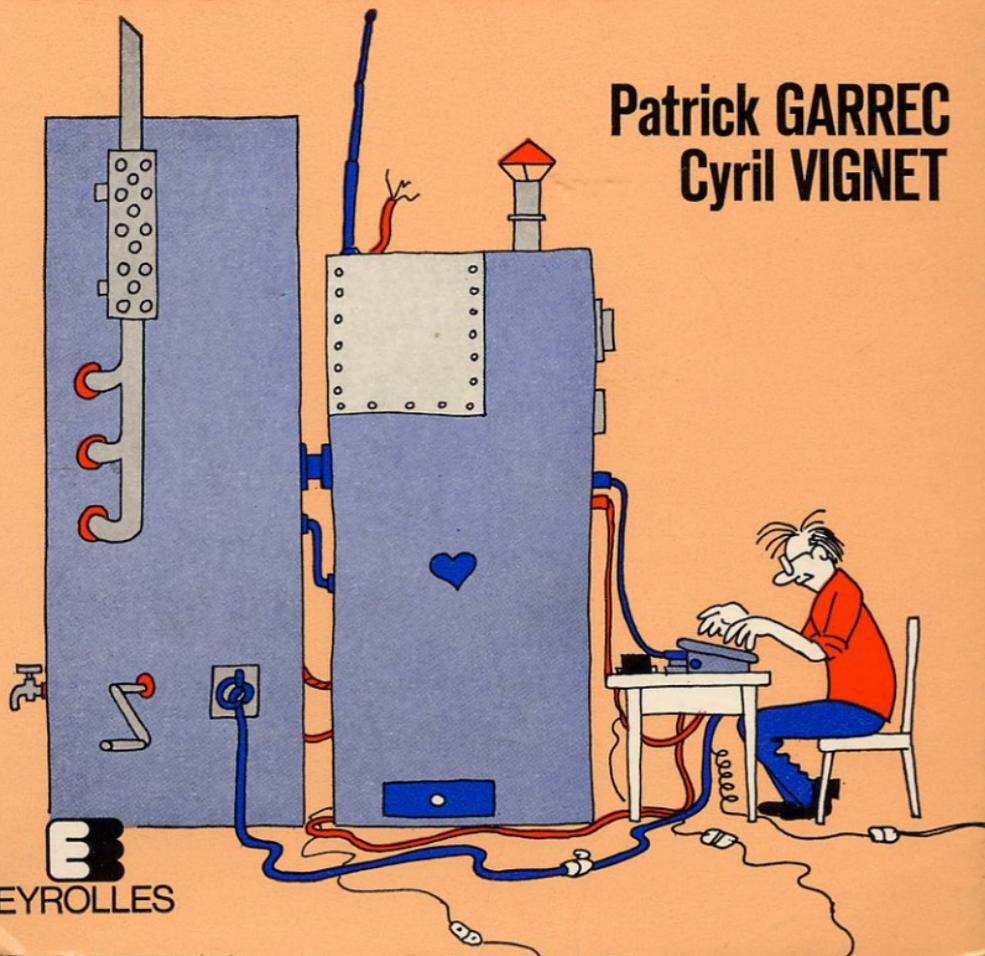


MICRO-ORDINATEURS



# DES EXTENSIONS A CONSTRUIRE POUR VOTRE ORIC ATMOS

Patrick GARREC  
Cyril VIGNET



**DES EXTENSIONS  
A CONSTRUIRE  
POUR VOTRE  
ORIC-ATMOS**

*à ma femme  
à toi,*

# DES EXTENSIONS A CONSTRUIRE POUR VOTRE ORIC-ATMOS

par

**Patrick GARREC**

*Ingénieur de la THOMSON-CSF  
Ancien élève de l'INSA de Rennes*

**Cyril VIGNET**

*Ingénieur informaticien  
Ancien élève de l'École Centrale  
des Arts et Manufactures*

Collection animée  
par Richard SCHOMBERG

«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 1<sup>er</sup> de l'article 40)».

«Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal».

© Éditions EYROLLES, 1984

  
EYROLLES

61, boulevard Saint-Germain — 75005 PARIS  
1985

## DANS LA MÊME COLLECTION

- SCHOMBERG - Le Basic Universel.  
 SCHOMBERG - Micro-ordinateurs : Comment ça marche ?  
 HERNANDEZ - Pascal par l'exemple.  
 NOLLET - La conduite du ZX 81.  
 PELLIER - La conduite du TRS 80.  
 LADEVIE - Votre gestion avec BASIC sur micro-ordinateur.  
 QUEINNEC - Langage d'un autre type : LISP.  
 PELLIER - Programmez vos jeux d'action rapide sur TRS 80.  
 ASTIER - La conduite de l'APPLE II.  
     Tome 1 : le Basic de l'APPLE II.  
     Tome 2 : le système graphique et l'assembleur de l'APPLE II.  
 MONTEIL - L'assembleur facile du 6502 et du 6510.  
 LEPAPE - L'assembleur facile du Z 80.  
 OROS et PERBOST - ZX 81 à la conquête des jeux.  
     - CASSETTE - ZX 81 à la conquête des jeux.  
     - CASSETTE N° 2 - 13 jeux 1 K.  
     - CP/M et sa famille.  
 PERBOST - Langage machine, trucs et astuces sur ZX 81.  
 DAX - La conduite du PC 1212 (ou TRS 80 pocket).  
 NOLLET - Apprenez à parler à votre ordinateur.  
 BICKING - La conduite du VIC 20.  
 TEJA - La conduite de l'IBM-PC.  
 MONTEIL - Télécommande avec votre micro-ordinateur.  
 PLOUIN - Tout sur les disques du TRS 80 modèles I et III.  
 SAGUEZ - La conduite du FX-702 P.  
 PELLIER - La conduite du PC 1500.  
 OROS et PERBOST - Jeux et applications pour ZX SPECTRUM.  
 GROS - Le grand livre du ZX SPECTRUM.  
 HARWOOD - La conduite du ZX SPECTRUM.  
 HARTNELL - Graphisme 3 D sur votre micro-ordinateur.  
 HARTNELL et JONES - Des extensions à construire pour votre ZX 81.  
 VULDY - Le Basic en gestion sur Apple II.  
 BOUQUEROD - La conduite du TI 99.  
 PINSON - Mon TI 99/4A.  
 WILLARD - Les fichiers en Basic sur micro-ordinateur.  
 CEYRAT - Pratiquez l'intelligence artificielle.  
 DELANNOY - VIC 20 à la conquête des jeux.  
 AUBERT - La conduite du T 07.  
 PERBOST et MASSE - La conduite de l'ORIC-1.  
 TERRAL - Premiers pas en LOGO.  
 ASTIER - La conduite du COMMODORE 64.  
 MONTEIL - Tome 1 : Basic, graphisme et son.  
 MONTEIL - Tome 2 : Langage-machine entrées/sorties et périphériques.  
 OROS - La conduite de l'ATARI 400/800.  
 WILLARD - TI 99 à la conquête des jeux.  
 KRUTCH - Expériences d'intelligence artificielle en Basic.

- ASTIER - ORIC-1 à la conquête des jeux.  
 PELLIER - Langage machine, trucs et astuces sur ZX SPECTRUM.  
 SAGUEZ et ANDRIEUX - Maîtrisez les interfaces de votre micro-ordinateur.  
 DE GEETER - Forth pour micros.  
 ASTIER - ATMOS à la conquête des jeux.  
 CROWTHER et HARTLEY - MO5 et TO7 à la conquête des jeux.  
 MONTEIL - Introduction à l'IBM-PC Junior.  
 OROS - La conduite des ATARI XL - 600 XL, 800 XL, 1245 XLD.  
 PERBOST - ZX SPECTRUM à la conquête des jeux.  
 BERNARD - L'assembleur facile du 6809.  
 GARREC et VIGNET - Des extensions à construire pour votre ORIC-ATMOS.

## LOGILIVRES EYROLLES (logiciels sur cassettes)

- PELLIER - Kamikaze (jeu pour ZX Spectrum).  
 PELLIER - Astéroïdes (jeu pour ZX Spectrum).  
 PELLIER - Othello/Isola (jeux pour ZX Spectrum).  
 PELLIER - Éditeur/Assembleur pour ZX Spectrum.  
 PERBOST et MASSE - VIC 20 version de base à la conquête des jeux.  
 HADDADI - Calcul des structures sur PC 1500/PC 2.  
 VANRYB et POLITIS - LYNX. Dictée musicale. Générateur de caractères.  
 ROSENTHAL - Résistance des matériaux sur ORIC-1 et ORIC-ATMOS.

## Avant-propos

*Qui n'a jamais été intrigué en déballant son dernier ordinateur ? Mais combien de questions sans réponses, après avoir lu le manuel ? Tout ce qui est présenté dans ce livre a été réalisé sur un ORIC-1 acheté banalement au revendeur du coin. Avec un peu de persévérance nous avons pu trouver le principe des Entrées-Sorties sur ORIC-1. La Sortie d'ORIC-ATMOS et de son excellent manuel a confirmé les résultats acquis.*

*Nous avons voulu, dans un premier temps, ne présenter que des interfaces propres : c'est-à-dire ne demandant aucune intervention à l'intérieur de l'ORIC. Mais la connaissance de l'engin aidant et le démon de l'informatique ressurgissant nous avons gonflé l'ORIC-1 en "presque" ORIC-ATMOS.*

*Ce livre peut-être une introduction à l'interfaçage d'un micro-ordinateur classique, l'ORIC étant pris comme exemple pratique. Nous avons voulu être le plus didactique possible : outre les connaissances du matériel ("Hardware") que vous pourrez acquérir, une partie est réservée à la programmation des circuits d'interfaces. En effet, le développement de l'électronique et de l'informatique met à la portée de l'amateur des circuits électroniques performants, pour le prix d'une place de cinéma.*

Si vous désirez être tenu au courant de nos publications, il vous suffit d'adresser votre carte de visite au :

Service « Presse » Éditions EYROLLES  
61, Boulevard Saint-Germain,  
75240 PARIS CEDEX 05,

en précisant les domaines qui vous intéressent. Vous recevrez régulièrement un avis de parution des nouveautés en vente chez votre libraire habituel.

*La rançon de la puissance de ces circuits est leur programmation: en effet une fois cablés ces composants ne fonctionnent pas, il est nécessaire d'écrire dans des registres (zones mémoires spécifiques au composant) pour qu'ils jouent leurs rôles.*

*Tout ceci ne sera pas présenté de façon abrupte mais progressivement à l'occasion du montage et de l'utilisation des différentes interfaces. Autrement dit, ce livre est un parcours logique de la première à la dernière page et nous vous conseillons de le lire avec méthode. Vous y gagnerez en compréhension, intérêt et connaissance globale.*

*D'autre part, nous avons voulu nous tourner résolument vers l'avenir: à côté de circuits d'utilisation classique mais éprouvés (certains diront "démodés" mais en électronique "démodé" veut dire qui n'existait pas il y a 10 ans!) nous avons mis en œuvre certains composants à peine commercialisés; il se peut que, lorsque vous lirez ces lignes, ces composants fussent déjà vieux! A ce propos, souvenez-vous que si certaines solutions ont été choisies quand nous avons réalisé ces interfaces, ce ne seront peut être pas les meilleures dans les années à venir. Quoiqu'il en soit tous les montages décrits fonctionnent, et cela même la mode ne pourra l'empêcher à l'avenir.*

*Un autre de nos soucis a été de limiter tout ce qu'on à l'habitude de nommer péjorativement la "quincaillerie", restant fidèle à nos amours et sachant que moins on fait d'électronique plus on fait d'informatique. Car ce livre se veut d'informatique et plus particulièrement de "Hardware", c'est-à-dire la frontière entre la science du fer à souder et celle de la boucle vicieuse.*

*Vous êtes prêts! Alors tournez la page.*

## Table des matières

<b>Avant-Propos</b> .....	IX
<b>Remerciements</b> .....	XIII
<b>1. Généralités sur ORIC</b> .....	1
1.1. Structure matériel et logiciel d'un micro-ordinateur .....	1
1.2. Structure de l'ORIC .....	2
1.3. Les différences entre ORIC-1 et ORIC-ATMOS .....	5
1.4. La ROM et les adresses du plan mémoire .....	6
1.5. Les circuits d'interfaces de l'ORIC .....	6
1.6. Le décodage de notre interface universelle .....	8
<b>2. PIA Interface adaptateur pour périphériques</b> .....	11
2.1. PIA 6821 de chez MOTOROLA .....	11
2.2. Programmation du boîtier en sortie .....	17
2.3. Programmation du boîtier en entrée .....	23
2.4. La manette de jeux .....	25
2.5. La conversion analogique-digitale: le voltmètre .....	28
2.6. Le lecteur programmeur d'EPROM .....	38
2.7. Le programmeur de 2764 .....	42
<b>3. Transmission des données en série</b> .....	45
3.1. ACIA 6850 de chez MOTOROLA .....	46
3.2. L'interface RS 232C .....	50
3.3. L'interface différentielle: RS 422 .....	60
3.4. Le Modem .....	63
3.5. Minitel .....	69

<b>4. Interface de puissance pour sortie "Centronics"</b> .....	<b>75</b>
4.1. Présentation de l'interface Centronics .....	76
4.2. La mise en œuvre sur ORIC-ATMOS .....	78
4.3. Les ordres BASIC associés .....	80
4.4. Interface de puissance pour Bus Centronics .....	81
4.5. Commande des canaux de puissance .....	86
4.6. Les programmes de mise en œuvre .....	90
4.7. Réalisation de la partie "Puissance" .....	93
4.8. Conclusions et conseils .....	99
<b>5. L'ORIC à cœur ouvert</b> .....	<b>103</b>
5.1. Et l'ORIC devient ATMOS .....	103
5.2. Rephasage de Ø2 .....	108
5.3. Pannes en vrac .....	113
<b>6. Conseils de réalisation pratique</b> .....	<b>115</b>
6.1. Comment utiliser les schémas .....	115
6.2. Un peu d'outillage n'a jamais manqué .....	118
6.3. Réalisation des alimentations .....	119
6.4. Conseils pratiques .....	122
<b>Annexe 1 : Adresses des périphériques et de leurs registres</b> .....	<b>124</b>
<b>Annexe 2 : La fonction TIME de l'ORIC et de l'ATMOS</b> .....	<b>125</b>
<b>Bibliographie</b> .....	<b>126</b>

## Remerciements

Nous tenons à remercier toutes les personnes qui par leurs soutiens matériel et moral ont rendu possible la réalisation de cet ouvrage et plus particulièrement mademoiselle N. GROSSET et Messieurs A. CACCAMO, R. SCHOMBERG, C. TOMASIN.

Nous tenons à souligner l'aide technique apportée par monsieur G. MARTINEZ de TEXAS INSTRUMENT France.

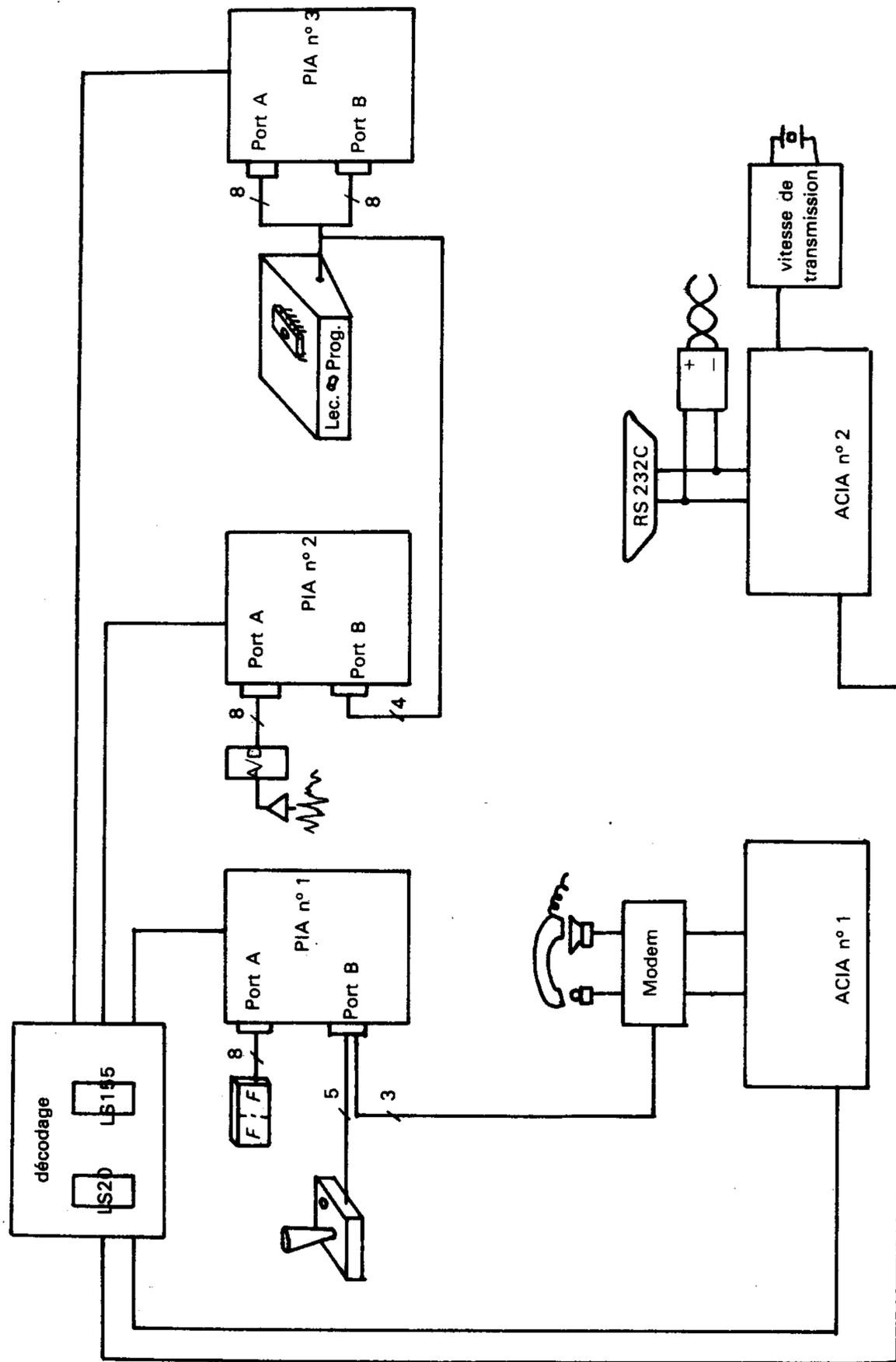


Schéma synoptique de l'interface universelle.

