leur prendre leurs données ?

Deux broches sont disponibles sur le circuit intégré Motorola 6821, ce qui nous fera quatre adresses seulement pour le port A et B.

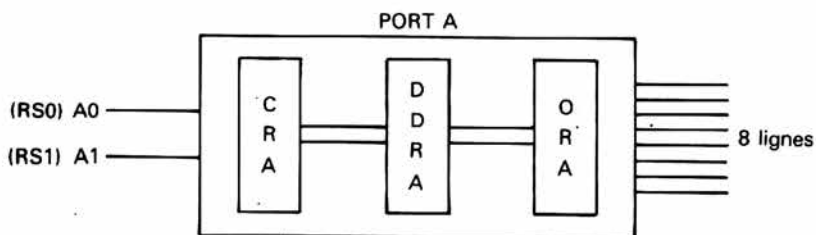
Ce sont les fils notés sur le schéma A 0 et A 1 avec les quatre niveaux suivants :

A0	A1	Adresse décim.	Adresse hexa.	
0	0	1020	03FC	Port A
1	0	1021	03FD	
0	1	1022	03FE	Port B
1	1	1023	03FF	

L'adressage du PIA est provoqué par un niveau bas sur la broche CS ce qui aura lieu aux adresses **1020 a 1030** par l'intermédiaire de circuits 74LS30, 74LS04, 74LS32.

1 - FONCTIONNEMENT DU PIA

Le cheminement pour mettre les lignes dans un état donné est un peu particulier et nécessitera toute l'attention du lecteur.



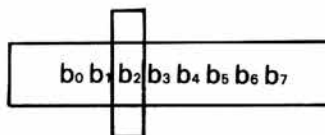
Nous prendrons l'exemple du port A ; sachant que pour le port B la démarche sera exactement identique.

Voyons en la structure :

3 parties distinctes interviennent :

- 1) le CRA Command Register A (registre de commande A).
- 2) le DDRA Data Direction Register A (registre de direction des données).
- 3) l'ORA Output Register A (registre de données A).

Le principal acteur est le registre de commande (CRA) (à l'adresse 1021) et essentiellement pour notre application le niveau du bit "b2".



Les autres sont utilisés à d'autres tâches d'interruption dont nous ne tiendrons pas compte.

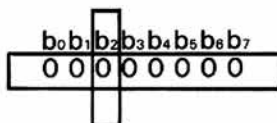
Si "b2" est au niveau 0, les données que nous mettrons à l'adresse immédiatement plus basse : 1020 seront des ordres de Direction : (sélection du DDRA). Un niveau "0" provoquera la mise en "entrée" du port A, un niveau "1" fixera les lignes en "sorties".

Prenons un exemple pratique :

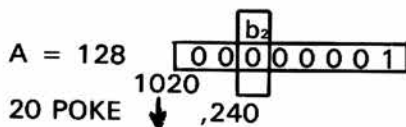
Nous désirons 4 entrées sur PA 0, PA 1, PA 2, PA 3 et 4 sorties pour le reste PA 4, PA 5, PA 6, PA 7.

Notre programme en BASIC sera le suivant :

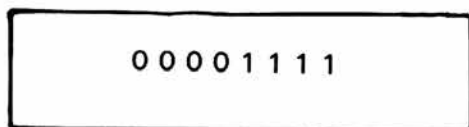
10 POKE 1021;0



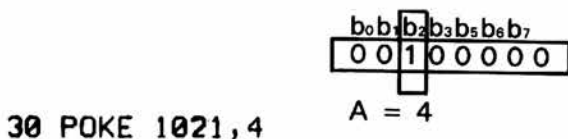
Nous choisissons d'agir sur le registre de Direction par la mise à 0 de b_2 : à noter qu'en POKANT 1021, 128 le résultat pour b_2 sera le même. Par exemple :



Les quatre derniers bits de poids élevés sont au niveau 1, ce qui fixe PA 4, PA 5, PA 6, PA 7 en **sorties**. Les 4 premiers bits sont à 0, les lignes sont configurées en **entrées**.



Il ne reste plus ensuite qu'à donner les valeurs aux sorties : Nous revenons à l'adresse 1021 du registre de Commande (CRA). Nous avons vu qu'il faudra mettre le bit " b_2 " à 1 cette fois.



Si nous voulons agir pour avoir un niveau haut sur PA 6 et un niveau bas sur PA 4, PA 5, PA 7, nous reviendrons sur l'adresse 1020.

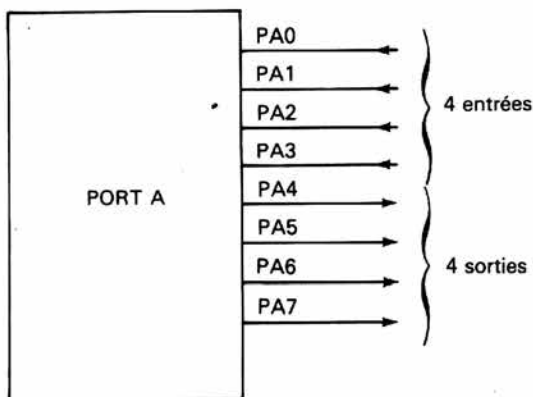
40 POKE 1020,64

0 0 0 0 0 0 1 0 A = 64

Ce qui met bien PA 6 à 1 et les autres à 0 (voir tableau de conversion binaire).

Nous aurons donc pour la configuration de quatre entrées-quatre sorties, avec PA 6 à 1, le programme global :

```
10 POKE 1021,0
20 POKE 1020,240
30 POKE 1021,4
40 POKE 1020,64
```



Le cheminement sera toujours identique :

- 1) atteindre le DDRA par le CRA,
- 2) assigner les entrées ou les sorties,
- 3) atteindre ORA par le CRA,
- 4) assigner les données.

Le port B fonctionne de manière absolument identique avec les deux adresses hautes :

1022 AU LIEU DE 1020

1023 AU LIEU DE 1021

2) LA LECTURE DES DONNÉES

A la mise sous tension, les ports du PIA sont configurés en entrées.
Tous les registres à "0".

Si l'on fait :

10 POKE 1021,4 : Accès au registre de données ORA.

20 PRINT PEEK (1020) ; lecture des données (;) crée un défilement horizontal.

30 GOTO 10.

Nous aurons toute une suite de 255, 255, 255 (FF en hexadécimal).
Les entrées étant par ailleurs à l'état haut ("1" logique) **pour le port A** seulement par des résistances pull-up. Par contre, le comportement du port B est différent. Effectuons le petit programme suivant :

10 POKE 1023,4 : accès au registre ORB.

20 PRINT PEEK (1022) ; lecture des données du port B.

Nous aurons une suite erratique du genre :

18, 0, 255, 128, 255, 12 ... voir tableau (sortie sur imprimante) A.
Les entrées sont en effet à haute impédance et se mettent à l'état haut ou bas selon les variations de charges d'entrées.

Dans les cas cités précédemment, nous lisons la totalité du port A ou B : toutes les lignes sont en entrées.

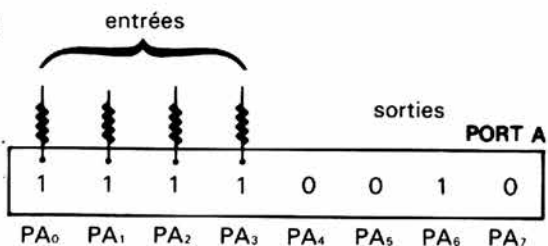
Reprenons le cas du paragraphe 1 où seules les 4 lignes PA 0, PA 1, PA 2, PA 3 sont configurées en entrées.

Si nous prenons les données globales, nous lirons "79", puisque nous cumulerons la lecture des 4 premières lignes + la valeur mise dans les sorties sur les quatre autres lignes.

Nous aurons ainsi 64 (PA 6 à 1) + 15 = 79.

Programme avec "MASQUE"

```
5 REM PA0,PA1,PA2,PA3 :ENTREES
6 REM PA4,PA5,PA6,PA7 :SORTIES
7 REM INTERROGATION DES ENTREES
10 POKE 1021,0
20 POKE 1020,240
30 POKE 1021,4
40 POKE 1020,64
70 A=PEEK(1020)
80 B=15 AND A
90 PRINT B;
100 GOTO 70
```

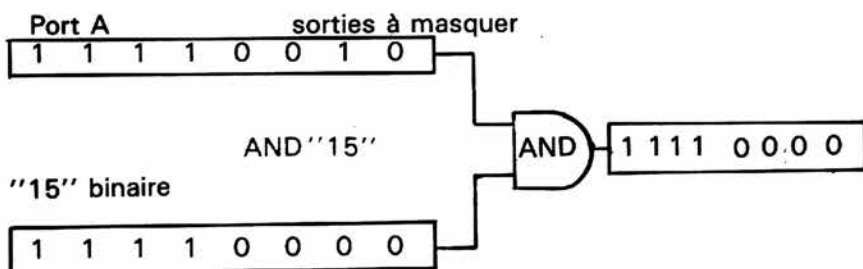


$$15 + 64 = 79$$

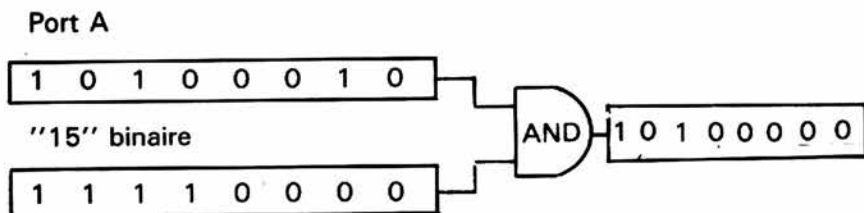
Tableau A

248 0 0 248 0 248 0 224 0 252 0 224 128
 255 224 128 255 0 0 0 248
 0 0 0 248 0 0 224 128 192 128 0 0 192 0
 0 240 248 0 192 0 0 240 0
 224

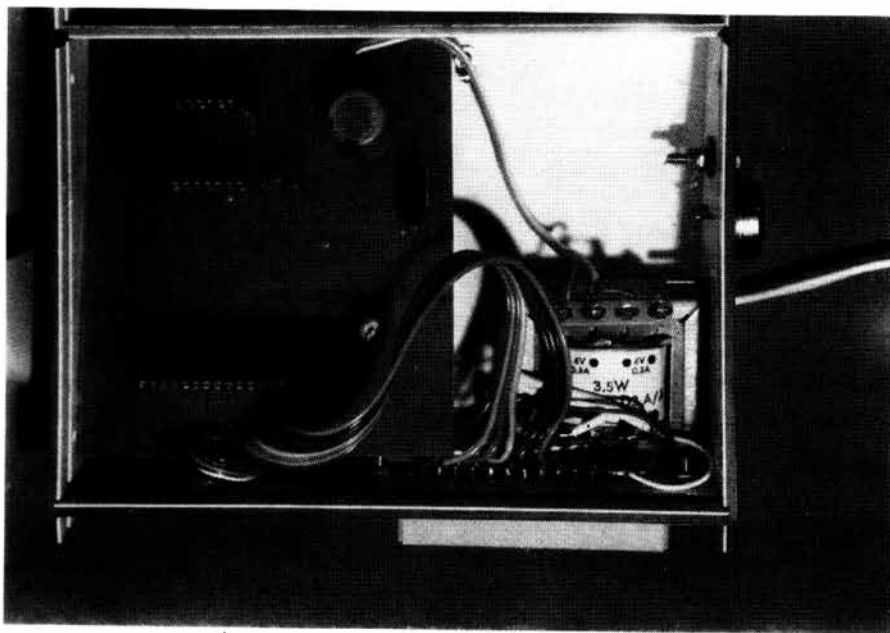
Afin de n'avoir à lire que la portion du port qui nous intéresse, on opère alors une opération logique AND qui permet de "masquer" la partie programmée en sorties en la mettant automatiquement à 0.



Si le Port A a ses entrées à 1010, nous aurons :



On pourra s'exercer à masquer d'autres parties en sachant que des "11111111" ne masquent plus rien et que "00000000" masque tout !



BOITIER 16 ENTRES / SORTIES

CHAPITRE III

RÉALISATION PRATIQUE DE LA CARTE D'INTERFACE 16 ENTRÉES/SORTIES

Nous avons vu dans le chapitre précédent le fonctionnement du PIA qui permet de dialoguer avec l'extérieur.

En abordant la réalisation pratique nous donnerons un certain nombre d'exemples concrets qui permettront au réalisateur de se familiariser avec une technique de programmation, somme toute très simple.

Quelques circuits intégrés de la série LS-TTL nous permettront de décoder quatre adresses hautes situées en ROM et inutilisées pour le fonctionnement BASIC.

Lorsque nous sélectionnons le PIA, la ROM est automatiquement déconnectée par un niveau bas sur la broche ROM DIS.

Nous avons vu que les quatre adresses utiles sont : 1020, 1021, 1022, 1023. Elles nous donneront accès aux registres de commande CRA et CRB, ainsi qu'à DDRA, ORA, DDRB et ORB. Nous avons vu plus haut leur signification exacte.

Cinq circuits intégrés sont utilisés :

- 2 portes NAND à huit entrées du type 74LS30,
- 4 inverseurs dans un boîtier 74LS04,
- 2 portes OR dans un boîtier 74LS32,
- 1 PIA 6821P.

La liaison avec l'ORIC est assurée par un connecteur 34 broches femelle sur l'arrière de l'appareil. On utilisera de préférence du fil en nappe pour l'interfaçage ainsi qu'une longueur raisonnable de fil de liaison afin de ne pas introduire de charge capacitive prohibitive sur le BUS du microprocesseur : une longueur de 30 à 40 cm paraît un maximum à ne pas dépasser.

Dans tous les cas, il n'est pas judicieux d'employer un circuit imprimé en liaison directe sur cette prise sur le plan mécanique.

Pour les PORTS de sortie toutes les solutions sont permises : le connecteur ou les bornes à vis (également une double rangée de dominos d'électricien de petit modèle). Les contacts doivent être sans problèmes ; il faudra également éviter les sources de court-circuits.

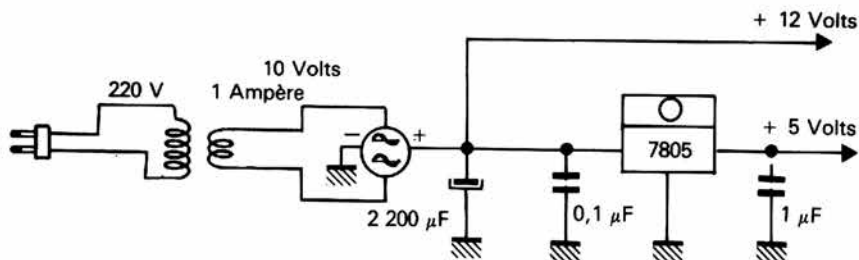
ALIMENTATION DE LA PLATINE

Pour des applications à faibles courants on pourra utiliser l'alimentation propre de l'ORIC : broche 33 du connecteur principal = + 5 Volts ; broche 34 = masse. Mais nous ne conseillons pas cette solution car le transformateur du microprocesseur risquerait d'être rapidement surchargé.

Nous préconisons plutôt la solution de l'alimentation séparée avec un régulateur 5 Volts et radiateur, afin d'effectuer les nombreuses manipulations que nous présenterons dans les chapitres suivants sans problèmes "alimentaires".

Un transformateur pouvant délivrer un à deux Ampères sous 12 Volts fera parfaitement l'affaire et s'intégrera facilement dans un boîtier de faibles dimensions.

La présence d'une tension de 12 Volts facilitera par ailleurs l'interfaçage avec des circuits C-Mos.



Alimentation séparée

PARTICULARITÉS DES PORTS A & B

Sous des apparences identiques, les deux PORTS A & B possèdent des caractéristiques un peu différentes qu'il n'est pas inutile de connaître, selon les besoins. La programmation en Direction et Données s'effectue de la même façon. Par contre ils diffèrent sur les points suivants :

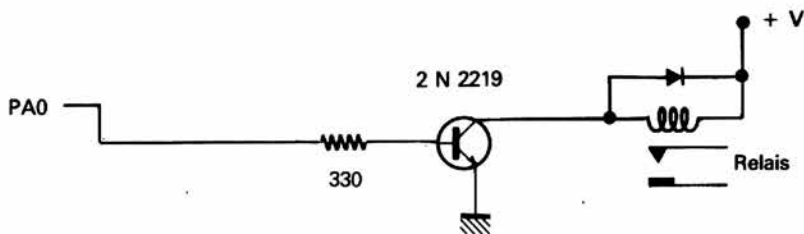
Le port A possède des résistances pull-up mettant les entrées au niveau "1" : PA0 à PA7.

Le port B est un 3 états : il a la capacité de se mettre à haute impédance en mode entrée et réclame moins de courant que le port A.

Dans tous les cas un RESET mettra à zéro tous les registres et les ports sont configurés en ENTRÉES.

Vu sous l'angle de l'utilisateur nous serons obligés dans la plupart des applications de recourir à un transistor ou un circuit intégré capable de fournir une certaine puissance : allumage d'ampoules, déclenchement d'un relais ...

Il est possible d'utiliser un autre circuit, le 6822, qui est compatible broche pour broche avec le 6821 et permettant l'interfaçage C-MOS direct en alimentant ses sorties par une tension pouvant atteindre 18 Volts : sorties à drain ouvert. Ce pourrait être une excellente solution pour driver des synthétiseurs de fréquence directement. Ce circuit est toutefois moins répandu que le 6821 et donc un peu plus onéreux.



LE CABLAGE DE LA PLATINE D'INTERFACE

La réalisation sur circuit imprimé facilite la mise en place générale. Les quatre petits circuits intégrés sont orientés de la même façon : le 6821 est dans le sens inverse.

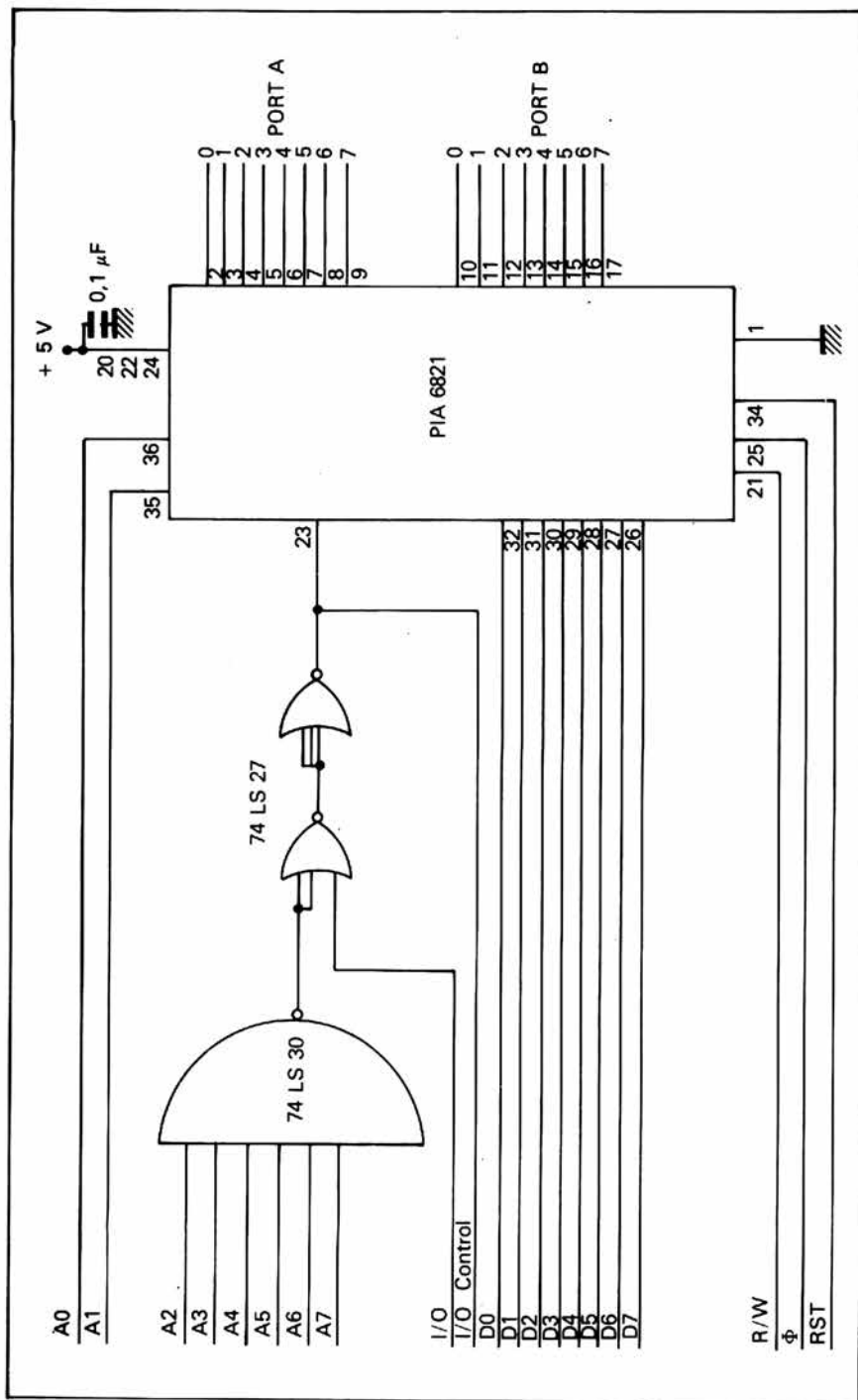
Les trois condensateurs de découplage ont une valeur peu critique, leur présence est cependant indispensable ! Les sorties port A et B se feront sur deux connecteurs séparés à l'arrière du boîtier.

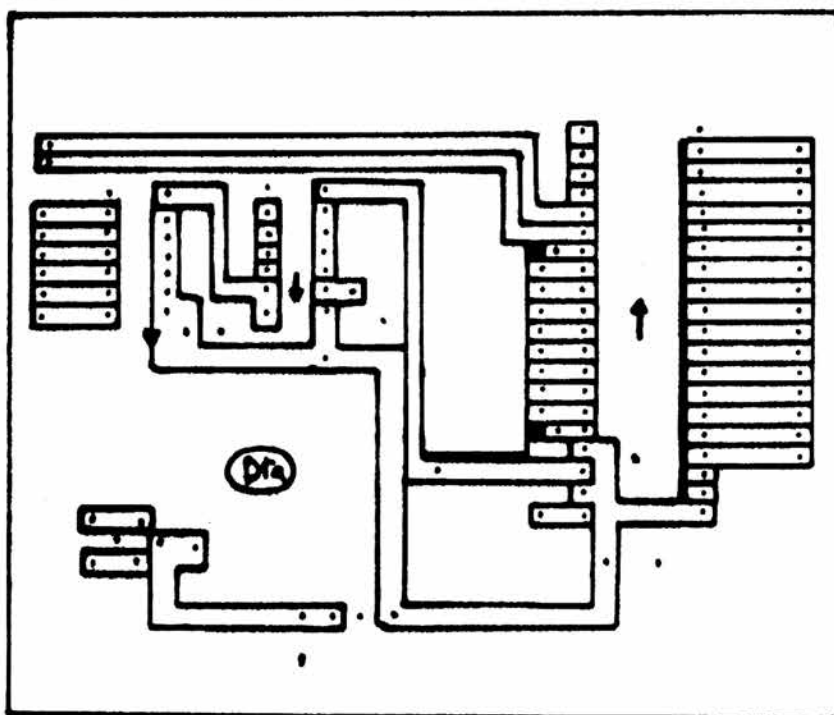
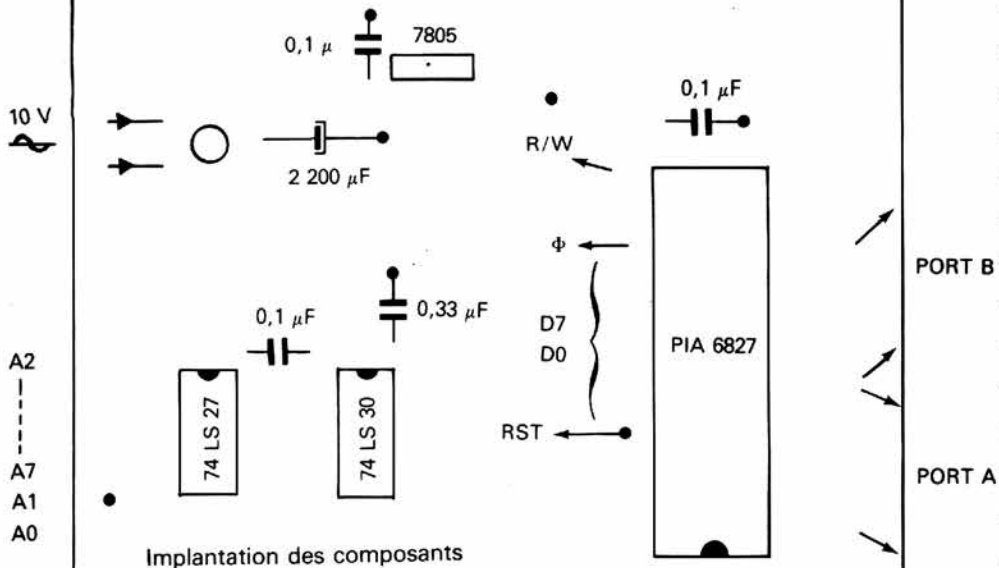
Le seul point critique est le raccordement des lignes d'adresses et de données sur le BUS de l'ORIC.

Il faudra ménager un minimum d'attention pour le réaliser. Notons que ce montage existe non seulement en KIT à assembler, mais également tout réalisé pour ceux qui n'auraient pas une certaine habitude du câblage électronique. L'utilisation d'un câble en nappe favorise, à notre avis, la réalisation pratique ainsi que l'esthétique.

De même, un montage soigneux et sans soudures excessives diminuera considérablement le risque de court-circuits qui peuvent être irrémédiables pour la vie du micro-ordinateur.

On évitera également les court-circuits sur les ports programmés en sorties ainsi que la présence d'un fer à souder "balladeur" lorsque le montage est sous tension... Moyennant ces quelques précautions, l'interface s'est montrée d'une résistance à toutes épreuves. D'ailleurs, pour le fonctionnement de montages sous 220 Volts ou 380 Volts, nous utiliserons plus loin une méthode très efficace : l'interfaçage par coupleurs optiques autorisant un isolement supérieur à 3 000 Volts.





CHAPITRE IV

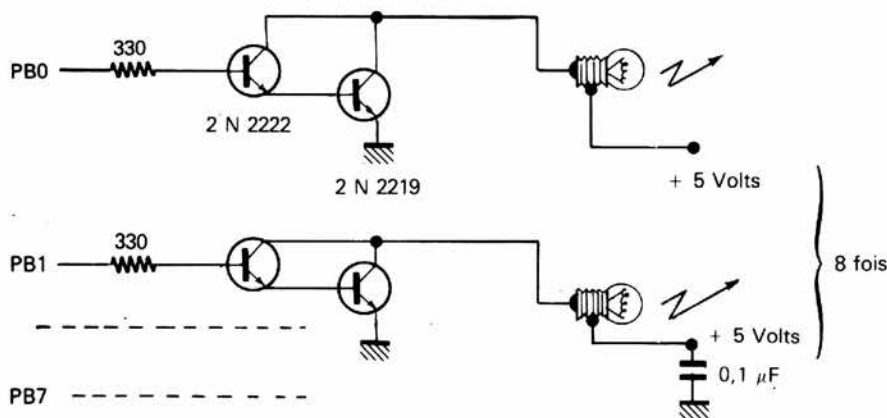
DES MOYENS SIMPLES : JEUX DE LUMIÈRES, COMMANDES DE RELAIS ...

Le lecteur qui n'est pas trop rompu aux techniques de programmation aura tout intérêt à se pencher sur ce module qui lui permettra d'accéder facilement par la suite à des sujets plus complexes.

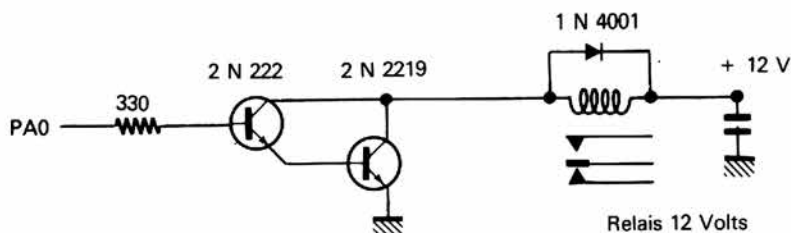
La carte que nous proposons comporte huit ampoules, drivées par deux transistors montés en Darlington et permettra de matérialiser les états du PIA en sorties.

Tous les effets sont permis : l'allumage séquentiel, le chenillard classique, l'éclairage "aléatoire" par une instruction RND, etc...

On pourra utiliser des ampoules de lampe de poche en veillant à leur tension d'utilisation et surtout à leur consommation : la plupart réclament aux alentours de 200 mA, ce qui, multiplié par huit, donnera déjà **1,6 A** Attention aux capacités de l'alimentation !



La carte enfichable ira indifféremment sur le port A ou B selon les adresses que l'on aura définies par programmation. La carte "relais" fonctionnera de la même façon que la carte des ampoules.



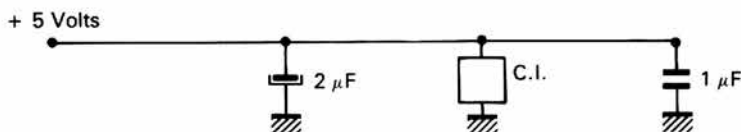
RÉALISATION DES CARTES

Le câblage de la plaque imprimée se fait en utilisant les deux faces : la partie supérieure sert essentiellement au plan de masse, les liaisons se faisant par les pistes du dessous.

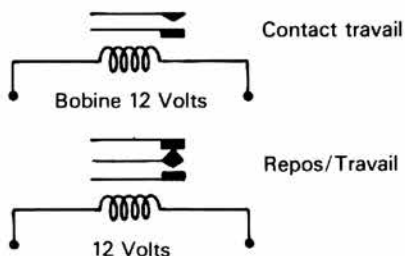
On utilisera huit résistances de 330 Ohms.

Des 2N2222 et 2N2219 forment l'étage d'amplification. Le refroidissement ne nécessite pas de radiateur. Les liaisons aux ampoules se feront impérativement comme indiqué : le culot de l'ampoule au collecteur du transistor et la vis au + 5 V afin d'éviter les court-circuits. Les ampoules sont en effet très rapprochées sur le circuit imprimé et les vis sont ainsi à un potentiel commun.

Il sera bon de mettre un ou deux condensateurs de découplage sur la platine (1 μ F, 16 V, par exemple).



La platine relais ne comporte que 4 relais. Les types peuvent être très variés selon les fournisseurs et les utilisations. Ils sont principalement caractérisés par la tension de fonctionnement de la bobine : ici 12 Volts, le pouvoir de coupure des contacts : 500 mA, 1 A, 10 A selon que l'on voudra fonctionner sur un train électrique ou un montage du type EDF ! Attention également à l'isolement.



Il existe des relais à contact travail, contact repos, contacts repos/travail au gré de l'utilisateur.

PROGRAMMATION

Nous avons déjà utilisé l'instruction POKE pour mettre une valeur à une adresse donnée. La lecture de la donnée, à l'inverse, s'effectue par l'instruction PRINT PEEK (adresse).

Exemple :

```
PRINT PEEK (1020) en décimal  
ou  
PRINT PEEK (#03FC) en hexadécimal.
```

Notons que l'ORIC donne directement la conversion du décimal en hexadécimal par :

```
PRINT HEX$ (1020)#03FC
```

Un court programme donnera la suite de conversion des 10 adresses :

```
10 FOR N= 1020 TO 1030  
20 PRINT N  
30 PRINT HEX$ (N)  
40 NEXT
```

Nous donnons en fin d'ouvrage une table de conversion décimal/hexadécimal/binaire de 0 à 255, c'est-à-dire sur 8 bits.

Ce tableau pourra être très utile pour ceux qui n'ont pas l'habitude de cette gymnastique chiffrée : elle permet de savoir (et de prévoir !) dans un chenillard, par exemple, l'ampoule qui sera allumée (bit à "1") et celle éteinte (bit à "0").

C'est ainsi que, si nous POKONS dans le registre des données du port B la valeur :

58 (déc.) = # 3A (hexa.),

cela correspondra en binaire dans la suite des 1 et des 0 à :

0 1 0 1 1 1 0 0

et ainsi seront allumées les ampoules 2, 4, 5 et 6, les autres étant éteintes.

On peut, par programme, POKER individuellement les adresses du port de sortie après les trois lignes d'initialisation. Mais il est plus intéressant d'utiliser une instruction READ avec une suite de DATAS.

Analysons le 1^{er} cas :

Un seule ampoule clignotante sur le port A, première ligne (PA0). Les trois premières lignes de programme initialisent le port, c'est-à-dire le mettent en position de sortie :

```
5 REM UNE SEULE AMPOULE CLIGNOTANTE
6 REM PORT A 0
10 POKE 1021,0
20 POKE 1020,255
30 POKE 1021,4
40 POKE 1020,1
50 WAIT 100
60 POKE 1020,0
70 WAIT 100
80 GOTO 40
```

10 POKE 1021,0	Choix d'action sur le registre de direction (B2 à 0)
20 POKE 1020,255	Toutes les lignes du port A en sorties. Notos qu'on aurait pu faire également POKE1020,1 : PA0 seule en sortie.
30 POKE 1021,4	Choix d'action sur le registre des données : (bit B2 à 1).
40 POKE 1020,1	Bit 0 du registre de données à 1, l'ampoule s'allume.
50 WAIT 100	Attente d'une seconde, position allumée : Wait 1 = 10 ms, 100 = 100 ms = 1 seconde.
60 POKE 1020,0	Bit 0 du registre des données à 0. Attente d'une seconde en position éteinte. Retour en début de programme : position allumée.

```
5 REM CHENILLARD sur le PORT A
6 REM PA0 à PA7
10 POKE 1021,0
20 POKE 1020,255
30 POKE 1021,4
```

```

40 FOR N=0 TO 7
50 READ A
55 PING
60 POKE 1020,A
70 WAIT 100
80 NEXT N
90 DATA 1, 2, 4, 8, 16, 32, 64, 128
100 RESTORE: GOTO 40

```

Le programme d'un chenillard n'a pas été fait avec des poke séparés, ce qui serait évidemment très fastidieux par cette méthode pour chaque ampoule.

On utilisera donc pour des programmations complexes, l'instruction READ suivie des DATAS.

Les trois premières lignes initialisent les lignes du port.

```

5 REM PROGRAMME EN LANGAGE MACHINE
6 REM ALLUMAGE DES DEUX PREMIERES AMPOU
LES
10 FOR N= 0 TO 20
20 READ D%
30 POKE #0400 +N,D%
40 NEXT
45 REM INITIALISATIONS DES SORTIES
50 DATA #A9,#00,#8D,#FD,#03,#A9,#FF,#8D
, #FC,#03,#A9,#04,#8D,#FD,#03
55 REM MISE A "1" DE PA0 ET PA1
60 DATA #A9,#03,#8D,#FC,#03,#60
70 REM APPEL DE LA ROUTINE
80 CALL #0400
100 FOR N= #0400 TO #0432
110 PRINTHEX$(PEEK(N));
120 NEXT

```

```

50 L'ordinateur lit le 1er data
55 Ping (ce n'est pas indispensable)
60 Va le mettre en donnée : 1
70 Attend une seconde

```

- 80 Passe au "N" suivant
- 90 Les datas suivent les "poids" binaires des bits de sortie : 1, 2, 4, 8, 16, 32, 64, 128
- 100 RESTORE. Cette instruction est indispensable pour reprendre la pile des datas si l'on recommence le défilement.
Sur cette instruction, l'ordinateur affichera "OUT OF DATA" : tous les datas ont été utilisés.

Si l'on veut constituer une suite d'allumage montant et descendant, on modifiera le programme selon la forme 1, 2, 4, 8, 16, 32, 64, 128, 64, 32 ...

Il faudra évidemment modifier la ligne 40 en conséquence : N = 0 TO 14.

On s'exercera à allumer les deux premières puis les ampoules 3 et 4 ... en essayant de comprendre ce qui se passe.

Un allumage "aléatoire" est tout-à-fait possible : le programme devient :

```
40 FOR A = INT(RND(1)*255)
50 POKE 1020,A
60 WAIT 25
70 GOTO 40
```

La mise en route de relais s'effectue de la même façon. Un "1" sur une sortie du PIA correspond au collage des relais. Nous verrons plus loin que la réponse du relais peut-être conditionnée par l'état d'un port mis en entrée.

Si nous voulons faire entrer en action le troisième relais du port B (PB2 à "1"), nous aurons le programme suivant :

```
10 POKE 1023,0
20 POKE 1022,255
30 POKE 1023,4
40 POKE 1022,1
```

Pour plus de rapidité dans l'exécution des ordres, il pourra être nécessaire d'exécuter les ordres en langage machine. Ce n'est d'ailleurs pas nécessaire pour l'allumage rapide des lampes ou l'action des relais, mais pour des applications d'automatismes complexes.

C'est le rôle de la routine écrite à partir de la ligne 50 en datas : Elle est mise dans l'ORIC à l'adresse de départ 0400 et appelée pour exécution par l'instruction CALL en ligne 80.

On notera l'équivalence des instructions :

A0	00	8D	FD	03	POKE 1021,0
A9	FF	8D	FC	03	POKE 1020,255
A9	04	8D	FD	03	POKE 1021,4
A9	03	8D	FC	03	POKE 1020,3

60 est l'instruction de retour de programme (RTS).

On pourra consulter les pages finales du manuel de l'ORIC pour plus de précisions ou, mieux, l'ouvrage de Rodney Zacks sur le 6502 (Sybex).

INTERFACE DE PUISSANCE/SECTEUR 220 VOLTS

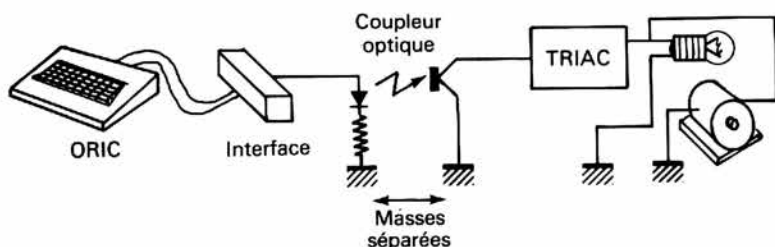
La carte d'ampoules clignotantes sera employée principalement pour "se faire la main" sur les procédés de programmation. On pourrait constituer une matrice lumineuse avec de nombreuses lampes pouvant autoriser le défilement de lettres publicitaires, changement de couleurs ... défilement de texte. L'imagination du lecteur fera le reste, tant est grande la possibilité de gestion par ordinateur.

Dès que nous désirons passer à la régulation de puissances beaucoup plus importantes on aura obligatoirement recours, soit à la carte RELAIS vue plus haut, soit à un pilotage par TRIACS pour des applications industrielles par exemple.

C'est la raison pour laquelle nous allons décrire un module d'extension à huit sorties pour 220 Volts :

La puissance autorisée sera d'environ 1 kW par voie, mais il est toutefois possible d'adjoindre des TRIACS beaucoup plus puissants selon les besoins de l'application en prévoyant également des conditions de refroidissement par radiateurs adéquats.

La question essentielle dans ce genre de montage est l'ISOLEMENT. Il n'est en effet pas possible de tolérer des retours du 220 Volts secteur sur la carte d'interface de commande, pas plus que sur l'ORIC. Pour cela on emploie une méthode d'isolement extrêmement efficace : le couplage OPTIQUE.



Il existe couramment sur le marché des circuits intégrés opérant une liaison de ce genre à l'intérieur de la puce elle-même et qui procurent des isollements de l'ordre de 2 000 à 7 000 Volts entre la carte d'interface et l'utilisation 220 Volts. Ce sera suffisant ! Pourvu qu'on ait pris soin de dissocier les plans de masse sur la carte du circuit imprimé...

Le principe de fonctionnement est simple :

Un transistor amplificateur permet d'attaquer un coupleur optique double : un HCPL2531.

En sortie un transistor du type 2N2222 attaque la gachette d'un TRIAC. Chacune des voies est répétée 8 fois en sachant qu'un boîtier de coupleur 2531 comporte deux unités optiques.

RÉALISATION PRATIQUE

Le montage aura intérêt à être dans un boîtier plastique. Comme dit plus haut, les masses sont totalement séparées entre entrées et sorties 220 Volts.

Dans notre réalisation nous avons effectué les sorties d'utilisation sur une barrette de dominos d'électricien : huit sorties indépendantes : triacs, quatre points communs pour le raccordement.

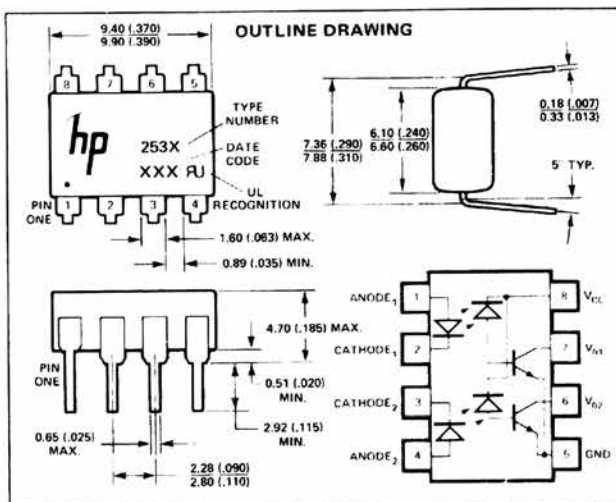
L'optocoupleur, ainsi que le 2N2222, réclament un tension d'alimentation continue entre 5 et 10 Volts. Il aurait été possible de l'obtenir à partir d'un redressement simplifié du 220 Volts.

Nous avons préféré cependant d'utiliser une technique plus "stable" avec transformateur séparé. Elle est d'ailleurs simplissime : un adaptateur 6,3 Volts, un redresseur et une capacité d'environ 2 200 uF.

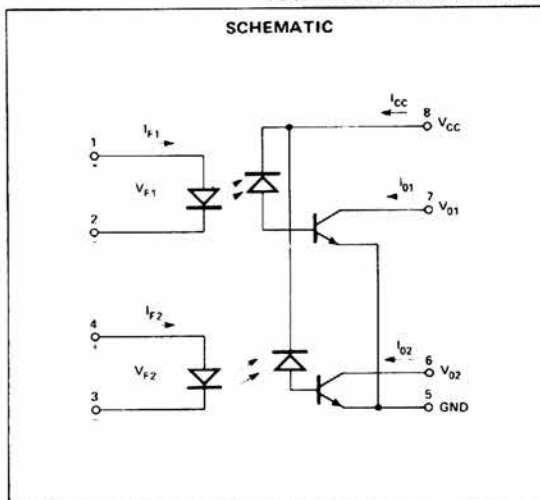
Principaux composants :