# 1

## Généralités sur ORIC

### 1.1. STRUCTURE CLASSIQUE D'UN MICRO-ORDINATEUR MATÉRIEL ET LOGICIEL

Reprenons tout d'abord certaines généralités sur les micro-ordinateurs. Depuis l'avènement des microprocesseurs, l'architecture générale d'un ordinateur personnel est sensiblement la même pour tous les modèles.

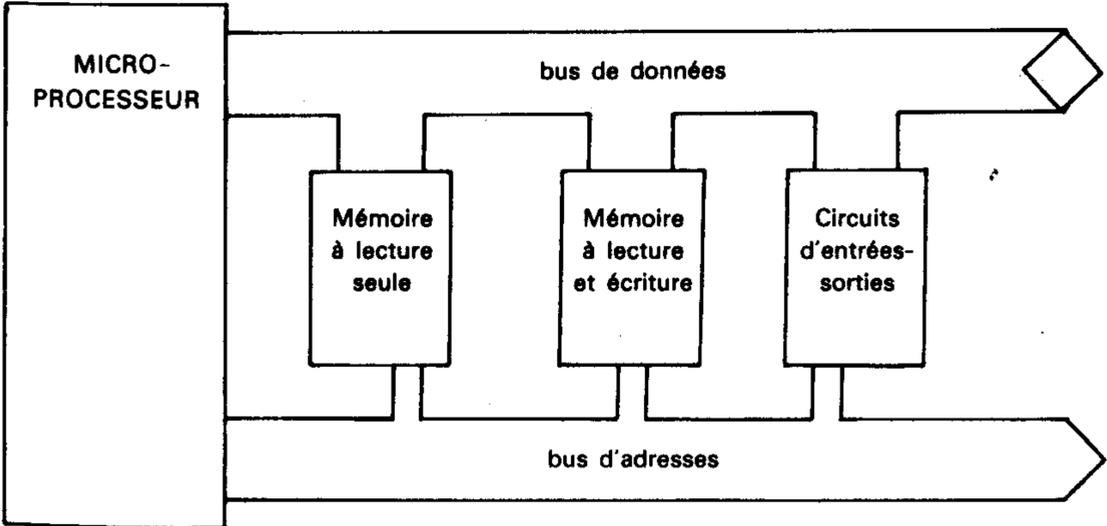


Fig. 1.1. — Structure de micro-ordinateur classique

Le microprocesseur séquence et exécute les instructions contenues dans la mémoire à lecture seule. Pour réaliser certaines instructions, il est nécessaire de connecter une mémoire à lecture et écriture (RAM). La transmission des informations se fait par le bus de données; tandis que la sélection des boîtiers et des cases-mémoires ou des registres internes des circuits se fait par le bus d'adresses et le circuit de décodage.

Enfin suivant le programme de la mémoire à lecture seule (appelé aussi ROM) il peut être nécessaire de dialoguer avec l'extérieur (le contraire serait étonnant mais pas impossible). Ceci se fait grâce à des circuits périphériques ou d'interfaces [circuits de gestion du clavier, affichage d'écran, etc].

## 1.2 STRUCTURE DE L'ORIC

Maintenant voyons un peu comment les concepteurs de l'ORIC ont agencé les boîtiers (fig. 1.2).

On retrouve la trilogie classique, CPU, RAM, ROM déservie par le bus d'adresses et le bus de données et un circuit spécifique ORIC. Avec les progrès technologiques d'intégration des composants l'emploi de circuits spécifiques se généralise pour les micro-ordinateurs des dernières années. Ce circuit élabore à partir des adresses les signaux de décodage, pilote et rafraîchit les 64 K octets de mémoire vive, et gère une partie de l'interface avec l'écran vidéo.

Puis pour communiquer avec l'extérieur l'ORIC utilise un circuit d'interface programmable le 6522 VIA (Versatile Interface Adapter). Ce circuit combine d'une certaine mesure, les performances d'une PIA (Paralell Interface Adapter) et d'une ACIA (Asynchronous Communication Interface Adapter).

Pour ceux qui ont un ORIC-ATMOS, tout ce qui suit est expliqué, quant au principe, au chapitre des entrées-sorties. Pour les possesseurs d'ORIC-1, ouvrez votre manuel à la page 161.

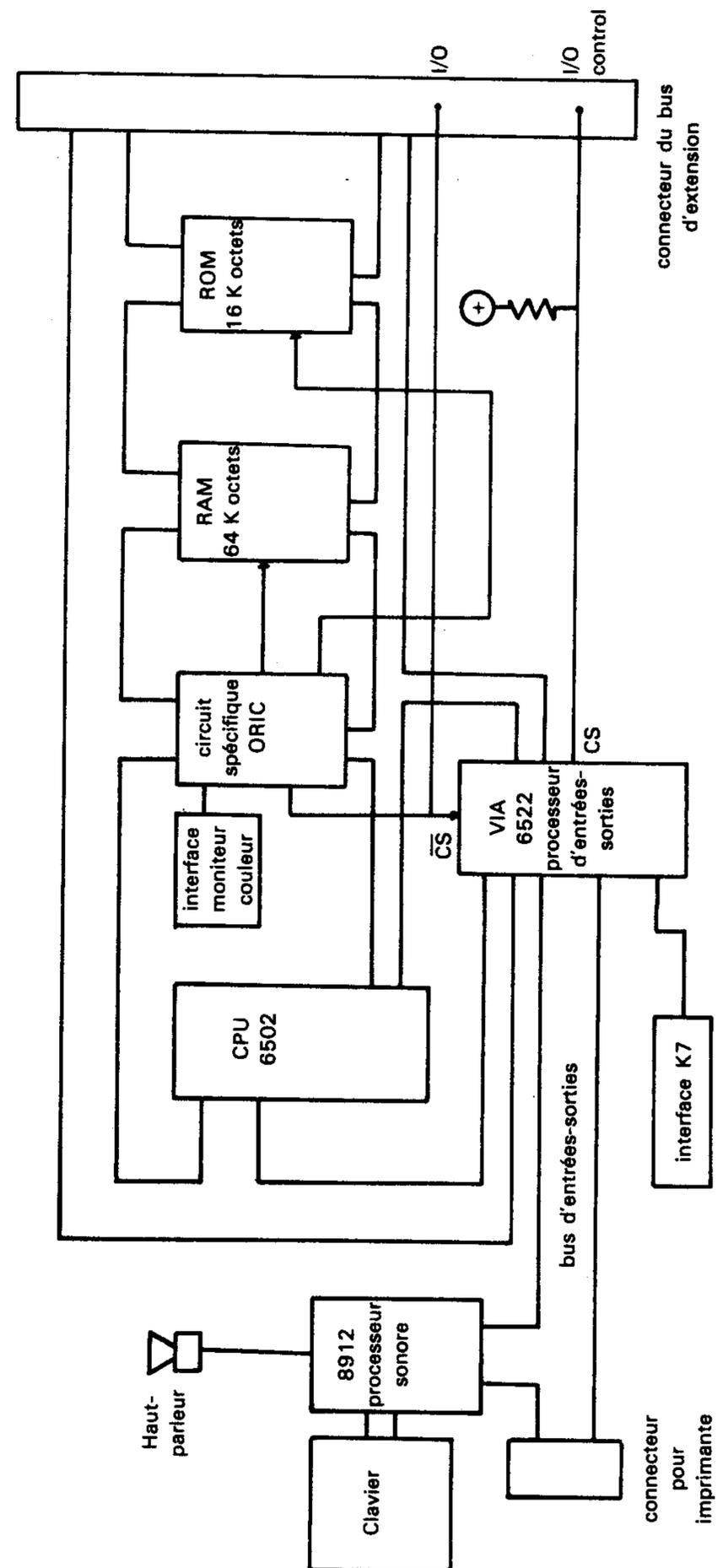


Fig. 1.2. — Schéma général de l'ORIC.

page 4	S/P → 4200	0500
page 3	adresses entrées/sorties	0400
page 2	variables en mode RUN	0300
page 1	Pile	0200
page 0	utilisée	0100
		0000

La page 3 qui commence à partir de #300 jusqu'à #3FF s'appelle entrées-sorties. Il serait juste de supposer que la VIA 6522 est sélectionnée pour ces adresses. Pour être sûr que cette intuition était juste, un petit coup d'œil sur le livre VISA POUR ORIC. On peut lire parmi les adresses utiles pour le magnéto #302 et pour le clavier #306 et #307. Or ces deux périphériques passent par la VIA pour dialoguer avec le CPU.

Or les Data sheets (feuilles de spécifications du constructeur) de la VIA expliquent que ce circuit comporte 16 registres; donc la VIA est adressée entre #300 et #30F.

Regardons de nouveau le synoptique général de l'ORIC. Pour bien comprendre deux fils ont été représentés. La VIA 6522 peut être sélectionnée par deux entrées :

CS (chip select) désélection du boîtier si CS=0.  
 $\overline{CS}$  désélection du boîtier si CS=1.

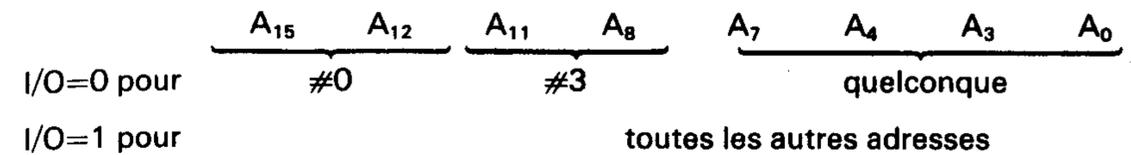
CS	$\overline{CS}$	
1	1	désélectionné
1	0	sélectionné
0	1	désélectionné
0	0	désélectionné

Différentes configurations possibles pour l'adressage de la VIA.

Quand le bus d'extension n'est pas utilisé une résistance connecte le CS au plus 5 V donc CS=1 (logique). La sélection du boîtier est déterminée par la broche  $\overline{CS}$  reliée à la broche I/O du circuit spécifique. Or

celui-ci met la broche I/O à zéro pour toutes adresses comprises entre #300 et #3FF.

Autrement dit, la broche I/O décode les adresses hautes :



Sachant que le logiciel de l'ORIC n'utilise que 16 registres dans la page 3 (correspondants aux 16 registres de la VIA). Les adresses de #310 à #3FF sont libres.

Pour des facilités de décodage nous utiliserons pour nos interfaces la plage d'adresses de #3F0 à #3FF.

La lecture du Manuel ORIC-ATMOS nous a conforté dans ce choix (fait plusieurs mois auparavant). En effet les périphériques ORIC utilisent ou utiliseront les adresses suivant un principe ascendant : c'est-à-dire en partant de #310 pour aller vers #3FF. Il y a donc très peu de chance que notre interface rentre en conflit avec les périphériques ORIC.

### 1.3. LES DIFFÉRENCES ENTRE ORIC-1 ET ORIC-ATMOS

A présent quelques mots sur la différence entre l'ORIC-1 et l'ATMOS. Il n'y a aucune différence matérielle entre les 2 versions : c'est-à-dire que le circuit imprimé est le même. La grande différence réside dans la ROM. Les 16 K de programme pour gérer le clavier, le BASIC, les entrées-sorties sont plus "finies" et plus évoluées sur l'ATMOS. Certaines erreurs ou omissions de l'ORIC ont été réparées. C'est cette similitude qui sera utilisée dans un des derniers chapitres pour transformer à moindre frais votre ORIC-1 en ORIC-"ATMOS".

L'autre changement réside dans le clavier qui a une touche supplémentaire (dont l'usage ne paraît pas encore défini).

## 1.4. LA ROM ET LES ADRESSES DU PLAN MÉMOIRE

Pour en revenir à l'utilisation des entrées-sorties les 2 ROM (ORIC-ATMOS) sont compatibles; une partie de nos réalisations ayant été réalisée et testée sur ORIC-1 et désormais utilisée sur notre "ATMOS".

## 1.5. LES CIRCUITS D'INTERFACE DE L'ORIC

### Le bus d'extension

L'interface proposée ne nécessitant aucune transformation de l'ORIC, voyons la liaison avec l'extérieur: elle se matérialise sous la forme d'un connecteur mâle 2 x 17 points pour cable plat (norme HE10 pas de 2,54 x 2,54).

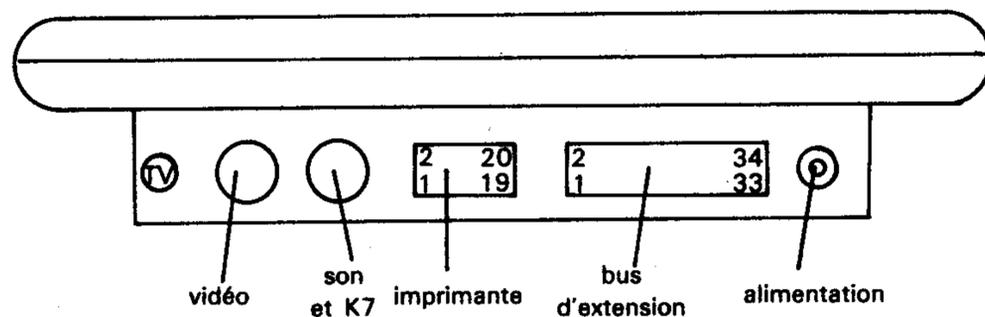


Fig. 1.3. — Face arrière de l'ORIC.

Voici maintenant la description complète du bus d'extension.

De A<sub>15</sub> à A<sub>0</sub>: Ces fils représentent le bus d'adresses.

De D<sub>7</sub> à D<sub>0</sub>: bus de données.

+ 5 volt: Alimentation de l'ORIC. Ne sera pas utilisé pour notre interface.

Masse: Masse électrique reliée à la masse électrique de l'interface.

I/O: Décodage de la page 3.

Memory Map	1	2	ROM disable
horloge Ø2	3	4	Reset
I/O	5	6	I/O control
R/W	7	8	IRQ
D2	9	10	D0
A3	11	12	D1
A0	13	14	D6
A1	15	16	D3
A2	17	18	D4
D5	19	20	A4
A5	21	22	D7
A6	23	24	A15
A7	25	26	A14
A8	27	28	A13
A9	29	30	A12
A10	31	32	A11
+ 5 volt	33	34	Masse

Fig. 1.4. — Bus d'extensions.

I/O control: Désélection de la VIA pour éviter les conflits "Hardware".

Horloge Ø2: Horloge du 6502. Utilisée pour synchroniser les circuits de l'interface.

R/ $\bar{W}$  (read/write): Indique si l'instruction du 6502 est une opération de lecture ou d'écriture.

Reset: Remise à zéro à la mise sous tension. N'a aucun rapport avec le bouton poussoir.

ROM disable: Broche qui peut être utilisée pour déconnecter la ROM. Les mini-disquettes ORIC utilisent cette broche car une autre ROM est placée à l'extérieur dans les mini-disquettes.

Memory Map: Permet de déconnecter la mémoire vive.

IRQ: Signal d'interruption vers le 6502. Méfiance, son utilisation n'est pas triviale.

Tous les signes présentés ici seront aussi utilisés pour tous les schémas.

## 1.6. LE DÉCODAGE DE NOTRE INTERFACE UNIVERSELLE

Voyons maintenant la méthode de décodage.

On a vu que les adresses  $A_{15}$  à  $A_8$  sont décodées pour #03 par I/O.

Le groupe d'adresses  $A_7A_6A_5A_4$  est décodé avec une porte NAND à quatre entrées (1/2 74LS20) dont la sortie est à zéro pour  $A_7A_6A_5A_4 \neq F$ . De plus cette sortie est reliée à la broche I/O control. Donc la VIA est désélectionnée pour  $A_7A_6A_5A_4 \neq F$  ce qui n'est pas problématique car lorsque le logiciel de l'ORIC utilise la VIA on a  $A_7A_6A_5A_4 = O$ . Le signal est ensuite inversé pour avoir un état actif à l'état haut (CS).

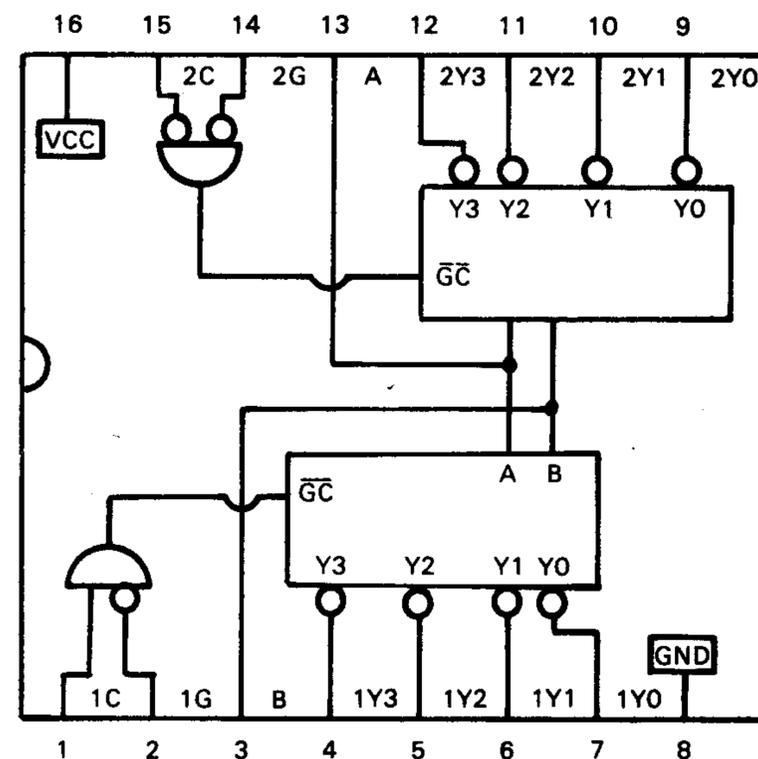
La deuxième partie du décodage est faite par un demi 74LS155 dont le schéma et la table de vérité sont donnés figure 1.5.

Vu le câblage, le signal I/O est propagé sur quatre sorties en fonction des combinaisons de  $A_3A_1$ . Le signal I/O est actif à l'état bas ( $\overline{CS}$ ).

Pour un boîtier ayant une entrée CS et une autre  $\overline{CS}$  (identique aux broches de sélection de la VIA) on a ainsi créé quatre emplacements mémoires sélectionnés ainsi que l'indique le schéma 1.6.

Les adresses  $A_1A_0$  ne sont pas utilisées pour le décodage mais pour la sélection des registres internes des circuits d'interfaces.

Pour réaliser le câblage de cette partie — ainsi que pour les autres — voir le dernier chapitre qui rappelle aussi les symboles employés dans les schémas.



Entrées			Sorties				
B	A	1G	1C	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

B	A	2G	2C	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

Fig. 1.5. — Schéma logique et table de vérité du 74LS155.

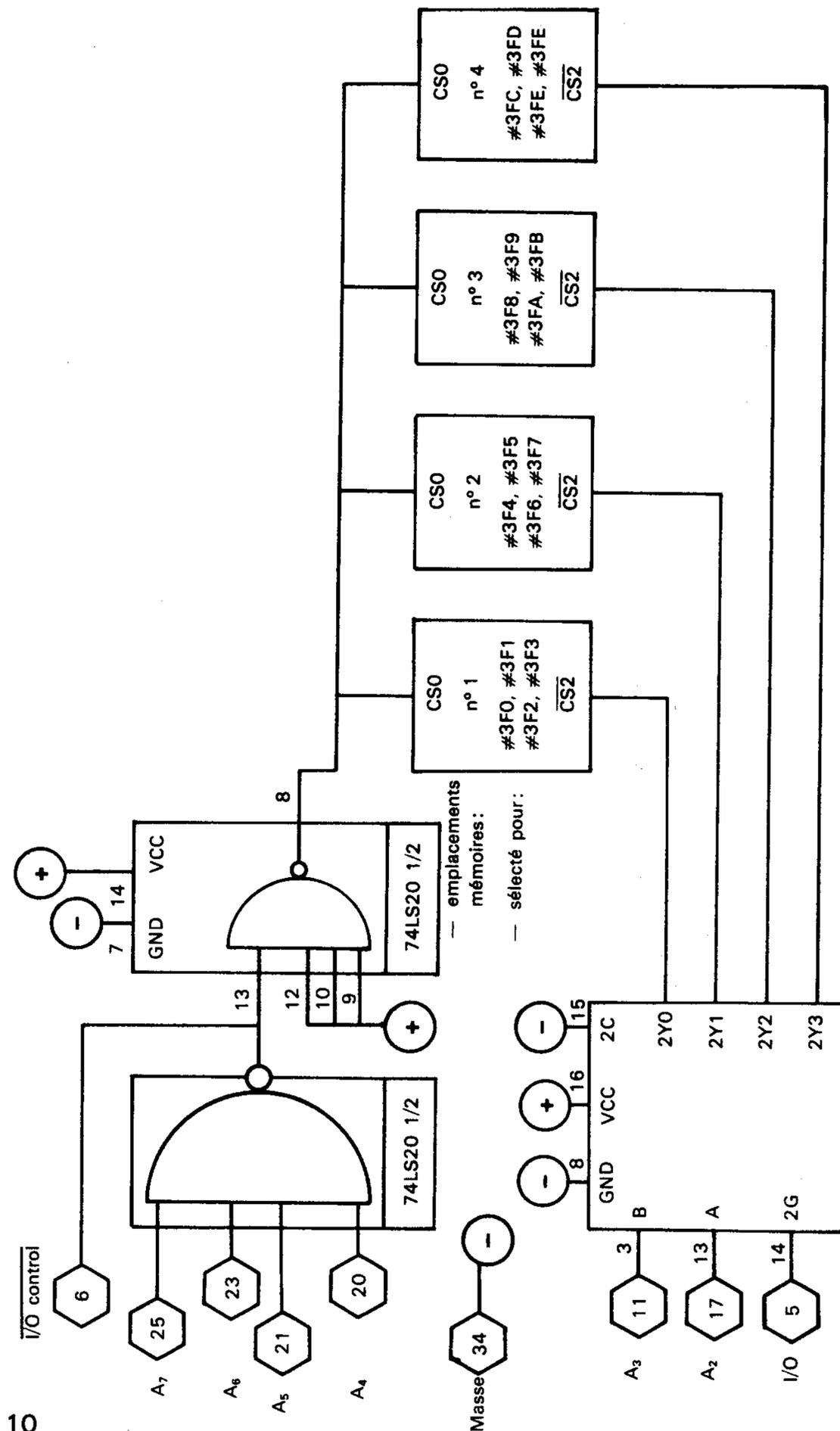


Fig. 1.6. — Schéma du décodage des 4 plans mémoires.

## 2

# PIA : Interface Adaptateur pour Périphériques

### 2.1. LA PIA EMPLOYÉE EST LE BOÎTIER 6821 DE CHEZ MOTOROLA

La PIA est un boîtier qui permet à l'ordinateur de converser avec l'extérieur, c'est en quelque sorte les sens tactiles et les mains de l'ordinateur.

Dans chaque boîtier, il y a 16 ports d'entrées-sorties répartis en :

- 8 ports de PA<sub>0</sub> à PA<sub>7</sub>,
- 8 ports de PB<sub>0</sub> à PB<sub>7</sub>.

Chaque port pouvant servir à commander un périphérique même si celui-là est une simple diode électroluminescente encore appelée LED.

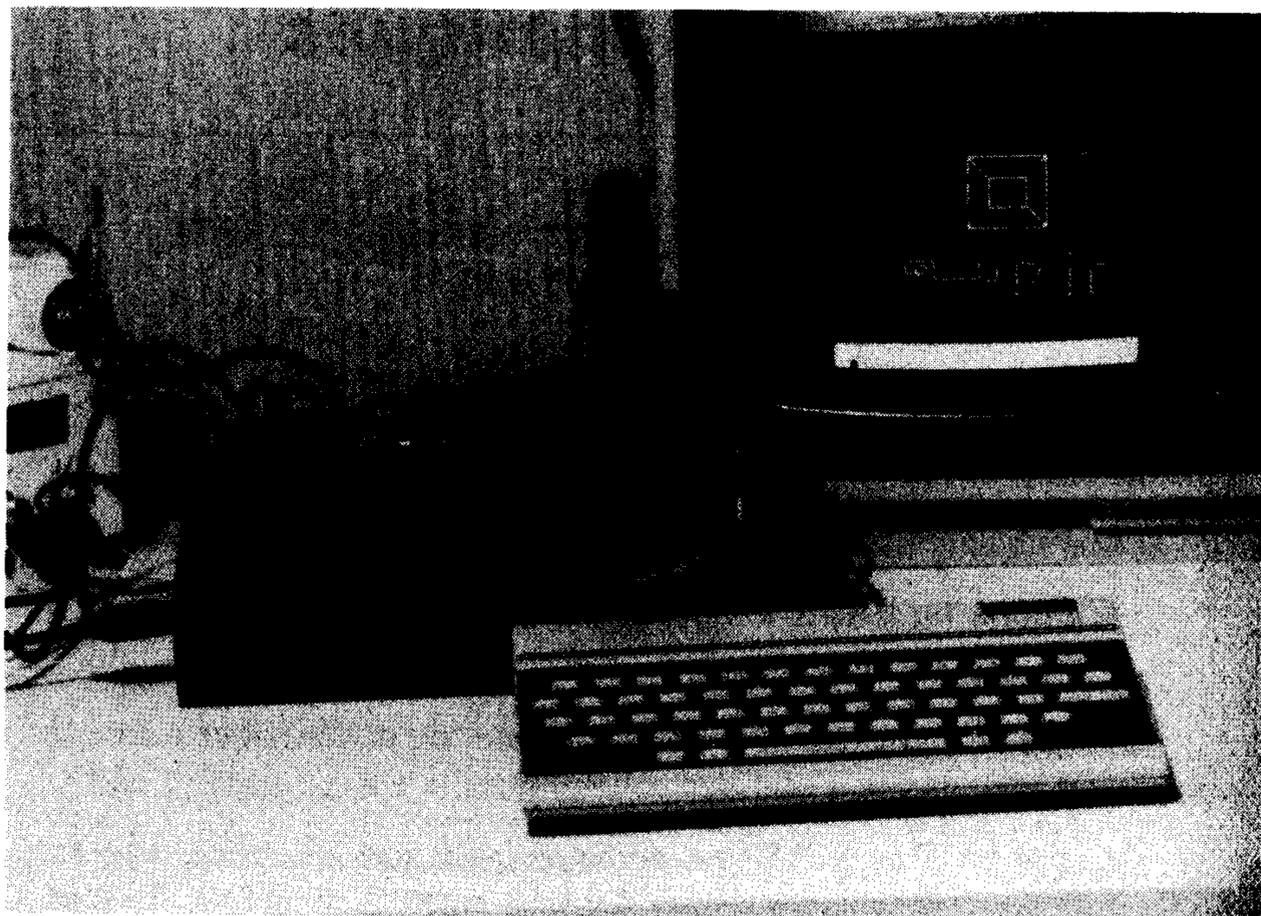
Chaque bit PA<sub>0</sub> à PA<sub>7</sub> est commandable soit en entrée, soit en sortie, indépendamment des autres. Le plus souvent, toutefois, ils sont tous programmés dans la même direction et dans ce cas on parle de ports parallèles.

Il est donc possible de connaître la position de huit interrupteurs simultanément si les ports sont déclarés en entrée et d'analyser la signification des différentes combinaisons possibles comme nous le verrons pour la manette de jeux. L'on peut aussi commander des automates ou des robots si les ports sont déclarés en sortie. Ces robots peuvent être des

robots ménagers comme une cafetière électrique ou une plaque de cuisinière.

Nous pouvons même effectuer la régulation du chauffage grâce à un capteur qui indiquera à l'un des ports (mis en lecture) la valeur de la température, l'ordinateur peut réagir en fonction de cette donnée et d'une autre valeur mise en mémoire, appelée consigne.

Il arrête le chauffage si la température dépasse la valeur de la consigne et le remet en marche lorsqu'elle y est inférieure. La consigne peut varier en fonction de l'heure et de la date; ce n'est pas la peine de chauffer autant lorsqu'on n'est pas chez soi!



### Description du boîtier proprement dit

Par rapport au calculateur, le boîtier de la PIA se comporte comme quatre cases mémoires, une fois le décodage effectué comme il est décrit dans les généralités. Il y a une case-mémoire, encore appelée registre de données, réservée au port A et une autre pour le contrôle de ce port; il en est de même pour le port B.

Le registre de données est une "boîte aux lettres" dans laquelle l'ORIC vient inscrire des données par un simple poke à l'adresse et l'on vient les lire grâce à un peek à la même adresse.

Le registre de contrôle sert entre autre à accéder au registre sens de transfert de la PIA.

Les entrées de sélection, encore appelées CS (chip select), servent à valider le boîtier. Si celui-là n'est pas validé, il est en haute impédance vis-à-vis du bus de données c'est-à-dire que tout ce qui arrivera sur la PIA alors que le boîtier ne sera pas sélectionné, ne perturbera pas le bus de données.

La figure 2.1 montre un schéma synoptique de la PIA.

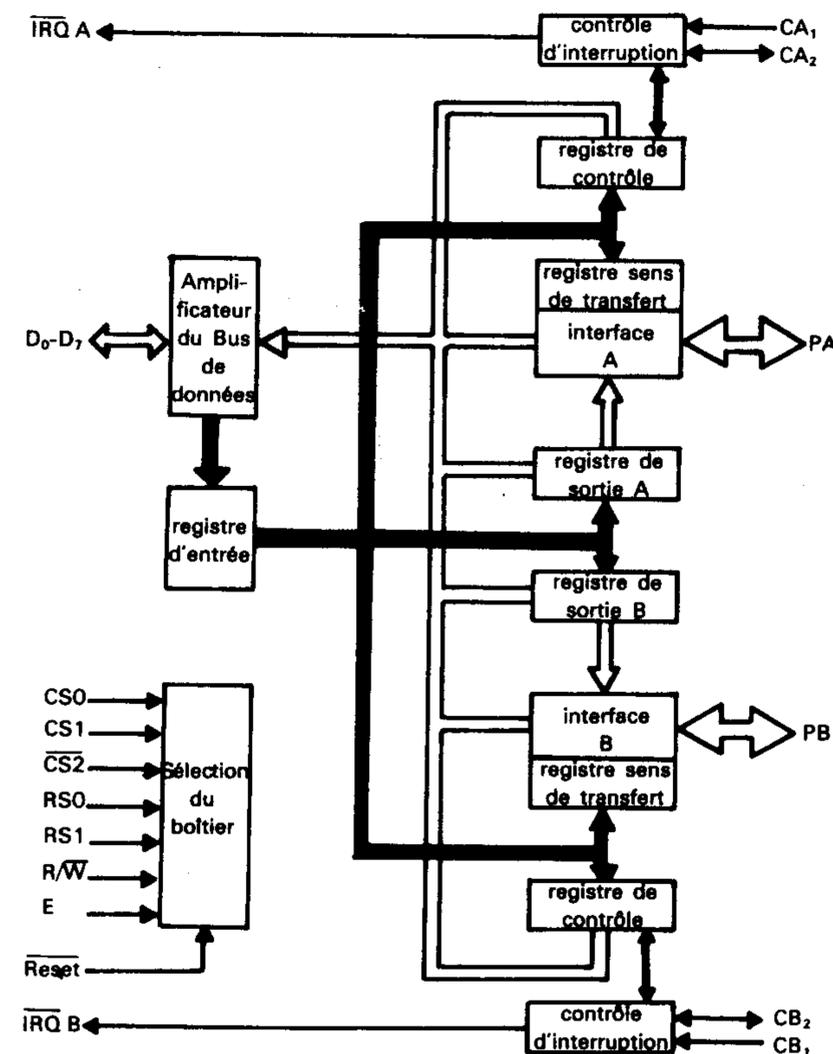


Fig. 2.1. — Schéma synoptique de la PIA.

Pour que le boîtier soit sélectionné, il faut que l'on ait les conditions suivantes :  $CS0=1$ ,  $CS1=1$ ,  $\overline{CS2}=0$ . Si l'une de ces conditions n'est pas vraie, le boîtier n'est pas adressé. Le schéma de câblage d'une PIA est donné en figure 2.2.

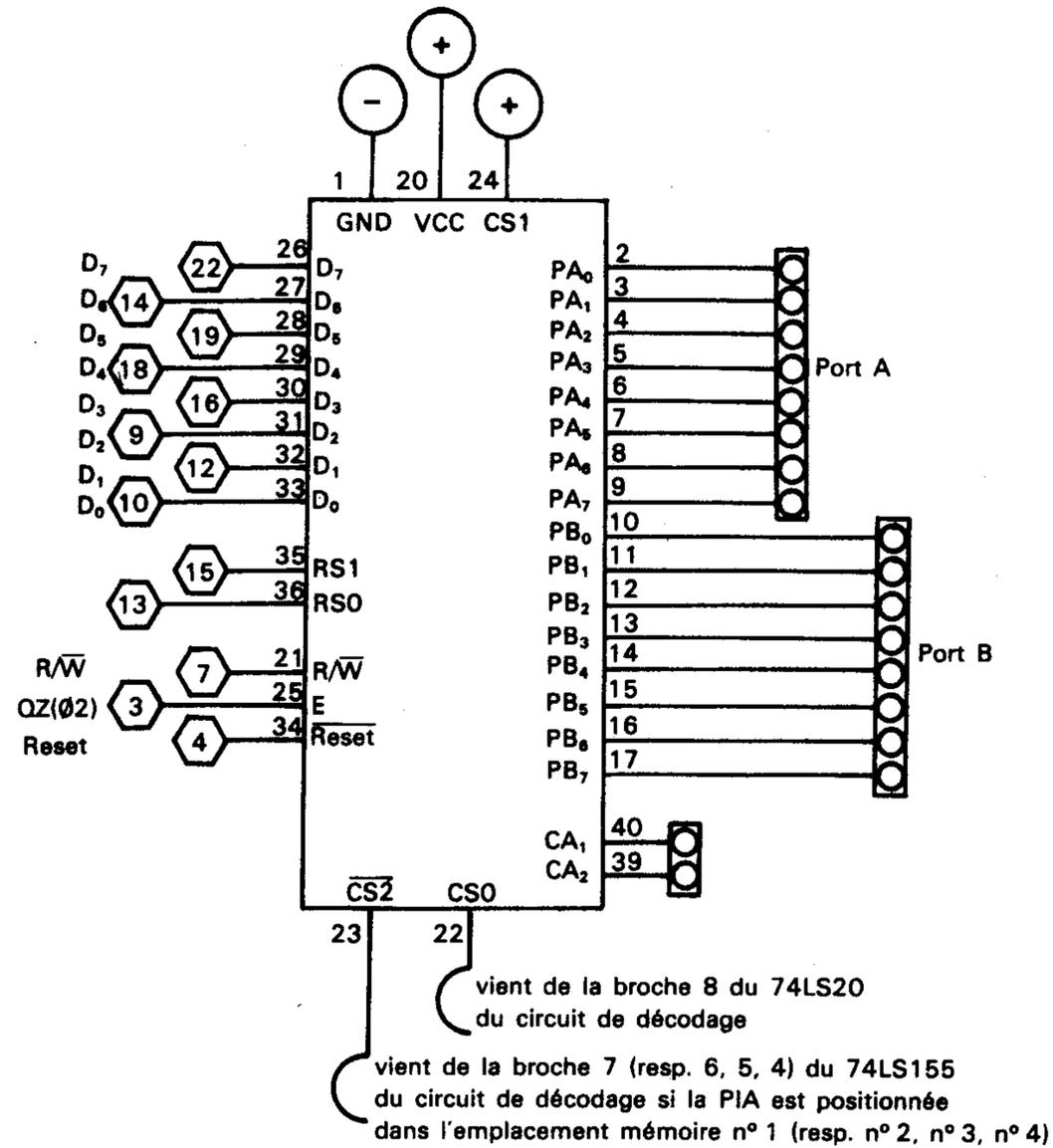


Fig. 2.2. — Schéma de câblage d'une PIA.

Une fois le boîtier sélectionné, le transfert des données pourra s'effectuer en écriture ou en lecture grâce au choix lecture/écriture et à l'aide de l'horloge  $\Phi_2$ . Les données doivent être stables pendant la durée de l'impulsion de l'horloge. Cette horloge est l'horloge  $\Phi_2$  de l'ORIC.