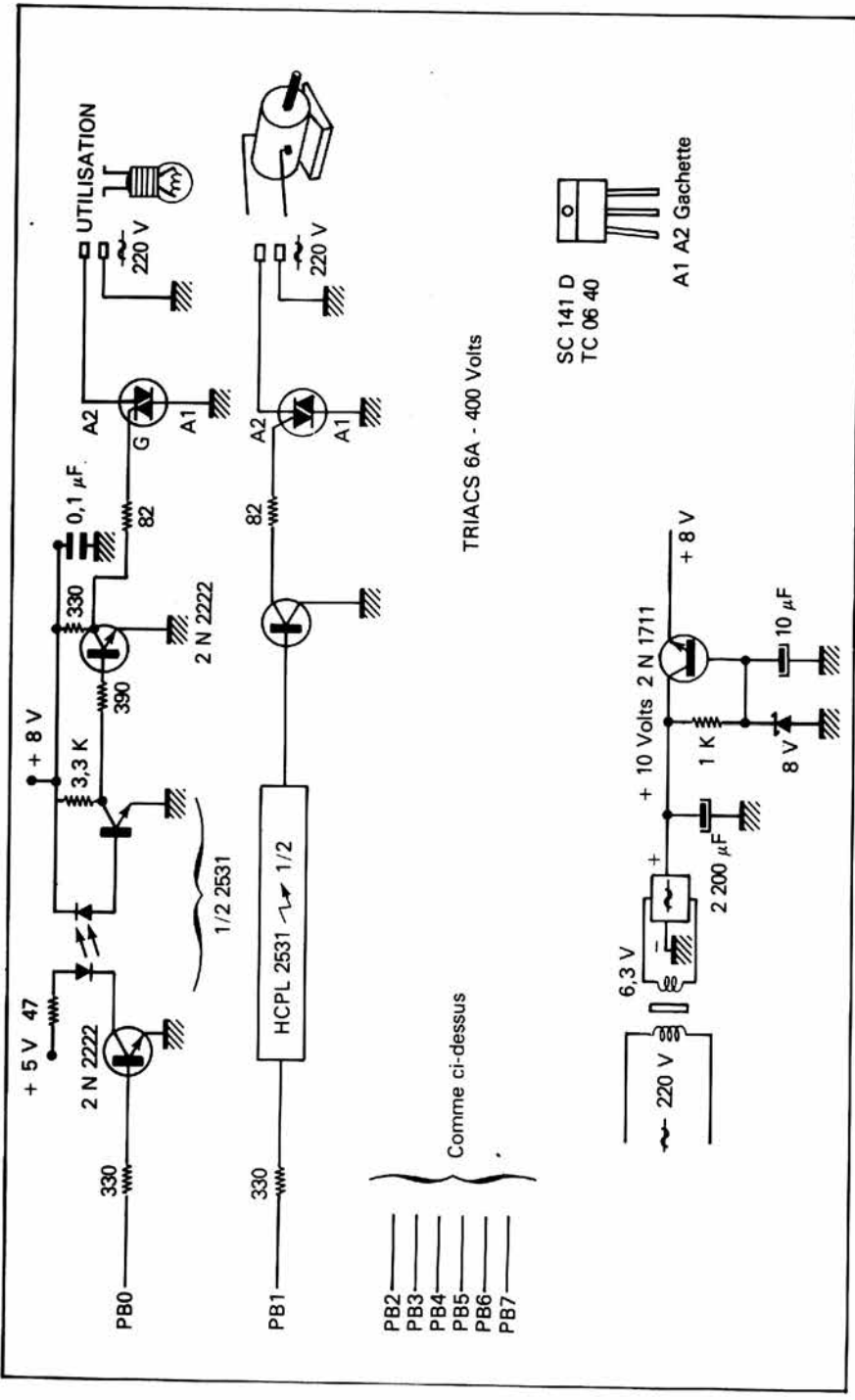
- 16 transistors 2N2222,

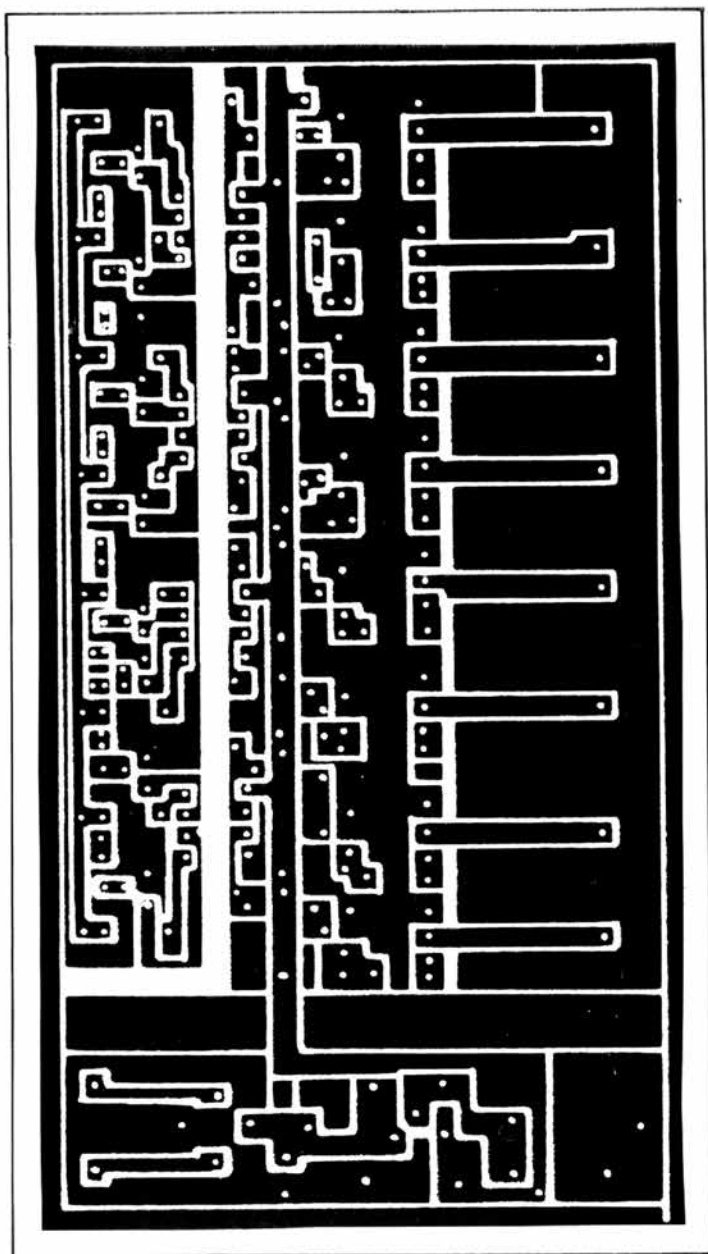
- 8 triacs 6 Ampères, 400 Volts,
- 4 optocoupleurs HCPL 2531 Hewlett Packard,
- 16 résistances 330 Ohms,
- 8 résistances 3,3 kOhms,
- 8 résistances 82 Ohms,
- 1 transformateur 220 V/6,3 V, 300 mA,
- 3 condensateurs de découplage 0,1 uF,
- 1 condensateur 2 200 uF, 16 V.

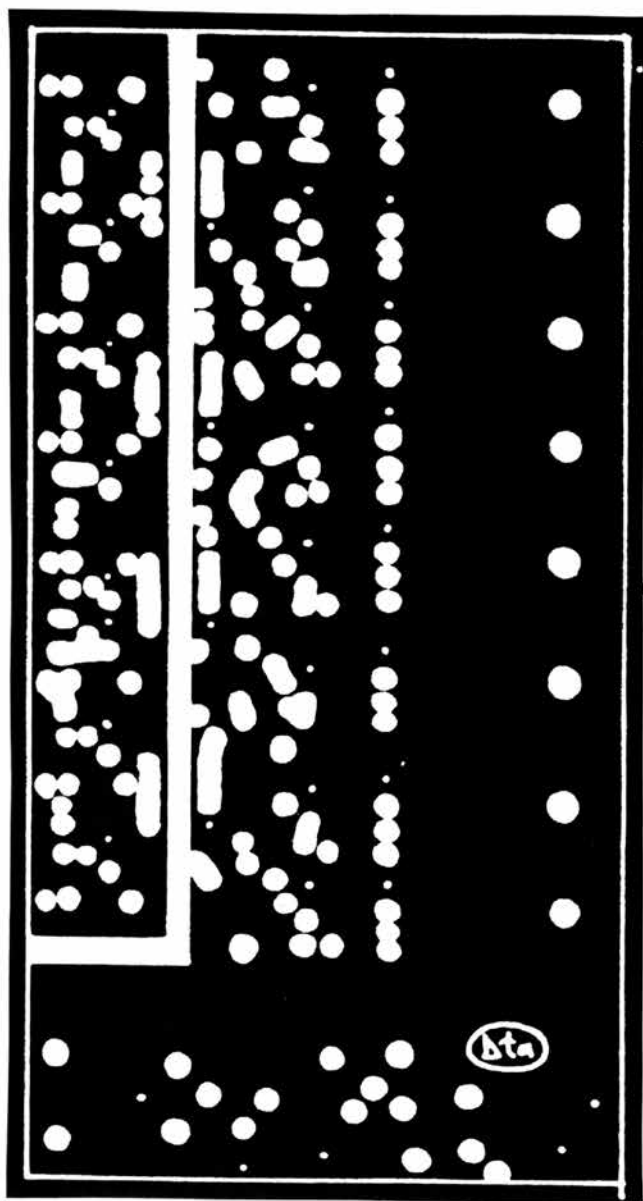


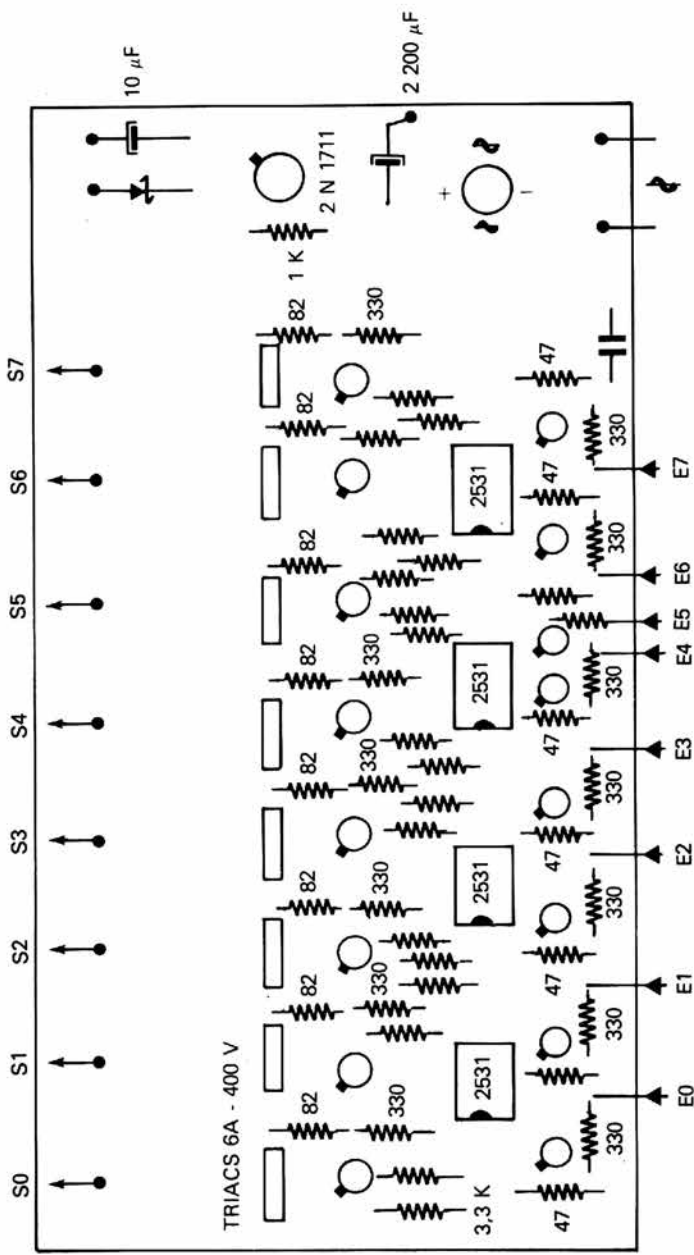
TECHNICAL DATA APRIL 1978

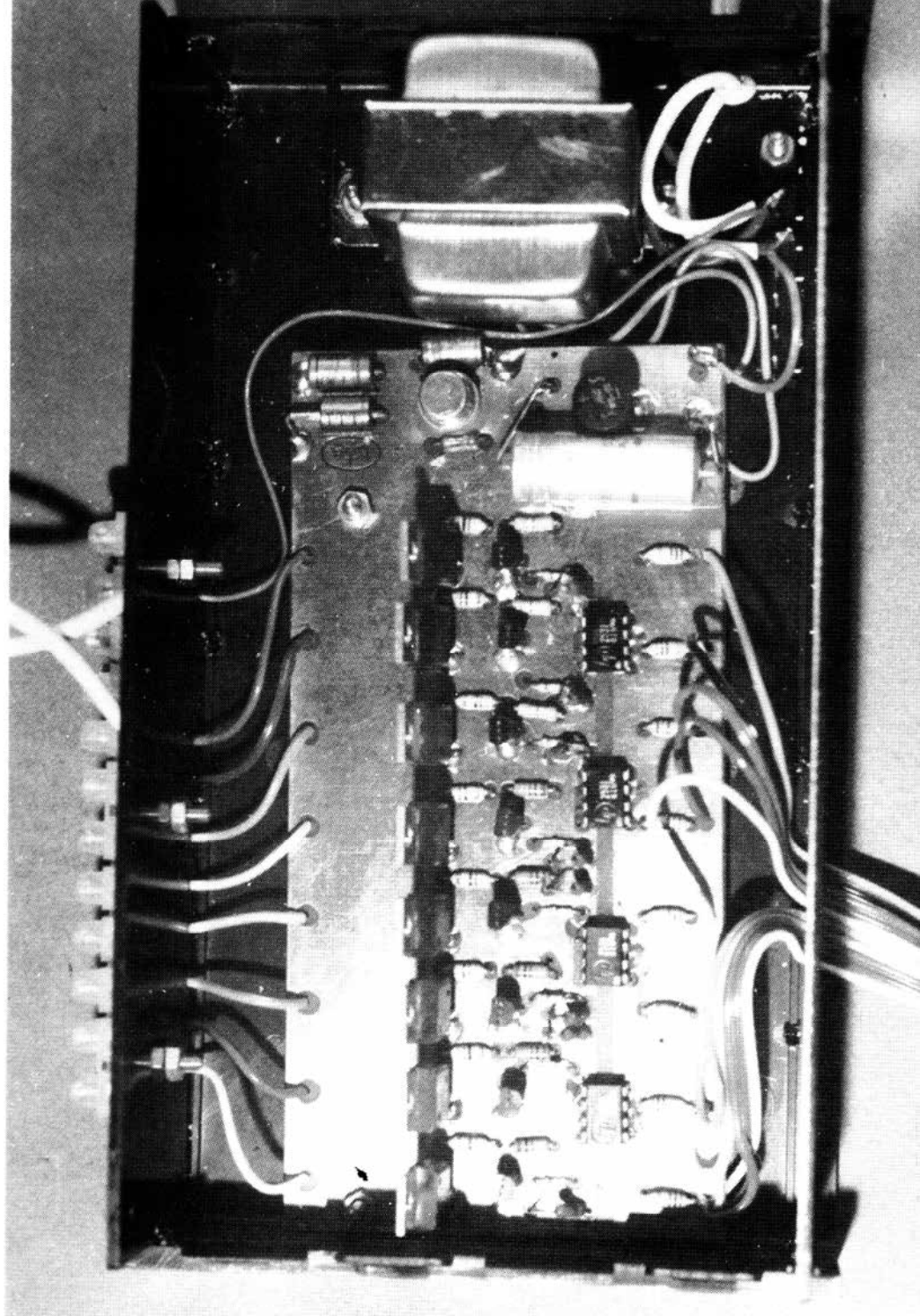












CHAPITRE V

LA PROGRAMMATION FACILE DES SYNTHÉTISEURS DE FRÉQUENCES

On sait qu'un certain nombre de générateurs basse fréquence, haute fréquence ou très haute fréquence sont asservis à une référence à quartz et qu'il est possible de faire varier leur accord par l'entremise de poids binaires sur les entrées de diviseurs programmables.

Il est ainsi possible de faire varier la fréquence de sortie d'un oscillateur de ce type, par bonds de 5 kHz, par exemple, sur une étendue de 30 MHz : ce sera le cas d'un récepteur du type "Aviation".

La grosse difficulté réside habituellement à donner une relation compréhensible par l'utilisateur entre l'affichage des poids binaires du synthétiseur et la lecture de la fréquence d'utilisation.

Toutes les capacités d'un ordinateur se trouvent bien mises à profit dans une telle configuration.

Il pourra, à partir de l'écriture sur l'écran de la fréquence demandée, convertir en binaire sur les sorties d'interface les ordres nécessaires. De cette façon également on pourra très facilement procéder à des mises en mémoire de fréquences prioritaires, au scanning de portions de bande pour la surveillance de fréquences, au déclenchement de routines d'enregistrement si telles fréquences sont modulées par un utilisateur précis ... les limites seront celles de l'imagination du réalisateur.

Un certain nombre de stations radioamateurs sont de plus en plus équipées de cette façon et de grandes marques commerciales de transceivers (émetteurs-récepteurs) ont une prise prévue pour ce type de raccordement informatisé.

Le confort d'une telle gestion par microprocesseur est indéniable puisqu'il permet en même temps la gestion du carnet de trafic, la sortie sur imprimante des contacts effectués et, éventuellement, le décodage automatique des signaux Morse ou Télétype (RTTY) !

Un point important est à considérer dans l'établissement d'un tel projet : le niveau de parasites et de spurious générés par le système ordinateur sur le récepteur de trafic dans les divers emplois que nous avons réalisés l'ORIC nous a donné toute satisfaction et les quelques raies parasites ne sont pas gênantes, pourvu qu'on utilise une bonne antenne extérieure reliée à l'appareil par un câble coaxial sérieux.

Nous donnerons, sans nous étendre de trop sur un sujet qui nous passionne par ailleurs, un exemple de réalisation simple dans le domaine de la basse fréquence.

Les lecteurs désireux d'approfondir la question pourront se reporter à la bibliographie en fin d'ouvrage.

PILOTAGE D'UN GÉNÉRATEUR BF DE 10 HZ A 1 MHZ

Il s'agit d'une platine très simple que pourront même réaliser ceux qui n'ont pas une grande habitude des circuits électroniques étant donné son absence totale de mise au point.

Un certain soin sera demandé dans l'exécution car il s'agit de circuits de technologie C-MOS, donc sensibles aux courants de fuite d'un fer à souder mal isolé. On veillera également à la bonne orientation des circuits intégrés. Cela nous paraît élémentaire.

C'est un générateur très stable dans la gamme des 10 Hz à 1 MHz, en signaux carrés (ou sinusoïdaux par l'adjonction de filtres à forte atténuation en passe-bas du type Reticon R5609).

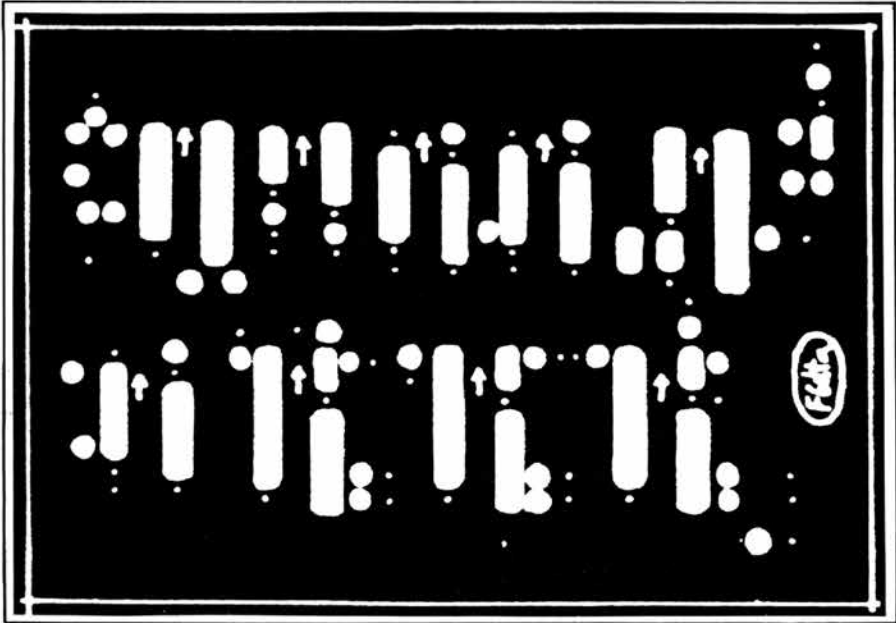
Il peut produire :

- 10 Hz à 1 kHz au pas de 1 Hz,
- 100 Hz à 10 kHz au pas de 10 Hz,
- 1 kHz à 100 kHz au pas de 100 Hz,
- 10 kHz à 1 MHz au pas de 1 kHz.

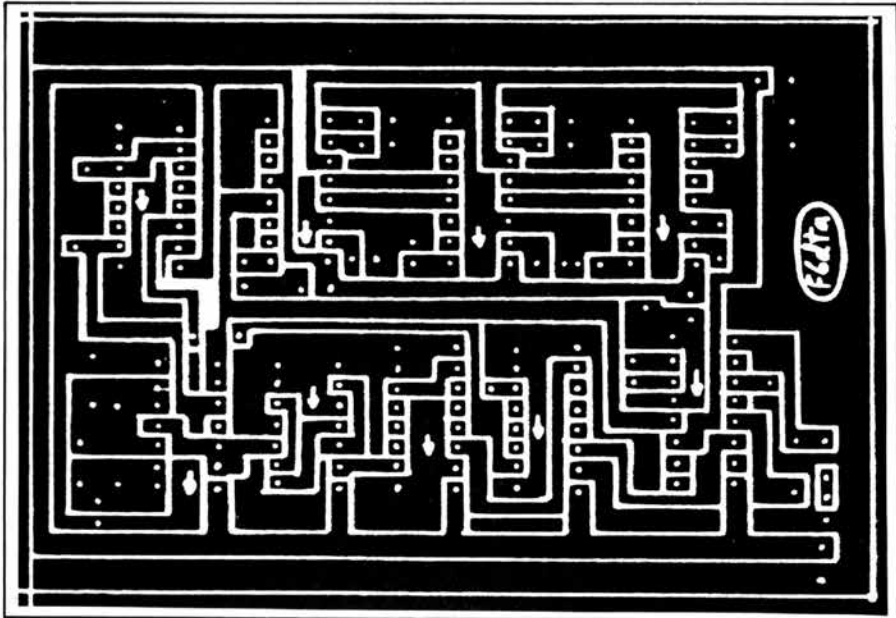
Les emplois d'un générateur BF sont multiples dans un laboratoire amateur ou professionnel : mesure de bande passante, étalonnages divers, modulation d'un émetteur pour réglages, etc...