Les deux lignes RS0 et RS1 servent à choisir l'un des six registres internes de la PIA, c'est-à-dire les deux registres de données PA et PB, les registres sens de transfert et les deux registres de contrôle CONT A et CONT B (tabl. I).

TABLEAU I. — Sélection des registres.

RS 1	RS 0	Choix du registre
0	0	Registre de données de la périphérie PA Registre sens de transfert de PA
0	1	Registre de contrôle de A
1	0	Registre de données de la périphérie PB Registre sens de transfert de PB
1	1	Registre de contrôle de B

Dans cette interface nous avons câblé 3 PIA. Voici les adresses des cases-mémoires utiles pour l'interface présentée :

TABLEAU II. — Adresses des registres des PIAs.

PIA n° 1	#3 F0	données	port A
	#3 F1	contrôle	
	#3 F2	données	port B
	#3 F3	contrôle	
PIA n° 2	#3 F4	données	port A
	#3 F5	contrôle	
	#3 F6	données	port B
	#3 F7	contrôle	
PIA n° 3	#3 F8	données	port A
	#3 F9	contrôle	
	#3 FA	données	port B
	#3 FB	contrôle	

Dans ces cases-mémoires rajoutées à l'ordinateur, l'ensemble du jeu d'instructions sera possible, c'est-à-dire que l'on pourra stocker des valeurs à l'aide des instructions poke#adresse ou venir prendre le contenu à l'aide de l'instruction A=Peek (#adresse).

### Détermination du sens de transfert des données

Pour initialiser le sens de transfert des ports de la PIA, nous utilisons un subterfuge qui justifie le fait de n'avoir que quatre cases-mémoires en tout dans la PIA.

Pour accéder au registre sens de transfert, il faut deux instructions. Il faut commencer par mettre à zéro le bit  $CRA_2$  du registre de contrôle du port A (respectivement le bit  $CRB_2$  pour le port B) puis aller inscrire dans le registre de données le sens de transfert que l'on a choisi, ceci individuellement pour chaque élément du port considéré. Pour pouvoir accéder ensuite au registre de données, il faut mettre à 1 le bit  $CRA_2$  du registre de contrôle de A (respectivement  $CRB_2$  pour le port B).

### 2.2. PROGRAMMATION DU BOÎTIER EN SORTIE

Nous allons prendre un exemple : les huit bits du port A de la PIA 1 vont être déclarés en sortie et vont servir à allumer des diodes (fig. 2.3).

Les LED sont commandées en inverse et sont en série avec une résistance de  $470 \Omega$  pour limiter le courant à 10 mA par diode c'est-à-dire que lorsque l'un des ports de la PIA sera forcé à zéro par programmation, la diode aura l'une de ses bornes au +5 V par l'intermédiaire de la résistance de limitation de courant et l'autre à la masse par le port du PIA.

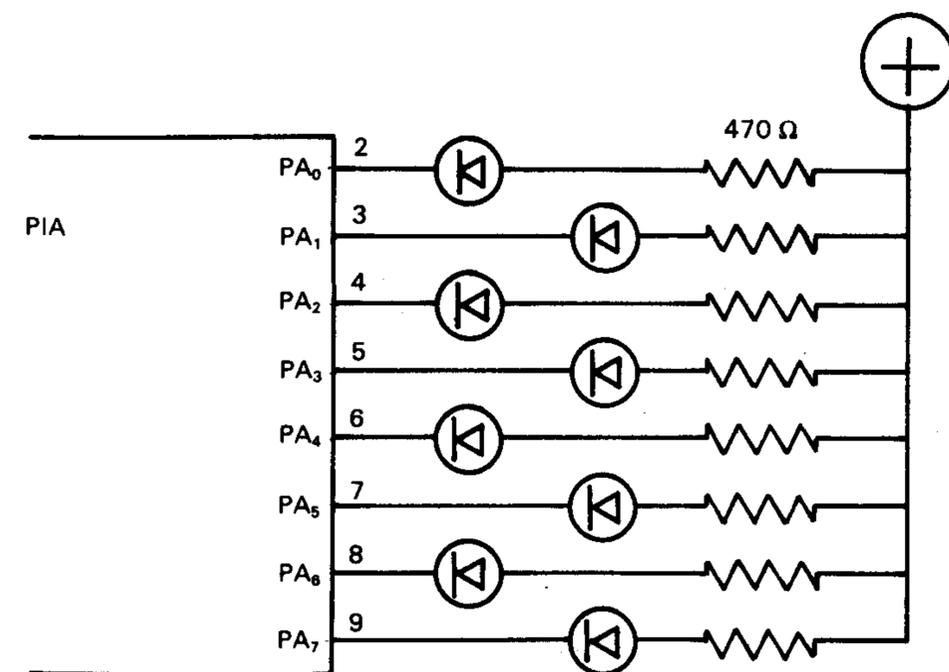


Fig. 2.3. — Commande de leds par la PIA.

A cause de cette différence de potentiel un courant de 10 mA circulera dans la LED qui pourra s'allumer.

Si le port de la PIA est à 1 c'est-à-dire au potentiel 5 V, les deux bornes de la diode seront alors au même potentiel, la diode sera éteinte.

La raison d'un tel montage est liée aux caractéristiques des boîtiers TTL qui ont un courant de sortie plus important à l'état bas qu'à l'état haut.

Pour la PIA le courant à  $V_0 = 1,5$  V est de 10 mA tandis qu'à l'état haut il n'est que de quelques  $\mu$ A.

Lorsque l'on met à 1 les bits du registre sens de transfert, le bit considéré est positionné en sortie. De même si on le met à zéro, il est positionné en entrée.

Registre sens de transfert	
0	entrée
1	sortie

*Exemple :*

Poke #3F1,0 met 0 dans le registre de contrôle ce qui va permettre d'accéder au registre de sens de transfert.

Poke #3F0,7 déclare les bits PA<sub>0</sub>, PA<sub>1</sub>, PA<sub>2</sub> en sortie et les bits PA<sub>3</sub>, PA<sub>4</sub>, PA<sub>5</sub>, PA<sub>6</sub>, PA<sub>7</sub> en entrée.

Car 7 s'écrit :

0 0 0 0 0 1 1 1 en hexadécimal  
 PA<sub>7</sub> PA<sub>6</sub> PA<sub>5</sub> PA<sub>4</sub> PA<sub>3</sub> PA<sub>2</sub> PA<sub>1</sub> PA<sub>0</sub>

Le but maintenant est de faire un chenillard à diodes électroluminescentes (ou LED).

Le programme sera le suivant :

```

1 REM  " "TEST DES LEDS"
10 POKE#3F1,0
20 POKE#3F0,255
30 POKE#3F1,255
40 POKE#3F0,I
45 PRINT I
50 I=2^N
60 N=N+1
70 IF N>8 THEN N=0 : I=0
80 WAIT 25
90 GOTO 40
  
```

**Programme 1. — Tests des Leds.**

*Ligne 10 :* Tous les bits du registre de contrôle sont mis à zéro donc le bit CRA<sub>2</sub> est aussi mis à zéro (les autres bits ne servent pas pour cette application : ils servent pour commander CA<sub>1</sub> et CA<sub>2</sub> et régler le fonctionnement en interruption). La mise à zéro de CRA<sub>2</sub> permet d'accéder au registre sens de transfert. L'instruction Poke# sert donc à mettre un 0 à l'adresse #3F1.

*Ligne 20 :* On veut déclarer tous les bits du port A de la PIA 1 en sortie, il faut donc mettre à 1 tous les bits du registre sens de transfert.

PA <sub>7</sub>	PA <sub>6</sub>	PA <sub>5</sub>	PA <sub>4</sub>	PA <sub>3</sub>	PA <sub>2</sub>	1	PA <sub>0</sub>	
1	1	1	1	1	1	1	1	= FF en hexadécimal or FF en hexadécimal = = 255 en décimal.

*Ligne 30 :* Lors du prochain accès, on voudra pouvoir écrire dans le registre de données : il faut donc positionner le bit CRA<sub>2</sub> à 1 pour pouvoir y accéder, les autres bits étant indifférents on peut les mettre à 1 aussi.

*Ligne 40 :* On va inscrire la valeur de I dans le registre de données. Lors du démarrage du programme, toutes les variables sont mises

à zéro donc I est nul. Tous les bits du port étant déclarés en sortie, si on met un zéro dans le registre de données, ils vont tous passer à zéro donc toutes les diodes vont s'allumer.

Il suffit d'écrire la valeur de I en hexadécimal et on connaît de cette manière quels seront les bits à 1 donc les diodes correspondantes éteintes et les bits à zéro donc les diodes correspondantes allumées.

*Exemple: I = 253*

253 s'écrit :

PA <sub>7</sub>	PA <sub>6</sub>	PA <sub>5</sub>	PA <sub>4</sub>	PA <sub>3</sub>	PA <sub>2</sub>	PA <sub>1</sub>	PA <sub>0</sub>	
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	
1	1	1	1	1	1	0	1	en binaire

seul le bit de poids 2<sup>1</sup> est à zéro, PA<sub>1</sub> est à zéro, seule la diode correspondant à PA<sub>1</sub> sera allumée.

*Ligne 50:* I prendra ses valeurs dans l'ensemble des puissances de 2 c'est-à-dire qu'il sera égal à :

2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
PA <sub>7</sub>	PA <sub>6</sub>	PA <sub>5</sub>	PA <sub>4</sub>	PA <sub>3</sub>	PA <sub>2</sub>	PA <sub>1</sub>	PA <sub>0</sub>
128	64	32	16	8	4	2	1

Quand I prendra l'une de ces valeurs, la diode correspondante au bit à 1 sera éteinte.

*Ligne 60:* On incrémente N de une unité.

*Ligne 70:* Pour ne pas sortir de l'ensemble de définition de I et donc risquer de faire un poke d'une valeur supérieure à 255, on remet I à zéro pour permettre la combinaison "toutes les LED allumées" ainsi que N à zéro.

*Ligne 80:* Temps d'attente facultatif.

*Ligne 90:* Rebouclage du programme en ligne 40.

Cette application bien qu'élémentaire permet de tester le câblage des PIA. De fait, il suffit de monter ces LED sur un connecteur enfi-

chable et d'essayer ce programme en changeant les adresses des registres de contrôle et des registres de données par ceux des ports à tester.

*Exemple:* Si on désire tester le port B de la PIA 2, on remplace dans le programme précédent l'adresse #3F0 par l'adresse #3F6 et l'adresse #3F1 par l'adresse #3F7.

Pour des utilisateurs plus fortunés, il est possible de remplacer les diodes par deux afficheurs à sept segments comme il est montré sur le schéma (fig. 2.4). La valeur de I sera alors affichée clairement en hexadécimal. Ces afficheurs sont des TIL 311 de chez TEXAS INSTRUMENT. Ils comportent un décodeur intégré et un verrouillage par bascules, consomment l'équivalent d'une charge TTL et disposent d'une alimentation séparée pour les diodes des afficheurs.

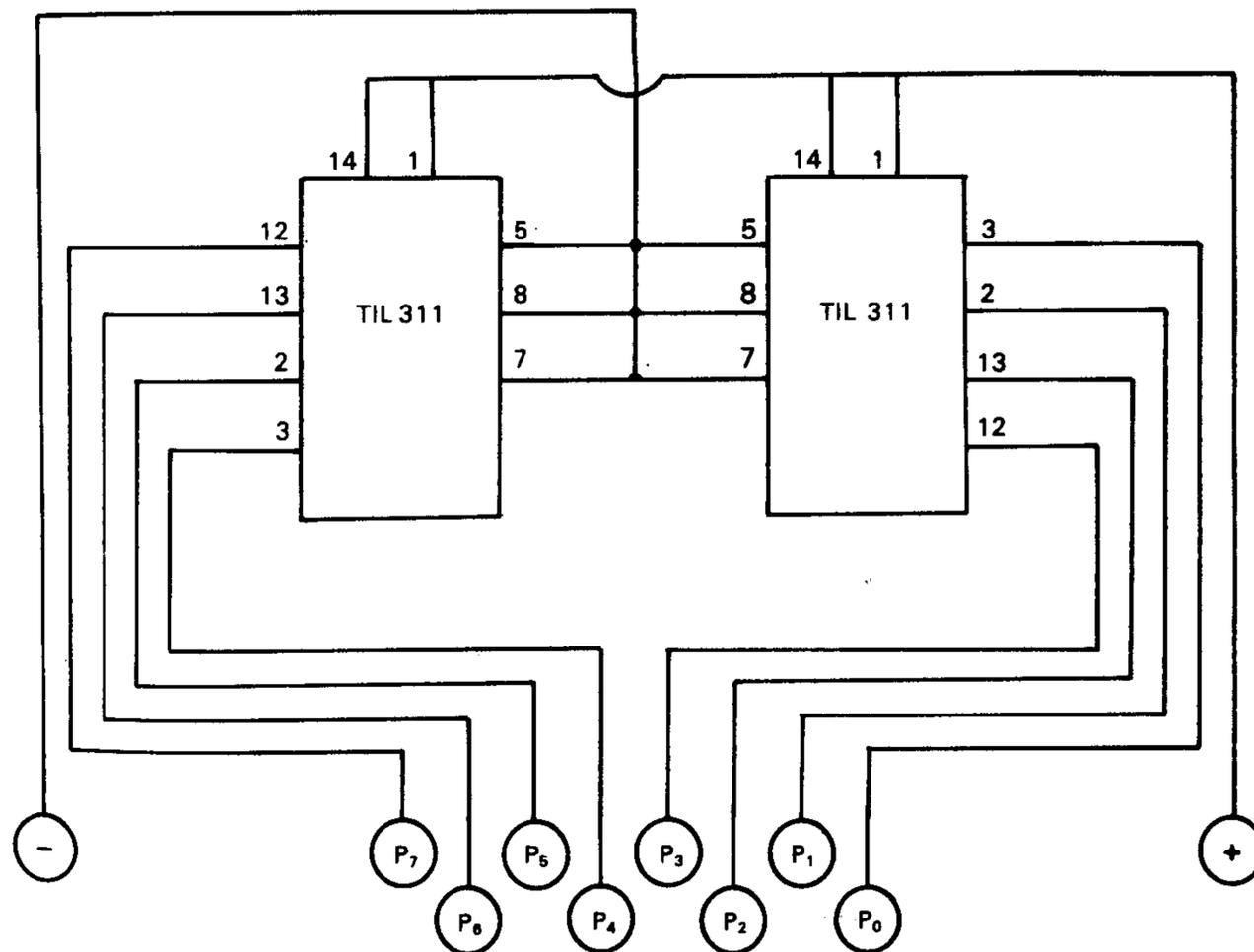


Fig. 2.4. — Câblage des afficheurs.

On veut fabriquer un convertisseur décimal en binaire sur huit bits.

Le programme est le suivant :

```
1 REM"TEST EN HEXADECIMAL"  
10 POKE#3F1,0  
20 POKE#3F0,255  
30 POKE#3F1,255  
40 INPUT"NOMBRE A CONVERTIR";I  
50 POKE#3F0,(255-I)  
60 GOTO40
```

### Programme 2. — Tests en hexadécimal.

*Ligne 10:* Positionner le bit  $CRA_2$  du registre de contrôle à 0 pour accéder au registre sens de transfert.

*Ligne 20:* Tous les bits sont déclarés en sortie.

*Ligne 30:* Remettre le bit  $CRA_2$  à 1 ce qui donne accès au registre de données.

*Ligne 40:* Entrée du nombre à convertir.

*Ligne 50:* Mettre dans le registre de données le complément à 1 du nombre à transcrire.

*Exemple:* Si  $I = 16$  en ligne 50 on aura  $255 - 16 = 239$

239 s'écrit 1 1 1 0 1 1 1 1

On aura donc la LED  $PA_4$  allumée et  $16 = 2^4$ .

*Ligne 60:* Renvoi à la ligne 40.

## 2.3. PROGRAMMATION DU BOÎTIER EN ENTRÉE

Nous allons maintenant utiliser la PIA pour effectuer une entrée. Pour cela, il faudra utiliser quelques interrupteurs. L'ordinateur ORIC viendra lire la PIA et affichera la valeur des interrupteurs.

Cette valeur pourra ensuite être utilisée à des fins ludiques telles que changer la fréquence d'un buzzer, bouger un curseur ou autres (fig. 2.5).

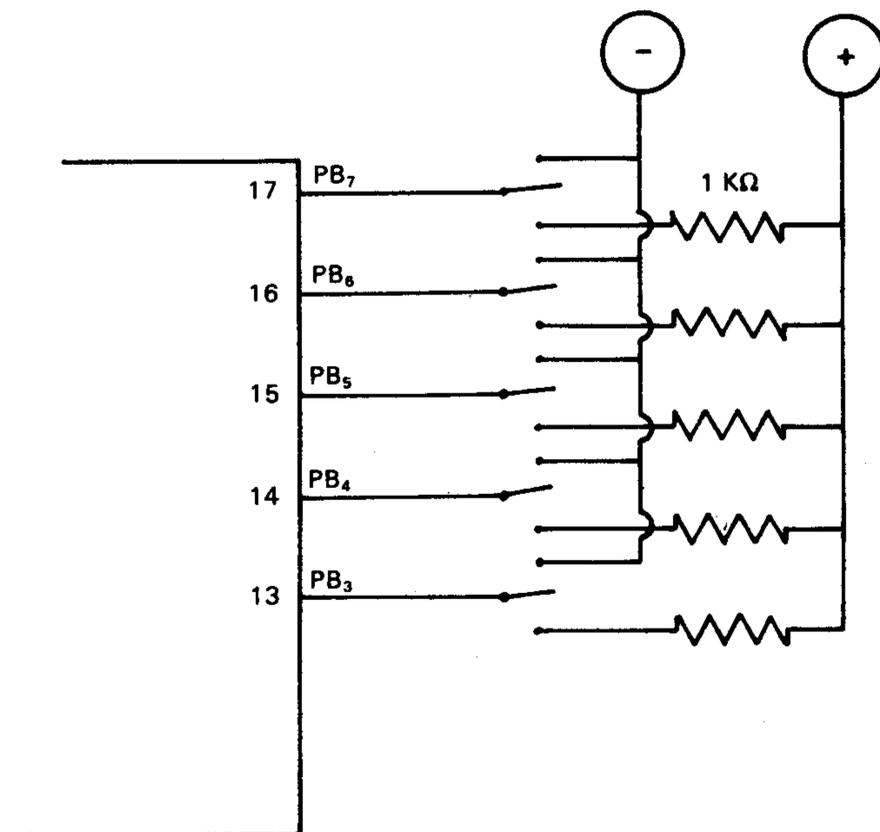


Fig. 2.5. — Lecture d'interrupteurs par PIA.

Nous utilisons les bits  $PB_3$ ,  $PB_4$ ,  $PB_5$ ,  $PB_6$ ,  $PB_7$  du port B de la PIA n° 1.

Lorsque l'interrupteur est ouvert, la résistance de rappel au +5 V force la ligne du port au 1 logique. Cette résistance est utile en cas de défaillance de l'interrupteur ; en cas de court-circuit, le courant est limité à quelques milliampères ; cette petite précaution permet d'éviter des fumées nauséabondes.

Le programme que nous vous proposons n'a qu'un but pédagogique. Nous allons effectuer la lecture des interrupteurs et l'afficher sur l'écran. Ces interrupteurs pourront prochainement être ceux de la manette de jeux et les informations seront alors utilisées de façon plus judicieuse.

```

1 REM"LECTURE DES INTERRUPTEURS"
10 POKE#3F3,0
20 POKE#3F2,7
30 POKE#3F3,255
40 A=PEEK(#3F2)
50 PRINTA
60 GOTO40

```

### Programme 3. — Lecture des interrupteurs.

*Ligne 10:* Positionner le bit CRB<sub>2</sub> du registre de contrôle de la PIA n° 1 à zéro pour permettre l'accès au registre sens de transfert des données.

*Ligne 20:* Positionner les ports PB<sub>7</sub>, PB<sub>6</sub>, PB<sub>5</sub>, PB<sub>4</sub>, PB<sub>3</sub> en entrée ; les ports PB<sub>0</sub>, PB<sub>1</sub>, PB<sub>2</sub> en sortie.

PB <sub>7</sub>	PB <sub>6</sub>	PB <sub>5</sub>	PB <sub>4</sub>	PB <sub>3</sub>	PB <sub>2</sub>	PB <sub>1</sub>	PB <sub>0</sub>	
0	0	0	0	0	1	1	1	= 7 en décimal
entrée					sortie			

Il suffit donc d'inscrire 7 dans le registre sens de transfert pour obtenir le résultat précité.

*Ligne 30:* Positionner le bit CRB<sub>2</sub> du registre de contrôle de la PIA n° 1 à 1 pour permettre cette fois-ci l'accès au registre de données. Les autres bits du registre de contrôle servant à contrôler des informations non utilisées pour cet exemple, les mettre tous à 1, ce qui revient bien à mettre 255 à l'adresse #3F3.

*Ligne 40:* A l'aide de l'instruction Peek, mettre dans A le contenu du registre de données de la PIA n° 1 du port B.

L'instruction Peek peut s'exécuter normalement sur le registre de données de la PIA ; celui-là est considéré comme une case-

mémoire un peu particulière car, dans ce cas, le contenu de la case-mémoire est la valeur des interrupteurs. Elle est donc chargée à l'extérieur.

*Ligne 50:* Afficher la variable A qui contient la position des interrupteurs, codée en binaire.

inter	inter	inter	inter	inter
5	4	3	2	1
PB <sub>7</sub>	PB <sub>6</sub>	PB <sub>5</sub>	PB <sub>4</sub>	PB <sub>3</sub>
1	0	0	0	1

Si seuls les interrupteurs 5 et 1 sont à 1, la valeur de A affichée sera  $2^7 + 2^1 = 128 + 2 = 130$ .

*Ligne 60:* Va effectuer une nouvelle lecture du registre de données.

Pour transposer ce programme et se servir d'un autre port d'une autre PIA, il suffit de changer les adresses du registre de contrôle et du registre de données par celles du port et de la PIA choisie.

Pour ce faire, il faut se référer aux adresses registres en page 16.

## 2.4. LA MANETTE DE JEUX

La manette de jeux choisie est de marque Archer (distribué par Tandy).

Cette manette est, en réalité, un ensemble de six interrupteurs agencés sous une forme de poignée. Deux des interrupteurs sont en parallèles : ce sont des interrupteurs de tir situés au sommet et au pied de la manette. Nous pouvons considérer que celui du pied de la manette est un bouton de rappel de tir. Les autres interrupteurs sont des microcontacts dissimulés dans la poignée de jeux : il y en a quatre (celui du haut, du bas, de droite et de gauche).

Le schéma de câblage est indiqué sur les figures 2.6 et 2.7.

(Pour ceux qui possèdent déjà la manette de jeux, il faut repérer les différents interrupteurs à l'aide d'un ohmmètre.)

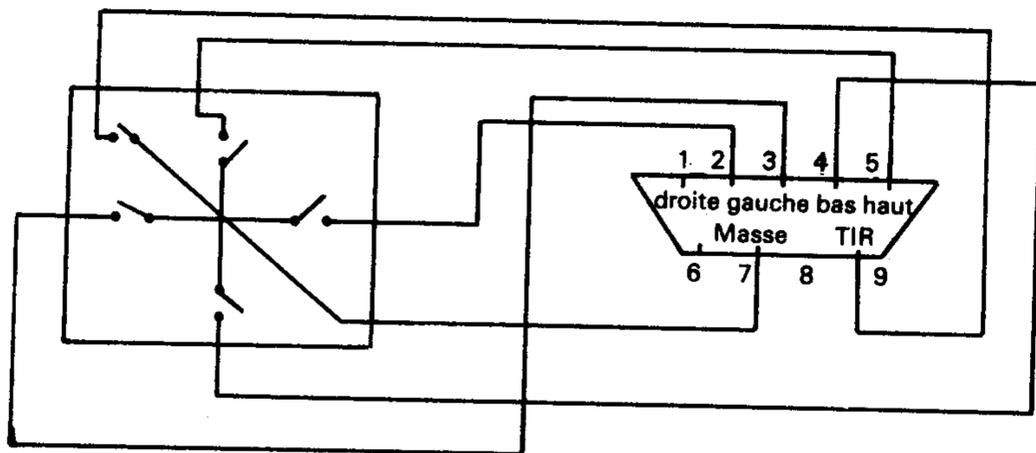


Fig 2.6. — Schéma d'une manette de jeu.

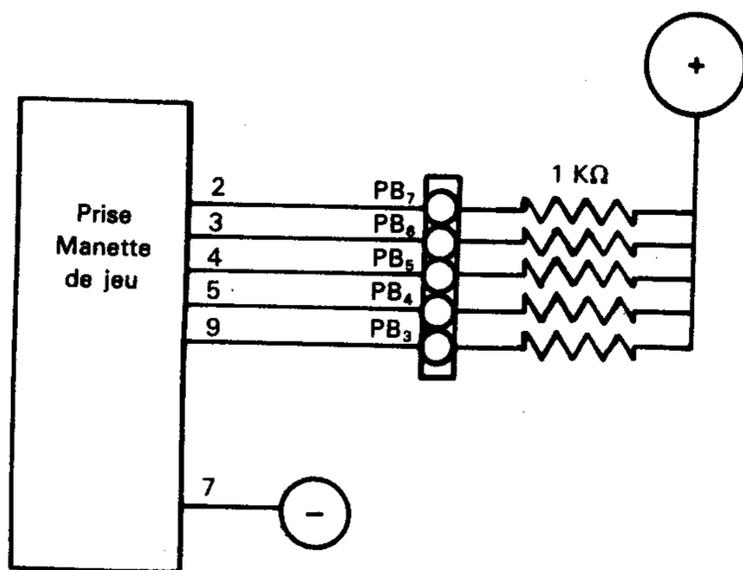


Fig. 2.7. — Câblage de la manette de jeu à la PIA.

On retrouve les résistances de rappel au 5 V de protection contre les court-circuits.

Le programme d'acquisition des informations de la manette de jeux est en tout point similaire à celui déjà réalisé pour la lecture des interrupteurs. En fait, on ne fait que lire des interrupteurs.

Placer la manette de jeux sur le port B de la PIA n° 1.

L'interrupteur vers la droite correspond à PB<sub>7</sub>.

L'interrupteur vers la gauche correspond à PB<sub>6</sub>.

L'interrupteur du bas correspond à PB<sub>5</sub>.

L'interrupteur du haut correspond à PB<sub>4</sub>.

L'interrupteur de tir correspond à PB<sub>3</sub>.

Si on veut mettre la manette sur un autre port ou une autre PIA, il suffit de changer les adresses des registres de contrôle et de données dans le programme suivant :

```

1 REM" MANETTE POUR DESSINER"
10 POKE#3F3,0
20 POKE#3F2,7
30 POKE #3F3,255
40 X=120
50 Y=100
60 HIRES
100 A=PEEK(#3F2)
110 B=INT(A/128)
120 C=INT((A-B*128)/64)
130 D=INT((A-B*128-C*64)/32)
140 E=INT((A-B*128-C*64-D*32)/16)
150 IF(A/16-INT(A/16))*2=0GOTO 300
200 X=X-B+C
210 Y=Y-D+E
250 IF X<0ORY<0ORX>239ORY>199 GOTO 10
260 CURSET X,Y,1
270 CLS:GOTO100
300 TEXT:CLS:GOTO10

```

#### Programme 4. — Manette pour dessiner.

Dans les variables A, B, C, D on récupère la valeur binaire indiquant, en logique inverse, si les interrupteurs sont fermés ou ouverts.

Suivant ces valeurs, on déplace le curseur vers la droite, le haut, le bas ou la gauche. Vous aurez transformé votre ORIC en ardoise magique.

Si vous désirez faire fonctionner ce programme plus rapidement, il faut que vous utilisiez l'assembleur du 6502.

Avec ce programme, vous avez le squelette de base nécessaire à l'articulation de tous vos jeux futurs.

Comme vous pouvez le constater, l'utilisation des PIA en programmation est relativement aisée dans les cas de figures énoncés précédemment.

## 2.5. LA CONVERSION ANALOGIQUE-DIGITALE : LE VOLTMÈTRE

Il y a des fonctions moins faciles à mettre en œuvre et que nous allons tâcher de vous exposer le plus clairement possible : ces fonctions sont les CA1, CA2, CB1, CB2 déclenchant sur fronts.

Elles vont nous servir à connecter un convertisseur analogique digitale qui vous sera utile lors de vos acquisitions futures.

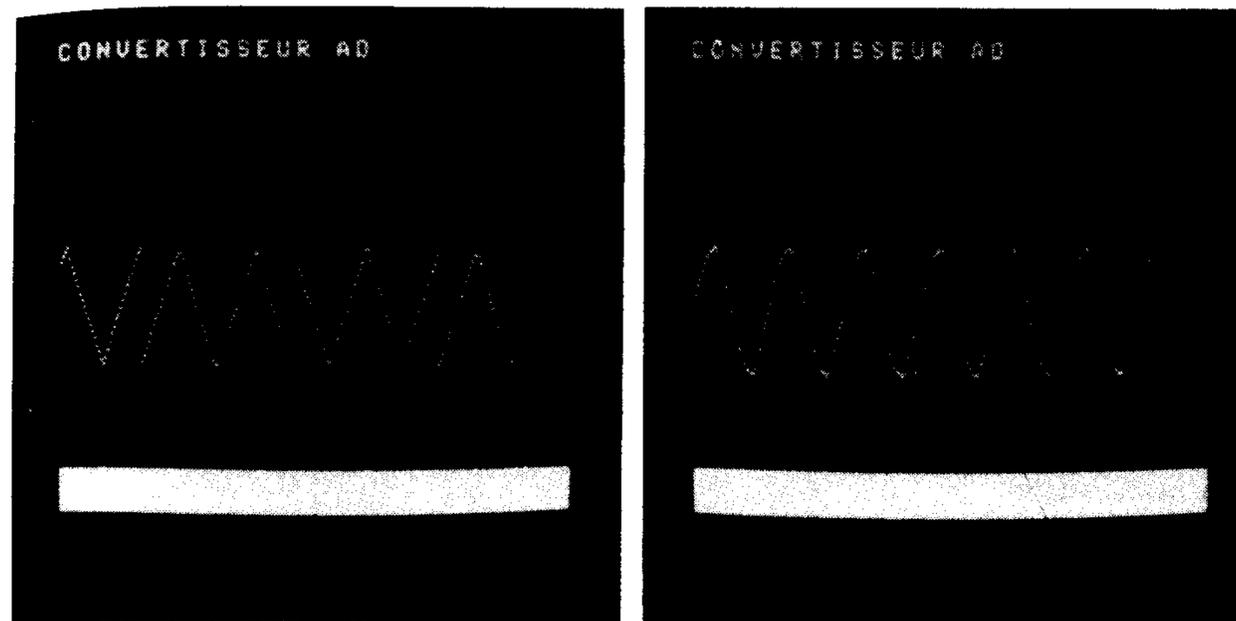
Pour l'instant, nous allons réaliser un voltmètre digital qui vous affichera la courbe sur l'écran si vous possédez un générateur sinusoïdal, sinon vous pourrez tester votre montage à l'aide d'un simple potentiomètre.

La plupart des capteurs ayant une sortie comprise entre +5 V et -5 V ou 0 V 10 V, le montage que nous vous proposons est idéal ; sachez que vous disposerez d'un résultat de conversion toutes les 25  $\mu$ s ce qui fait une fréquence de 40 000 Hz !

Le théorème de Shannon nous indiquant que pour reconstituer un signal il faut l'échantillonner à deux fois sa fréquence au moins, pratiquement le coefficient est de quatre ce qui donne une bande passante de 10 kHz.

Pour faire une acquisition à l'aide de l'ordinateur et la ranger en mémoire, il faut une vingtaine d'instructions en assembleur soit environ 30  $\mu$ s c'est donc le microprocesseur qui va limiter la vitesse.

Le convertisseur que nous allons vous proposer d'utiliser est de AD 570 de chez ANALOG DEVICE.



Sur l'entrée analogique nous ajoutons un ampli opérationnel câblé en suiveur pour adapter l'impédance. L'ampli opérationnel choisi est un TL 81 de chez TEXAS INSTRUMENT (schéma de câblage fig. 2.8).

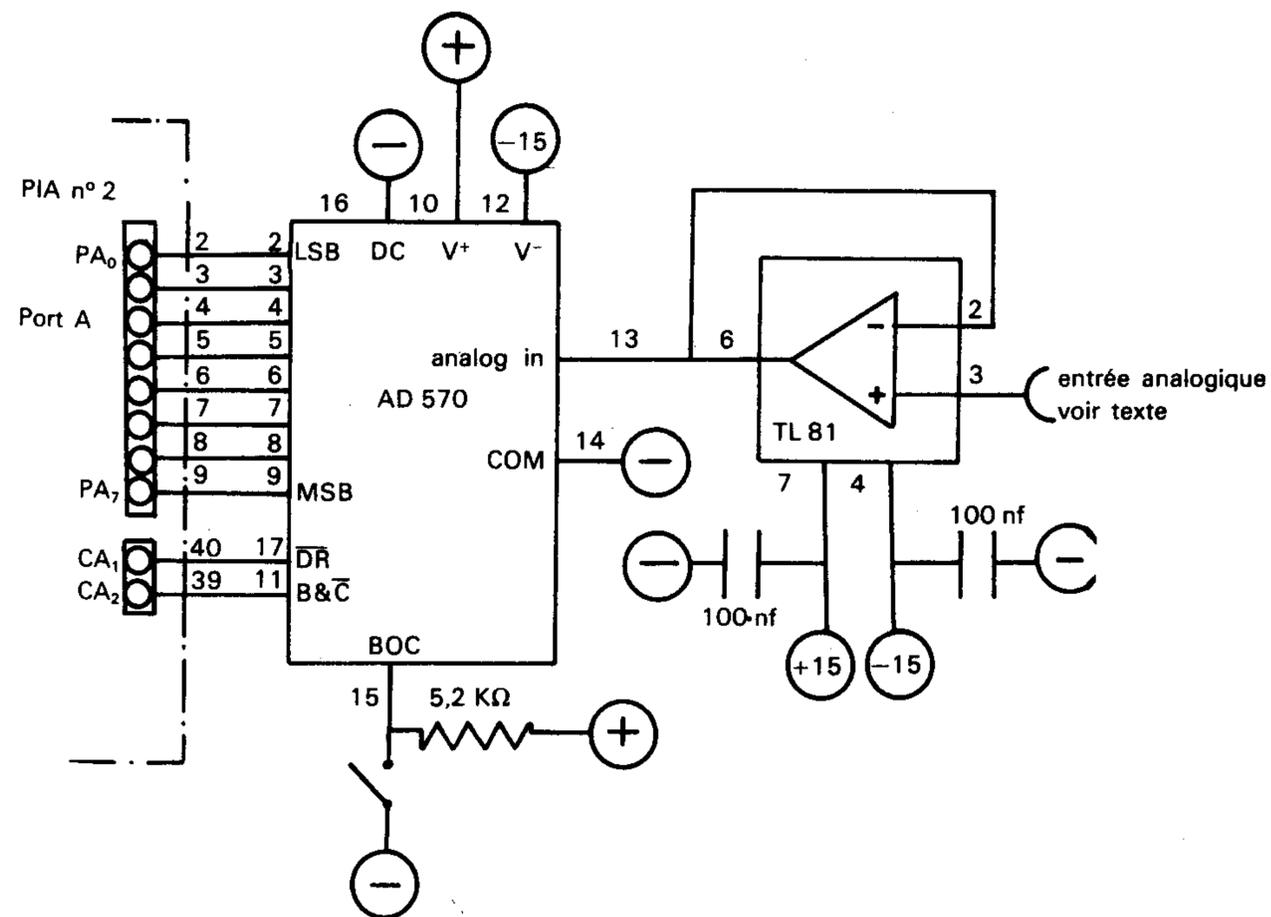


Fig. 2.8. — Câblage du convertisseur analogique digital AD 570.

Nous avons décidé d'employer la PIA n° 2 pour effectuer l'acquisition des données du convertisseur. Celui-là code la tension sur huit bits ce qui nous donne une précision de codage de 1/256.

L'excursion totale permise de la tension en entrée est de 10 Volts, on a donc 0,02 V d'erreur de codage.

Pour gérer les signaux spéciaux nécessaires à la conversion, nous avons employé les sorties CA1 et CA2 de la PIA n° 2.

La mise en œuvre de ces signaux est plus délicate au niveau de leur initialisation.