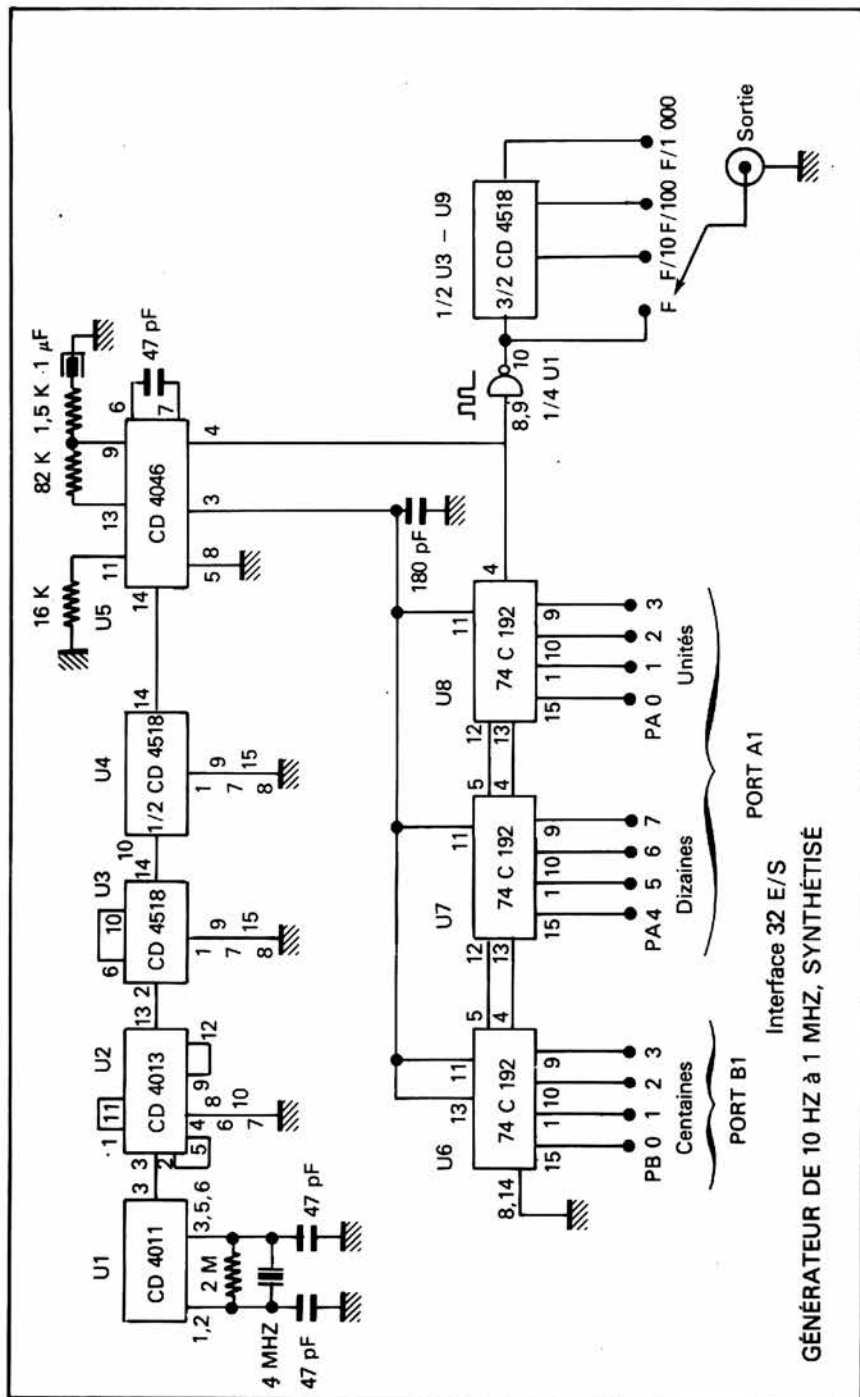
Le fait de réaliser cet ensemble sous forme synthétisée offre plusieurs avantages :

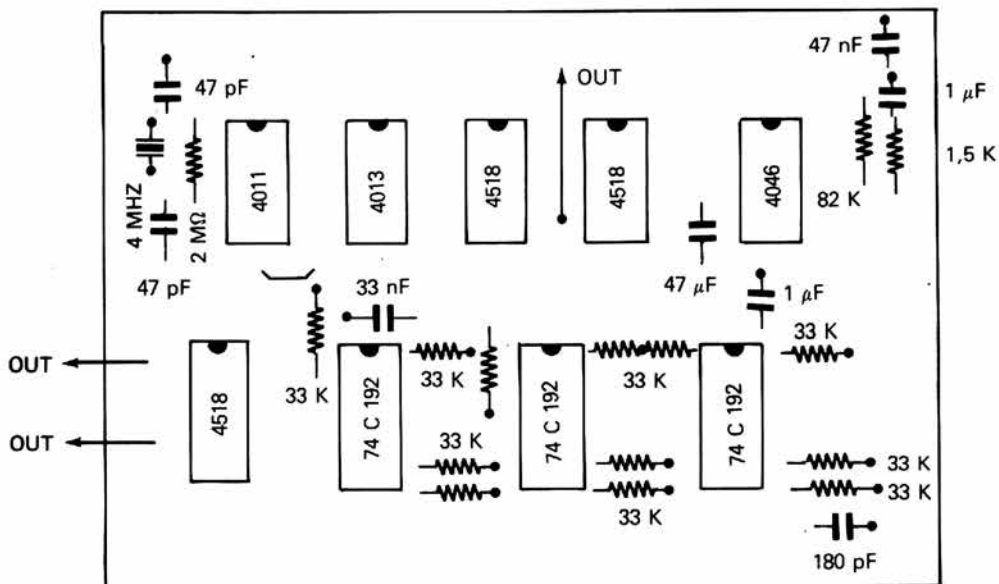
- grande stabilité en fréquence, due à un oscillateur (VCO) verrouillé sur un quartz étalon ;
- précision de la lecture au Hertz près ;
- balayage de plages de fréquences très précis ;
- pilotage très aisé par un micro-ordinateur doté d'une interface appropriée - nous retrouvons l'utilisation de l'ORIC.



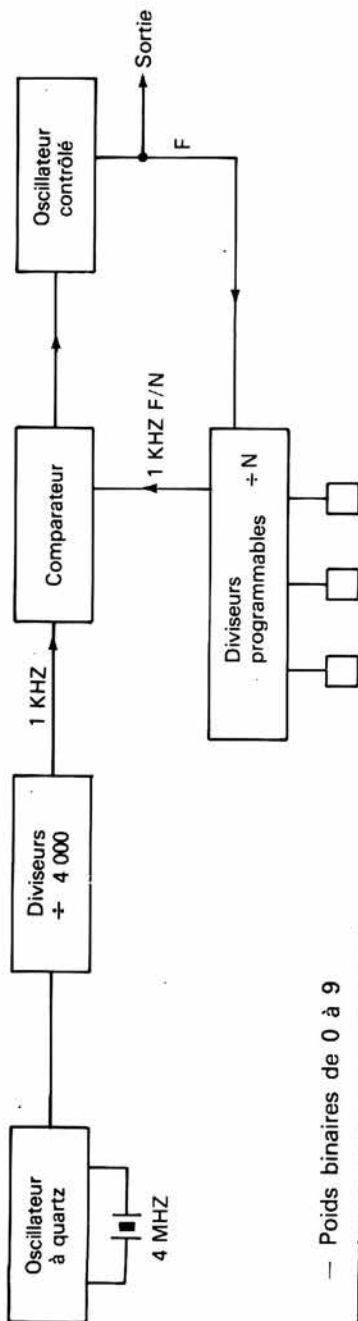
GÉNÉRATEUR SYNTHÉTISÉ 4046 - 74C192







Implantation Générateur 10 HZ, 1 MHz



— Poids binaires de 0 à 9

	A	B	C	A
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1

SOUS-GAMMES

- 10 KHZ à 1 MHz au pas de 1 KHZ
- 10 KHZ à 100 KHZ au pas de 100 HZ
- 100 HZ à 10 KHZ au pas de 10 HZ
- 10 HZ à KHZ au pas de 1 HZ
- 1 KHZ

PROGRAMMATION DE LA FRÉQUENCE

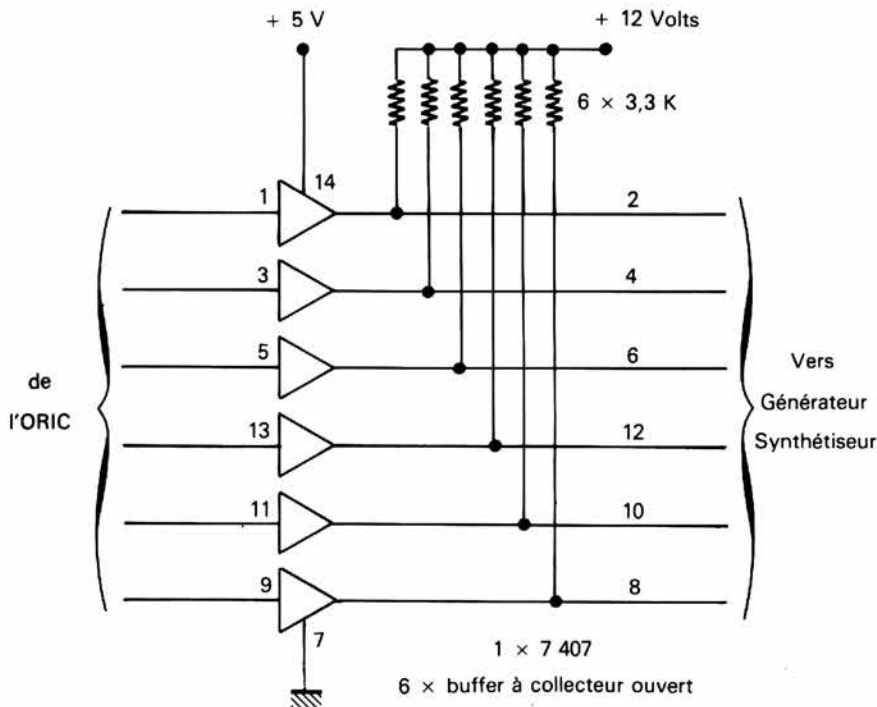
Les diviseurs programmables sont au nombre de trois et permettent une division de 10 à 999.

Leur "programmation" est réalisée par des niveaux hauts ou bas sur leurs entrées (broches 15, 1, 10 et 9) qui correspondent respectivement aux poids binaires 1, 2, 4, et 8.

Quatre interrupteurs peuvent remplir ce rôle pour chaque compteur, mais il est évidemment beaucoup plus confortable de mettre au travail l'ORIC avec son interface.

Deux solutions sont possibles sur le plan pratique et concernent des problèmes d'alimentation et de couverture en fréquences.

- Alimenter la platine du générateur en 5 Volts, autrement dit à la même tension que celle des ports de sortie du PIA. Pas de problème de compatibilité de niveaux, mais il sera difficile d'obtenir toute la plage de fréquences que nous avons annoncée.
- Alimenter le générateur en 12 Volts. Dans ce cas il faudra utiliser deux circuits intégrés supplémentaires du genre 7407 à collecteurs ouverts selon le schéma très simple suivant :



La commutation de gammes peut évidemment se faire de façon totalement automatique par les quatre ports restants sur PB4, PB5, PB6 et PB7.

PROGRAMME DE GESTION

Il permet d'accomplir les fonctions suivantes :

- mise en route d'une fréquence,
- balayage d'un point à un autre (scanning),
- prise en compte de quatre fréquences mémorisées.

Les adresses utilisées correspondent à l'interface 32 entrées/sorties que nous décrivons en fin d'ouvrage :

adresse 1016 : PORT A 1

1018 : PORT B 1

Les fréquences admissibles se situent entre 3 et 999 kHz. Les deux chiffres de droite sont stockés sur le Port A, le chiffre de gauche sur le port B (quatre bits seulement).

On remarquera à la ligne 20 que le nombre décimal est converti en hexadécimal afin de conserver les poids convenables sur les huit bits de A.

Un appui sur la lettre "E" fait revenir au programme principal.

Le scanning s'opère en ligne 100 : fréquence basse, fréquence haute, valeur d'incrémententation.

De même, en 200, nous avons la prise en compte de quatre fréquences. Il sera évidemment très facile d'en augmenter la capacité !

La routine en 1000 effectue la conversion affichage/poids binaires sur les ports A et B.

SYNTHÉTISEUR VHF 144-146 MHZ

Nous donnons ci-après un exemple simple de gestion d'un transceiver VHF avec la conversion fréquence : poids binaires.

On constatera le peu de lignes de programmation pour fournir un calcul qui serait, sans cela, très complexe.

Il permet, en outre, l'incrémententation (en plus ou en moins) de 25 kHz :

Touche "L" = +25 kHz

Touche "M" = -25 kHz

Un seul port est utilisé : port A de l'interface 16 E/S. Les caractéristiques du transceiver utilisé sont les suivantes : 144 à 146 MHz.

```

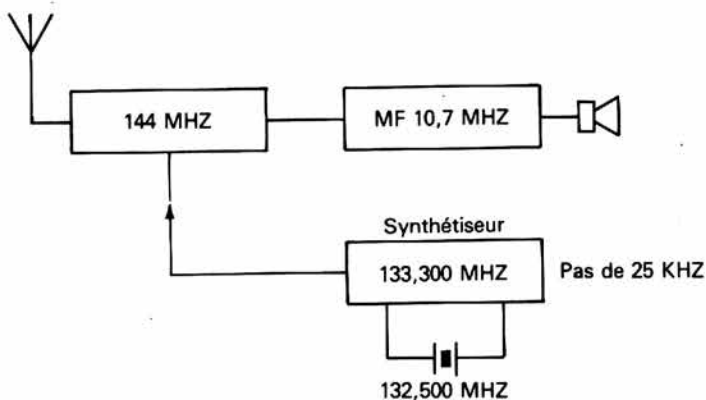
2 GOTO 500
5 CLS
10 PRINT "FREQUENCE ?"
15 INPUT A$
16 G$=RIGHT$(A$,2)
20 A=VAL("#"+G$) : POKE 1016,A
25 PRINT : PRINT : PRINT
30 PRINT A$;"");"KHZ"
35 PRINT : PRINT : PRINT
40 B=(VAL(A$)-VAL(G$))/100 : POKE 1018,B
50 IF A$="E" THEN 2
60 GOTO 10
100 REM ----- SCANNING -----
110 PRINT "FREQUENCE BASSE"
120 INPUT H
130 PRINT "FREQUENCE HAUTE"
140 INPUT I
150 PRINT "VALEUR DU PAS"
155 INPUT P
170 D=H
180 D=D+P
185 IF KEY$="E" THEN 2
190 IF D=>I THEN 170
192 A$=STR$(D) : GOSUB 1000
195 WAIT 25 : GOTO 180
200 REM ----- MISE EN MEMOIRE -----
210 PRINT "FREQ.1" : INPUT F1
220 PRINT "FREQ.2" : INPUT F2
230 PRINT "FREQ.3" : INPUT F3
240 PRINT "FREQ.4" : INPUT F4
250 CLS : PRINT : PRINT : PRINT : PRINT
260 PRINT "FREQUENCE 1" : A$=STR$(F1) :
  GOSUB 1000
265 GET S$
270 PRINT "FREQUENCE 2" : A$=STR$(F2) :
  GOSUB1000
272 GET S$
275 PRINT "FREQUENCE 3" : A$=STR$(F3) :
  GOSUB 1000
280 GET S$

```

```

285 PRINT "FREQUENCE 4" : A#=STR$(F4) :
    GOSUB 1000
286 GET S$
290 IF S#="E" THEN GOTO 2
295 GOTO 250
495 END
500 CLS : PAPER 3 : INK 1
501 POKE 1017,0 : POKE 1016,255 :
    POKE 1017,4 : POKE 1016,0
502 POKE 1019,0 : POKE 1018,255 :
    POKE 1019,4 : POKE 1018,5
505 PRINT : PRINT : PRINT : PRINT
510 PRINT "/////////////////////"
515 PRINT "    1 / FREQUENCE                "
520 PRINT "    2 / SCANNING                "
525 PRINT "    3 / MEMOIRES                "
530 PRINT "/////////////////////"
535 GET X$
540 X=VAL(X$) : ON X GOTO 10,100,200
1000 G#=RIGHT$(A#,2)
1020 A=VAL("#"+G$) : POKE 1016,A
1030 PRINT A$;"")"KHZ"
1040 B=(VAL(A$)-VAL(G$))/100 : POKE 1018,B
1060 RETURN
1070 REM ==> COPYRIGHT MICHEL LEVREL
1080 REM --- MARS 1984

```

Moyenne fréquence 10,7 MHz. Oscillateur quartz 132,5 MHz. Pas de 25 kHz.

Le synoptique est le suivant : Fig. A.

Il pourra être adapté à tout autre cas bien sûr.

On obtient de ce fait l'affichage immédiat de la fréquence avec un confort ... total.

La ligne 85 est indispensable pour compenser l'arrondi automatique de l'ORIC à la valeur inférieure, sans quoi les incréments ne se font plus que tous les deux pas.

On pourra reprendre les données du programme précédent pour y adjoindre un scanning, des mémoires et, en utilisant l'une des lignes du port B, réaliser un arrêt automatique du balayage sur une fréquence occupée.

Par appel d'un fichier, on constituera en même temps la gestion des cartes QSL, le calcul des distances entre deux stations et le décodage automatique des signaux MORSE ! Toutes choses peu prévisibles il y a dix ans.

```
5 REM SYNTHETISEUR UHF ; Moyenne freque
nce 10.7Mhz;Pas de 25 KHz;Oscill>quartz
10 POKE 1021,0
```

```

20 POKE 1020,255
30 POKE 1021,4
40 POKE 1020,0
50 CLS:PRINT:PRINT:PRINT:PRINT
60 INPUT "FREQUENCE? (en MHz)";A$
70 F= VAL(A$)-10.7
80 U = (F-132.5)/0.025
85 IF U>(INT(U)+0.9)THEN U=U+1
90 N$="#" +STR$(U):N=VAL(N$)
100 POKE 1020,N
105 GET B$
106 IF B$="L" THEN A$=STR$(VAL(A$)+0.02
5):PRINTA$ :GOTO70
107 IF B$="M" THEN A$=STR$(VAL(A$)-0.02
5):PRINTA$ :GOTO70
110 GOTO60
120 REM Copyright M.LEUREL F6DTA.

```

CHAPITRE VI

TECHNIQUES DE COMMANDES D'AUTOMATISMES

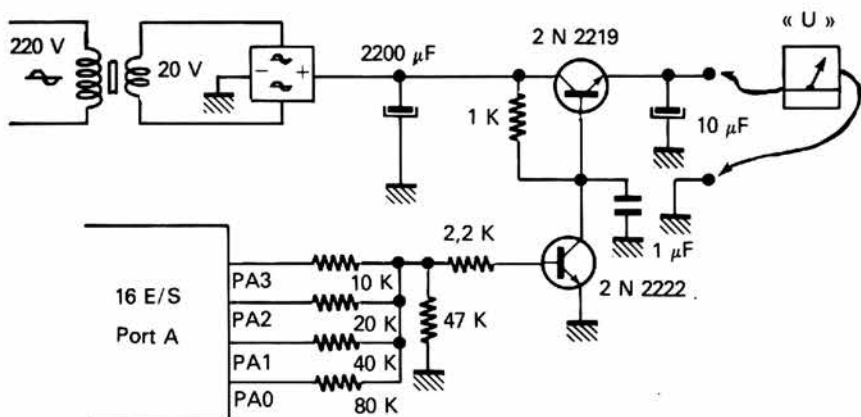
Nous avons vu dans les chapitres précédents comment piloter des systèmes d'ampoules, des triacs pour des montages sur le secteur ou gérer les commandes d'un récepteur ou transceiver synthétisé. Il est bien sûr possible d'étendre les possibilités d'automatismes à d'autres domaines divers. Nous allons faire l'exploration de quelques unes d'entre-elles.

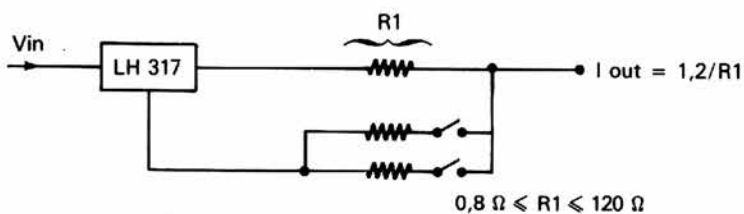
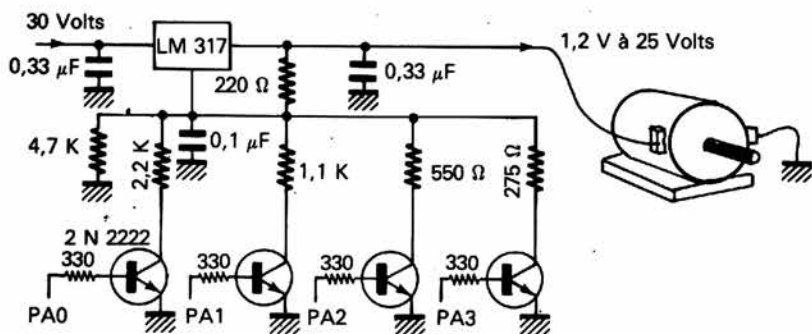
ALIMENTATION PROGRAMMABLE

Les possibilités de contrôle de la tension d'alimentation et /ou de l'intensité débitée deviennent très faciles. Elles permettent l'ajustement en fonction des conditions d'exploitation, le contrôle de la vitesse de moteurs à courant continu ou la coupure brutale de la tension (electronic shutdown).

Pour cela il est possible d'utiliser le pouvoir de coupure de petits relais ou, plus simplement, des transistors de commutation. On utilisera de préférence dans cette fonction des régulateurs intégrés dans un boîtier TO3 ou TO220 permettant déjà à eux seuls de fournir plus d'un Ampère. Il n'y a que l'embaras du choix :

LM309 (5 Volts et au-delà), LM317 (1,2 V à 25 Volts),... Consulter les DATA de National Semiconductors par exemple.





PROGRAMME DE GESTION

Nous utilisons le PORT A de la carte 16 Entrées/Sorties.

Lignes 10-30 : toutes les lignes sont des sorties ;
40 : variation toutes les secondes de la valeur mise à l'adresse 1020 ;
45 : écriture du nombre N ;
80 : retour en début de programme.

```
5 REM VARIATION PROGRÈSSIVE DE LA TENSIO
N DE SORTIE
10 POKE 1021,0
20 POKE 1020,255
30 POKE 1021,4
40 FOR N=0 TO 15
45 PRINTN;
50 POKE 1020,N
60 WAIT 100
70 NEXT N
80 GOTO 40
```

Le programme suivant permet la commande de la vitesse d'un moteur par examen du clavier.
Entrer un nombre entre 0 (vitesse maximale) et 9 (vitesse minimale).
Un appui sur "E" (End) provoque l'arrêt du programme.

```
5 REM VARIATION VITESSE MOTEUR AVEC EXAM
EN DU CLAVIER
10 POKE 1021,0
20 POKE 1020,255
30 POKE 1021,4
40 POKE 1020,8
45 PRINTN;
50 GET A$
60 IF A$="E" THEN END
70 A = VAL(A$)
```

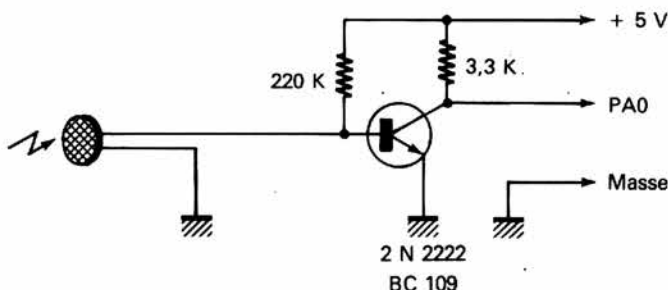
80 POKE 1020, A
90 PRINT A;
100 GOTO 50

PROCÉDÉS PHOTOSENSIBLES

Il sera très souvent utile, pour des opérations de comptage d'objets, de déclenchement sélectif d'alarmes, d'utiliser un procédé photosensible : lumière du jour, infrarouge...

Nous proposons une manipulation très intéressante à partir d'une photorésistance qui pourra être indifféremment du type LDR03, LDR05, MPY12.

Le montage est le suivant :



La tension sur le collecteur du transistor sera sensiblement égale à 5 Volts lorsque la cellule est éclairée. Elle passe à 0 dans l'obscurité. Si nous relierons directement le collecteur à PA0 de la platine 16 E/S, nous pourrions lire l'état du port A par l'instruction PEEK(1020), les initialisations étant faites par les lignes 10 à 30.

Ligne 10 : accès au registre de direction,
Ligne 20 : toutes les lignes sont des entrées,
Ligne 30 : accès au registre des données.

Le port A possédant des résistances pull-up (voir ch. IV), lorsque la cellule sera éclairée nous lirons 255 (FF) sur le port en entrées. Si le poids PA0 passe à 0, nous devrions lire $255 - 1 = 254$.