Comme nous le montre le schéma de l'AD 570, en dehors des diverses tensions d'alimentation et du bus de données connecté sur le port A de la PIA n° 2, il existe trois broches supplémentaires à signification non évidente.

La patte n° 15 sert à choisir la gamme de tension à convertir ; elle s'appelle bipolaire contrôle.

— Interrupteur fermé, on convertit une tension comprise entre 0 et 10 V.

— Interrupteur ouvert, on convertit une tension comprise entre -5 V et +5 V.

La résistance de 5,2 KOhms est une résistance de rappel au +5 V pour fixer la tension et éviter les court-circuits trop violents.

La broche  $\overline{DR}$  correspond à la sortie "data ready".

La broche B and  $\overline{C}$  correspond à l'entrée "blank and convert".

Ces deux signaux sont expliqués par le chronogramme (fig. 2.9).

Sur ce schéma on constate que, sur le front descendant de B and  $\overline{C}$ , la conversion analogique digitale commence. 25  $\mu$ s plus tard, le signal  $\overline{DR}$  indique que les données sont prêtes et que l'on peut les lire. On fait, alors, une lecture du port de donnée et on vient faire un blanking en mettant CA2 à 1 ; 1,5  $\mu$ s après le front montant de CA2, le signal  $\overline{DR}$  remonte à 1, on peut alors relancer une nouvelle conversion en abaissant CA2.

CA2 est donc utilisé comme un simple port d'entrée-sortie.

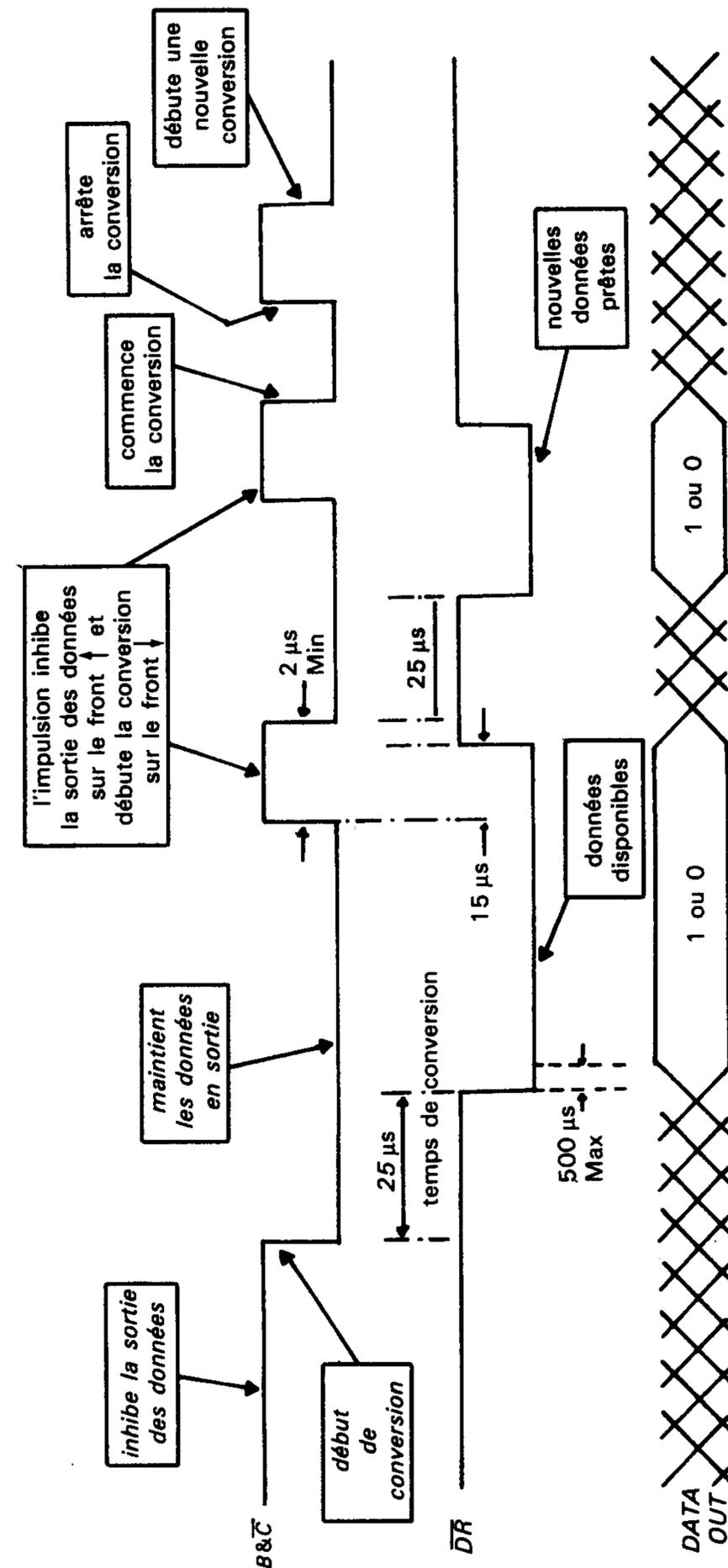


Fig. 2.9. — Chronogramme de l'AD 570.

Le tableau de contrôle de CA2 comme sortie est le suivant si on appelle CRA<sub>0</sub> à CRA<sub>7</sub> les bits du registre de contrôle de la PIA A.

TABLEAU III. — Tableau de fonctionnement de CA2.

CRA <sub>5</sub>	CRA <sub>4</sub>	CRA <sub>3</sub>	Mis à 0	CA2	Mis à 1
1	0	0	bas sur la transition négative de la première impulsion $\emptyset_2$ qui suit une lecture du registre de données de A.		haut quand l'indicateur d'interruption CRA <sub>7</sub> est mis à 1 par une transition active du signal CA1.
1	0	1	bas sur la transition négative de la première impulsion $\emptyset_2$ qui suit une lecture du registre de données de A.		haut sur la transition négative de la première impulsion $\emptyset_2$ qui se produit tandis que le circuit est désélectionné.
1	1	0	bas quand CRA <sub>3</sub> est mis à 0 par une écriture du registre de contrôle A.		toujours bas tant que CRA <sub>3</sub> est à 0. Passe à l'état haut quand CRA <sub>3</sub> est mis à 1 par une écriture du registre de contrôle de A.
1	1	1	toujours haut tant que CRA <sub>3</sub> est à 1. Passe à l'état bas quand CRA <sub>3</sub> est mis à zéro par une écriture du registre de contrôle de A.		haut quand CRA <sub>3</sub> est mis à 1 par une écriture du registre de contrôle de A.

Ce qu'il faut donc faire, c'est mettre CA2 à 0 ce qui donne :

CRA<sub>5</sub> CRA<sub>4</sub> CRA<sub>3</sub>  
1 1 0

Puis mettre CA2 à 1 ce qui donne :

CRA<sub>5</sub> CRA<sub>4</sub> CRA<sub>3</sub>  
1 1 1

Dans notre cas nous aurons toujours CRA<sub>5</sub> = CRA<sub>4</sub> = 1 et CRA<sub>3</sub> = position souhaitée de CA2. Avec CA1 on veut pouvoir détecter le data ready ( $\overline{DR}$ ), on va donc mettre CA1 en entrée.

Le tableau de contrôle des entrées d'interruption CA1 est le suivant.

On appelle toujours CRA<sub>0</sub> à CRA<sub>7</sub> les bits du registre de contrôle de A.

TABLEAU IV. — Tableau de fonctionnement de CA1.

CRA <sub>1</sub>	CRA <sub>0</sub>	Transition active de l'entrée d'interruption CA1	Indicateur d'interruption CRA <sub>7</sub>	Demande d'interruption du MPU IRQA
0	0	↓ active	mis à 1 sur ↓ de CA1	inhibé, IRQ reste à l'état haut.
0	1	↓ active	mis à 1 sur ↓ de CA1	passé à l'état bas quand l'indicateur CRA <sub>7</sub> passe à l'état haut.
1	0	↑ active	mis à 1 sur ↑ de CA1	inhibé, IRQ reste à l'état haut.
1	1	↑ active	mis à 1 sur ↑ de CA1	passé à l'état bas quand l'indicateur CRA <sub>7</sub> passe à l'état haut.

Le front actif de DR est un front ↓ (descendant); la ligne IRQ n'étant pas connectée, on va l'inhiber: la combinaison choisie est CRA<sub>1</sub> = CRA<sub>0</sub> = 0.

Il suffira de scruter le registre de contrôle de A et voir si le bit CRA<sub>7</sub> passe à 1 pour savoir que le signal DR est passé à zéro.

Si nous résumons un peu, la signification des bits du registre de contrôle de A dans le cas étudié est la suivante (tabl. V):

TABLEAU V. — Le registre de contrôle.

$CRA_7$	$CRA_6$	$CRA_5$	$CRA_4$	$CRA_3$	$CRA_2$	$CRA_1$	$CRA_0$
0 → les données ne sont pas prêtes 1 → les données sont prêtes	indifférent	1	1	1 → CA2 mis à 1 0 → CA2 mis à 0	1 → accès au registre de données 0 → accès au registre de contrôle	0	0

Le programme que nous allons faire maintenant va afficher en continu la valeur de la tension analogique connectée à l'entrée de l'ampli opérationnel suiveur TL 81. Si vous mettez un potentiomètre de 5 KOhms sur l'entrée analogique (fig. 2.10).

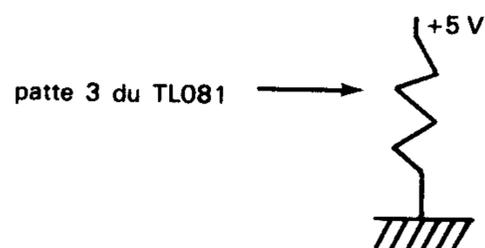


Fig. 2.10. — Montage de test pour l'AD 570.

Vous aurez la possibilité de visualiser les variations de tension occasionnées par votre potentiomètre.

Pour les applications futures vous pourrez, par exemple, connecter un LM 35 de chez NATIONAL SEMICONDUCTEUR sur l'entrée analogique. Le LM 35 donne en sortie une tension directement proportionnelle à la température ambiante. Ce boîtier n'est pas encore disponible sur le marché en juin 84. Nous vous donnons donc ses caractéristiques techniques en préliminaire ainsi que les raisons qui ont guidé notre choix.

## Le LM 35 de chez NATIONAL SEMICONDUCTEUR

L'avantage du LM 35 par rapport aux autres capteurs de température, c'est sa simplicité de mise en œuvre et sa bonne précision.

La courbe de réponse du composant est linéaire et commence à 0°C, la pente est de 10 mV/°C, la consommation du boîtier est de 60 μA ce qui est négligeable et entraîne une dissipation parasite infime à l'intérieur du boîtier.

La précision de la température est de 0,5°C environ.

La sortie U sortie est à connecter à l'entrée de votre voltmètre. Dans ce cas, en multipliant par 100 la valeur mesurée, c'est la valeur directement en degré celcius qui peut être affichée.

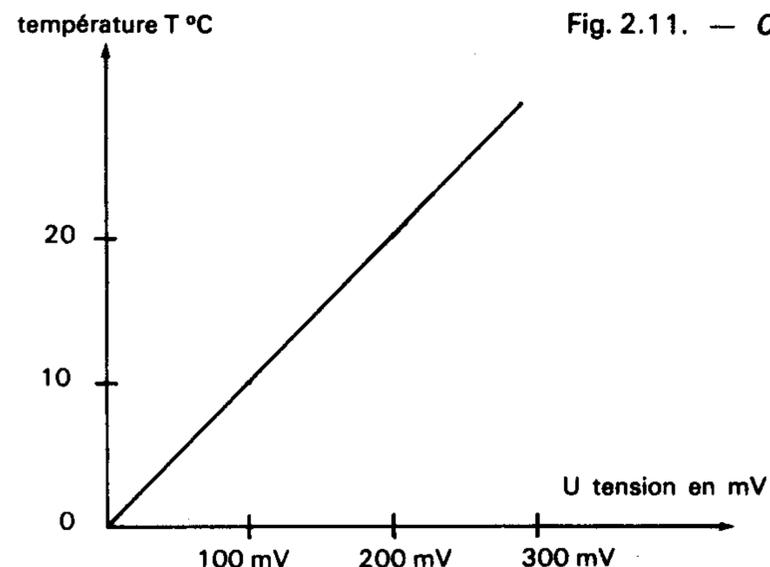


Fig. 2.11. — Courbe de tension du LM 35.

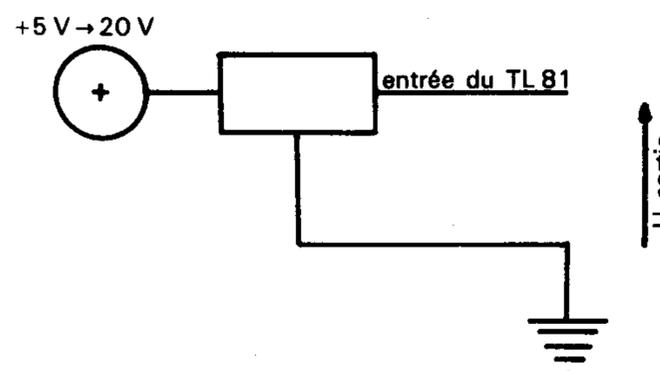


Fig. 2.12. — Câblage du LM 35.

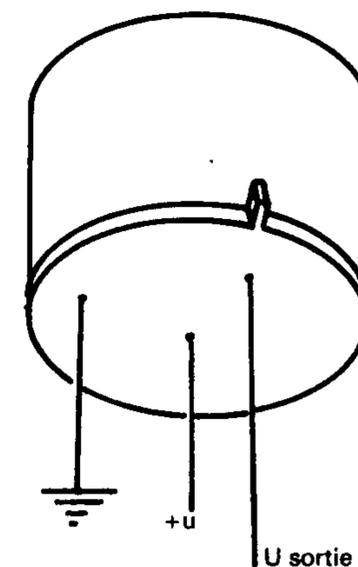


Fig. 2.13. — Vue en perspective du LM 35.

```

1 REM"CONVERTISSEUR ANALOGIQUE DIGITAL 8 BITS"
5 HIRES
10 POKE#3F5,48
20 POKE#3F5,56
30 REM PORT A EN ENTREE
40 POKE#3F4,0
50 POKE#3F5,60
60 A=PEEK(#3F4)
70 REM ABAISSER CA2
80 POKE#3F5,52
90 REM REGARDE SI LES DONNEES SONT PRETES
100 A=PEEK(#3F5)
110 IF A<128 THEN GOTO100
120 REM INHIBITION
130 B=PEEK(#3F4)
140 POKE#3F5,56
150 X=150-B/25.6
160 Y=Y+1
170 CURSETY,X,1
180 IF Y>210 THEN TEXT :CLS:HIRES:Y=0
190 PRINT(150-X)/10
200 GOTO70

```

### Programme 5. — Convertisseur AD 8 bits.

*Ligne 10:* Mettre  $CRA_3$  à 0 d'où  $CA_2$  est à 0.  $CRA_2 = 0$  pour accéder au registre de sens de transfert.

Si l'on regarde le tableau V, on voit qu'il faut mettre #30 en hexadécimal dans le registre de contrôle de A or #30 donne 48 en décimal.

*Ligne 20:* Mettre  $CRA_3 = 0$  ce qui va positionner  $CA_2$  à 1 et effectuer le blanking, le mot de contrôle devient #38 soit 56 en décimal.

*Ligne 40:* Déclarer l'ensemble du port A en entrée ce qui oblige à mettre 0 dans le registre sens de transfert des données.

*Ligne 50:* Mettre  $CA_2$  à 1 et sélectionner le registre de données.

*Ligne 60:* Par la lecture du registre de données, le bit  $CRA_7$  du registre de contrôle se remet automatiquement à zéro.

*Ligne 80:* Abaisser  $CA_2$  ce qui lance une conversion (cf. tableau V).

*Ligne 100:* Regarder si les données sont prêtes car si  $\overline{DR}$  passe à zéro (l'entrée  $CA_1$  étant active sur front descendant), le bit  $CRA_7$  du registre de contrôle va passer à 1.

*Ligne 110:* Boucler tant que les données ne sont pas prêtes;  $CRA_7$  à 1 donne  $2^7 = 128$ , il faut donc tester si A est inférieur à 128.

*Ligne 130:* Lire le registre de données remet implicitement le bit  $CRA_7$  à zéro. Dans la variable B nous disposons du résultat de la conversion.

*Ligne 140:* Remonter  $CA_2$  à 1 pour effectuer le blanking; 1,5  $\mu$ s plus tard, le signal data ready remonte automatiquement (cf. tabl. V).

*Ligne 150:* Calculer la valeur à afficher.

*Ligne 160 à 190:* Dessiner sur l'écran la nouvelle valeur, incrémenter le compteur, imprimer la valeur en volts, se positionner pour une nouvelle conversion.

REMARQUE 1: Ce programme est en Basic. Pour l'accélérer, il faut impérativement le traduire en langage machine.

REMARQUE 2: L'instruction `Poke#3F4,#30` n'est pas interprétée correctement par l'ORIC, il faut nécessairement convertir #30 en décimal: l'instruction `Poke` s'écrit impérativement sous la forme `Poke#adresse` en hexa, nombre en décimal.

REMARQUE 3: Le montage que nous avons réalisé peut être facilement étendu à un convertisseur plus performant: le AD 571 qui convertit sur 10 bits en 15  $\mu$ s. Les signaux de contrôle sont les mêmes, il suffit de consacrer 2 bits d'un autre port programmé en entrée pour acquérir les 2 bits manquants. Nous avons laissé libre pour cela les 2 bits de poids faible du port B de la PIA n° 2. [Le AD 570 est en fait construit comme un AD 571 mais il n'est testé — par le constructeur — et garanti que pour 8 bits. Essayez toujours avec 10 bits, qui sait...]

## 2.6. LE LECTEUR PROGRAMMATEUR D'EPROM

Nous avons réalisé :

- un lecteur de 2716 (2 K × 8),
- un lecteur de 2732 (4 K × 8),
- un lecteur et un programmeur de 2764.

Sur le schéma 2.14 nous avons présenté les brochages des différents types d'EPROM (erasable programmable read only memory).