Tous les dispositifs d'alarme ou de surveillance sont ainsi possibles par détection du "254".

Alarme sonore.

Alarme graphique avec identification de l'entrée en action et, éventuellement, consignes et directives. Les possibilités sont sans limites avec 256 combinaisons, depuis aucune alarme jusqu'à toutes les alarmes en route !

```
10 POKE 1021,0
20 POKE 1020,0
30 POKE 1021,4
40 A:=PEEK(1020)
41 PRINTA:
45 IF A=254 THEN GOTO 100
60 GOTO 40
100 HIRES:PAPER3:INK 1
110 CURSET 120,100,1
120 FOR N=1 TO 5
130 CIRCLE 18*N,1
140 NEXT N
150 GOTO 40
```

PROGRAMME ÉLÉMENTAIRE DE SIRÈNE

pouvant être appelé par une instruction GOTO ou GOSUB :

```
10 REM //SIRENE\\
20 FOR N=3 TO 5
30 FOR A=1 TO 12
40 MUSIC 1,N,A,15
60 NEXT A,N
70 GOTO 20
```

ALARME GRAPHIQUE

avec identification :

"ALARME POSTE 5"

avec éventuellement consignes d'appel téléphonique par exemple.

```
10 POKE 1021,0
20 POKE 1020,0
30 POKE 1021,4
40 A=PEEK(1020)
50 PRINTA;
60 IF A=254 THEN GOTO 100
70 GOTO 40
100 CLS :PAPER3:INK0
110 PLOT 10,15,"ALARME POSTE 5"
120 PING:WAIT 75
130 CLS:GOTO 40
```

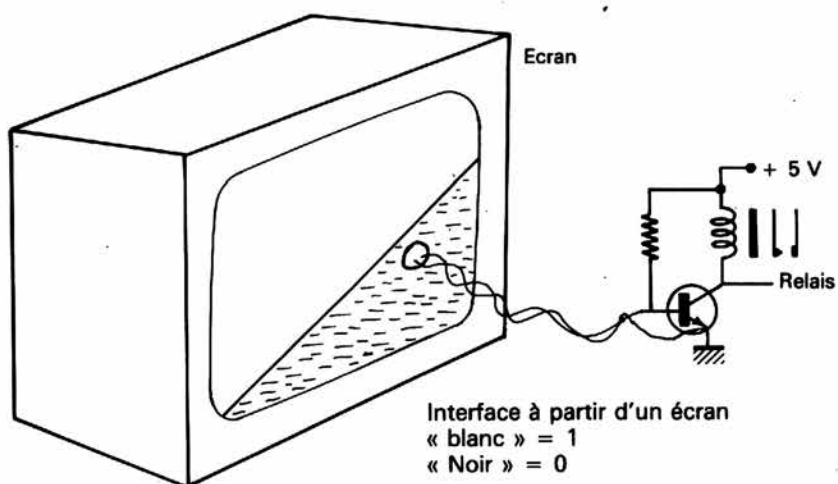
5 REM // DETECTION PAR PHOTORESISTANCE\\

```
10 POKE 1021,0
20 POKE 1020,0
30 POKE 1021,4
40~A= PEEK(1020)
41 PRINTA;
45 IF A=254 THEN GOTO 100
60 GOTO 40
90 REM ALARME
100 FOR N=3 TO 5
105 PLAY 1,0,0,0
110 FOR A=1 TO 12
120 MUSIC 1,N,A,15
130 NEXTA,N
135 PLAY0,0,0,0
140 GOTO 40
```


Un domaine d'application à exploiter pourrait être très pratique : nous avons mis la même photorésistance retenue par du sparadrap sur l'écran TV. En noircissant totalement l'écran par un PAPER 0, il est facile de faire apparaître un cube blanc par le petit programme suivant :

10 LORES 0 : PAPER 0
20 PLOT 15,10,23

Lorsque le cube se déplace, on remarque qu'à son passage devant la cellule, le niveau de PA0 passe à 1 tandis qu'il est à 0 dans les autres cas. Nous pourrions imaginer une détection de niveau du noir par rapport au blanc sur l'écran, réaliser une interface très simple pour la manipulation d'un émetteur si le cube est modulé en durée courte (point) et en durée longue (trait), etc... Il suffirait de remplacer la charge de 3,3 k du 2N2222 par un relais miniature : c'est une suggestion...



POIGNÉE DE TÉLÉCOMMANDE

Poignée de jeux (Joystick)
Poignée de dessin

Dès que nous sommes en possession d'une interface avec possibilité de prendre en compte ce qui se passe à ses entrées, il nous sera possible d'effectuer des télécommandes diverses, soit pour mener à

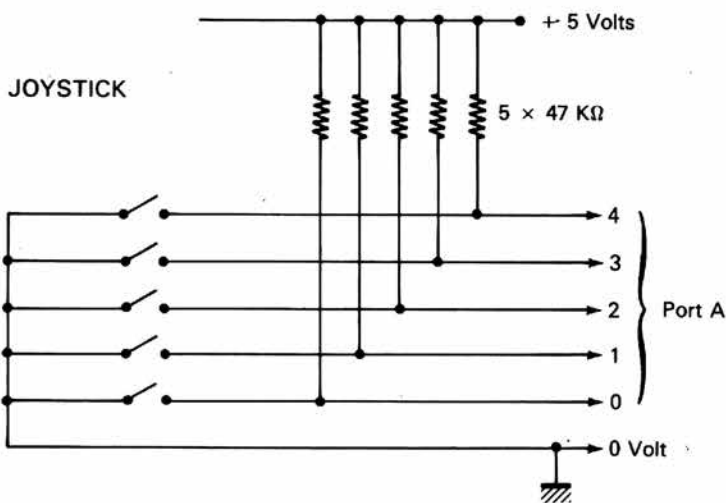
bien des automatismes mécaniques, soit pour les jeux d'arcades et surtout l'établissement de dessins en mode HIRES (haute résolution).

Nous avons à notre disposition une poignée du type ATARI : c'est celle-là que nous décrivons.

Les liaisons sont très simples. Les résistances pull-up ne sont pas nécessaires si l'on utilise la lecture du port A mais elles sont incluses dans la poignée et, dans tous les cas, ne gênent pas (5 résistances de 47 kOhms).

Elle possède huit degrés de liberté et un "fire" (bouton rouge).

Sept fils de liaison sont nécessaires : deux pour l'alimentation (masse - 5 Volts), les cinq autres concernent les lignes du PIA (PA0 à PA4).



- Bit 0 : gauche
- Bit 1 : droite
- Bit 2 : haut
- Bit 3 : bas
- Bit 4 : Fire : Zap

Le bit 4 « Fire » est programmé pour effacer le dessin.

Utilisation en poignée de dessin :

L'utilisation en poignée de dessin permet de tracer pratiquement ce que l'on veut sur l'écran avec une grande rapidité.

C'est la technique idéale pour les plus jeunes qui n'ont même pas besoin de connaître le BASIC.

```
1  REM *** POIGNEES DE JEUX ***
2  REM ***   POIGNEE       ***
3  REM ***   DE DESSIN    ***
5  HIRES : PAPER 3 : INK 4
6  X=100 : Y=100
10 POKE 1021,0
20 POKE 1020,0
25 POKE 1021,4
30 A=PEEK(1020)
40 CURSET X,Y,1
50 IF A=253 THEN X=X+1
60 IF A=254 THEN X=X-1
70 IF A=251 THEN Y=Y-1
80 IF A=247 THEN Y=Y+1
90 IF A=249 THEN X=X+1 : Y=Y-1
100 IF A=245 THEN X=X+1 : Y=Y+1
110 IF A=246 THEN X=X-1 : Y=Y+1
120 IF A=250 THEN X=X-1 : Y=Y-1
130 IF A=239 THEN ZAP : GOTO 5
140 GOTO 30
```

Transformation de la poignée de jeux en "manipulateur électronique"

```
1  CLS : PRINT : PRINT : PRINT : PAPER 3 : INK 1
2  PRINT "MANIPULATEUR ELECTRONIQUE"
5  B=10
10 POKE 1021,0
```

```

20 POKE 1020,0
30 POKE 1021,4
40 A=PEEK(1020)
50 IF A=254 THEN GOSUB 100
60 IF A=253 THEN GOSUB 200
70 GOTO 10
100 PLAY 1,0,0,0
105 MUSIC 1,4,2,15
110 WAIT B
120 PLAY 0,0,0,0
130 RETURN
200 PLAY 1,0,0,0
205 MUSIC 1,4,2,15
210 WAIT B*3
220 PLAY 0,0,0,0
230 RETURN

```

Utilisation de la poignée en automatique

Il faudra, dans ce cas, prendre les informations de la poignée sur le port A, comme plus haut et POKER les instructions en SORTIES sur le port B, par exemple. Ce pourra être, par exemple, l'avancement par incrémentation successive d'un moteur pas à pas. L'élévation d'un bras d'un robot ... toutes les capacités de traitement et de conversion de l'ordinateur sont ici possibles : conversion linéaire/logarithmique : translation/rotation, etc...

Dans le programme précédent nous avons converti très simplement notre poignée de jeux en manipulateur générant des sons par exemple.

HORLOGE A QUARTZ

Le déclenchement d'automatisme peut se faire à partir d'une horloge ... constituée par le micro-ordinateur lui-même.

Nous devons dire que ce n'est pas une solution économique, mais ça marche !

```

1  REM =====
2  REM ** HORLOGE A QUARTZ **
3  REM ** PAR PASCAL LEVREL **
4  REM **      ET      **
5  REM ** MICHEL LEVREL **
6  REM =====
7  CLS : PRINT : PRINT : GOTO 1000
10 LORES 0
11 PAPER 0 : INK 3
12 REM -----
13 REM ** DESSIN DE L'HORLOGE **
14 REM -----
20 FOR N=0 TO 20
30 PLOT N+10,0,18
50 NEXT N
60 FOR N=0 TO 7
70 PLOT 30,N+8,18
80 NEXT N
90 FOR N=0 TO 20
100 PLOT N+10,15,18
110 NEXT N
120 FOR N=0 TO 7
125 PLOT 10,N+8,18
135 PLOT 14,17,CHR$(72)
136 PLOT 20,17,CHR$(77)
137 PLOT 26,17,CHR$(83)
138 PLOT 13,5,"HORLOGE A QUARTZ"
140 NEXT N
141 REM -----
142 REM ** CALCUL DE PASSAGE      **
143 REM ** DES SECONDES AUX      **
144 REM ** MINUTES ET AUX HEURES **
145 REM -----
150 FOR R=1 TO 50
160 PLOT 13,12,CHR$(R) : REM DIZAINE H.
170 FOR S=J TO 57
180 PLOT 15,12,CHR$(S) : REM HEURE
190 FOR T=K TO 53
200 PLOT 19,12,CHR$(T) : REM DIZAINE M.

```

```

210 FOR U=L TO 57
220 PLOT 21,12,CHR$(U) : REM MINUTE
230 FOR V=48 TO 53
240 PLOT 25,12,CHR$(V) : REM DIZAINE S.
250 FOR W=48 TO 57
260 PLOT 27,12,CHR$(W) : REM SECONDE
270 WAIT 100
275 IF R=50 AND S=52 THEN GOTO 450
280 NEXT W
300 NEXT V
310 NEXT U
315 U=48 : L=48
320 NEXT T
325 T=48 : K=48
331 NEXT S
335 S=48 : J=48
340 NEXT R
345 R=48 : I=48
450 R=48 : S=48 : I=48 : J=48
460 GOTO 90
501 REM -----
502 REM ** MISE A L'HEURE **
503 REM **      DE      **
504 REM **  L'HORLOGE  **
505 REM -----
1000 PRINT"DIZAINE D'HEURE ?" : INPUT I$
1010 I=ASC(I$)
1020 PRINT"HEURE ?" : INPUT J$
1030 J=ASC(J$)
1040 PRINT"DIZAINE DE MINUTES ?" : INPUT K$
1050 K=ASC(K$)
1060 PRINT"MINUTES ?" : INPUT L$
1070 L=ASC(L$)
1080 GOTO 10

```

CHAPITRE VII

CONVERSION DIGITALE/ANALOGIQUE ET ANALOGIQUE/DIGITALE

Afin d'obtenir un traitement possible entre l'ordinateur et certains phénomènes physiques, il sera indispensable d'établir une conversion soit du domaine de l'ordinateur (qui traite des nombres binaires) pour les convertir en une tension variable : par exemple traduire les poids de 0 à 255 en une tension variable entre 0 et 10 Volts, ou, à l'inverse, convertir une température (matérialisée par une différence de potentiel) par une information consommable pour le traitement par ordinateur.

Les échanges peuvent donc se faire dans les deux sens et les applications sont nombreuses :

- commande de la vitesse d'un moteur électrique,
- ajustement de la fréquence d'accord d'une tête haute fréquence par varicap,
- systèmes de régulation de chauffage,
- pilotage d'un générateur BF contrôlé par une tension,
- télécommande en site et azimuth d'un groupe d'antennes pour la poursuite en satellite ou le pointage d'un télescope...
- bancs de mesures automatiques, etc...

Nous voyons que la conversion analogique/digitale concerne tous les échanges de phénomènes physiques tels que les comparaisons de tension, la prise en charge d'informations sur les capteurs : température, pression, humidité, couples... Les applications en régulation sont évidentes du fait des rétro-actions possibles, les relevés avec visualisation sur écran deviennent élémentaires avec stockage sur disquettes, bandes ou imprimantes.

L'ordinateur devient un auxiliaire précieux pour le traitement de l'information, les comparaisons multiples, l'établissement de courbes et de relevés.

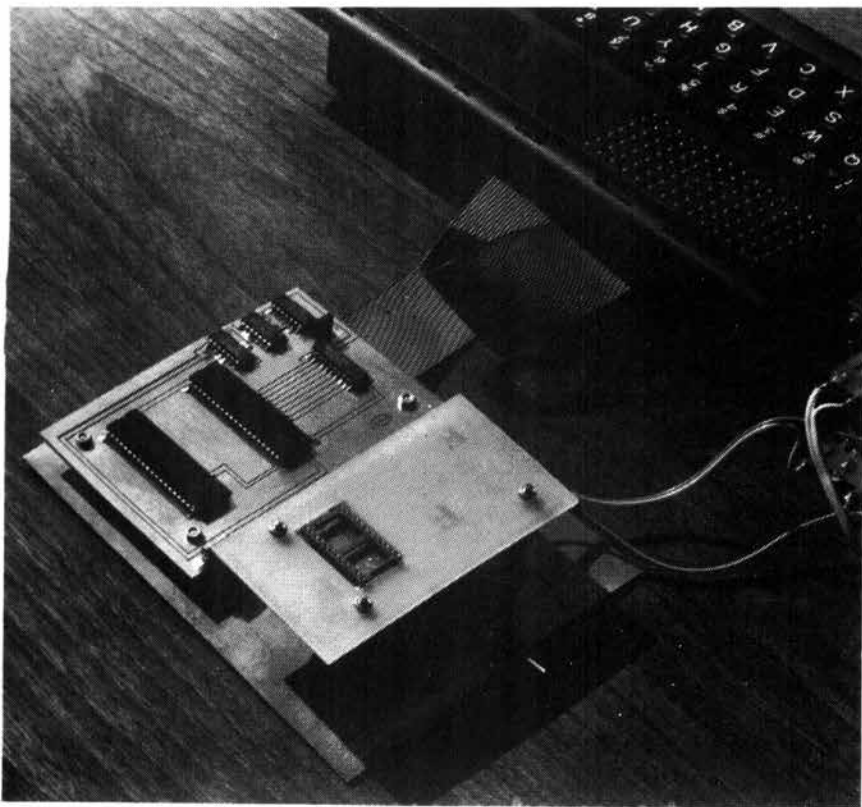
Des conditions du type FOR..., IF...THEN...ELSE , REPEAT...UNTIL peuvent s'établir et, avec une horloge en temps réel et la carte "relais" que nous avons réalisée au chapitre II, de nombreuses télécommandes pourront être mises en place avec n'importe quel protocole de fonctionnement.

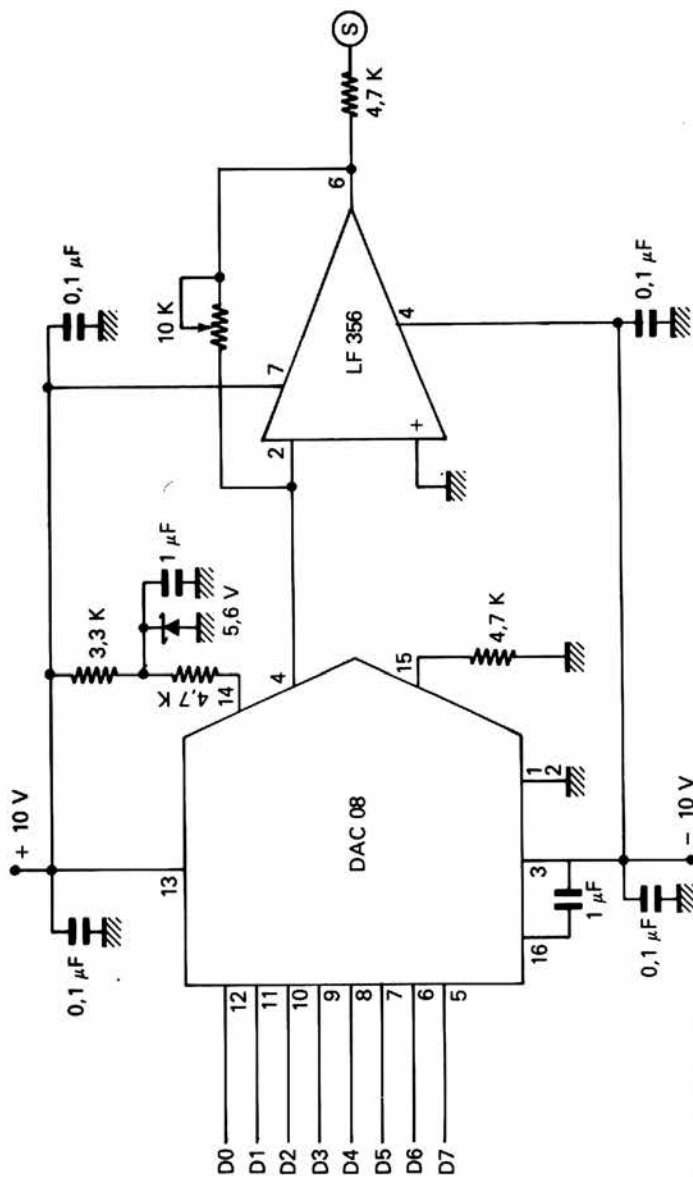
Un banc de mesure peut être établi pour des opérations répétitives avec prises d'information et de contrôle. De même, le relevé automatique de courbes de réponse pour un filtre ou un amplificateur basse

fréquence avec le relevé du tracé sur écran.

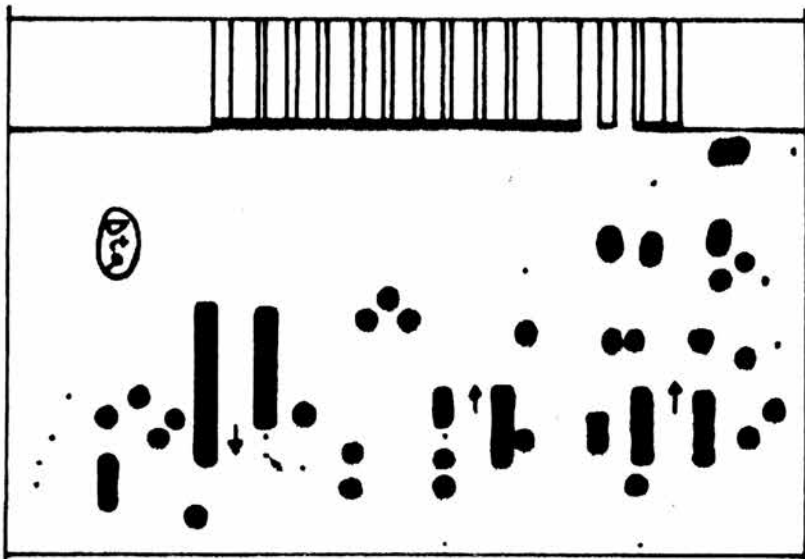
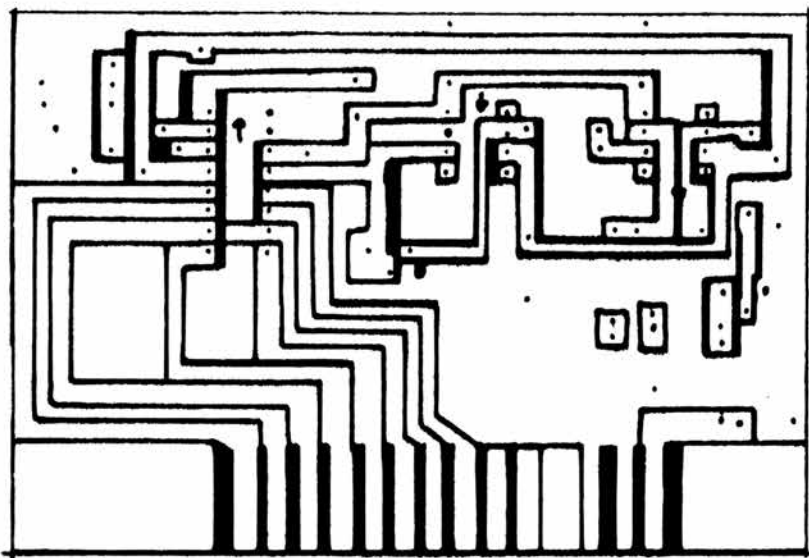
Le générateur synthétisé est piloté par l'un des ports de la carte d'interface, tandis que le convertisseur analogique/digital effectue la mesure de la tension de sortie.

Afin de piloter l'ensemble synthétiseur et convertisseur A/N, il faudra plus de 16 lignes sur l'interface de commande. C'est pour cette raison que nous avons développé une carte de 32 Entrées/Sorties.





Convertisseur
 Digital-Analogique



```

10 REM CONVERSION DIGITALE/ANALOGIQUE
15 REM TENSION EN DENT DE SCIE
20 POKE 1021,0
30 POKE 1020,255
40 POKE 1021,4
50 FOR N=0 TO 255
60 POKE 1020,N
70 NEXT N
80 GOTO 50

```

Nous donnons toute une suite de petits programmes d'application qui permettront de réaliser :

une conversion digitale/analogique : à chaque poids binaire de 0 à 255, correspond une tension échelonnée entre 0 et 5 Volts.

La ligne 80 fait recycler le processus.

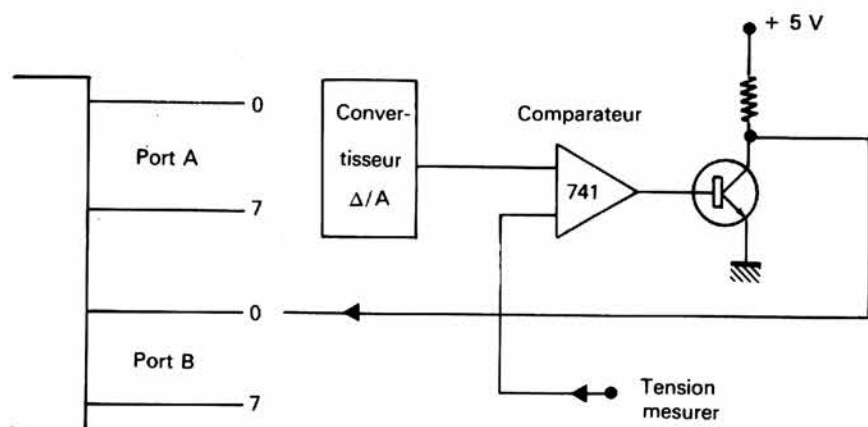
```

1  REM CONVERSION ANALOGIQUE/DIGITALE
5  POKE 1023,0 : POKE 1022,0 : POKE 1023,4
10 POKE 1021,0
20 POKE 1020,255
30 POKE 1021,4
35 POKE 1020,0
40 FOR N=0 TO 255
50 POKE 1020,N
60 A=PEEK(1022)
70 A=A AND 1
80 IF  A=1 THEN C=(N*5)/255 : PRINTC : END
90 NEXT N

```

Le second programme permet de "mesurer" une tension en détectant le niveau présent en sortie du transistor 2N2222 (comparateur

de tension) : Port Bo. Si PBO = 1 on arrête le processus. La valeur C est affichée après conversion.



```

10 REM CONVERSION DIGITALE/ANALOGIQUE
15 REM TENSION EN DENT DE SCIE
20 POKE 1021,0
30 POKE 1020,255
40 POKE 1021,4
50 REM VERSION RAPIDE : LANGAGE MACHINE
80 REM MISE EN MEMOIRE #0400
100 FOR Y=0 TO 17
110 READ A
120 POKE #0400+Y,A
130 NEXT Y
140 DATA #A9,#00,#8D,#FC,#03,#EE,#FC,#03,
        #AD,#FC,#03,#C9,#00,#D0,#F6,#4C,
        #00
150 DATA #04
160 REM APPEL DU PROGRAMME
200 CALL #0400

```

L'utilisation du langage BASIC prend beaucoup plus de temps que l'application du langage MACHINE. Nous avons établi le programme équivalent en l'implantant à l'adresse #0400.

Dans le programme suivant un compte de salves est produit par l'instruction REPEAT ... UNTIL et le retour à la routine en langage machine.

```
10 REM CONVERSION DIGITALE/ANALOGIQUE
15 REM TENSION EN DENT DE SCIE
20 POKE 1021,0
30 POKE 1020,255
40 POKE 1021,4
50 REM VERSION RAPIDE : LANGAGE MACHINE
60 D=0
80 REM MISE EN MEMOIRE #0400
100 FOR Y=0 TO 17
110 READ A
120 POKE #0400+Y,A
130 NEXT Y
140 DATA #A9,#00,#8D,#FC,#03,#EE,#FC,#03,
        #AD,#FC,#03,#C9,#00,#D0,#F6,#4C,
        #00
150 DATA #04
160 REM SALVES DE 5 DENTS DE SCIE
180 REPEAT
190 D=D+1
200 CALL #0400
205 UNTIL D=5
206 WAIT 100
210 WAIT 100 : D=0 : GOTO 180
```

CHAPITRE VIII

INTERFACE 32 ENTRÉES/SORTIES

L'interface qui suit ne correspond pas à une orientation vers le "gigantisme" mais constitue une réponse à un certain nombre de besoins spécifiques grands consommateurs de ports de sorties/entrées tels que gestion de bancs de mesure avec synthétiseur de fréquence et convertisseur digital/analogique, systèmes d'alarme avec scrutation de détecteurs de température, programmeur d'Eproms, etc...

Nous verrons que le circuit reste d'une grande simplicité en hardware et qu'il n'occupe au total que huit lignes d'adresses entre 03F8 et 03FF (1016 à 1023).

Par ailleurs, la gestion des ports de sorties est exactement identique à celle de l'interface 16 E/S puisqu'ils utilisent le même circuit intégré : le MC6821.

Quelques portes NAND sur A2 viennent gérer le choix entre sélection du 1^{er} 6821 ou le deuxième, suivant l'état haut ou bas de cette ligne d'adresse.

Tout le reste est évident et un circuit imprimé rend d'autant plus facile la réalisation de cette carte aux énormes possibilités.

Un circuit 74LS245 permet de ne pas surcharger le BUS de données, de telle sorte que l'ORIC ne voit qu'une seule entrée/sortie de D0 à D7 malgré les deux 6821.

Dans l'application présente, on emploiera indifféremment des 6821, 68A21 ou 68B21, voire même des 6822 (Cf chap. 16 E/S).

Nous ne donnons pas de conseils particuliers pour la réalisation pratique autres qu'un soin méticuleux dans les liaisons. On utilisera du circuit imprimé double-face avec deux straps de liaison pour les raccords de masse sur le dessous du circuit. On notera de même que certaines broches des circuits intégrés sont à souder sur le dessus et sur le dessous. Dans le cas d'utilisation de supports pour les 6821 (ce que nous conseillons si l'on craint les accidents sur les sorties en cours d'utilisation), il faudra employer des supports du type "tulipe" permettant la soudure des deux côtés.

Comme dans les précédents montages, les liaisons de sorties pourront se faire soit sur des connecteurs, soit sur des borniers à vis ou sur des dominos miniatures d'électricien. Le choix est d'importance si l'on veut rendre plusieurs modules compatibles par enfichages rapides.

Adressage des ports

Nous nous trouvons en possession de 32 lignes d'entrées/sorties distribuées sur quatre ports de huit lignes chacun.

On peut sélectionner soit des entrées, soit des sorties par le jeu de la programmation du registre de direction :

DDRA1/DDR1 ; DDRA2/DDR2.

La distribution adresses hexadécimales - adresses décimales - ports est la suivante :

1016	03F8	PORT A1
1017	03F9	
1018	03FA	PORT B1
1019	03FB	
1020	03FA	PORT A2
1021	03FD	
1022	03FE	PORT B2
1023	03FF	

Il faut se souvenir que tout accès au registre de direction ou de données pour chaque port est subordonné à l'état du bit B2 du registre de commande (chiffre impair de l'adresse décimale) :

registre CRB1
adresse 1019 b0 b1 b2 b3 b4 b5 b6 b7
03FB

Si B2 = 0 accès au registre de direction

Si B2 = 1 accès au registre de données

Nous désirons mettre le port B1 en entrées et le port A2 en sorties et 4 entrées/4 sorties sur B2 :

- Port B1 en entrées :

POKE 1019,0 accès au registre de direction
POKE 1018,0 toutes les lignes sont des entrées

- Port A2 en sorties :

POKE 1021,0 accès au registre de direction
POKE 1020,255 toutes les lignes sont des sorties

- Port PB2 en 4 entrées/sorties :

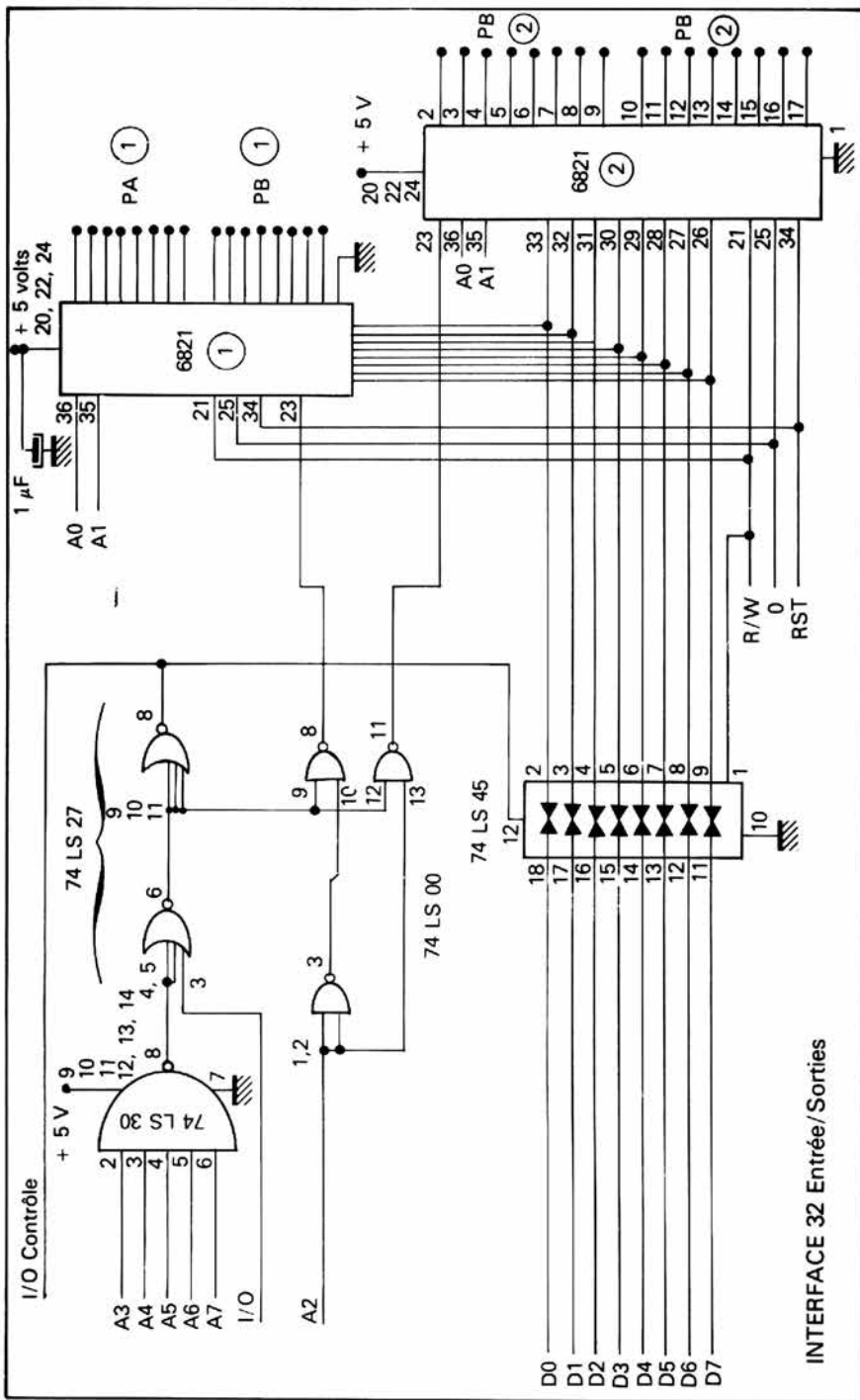
POKE 1023,0 accès au registre de direction
POKE 1022,240 PB2 0 à 3 en entrées
PB2 4 à 7 en sorties

On se reportera éventuellement à la correspondance du tableau binaire en fin d'ouvrage :

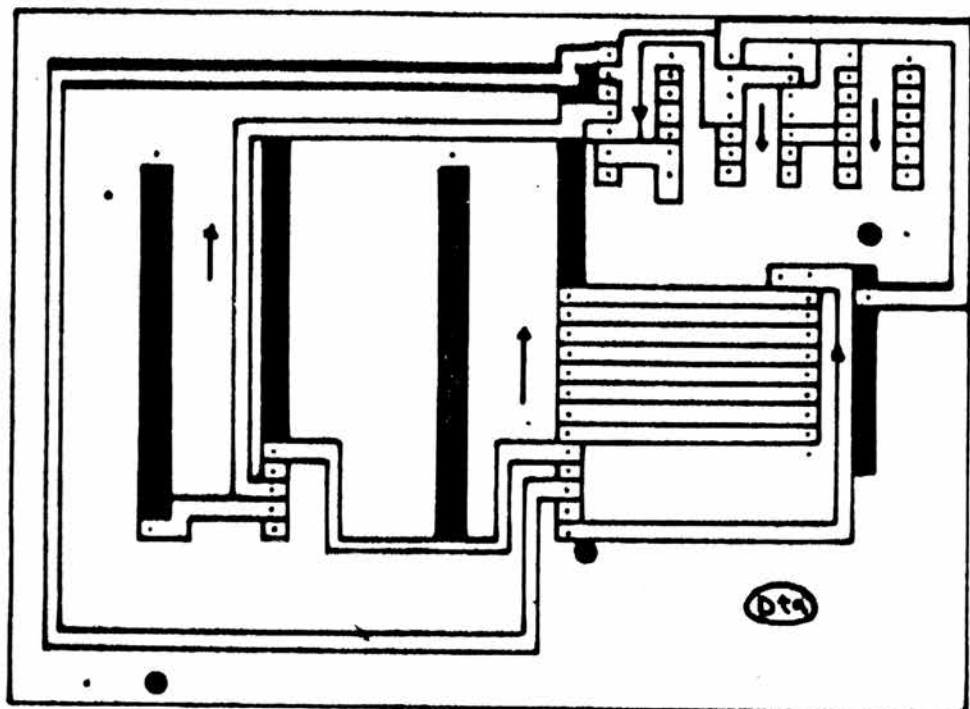
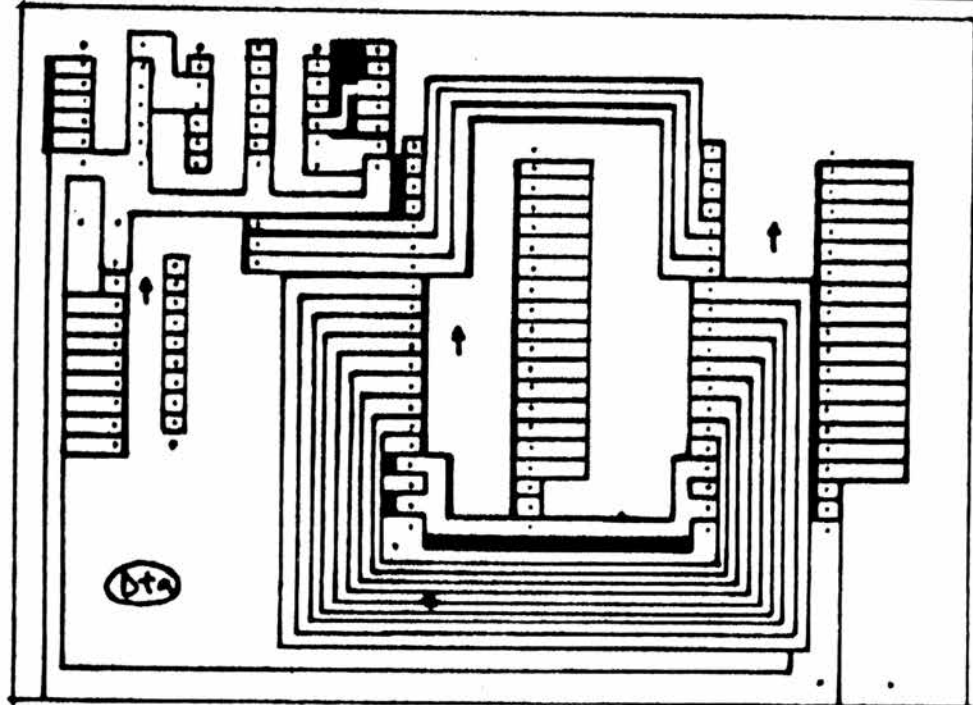
0 met tous les bits du registre à 0
255 tous les bits à 1
240 les quatre premiers bits à 0, les 4 derniers à 1
etc.. Un "1" sur le registre de direction correspond à la mise en SORTIE de cette ligne.
Un "0" met la ligne en **entrée**.

En utilisant la carte d'ampoules du chapitre III, nous aurons le programme suivant sur le Port b1 :

```
10 REM chenillard
20 POKE 1019,0
30 POKE 1018,255
40 POKE 1019,4
50 FOR N= 0 TO 7
60 READ A
70 POKE 1018,A
80 WAIT 25
90 NEXT N
100 DATA 1,2,4,8,16,32,64,128
110 RESTORE: GOTO 40
```

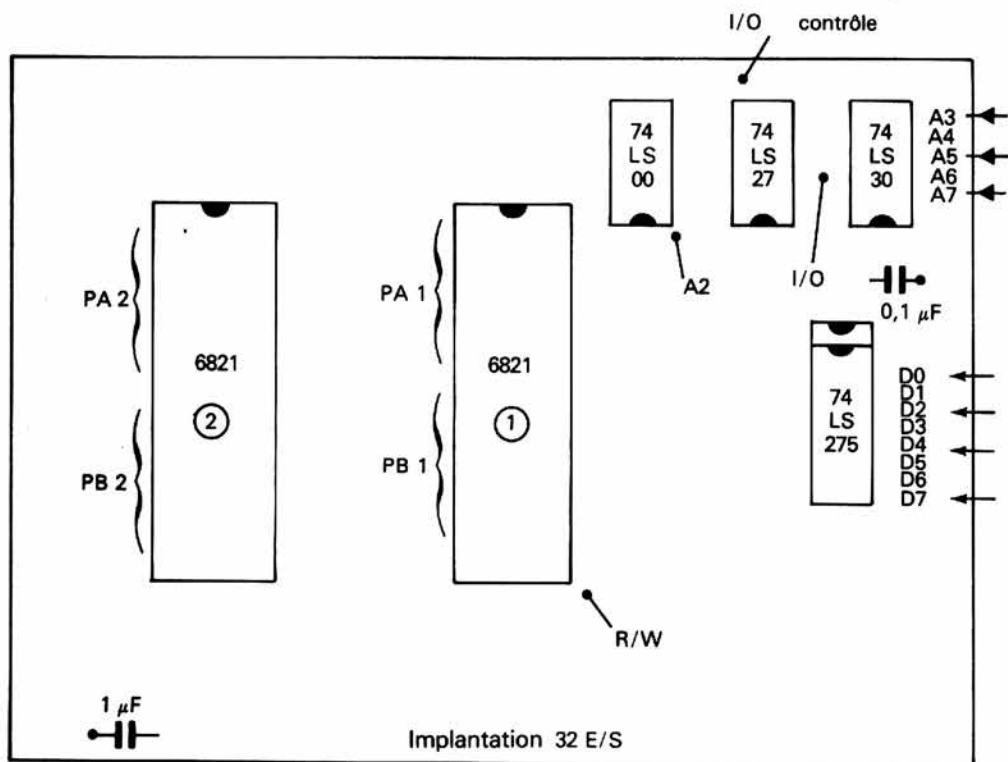



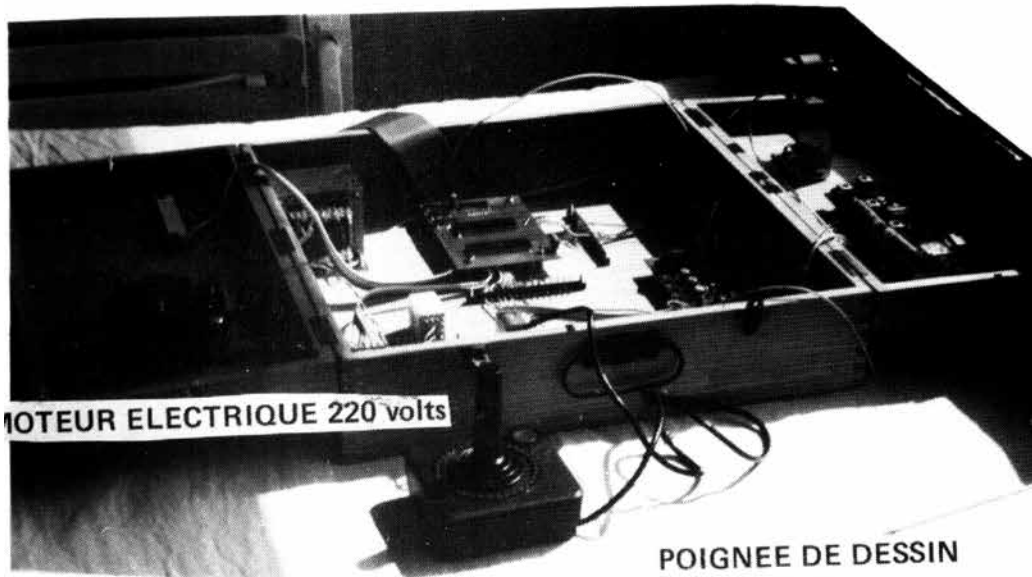
INTERFACE 32 Entrée/Sorties



ADDRESSES INTERFACE 32 E/S

1016 #3F8
 1017 #3F9
 1018 #3FA
 1019 #3FB
 1020 #3FC
 1021 #3FD
 1022 #3FE
 1023 #3FF

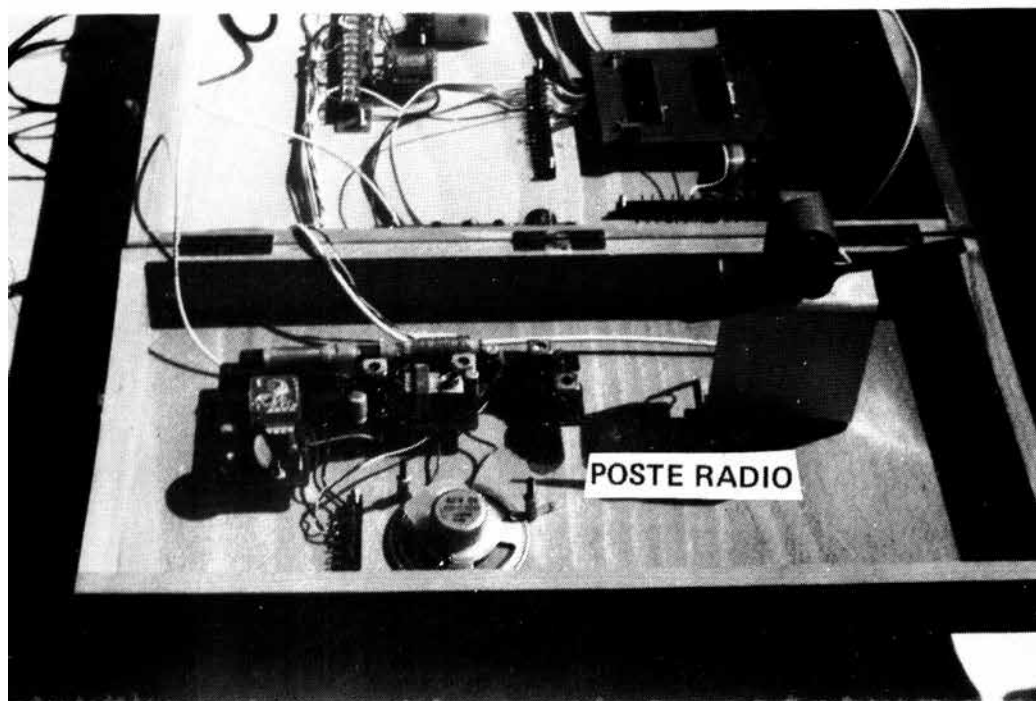




MOTEUR ELECTRIQUE 220 volts

POIGNEE DE DESSIN

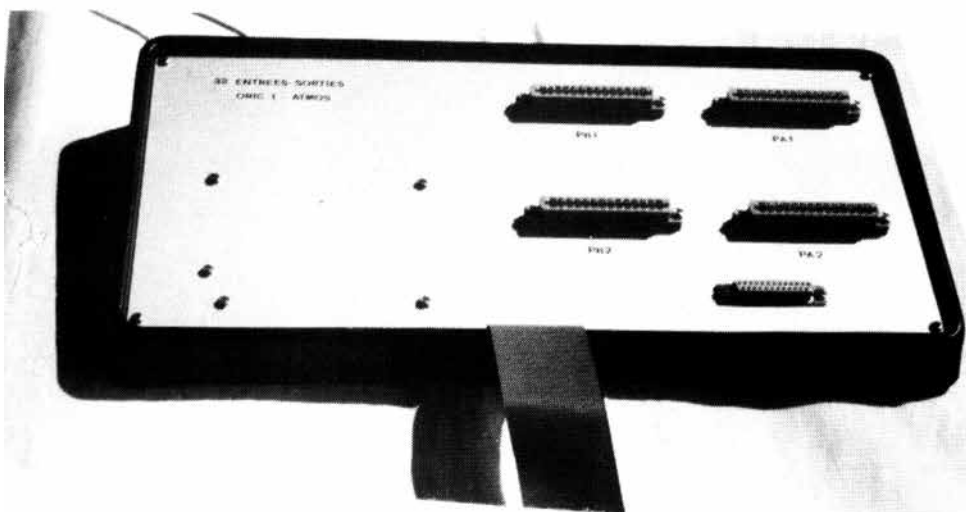
BOITE D'APPLICATIONS A PARTIR
D'UNE INTERFACE 32 ENTRES
SORTIES



POSTE RADIO



VUES DIVERSES D'APPLICATIONS
INDUSTRIELLES DE L'INTERFACE
32 ENTREES / SORTIES



APPLICATION A UN PROGRAMMATEUR D'EPROMS

On constatera facilement le peu d'organes annexes ! Un seul transistor et un régulateur... L'essentiel tient dans le programme de gestion qui permet :

- La recopie d'une Eprom déjà programmée et sa mise en mémoire dans l'ordinateur.
- La visualisation du listing sur les 2048 octets.
- La programmation d'une Eprom à partir de la mémoire vive de l'ORIC
- Examen de mémoire adresse par adresse et modification des octets par appui sur "M".