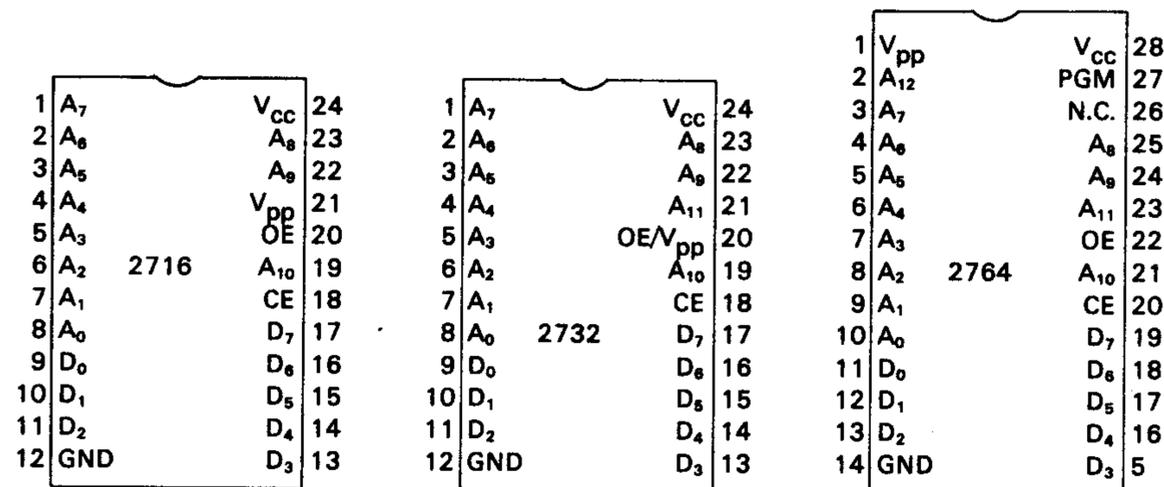


Fig. 2.14. — Brochage des EPROM.

Il y a très peu de différence entre les brochages des 2716 et des 2732. Les seules pattes 20 et 21 sont différentes. Pour la lecture, on connecte la patte 20 à la masse. La patte 21 correspond à A<sub>11</sub> pour les 2732 et à V<sub>pp</sub> pour la 2716. Il suffira donc de mettre V<sub>pp</sub> à 1 quand on lira les 2716 par rapport au programme de lecture de la 2732, seuls les 2 K octets de poids fort seront adressés.

Le schéma de connection 2.15 indique le numéro des ports à employer pour avoir un programme directement compatible sans aucun changement.

En guise de support 24 broches, nous vous conseillons d'utiliser un support à efforts d'insertion nuls. Le risque de tordre les pattes avec un tel support est réduit à néant. La vitesse avec laquelle on peut enlever et remettre une EPROM dans ce support est sans commune mesure plus élevée; le seul inconvénient, c'est son prix!

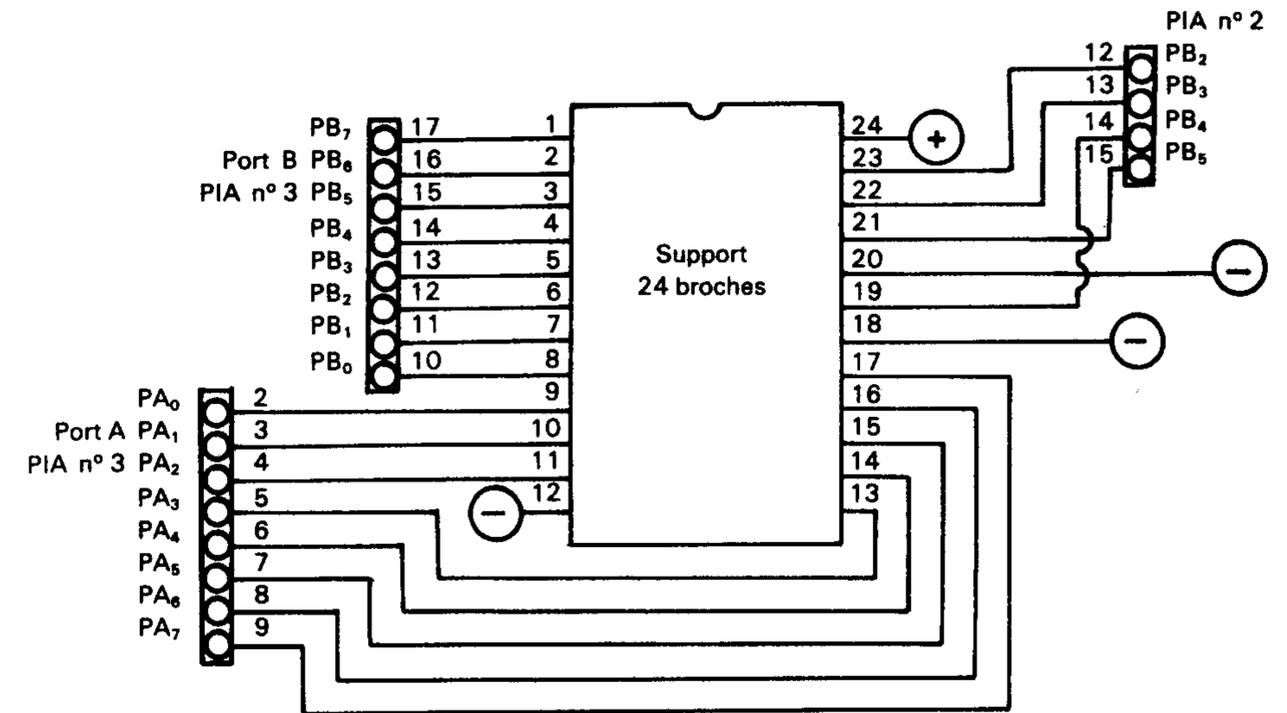


Fig. 2.15. — Schéma de câblage du lecteur de 1716-2732.

```

1 REM" LECTEUR D'EPROM 2716-2732"
2 GOT0990
3 CLS
5 REM PIA NO 3 PORT B EN SORTIE
6 REM ADRESSES BASSES
7 REM
10 POKE#3FB,0
20 POKE#3FA,255
30 POKE#3FB,255
34 REM
35 REM PIA NO 3 PORT A EN ENTREE
36 REM DONNEES
37 REM
40 POKE#3F9,0
50 POKE#3F8,0
60 POKE#3F9,255
64 REM
65 REM PIA NO 2 PORT B EN SORTIE
66 REM ADRESSES HAUTES
67 REM
70 POKE#3F7,0
80 POKE#3F6,253
    
```

```

90 POKE#3F7,255
200 FOR N=U TO 60 STEP 4
250 POKE#3F6,N
300 FOR M=0 TO255
350 POKE#3FA,M
400 A=PEEK(#3F8)
450 PRINTHEX$( (N-U)/4 ); :PRINTHEX$(M); :PRINTTAB(20)HEX$(A)
500 NEXTM
600 NEXTN
990 CLS
1000 PRINT"2716 OU 2732 "
1010 PRINT
1020 PRINT "0 POUR 2716"
1030 PRINT
1040 PRINT" 1 POUR 2732 "
1050 GETA$
1060 IFA$="0"THEN LET U=32:GOTO2
1070 IFA$="1"THEN LET U=0:GOTO2
1080 GOTO1020

```

**Programme 6. — Lecteur de 2716 ou 2732.**

*Ligne 2:* Envoi en ligne 990 pour le Menu.

*Ligne 10:* Positionner le registre de contrôle pour avoir accès au registre sens de transfert du port B de la PIA n° 3 (cf. tabl. I).

*Ligne 20:* Déclarer tous les bits du port B de la PIA n° 3 en sortie.

*Ligne 30:* Positionner le registre de contrôle pour avoir accès au registre de données du port B de la PIA n° 3:

- à PB<sub>0</sub> correspondra le bit d'adresse A<sub>0</sub>,
- à PB<sub>7</sub> correspondra le bit d'adresse A<sub>7</sub>.

*Lignes 40 à 60:* Le port A de la PIA n° 3 est positionné en entrée. Il correspondra aux bits de données de l'EPROM:

- PA<sub>0</sub> correspondra au bit de données D<sub>0</sub>,
- PA<sub>7</sub> correspondra au bit de données D<sub>7</sub>.

*Lignes 70 à 90:* PB<sub>2</sub> à PB<sub>5</sub> du port B de la PIA n° 2 sont déclarés en sortie, ce qui correspond à  $2^2 + 2^3 + 2^4 + 2^5 = 60$  en décimal.

- PB<sub>2</sub> correspond à l'adresse A<sub>8</sub>,
- PB<sub>3</sub> correspond à l'adresse A<sub>9</sub>,

- PB<sub>4</sub> correspond à l'adresse A<sub>10</sub>,
- PB<sub>5</sub> est connecté sur la broche 21 du support. Si on veut lire une 2732, il se comporte comme le bit d'adresse A<sub>11</sub>. Si on veut lire une 2716, il est alors connecté sur la broche V<sub>pp</sub> comme le montre la figure 2.14. Il faut le figer à 1 dans ce cas c'est-à-dire mettre le bit n° 5 à 1 ( $2^5 = 32$ ).

*Lignes 300 à 500:* Incrémenter les adresses A<sub>0</sub> à A<sub>7</sub>, lire les données et les afficher.

*Ligne 200 et 250:* Incrémenter ensuite les adresses de poids fort. Nous avons utilisé les bits PB<sub>0</sub> et PB<sub>1</sub> à une autre finalité que le lecteur d'EPROM.

C'est pourquoi il faut incrémenter l'indice H par 4 à chaque fois:

$$PB_2 \rightarrow 2^2 \rightarrow 4$$

Pour les 2716, nous figeons A<sub>11</sub> à 1 ce qui revient à commencer la boucle D<sub>0</sub> avec le bit A<sub>11</sub> à 1 soit PB<sub>5</sub> à 1 c'est-à-dire  $U = 2^5 = 32$ .

Si vous voulez transformer votre lecteur en programmeur, il faut appliquer les modifications du schéma 2.16 et vous référer au manuel du fabricant de l'EPROM que vous désirez programmer.

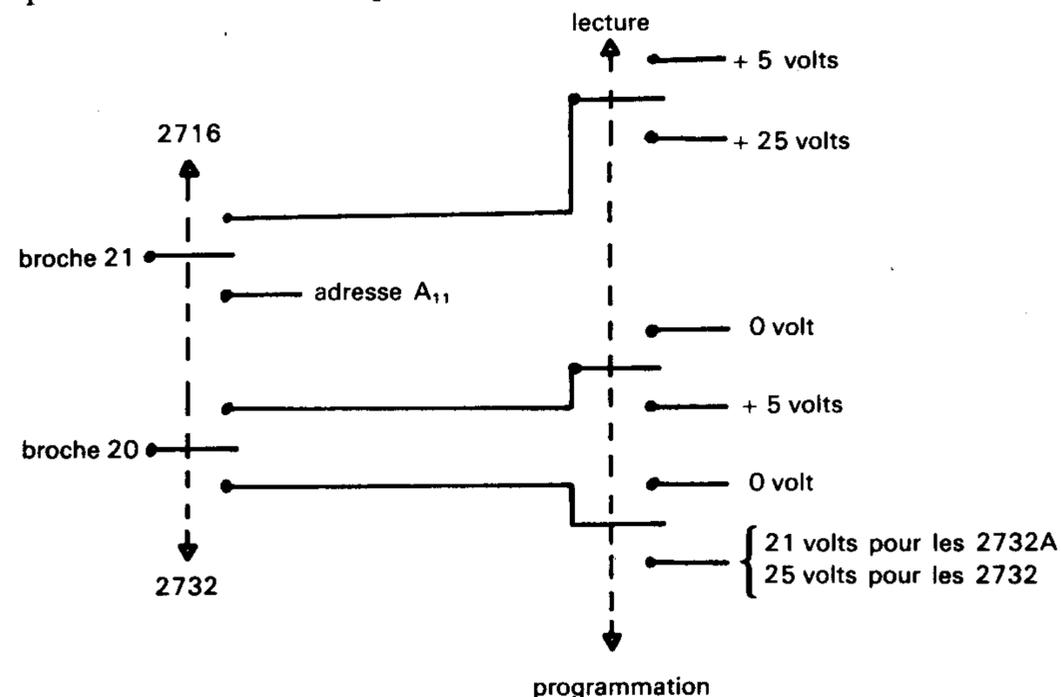


Fig. 2.16. — Modification pour réaliser un programmeur de 2716-2732.

Attention: Les tensions de programmation dépendent de chaque constructeur d'EPROM. Reportez-vous à la doc. du fabricant.

Nous allons vous indiquer comment programmer les 2764 car cela vous sera utile pour la suite de cet ouvrage.

Vous pourrez à l'aide de ce programmeur transformer votre ORIC en "ATMOS" ou bien apporter des modifications au contenu de vos EPROM de BASIC.

## 2.7. LE PROGRAMMATEUR DE 2764

Si vous avez réussi à nous lire jusqu'à maintenant, le câblage du schéma suivant ne va pas vous effrayer (fig. 2.17).

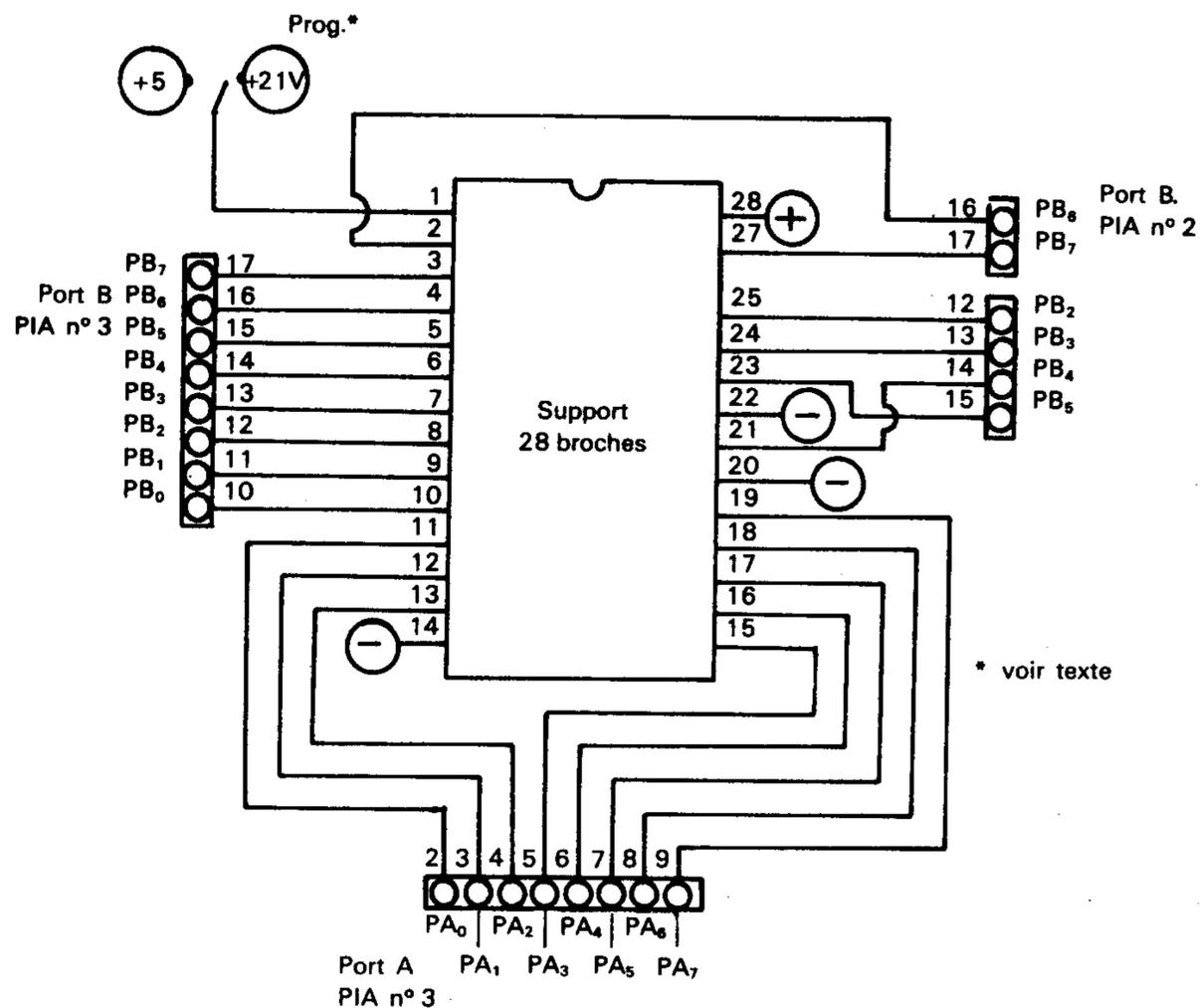


Fig. 2.17. — Lecteur-programmeur de 2764.

Le programmeur d'EPROM nécessite une tension de 21 V ou de 25 V suivant la marque d'EPROM utilisée. Lorsque vous achetez une

EPROM, renseignez-vous auprès de votre revendeur sur la tension exacte de programmation. Il n'est pas possible ici de donner la liste des différentes tensions suivant les constructeurs pour des raisons d'évolutions constantes des produits proposés.

La précision des tensions est primordiale. Nous vous donnons un exemple de réalisation d'alimentation stabilisée en peu de boîtiers au dernier chapitre.

Le programme de brûlage des 2764 est donné avec une forme conversationnelle un peu spartiate, c'est volontaire. Chacun pourra ainsi développer à son gré sa propre version.

```

1 REM"PROGRAMMATEUR DE 2764"
5 CLS
10 POKE#3FB,0
20 POKE#3FA,255
30 POKE#3FB,255
35 REM
40 POKE#3F9,0
50 POKE#3F8,255
60 POKE#3F9,255
65 REM
70 POKE#3F7,0
80 POKE#3F6,252
90 POKE#3F6,255
95 POKE#3F6,255
100 INPUT"ADRESSE EN DECIMAL";NN
105 IF NN<0 OR NN>8191 THEN GOTO1000
110 M=NN-INT(NN/256)*256
120 N=(NN-M)/64
130 POKE#3F6,N+128
140 POKE#3FA,M
150 INPUT"DONNEE EN DECIMAL";DA
160 PRINT"OK?";
170 GETA#
180 IFA#<>"0" THEN PRINT"NON PROGRAMME ":GOTO100
190 CALL#E76A
200 REM POUR ORIC1 FAIRE CALL#E6CA
210 POKE#3F6,N
    
```

```
220 FORI=1T025
230 NEXTI
240 POKE#3F6,N+128
250 CALL#E93D
260 REM POUR ORICI FAIRE CALL#E804
270 PRINT"PROGRAMME"
280 GOTO100
1000 PRINT"ADRESSE HORS LIMITES"
```

#### Programme 7. — Programmeur de 2764.

REMARQUE: Les tensions élevées doivent impérativement être connectées aux bons endroits sinon le résultat est garanti: vous pouvez utiliser votre composant comme élément décoratif ou comme générateur de fumées.