Distribution des rôles sur l'interface :

Le port A1 est attribué aux données. Dans le sens sortie pour la programmation de l'Eprom, dans le sens entrée pour la recopie.

Le port B1 concerne l'impulsion de 50 ms, positive, sur PB0. PB1 = OE ; BP2 = 25 Volts pulsés. A l'état 1, la tension de sortie est de 5 Volts. A l'état 0 la tension est de 25 Volts.

Le programme permet de suivre facilement le déroulement des opérations. Les adresses basses de l'Eprom sont sur le port A2 de A0 à A7, le reste est sur le port B2.

Un petit tableau récapitule les états des broches 18, 20 et 21. Elles sont prises en charge par le logiciel. Nous avons prévu cependant un interrupteur séparé qui court-circuite les diodes sur la base du 7805. Ainsi, une tension de 5 Volts max. pourra dans tous les cas parvenir à VPP durant les manœuvres préparatoires. L'affichage se fait par "tranches" de 256 octets.

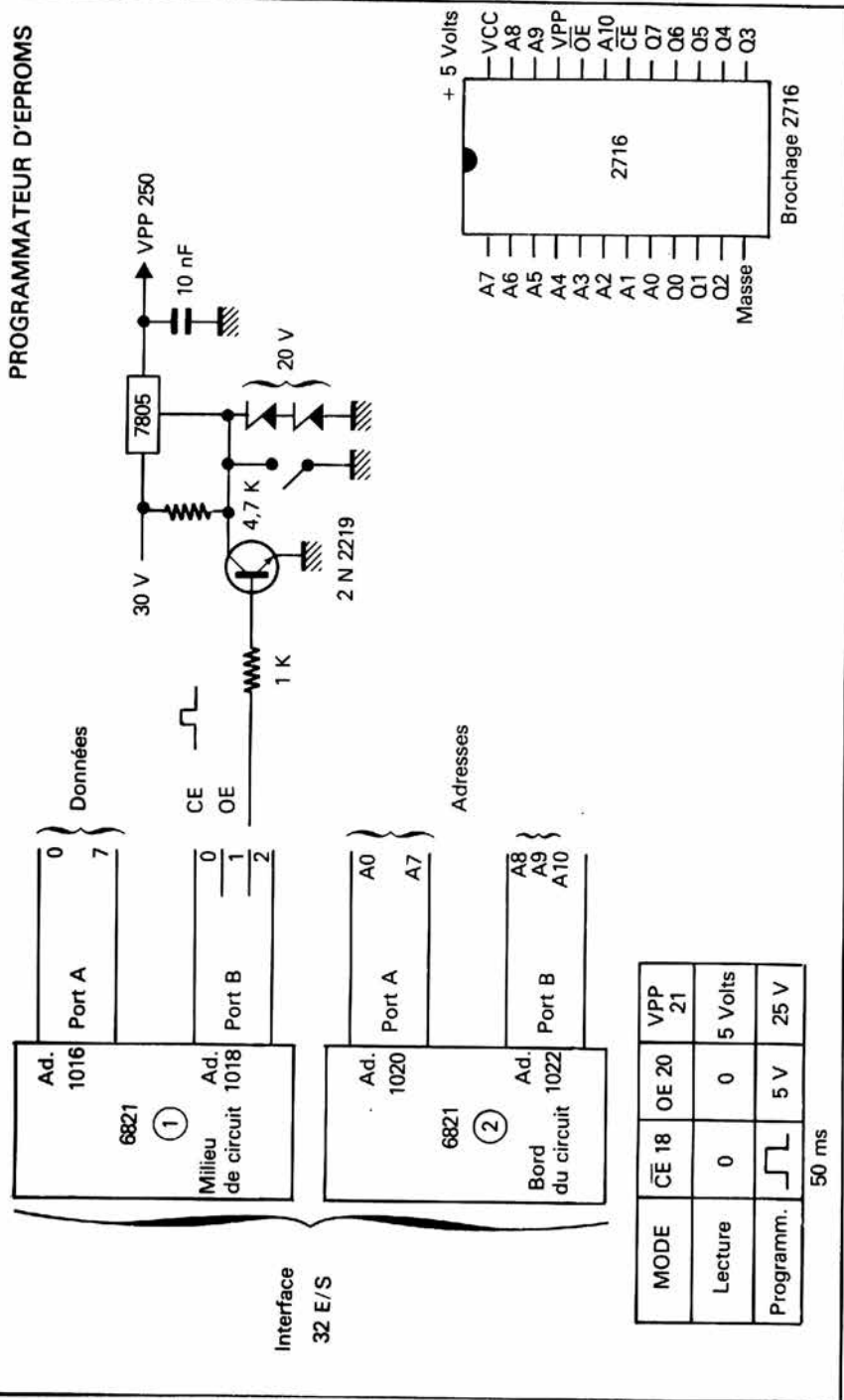
Le chargement d'Eprom en RAM s'effectue de la ligne 200 à 330. Nous avons réservé une zone mémoire pour le langage machine. La recopie en RAM est en 30720 et suivantes (Hex. °°).

Le listing complet de la mémoire vive (30720 + 2047 octets) a lieu en 600-620. Il s'agit de représentation hexadécimale. Les personnes plus accoutumées à des octets décimaux pourront supprimer l'instruction HEX\$ (ligne 610).

L'examen de la mémoire pas-à-pas avec l'adresse correspondante est en 700-1010.

On pourra modifier les octets en mémoire vive par appui sur la touche "M" Modification.

PROGRAMMATEUR D'EPROMS



On corrige l'octet ; un appui sur RETURN provoque la prise en compte de la modification et l'écriture de l'octet (avec son adresse) corrigé. L'appui sur la touche "espace" provoque le défilement des octets suivants, etc...

Programmation : 8 à 160

Tous les ports sont mis en sorties.

POKE1018,6 met CE à 0 ; OE à 5 et VPP à 5 Volts.

On applique alors le 25 Volts (ouverture de l'interrupteur court-circuitant le transistor 2N2219, plus précisément, cette manœuvre "autoriserà" le 25 Volts pulsé. En effet, PB1 (ligne 2) est à l'état 1, donc le 2N2219 est un court-circuit pour les diodes.

Les octets programmés sont affichés (ligne 100).

La durée de programmation des 2048 octets est d'environ quatre minutes (3 minutes 48 secondes exactement !). Il n'est pas nécessaire, à notre avis, d'aller plus rapidement et de faire intervenir un fonctionnement en langage machine.

Attention, ce programme ne fonctionne pas sans l'interface raccordée. Essayez, par exemple, POKE 1019,0 sur votre ordinateur. Le curseur se bloque tout simplement. Tout rentre dans l'ordre après branchement puisque l'accès en page 3 (par la broche I/O et I/O controle) désaccouple le VIA interne.

Réalisation

Nous avons prévu un petit circuit imprimé qui sert de support à l'Eprom 2716 et de câblage de l'alimentation. On pourra tout aussi bien se servir de plaquette à trous en epoxy. Un minimum de soin dans le câblage permettra un fonctionnement immédiat de l'appareil. Les microprocesseurs sont toujours pleins de raccords et les fils en nappe sont une aide précieuse pour l'ordre et l'esthétique.

```
5 HIMEM 28672
```

```
6 GOTO 500
```

```
8 CLS
```

```
10 A=0:B=255:C=4:D=0
```

```
15 REM TOUS LES PORTS EN SORTIES
```

```
20 FOR N=0 TO 8 STEP 2
```

```
30 POKE 1017+N,A:POKE 1016+N,B:POKE 101
```



```

7+N,C:POKE 1016+N,D
 40 NEXT N
 45 POKE 1018,6: REM CE=0 OE=5 UPP=5
 46 PRINT:PRINT:PRINT:PRINT"APPLIQUEZ LE
 25 VOLTS"
 47 PRINT:PRINT"APPUYEZ SUR UNE TOUCHE "
:GET A$
 50 FOR E= 0 TO 7
 60 POKE 1022,E
 70 FOR F=0 TO 255
 80 POKE 1020,F
 90 DO=PEEK(30720+(E*256)+F)
100 PRINTEX$(DO);
110 POKE 1016,DO
120 POKE 1018,3:REM CE=5 OE=5 UPP=0 (25
U)
130 WAIT 5 :REM 50 mS
140 POKE 1018,6:REM CE=0 OE=5 UPP=5
150 NEXT F:PRINT:PRINT:PRINT"   ":PING:
NEXT E
155 PRINT:PRINT:PRINT:PRINT"COUPEZ LE 2
5 VOLTS"
160 END
200 A=0:B=255:C=4:D=0
210 FOR N=0 TO 8 STEP 2
220 POKE 1017+N,A:POKE1016+N,B:POKE 101
7+N,C:POKE 1016+N,D
230 NEXT N
240 POKE 1017,0:POKE 1016,0:POKE 1017,4
:REM PORT A1 EN ENTREE
245 REM POSITION CS
250 FOR E=0 TO 7
260 POKE 1022,E
270 FOR F=0 TO 255

```

```

280 POKE 1020,F
285 POKE 1017,4
290 DO= PEEK(1016)
300 PRINT HEX$(DO);
310 POKE30720 +(E*256)+F,DO
320 NEXT F:PING:PRINT:PRINT:PRINT:NEXT
E
330 END

500 CLS:PAPER 3:INK 0:PRINT:PRINT:PRINT

510 PRINT" *****
**"
520 PRINT" * 1/CHARGEMENT EPROM\RAM
530 PRINT" * 2/LISTING "
540 PRINT" * 3/PROGRAMMATION "
545 PRINT" * 4/EXAMEN MEMOIRE "
550 PRINT" *****
**"
560 GET A$
570 IF A$<>"1"AND A$<>"2"AND A$<>"3"AND
A$<>"4" THEN 560
580 X=VAL(A$): ON X GOTO 200,600,8,700
600 FOR N=0 TO 2047
610 PRINTHEX$(PEEK(30720+N));
620 NEXT N
700 FOR N=0 TO 2047
710 G=30720+N
715 PRINTHEX$(G);" ";HEX$(PEEK(30720+N
))
720 GET A$
725 IF A$="E" THEN GOTO 5
726 IFA$="M" THEN PRINT HEX$(G);" ";HE
X$(PEEK(30720+N)):INPUT B$: GOTO 1000
730 NEXT N

```

1000 B\$="#"+B\$

1010 POKE 30720+N,VAL(B\$):GOTO 715

Composants de la carte 32 E/S

- 2 PIA Motorola 6821
- 1 74LS245
- 1 74LS00
- 1 74LS27
- 1 74LS30
- 2 Condensateurs 1,5 uF, 10 Volts ; 0,1 uF, 35 Volts.
- 1 Circuit imprimé double face.

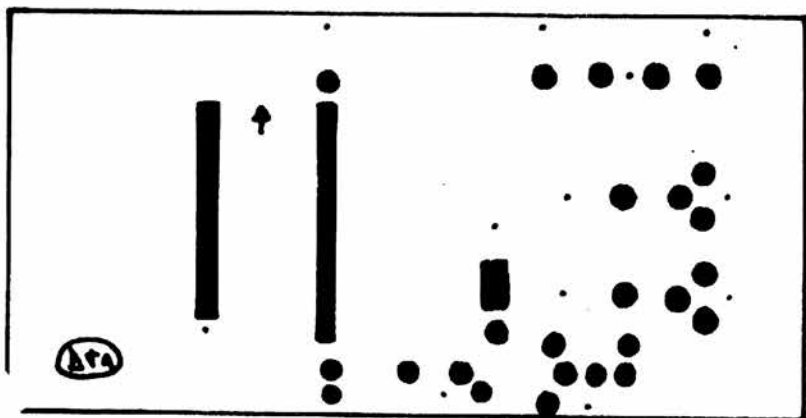
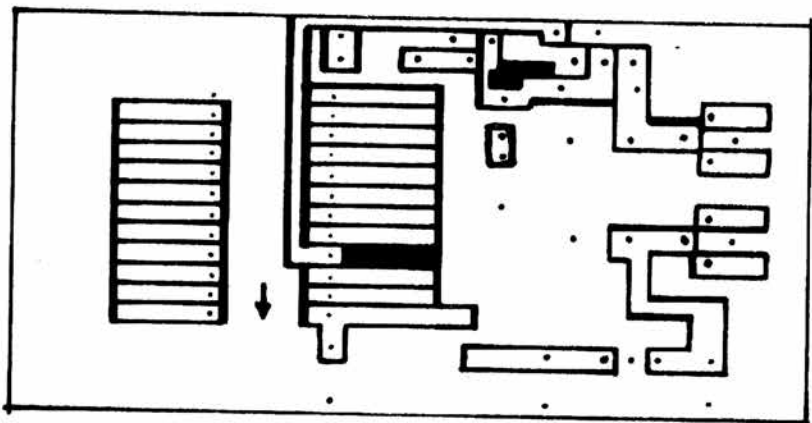


TABLEAU DE CONVERSION

Décimal / Binaire / Hexadécimal

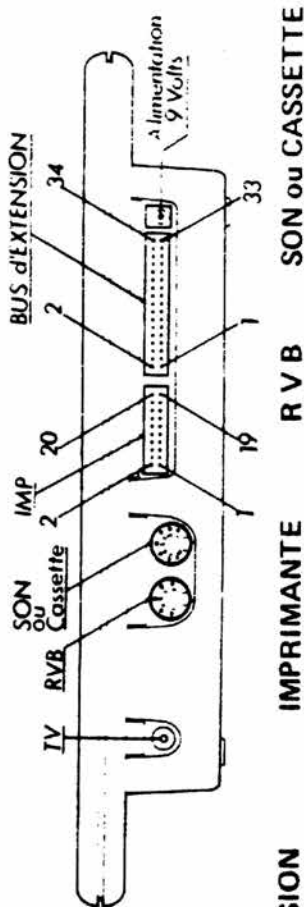
DEC	BINAIRE	HEX
0	00000000	00
1	00000001	01
2	00000010	02
3	00000011	03
4	00000100	04
5	00000101	05
6	00000110	06
7	00000111	07
8	00001000	08
9	00001001	09
10	00001010	0A
11	00001011	0B
12	00001100	0C
13	00001101	0D
14	00001110	0E
15	00001111	0F
16	00010000	10
17	00010001	11
18	00010010	12
19	00010011	13
20	00010100	14
21	00010101	15
22	00010110	16
23	00010111	17
24	00011000	18
25	00011001	19
26	00011010	1A
27	00011011	1B
28	00011100	1C
29	00011101	1D

DEC	BINAIRE	HEX
30	00011110	1E
31	00011111	1F
32	00100000	20
33	00100001	21
34	00100010	22
35	00100011	23
36	00100100	24
37	00100101	25
38	00100110	26
39	00100111	27
40	00101000	28
41	00101001	29
42	00101010	2A
43	00101011	2B
44	00101100	2C
45	00101101	2D
46	00101110	2E
47	00101111	2F
48	00110000	30
49	00110001	31
50	00110010	32
51	00110011	33
52	00110100	34
53	00110101	35
54	00110110	36
55	00110111	37
56	00111000	38
57	00111001	39
58	00111010	3A
59	00111011	3B

DEC	BINAIRE	HEX
60	00111100	3C
61	00111101	3D
62	00111110	3E
63	00111111	3F
64	01000000	40
65	01000001	41
66	01000010	42
67	01000011	43
68	01000100	44
69	01000101	45
70	01000110	46
71	01000111	47
72	01001000	48
73	01001001	49
74	01001010	4A
75	01001011	4B
76	01001100	4C
77	01001101	4D
78	01001110	4E
79	01001111	4F
80	01010000	50
81	01010001	51
82	01010010	52
83	01010011	53
84	01010100	54
85	01010101	55
86	01010110	56
87	01010111	57
88	01011000	58
89	01011001	59
90	01011010	5A
91	01011011	5B
92	01011100	5C
93	01011101	5D

DEC	BINAIRE	HEX
94	01011110	5E
95	01011111	5F
96	01100000	60
97	01100001	61
98	01100010	62
99	01100011	63
100	01100100	64
101	01100101	65
102	01100110	66
103	01100111	67
104	01101000	68
105	01101001	69
106	01101010	6A
107	01101011	6B
108	01101100	6C
109	01101101	6D
110	01101110	6E
111	01101111	6F
112	01110000	70
113	01110001	71
114	01110010	72
115	01110011	73
116	01110100	74
117	01110101	75
118	01110110	76
119	01110111	77
120	01111000	78
121	01111001	79
122	01111010	7A
123	01111011	7B
124	01111100	7C
125	01111101	7D
126	01111110	7E
127	01111111	7F

LES SORTIES DE L'ORIC



BUS d'EXTENSION

RVB

IMPRIMANTE

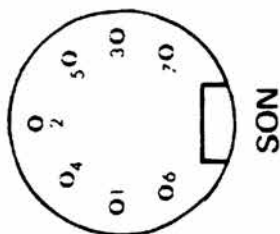
BUS d'EXTENSION

- 1 Sortie
- 2 Masse
- 3 Entrée
- 4 Son
- 5 Son
- 6 Contact du relais
- 7 Contact du relais

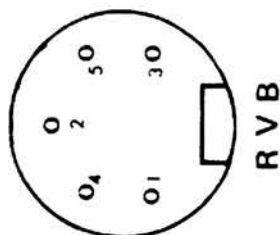
- 1 Rouge
- 2 Vert
- 3 Bleu
- 4 Synchro
- 5 Masse

- STB 1 2
- D0 3 4
- D1 5 6
- D2 7 8
- D3 9 10
- D4 11 12
- D5 13 14
- D6 15 16
- D7 17 18
- ADK 19 20

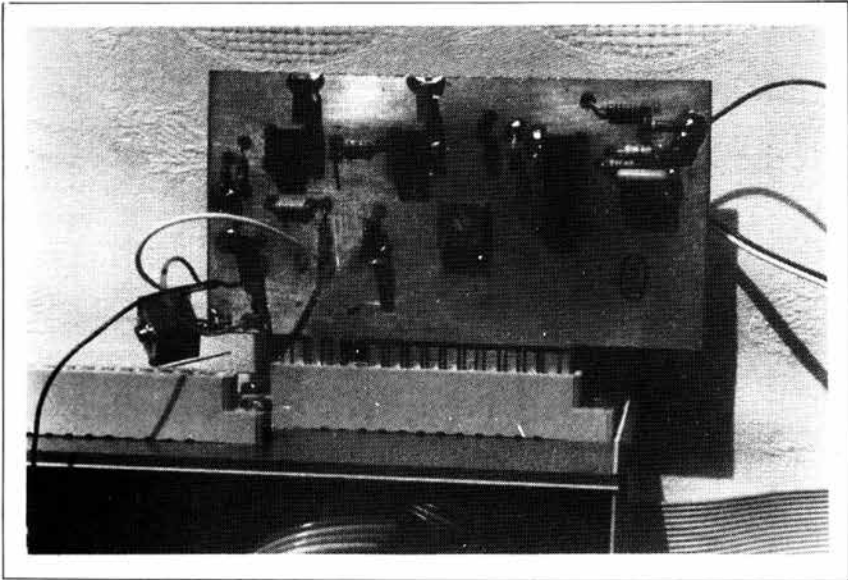
- ROMDIS STB 1 2
- Reset D0 3 4
- I/O Control D1 5 6
- IRQ D2 7 8
- D0 D3 9 10
- D1 D4 11 12
- D6 D5 13 14
- D3 D6 15 16
- D4 D7 17 18
- A4 D5 19 20
- D7 A5 21 22
- A15 A6 23 24
- A14 A7 25 26
- A13 A8 27 28
- A12 A9 29 30
- A11 A10 31 32
- +5V 33 34 Terre ou masse



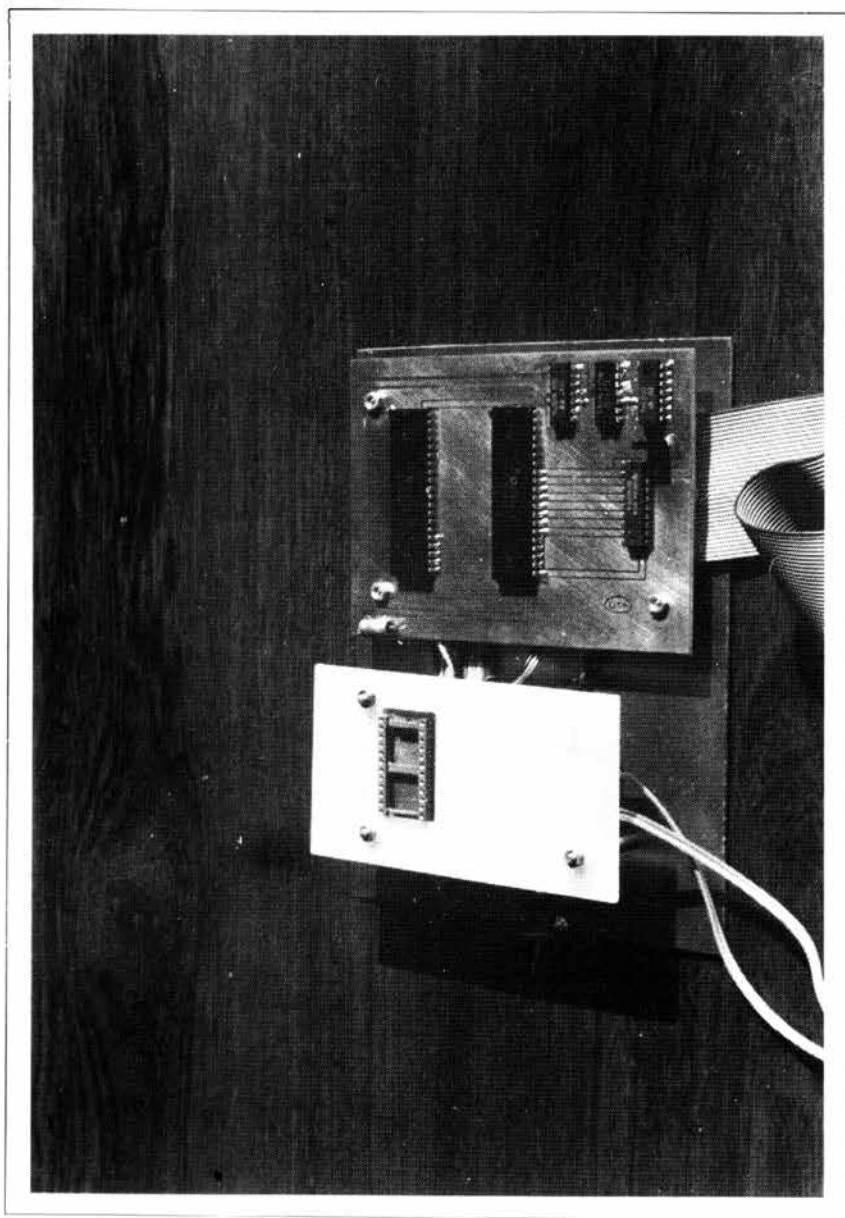
SON



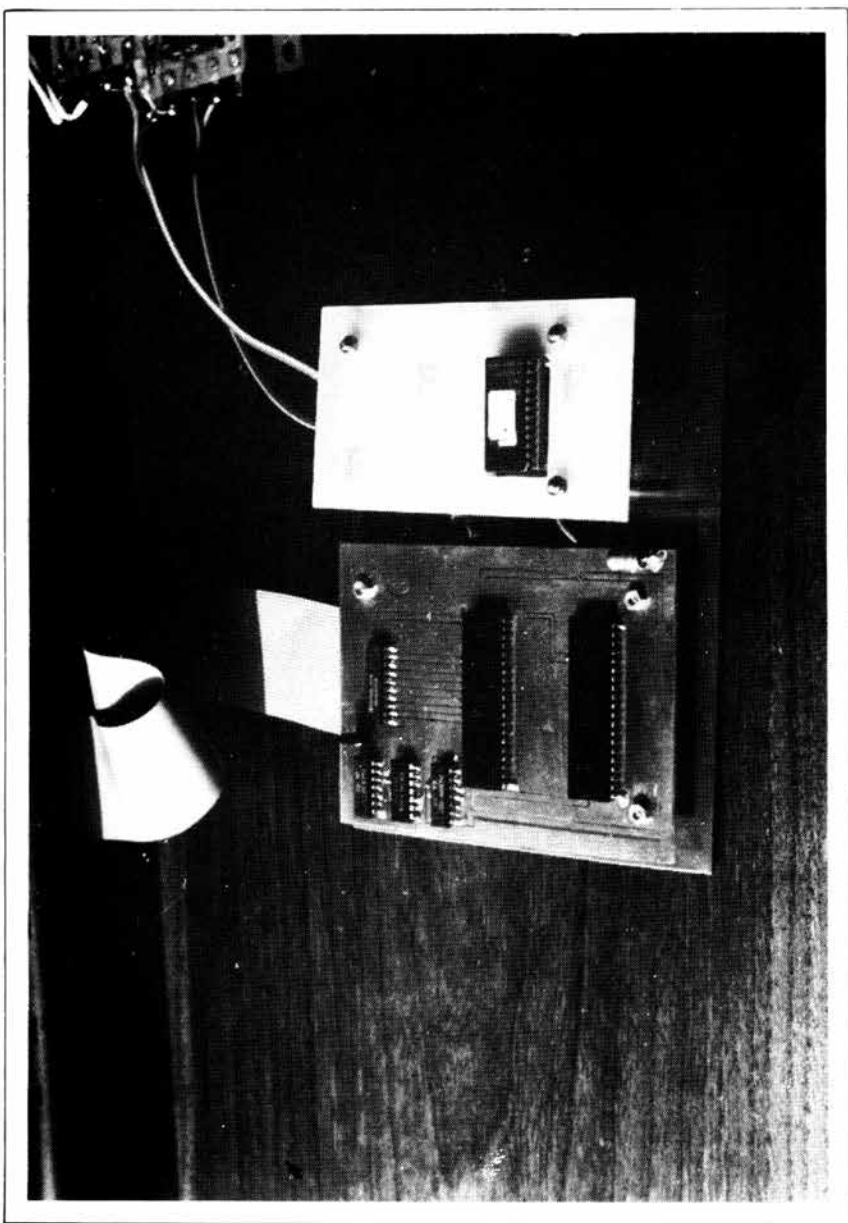
RVB



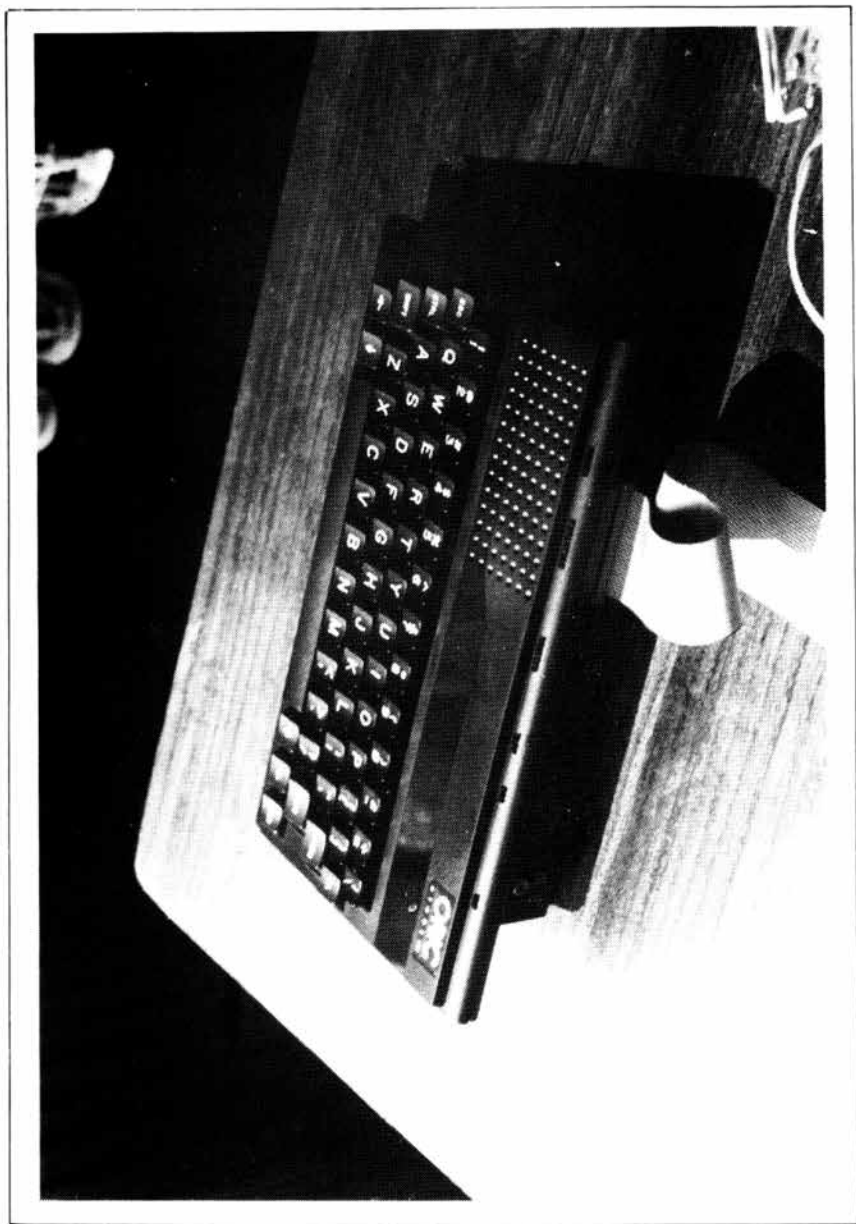
Montage d'essai pour la carte de conversion D/A et A/D.



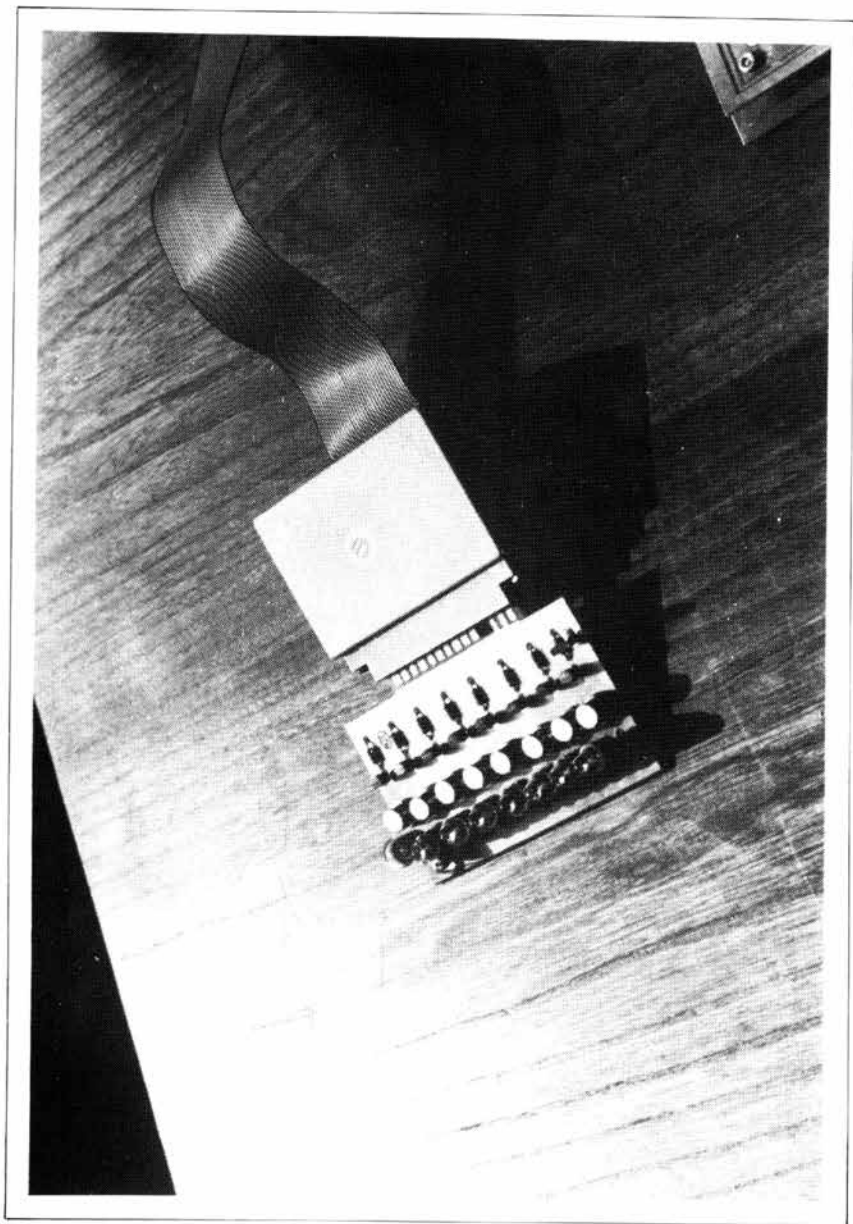
Le support EPROM aura intérêt à être du type "à levier" ou à force d'insertion nulle.



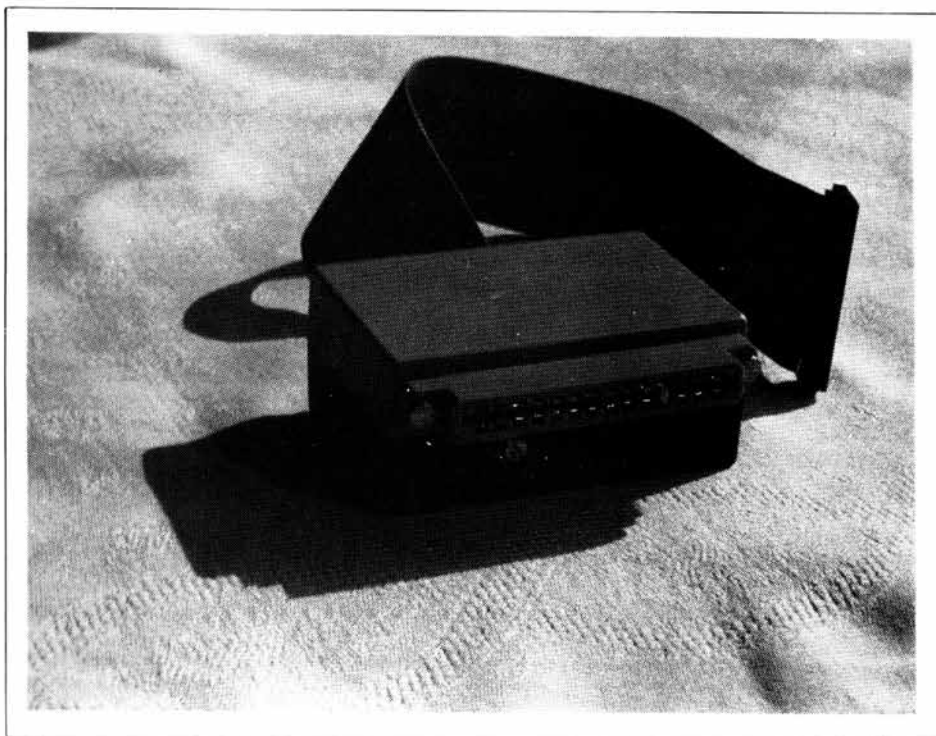
*Interface 32 entrées/sorties.
Recopie de mémoire EPROM. Chargement en RAM.
Programmateuf.*



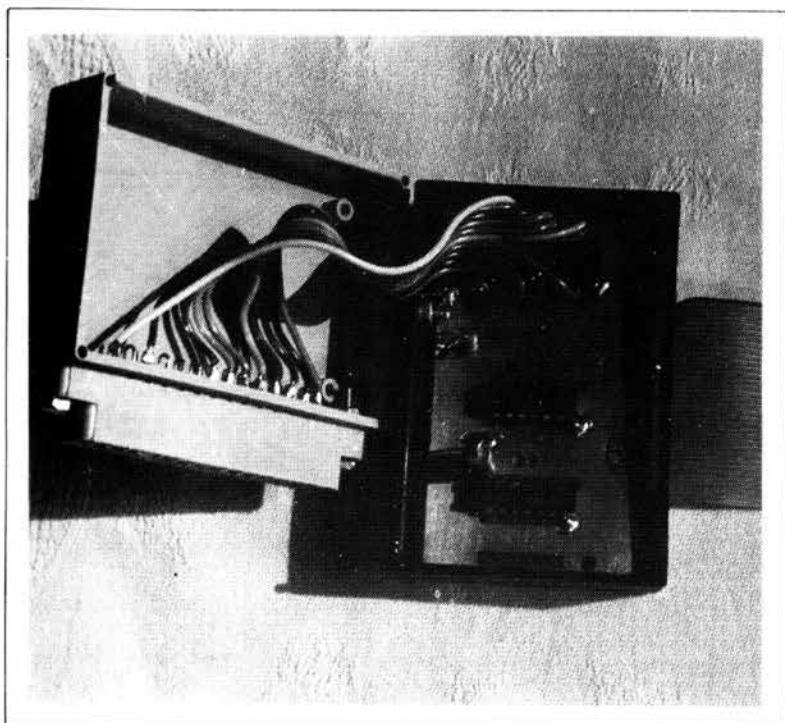
*Raccordement de l'interface au bus 34 broches.
Tous les montages décrits sont compatibles ORIC-1 et ATMOS.*



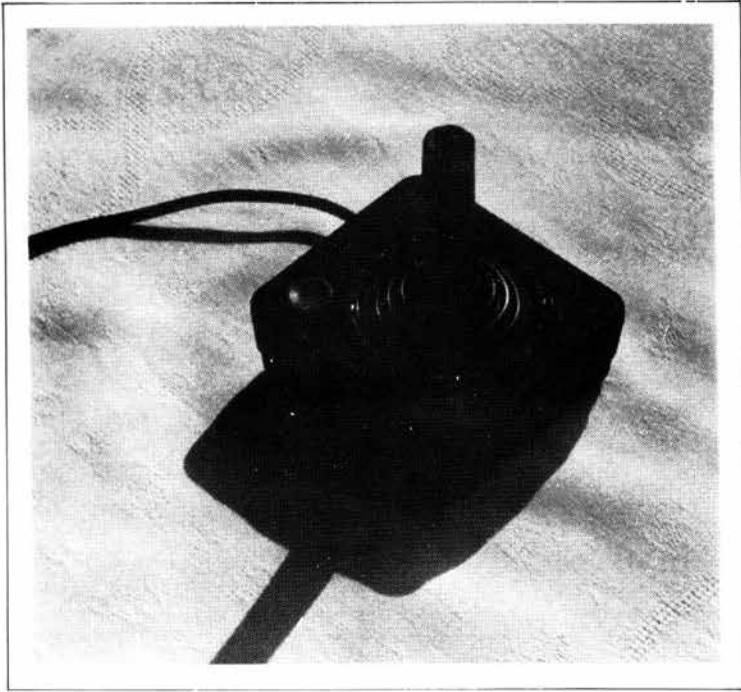
Interface 8 sorties et carte de huit ampoules.



Interface 8 sorties. Boîtier plastique. Connecteur 34 broches.



Raccordement interne du connecteur avec câble en nappe.



Poignée de jeux ou poignée de dessin.

TABLE DES MATIÈRES

INTRODUCTION

CHAPITRE I

La communication avec l'extérieur.
L'interface entrées/sorties.
8 sorties programmables.
Utilisation et exemples pratiques.

CHAPITRE II

Une interface à 16 lignes de communication.
Constitution du 6821. Fonctionnement.
Lecture des données du PIA.

CHAPITRE III

Réalisation pratique de la carte.
Particularités des ports A et B.
Câblage pratique.

CHAPITRE IV

Des moyens simples : jeux de lumières, commandes de relais.
Programmes d'application.
Interface de puissance "secteur". 8 sorties sur 220 V, 1 500 Watts.
Isolement par optocoupleur.

CHAPITRE V

ORIC : La programmation facile des synthétiseurs de fréquence.
Pilotage d'un générateur BF de 10 Hz à 1 MHz.
Réalisation pratique. Programme de gestion.
Application à un synthétiseur 144-146 MHz.

CHAPITRE VI

Techniques de commande d'automatismes.
Alimentation programmable. Alarme graphique.
Poignée de jeux (joystick).
Poignée de dessin. Horloge à quartz. Convertissez votre ORIC en horloge.

CHAPITRE VII

Conversion digitale/analogique et analogique/digitale. Circuit imprimé.
Programmes d'exploitation.

CHAPITRE VIII

Interface 32 entrées/sorties. Gestion d'automatismes complexes. Réalisation pratique. Application à un programmeur d'Eproms. Recopie de mémoires. Listing. Modification en RAM. Programmation de 2716.

— OUVRAGES PARUS AUX ÉDITIONS SORACOM —

La Guerre des Ondes

de F. Mellet et S. Faurez

Alimentations de puissance

Collection Sélection de montages

Transat Terre-Lune

*Union pour la Promotion de la
Propulsion Photonique*

QSO en radiotéléphonie (français-anglais)

de L. Sigrand

Interférences TV (QRM TV)

2ème édition

de F. Mellet et K. Pierrat

A l'écoute des radiotélétypes

2ème édition

de J.L. Fis

Technique radio pour l'amateur

3ème édition

de F. Mellet et S. Faurez

Communiquez avec votre ZX81

2ème édition

de D. Bonomo et E. Dutertre

Télévisions du monde

de P. Godou

Le radioamateur et la QSL

de G. Lelarge

Technique de la B.L.U.

2ème édition

de G. Ricaud

La réception des satellites météo

de L. Kuhlmann

Les synthétiseurs de fréquences

de M. Levrel

Expédition

Pôle Nord Magnétique 1983

de M. Uguen

Visa pour ORIC

de F. Blanc et F. Normant

Forth pour ORIC

d'Oric France

Programmes pour votre ORIC

de E. Jacob et J. Portelli

Communiquez avec votre ORIC

de D. Bonomo et E. Dutertre

Concevoir un émetteur expérimental

de P. Loglisci

Apprendre l'électronique sur ORIC

de P. Beaufiles

théoric

LA REVUE DES PASSIONNÉS D'ORIC

UNE VISITE CHEZ ORIC
PRODUCTS INTERNATIONAL
L'ATMOS AU BANC D'ESSAI
JEU: LE MOT LE PLUS LONG

tous les 2 mois
chez votre
marchand
de journaux

GAGNEZ UN
VOYAGE AUX
CANARIES



BIMESTRIEL - N°1 - 20F - MARS 1984

CHAQUE MOIS N'OUBLIEZ PAS*

MEGAHERTZ



***Pour ne pas l'oublier, ABONNEZ VOUS !**

**MEGAHERTZ : SORACOM - 16, av. Gros Malhon
35000 RENNES - Tél. : (99) 54.22.30**

DEMANDEZ NOUS UN EXEMPLAIRE DE PRESSE GRATUIT.

Composition : FIDELTEX

Maquette : SORACOM

Impression : VAN DEN BRUGGE

N° d'Editeur : 028

Dépôt légal : Avril 1984