#### Quelques explications sur ce programme

Pour "brûler" une EPROM il faut mettre à zéro la broche V pendant 50 ms minimum et 55 ms maximum. Le port PB<sub>7</sub> de la PIA n° 2 est relié à V<sub>pp</sub>. La boucle d'attente est effectuée par les lignes 220 et 230. L'instruction call#E76A sert à interdire les interruptions du clavier; le clavier est donc inhibé après cette instruction. Sa situation est rétablie par l'instruction call#E93D. Si les interruptions étaient autorisées pendant la boucle d'attente, la temporisation ne serait plus exacte!

## 3

# Transmission des données en série

### POURQUOI TRANSMETTRE ?

Dans toutes les applications que l'on a vues pour l'instant, de tous les périphériques utilisés, celui qui est physiquement le plus éloigné de l'ordinateur est la manette de jeu.

Tous les échanges entre les différents circuits décrits jusqu'à présent se faisaient par l'intermédiaire de bus parallèles. Pour faire circuler un octet D<sub>7</sub> à D<sub>0</sub> d'un composant à l'autre on utilise 8 fils différents. Cette solution, pratique et rapide, ne peut plus être utilisée dans le cas d'une communication à longue distance, surtout pour des raisons de prix de revient; pour prendre un exemple d'usage courant, le chemin de fer transibérien ne comporte qu'une seule voie et non plusieurs de front. Beaucoup moins rapide ce système à l'avantage d'être d'un prix inférieur à l'utilisation du parallélisme.

Comme le train, les bits d'information iront sur l'unique voie de communication les uns après les autres. Ce principe est connue sous le nom de transmission série.

Sachant que de nombreux périphériques lourds (ordinateurs, imprimantes, tables traçantes, télétypes...) utilisent le mode de communication série, nous allons étudier le moyen de convertir les données — parallèle au niveau du bus du 6502 — en données série.

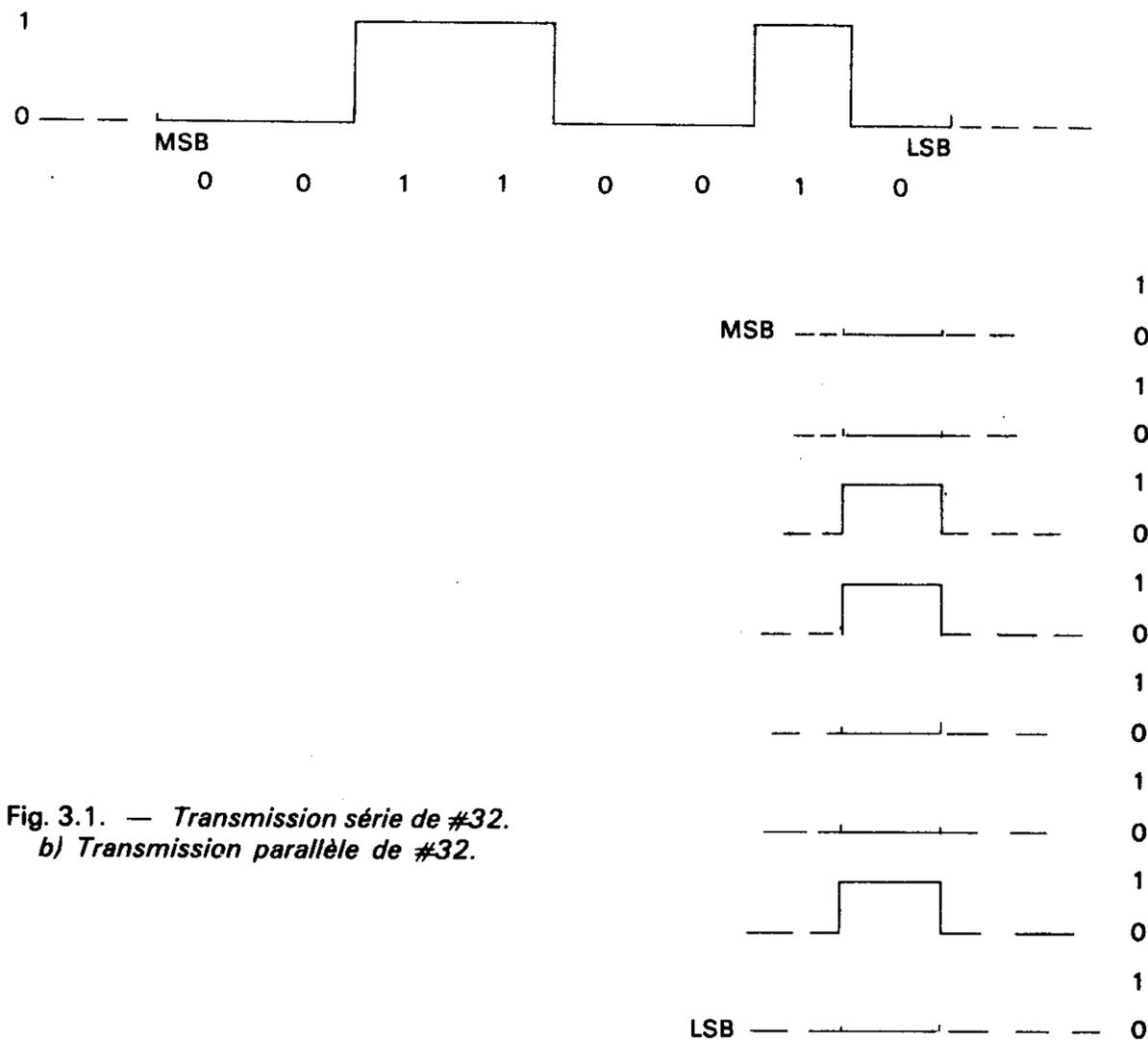


Fig. 3.1. — Transmission série de #32.  
b) Transmission parallèle de #32.

### 3.1. ÉTUDE DE L'ACIA 6850

L'ACIA (ou en français adaptateur pour communications asynchrones) est un circuit qui réalise la mise en série de données parallèles.

L'ACIA se comporte vis-à-vis de l'architecture du micro-ordinateur comme deux cases-mémoire (à la différence de la PIA qui paraît comme quatre cases-mémoire).

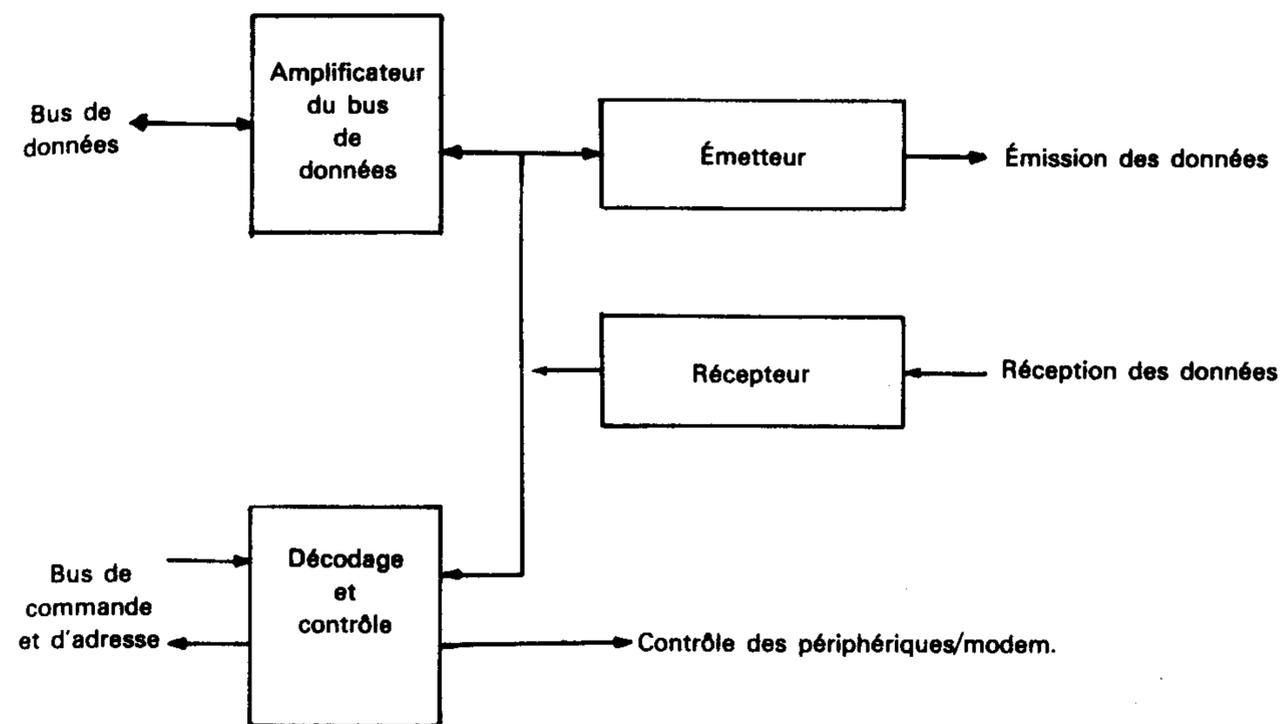


Fig. 3.2. — Synoptique fonctionnel de l'ACIA.

Le premier registre correspond :

- au registre de contrôle en écriture,
- au registre d'état en lecture.

La programmation du registre de contrôle est résumé dans les tableaux suivants :

TABLEAU I

<i>RC1</i>	<i>RC0</i>	<i>Fonction</i>
0	0	horloge externe
0	1	horloge externe ÷ 16
1	0	horloge externe ÷ 64
1	1	initialisation

TABLEAU II

RC4	RC3	RC2	Fonctions		
			nb de bits transmis	Parité	nb de bits d'arrêt
0	0	0	7	paire	2
0	0	1	7	impaire	2
0	1	0	7	paire	1
0	1	1	7	impaire	1
1	0	0	8	non	2
1	0	1	8	non	1
1	1	0	8	paire	1
1	1	1	8	impaire	1

Les bits RC5, RC6 et RC7 servent à gérer les interruptions : n'étant pas utilisés ils seront mis à zéro.

Le deuxième tableau donne les formats possibles des données transmises.

Si vous voulez envoyer des mots en ASCII, ils auront 7 bits. Le bit de parité est un bit supplémentaire qui indique le nombre de 1 qui compose la donnée — il est généré automatiquement par l'ACIA. Ce bit est comparé au niveau du périphérique avec la parité — recalculée — de la donnée correspondante : s'il y a une différence la donnée est réputée mauvaise. Le bit de parité est le premier pas dans la détection (et la correction) des erreurs de transmission.

Les bits d'arrêt préviennent le périphérique de la fin du message.

Le bit de début est généré à chaque transmission par l'ACIA.

Cette transmission (ou réception) correspond à la programmation des bits du registre de contrôle RC4 RC3 RC2 = 000. L'utilisation de RC1 et RC0 sera expliquée à l'occasion de la réalisation de l'interface compatible RS 232C.

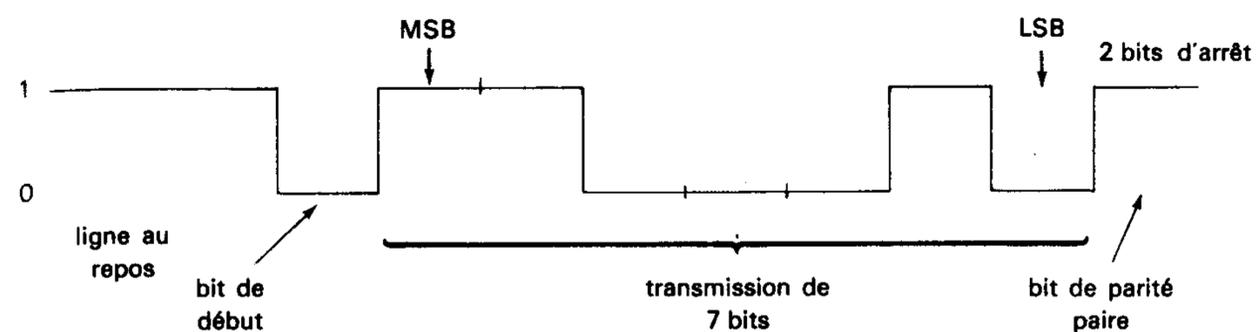


Fig. 3.3. — Transmission de l'ACIA.

Donnons de façon abrupte la constitution du registre d'état :

Bit	Fonction
0	registre de réception plein (RDRF)
1	registre de transmission vide (TDRE)
2	perte de la porteuse de données (DCD)
3	inhibition de l'émetteur (CTS)
4	erreur de format (FE)
5	récepteur en surcharge (OVRN)
6	erreur de parité (PE)
7	demande d'interruption (IRQ)

Les abréviations entre parenthèses sont les initiales des traductions anglaises.

Chacun de ces bits sera expliqué par la suite.

Le deuxième registre de l'ACIA est le registre de transmission en écriture, et le registre de réception en lecture.

Passons à la réalisation des interfaces séries.

### 3.2. RÉALISATION D'UNE INTERFACE AUX NORMES EIA RS 232C

#### Le décodage des ACIA

Nous avons vu au premier chapitre que les deux circuits de décodage nous donnaient accès à quatre plages mémoires de quatre registres chacune. Nous utiliserons la plage n° 4 : #3FC, #3FD, #3FE, #EFF.

Or on vient de voir que l'ACIA est un circuit à deux registres, donc seul le fil d'adresse  $A_0$  suffira pour décodé ces registres (contrairement à la PIA qui exige les fils  $A_0$  et  $A_1$ ).

Nous allons donc utiliser le fil d'adresse  $A_1$  pour partager en deux la plage n° 4 (voir schéma 3.4).

Il faut savoir que l'ACIA possède trois broches d'entrée pour le décodage du boîtier :

$\overline{CS2}$   
CS0  
CS1

$\overline{CS2}$  et CS0 sont utilisés par le décodage de la plage n° 4 (dans le cas de la PIA, CS1 n'est pas utilisé). L'état actif de CS1 étant l'état haut il nous faut réaliser la fonction logique suivante avec l'adresse  $A_1$  :

TABLEAU III

$A_1$	ACIA n° 1 CS1	ACIA n° 2 CS1
0	1	0
1	0	1

Le tout lorsque la plage n° 4 est sélectionnée c'est-à-dire pour  $A_2 = 1$  et  $A_3 = 1$ .

On note que le signal désiré sur CS1 ACIA n° 2 est le même que  $A_1$ , et celui de CS1 ACIA n° 1 est l'inverse. Il serait possible d'utiliser un inverseur (74LS04) mais cela nécessiterait le câblage d'un troisième boîtier de décodage pour l'interface. Devant cette inflation de composants nous avons recours à un stratagème. Reportez-vous au schéma et table de vérité du 74LS155.

Pour  $A_2 = 1$  et  $A_3 = 1$ , c'est-à-dire  $A = B = 1$  et  $1G = 0$  on a :

1C	1Y3
H	L
L	H

La fonction inverse a été réalisée, or ce demi-boîtier n'est pas utilisé. N'est-ce pas merveilleux ? (Voir figure 3.4 pour ce complément de câblage.)

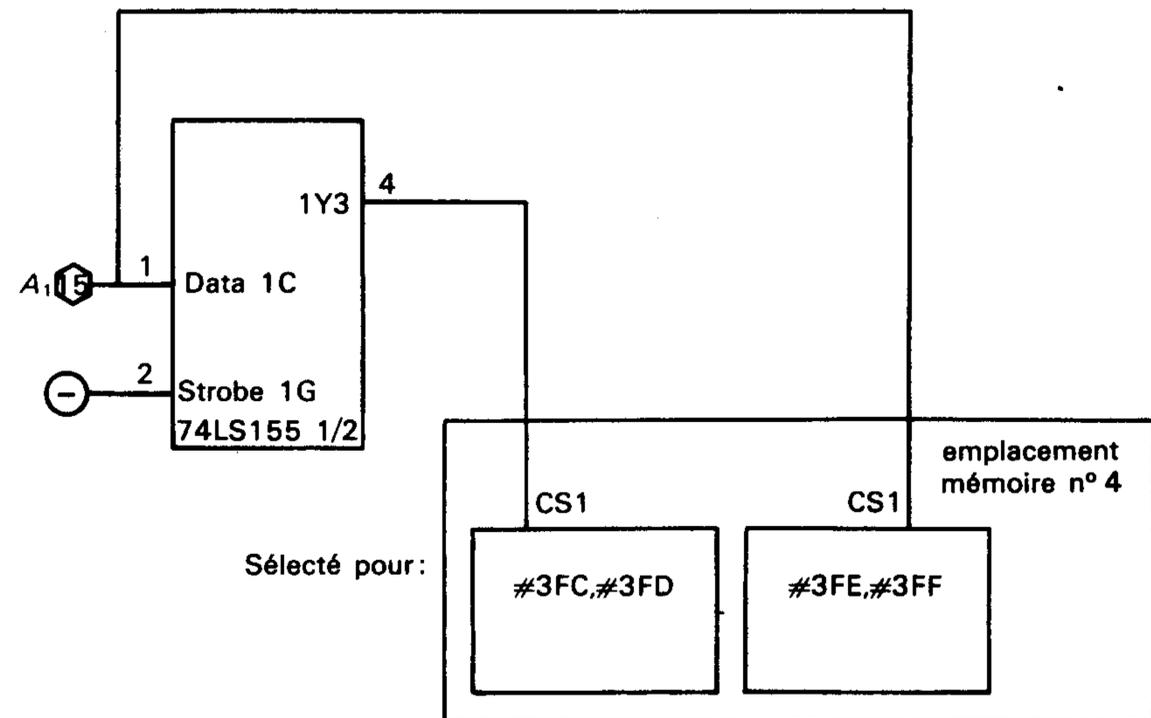


Fig. 3.4. — Décodage des deux ACIA.

Maintenant vous pouvez passer au câblage de l'ACIA suivant la figure 3.5 en n'oubliant pas de lire attentivement les remarques quant au décodage du boîtier.

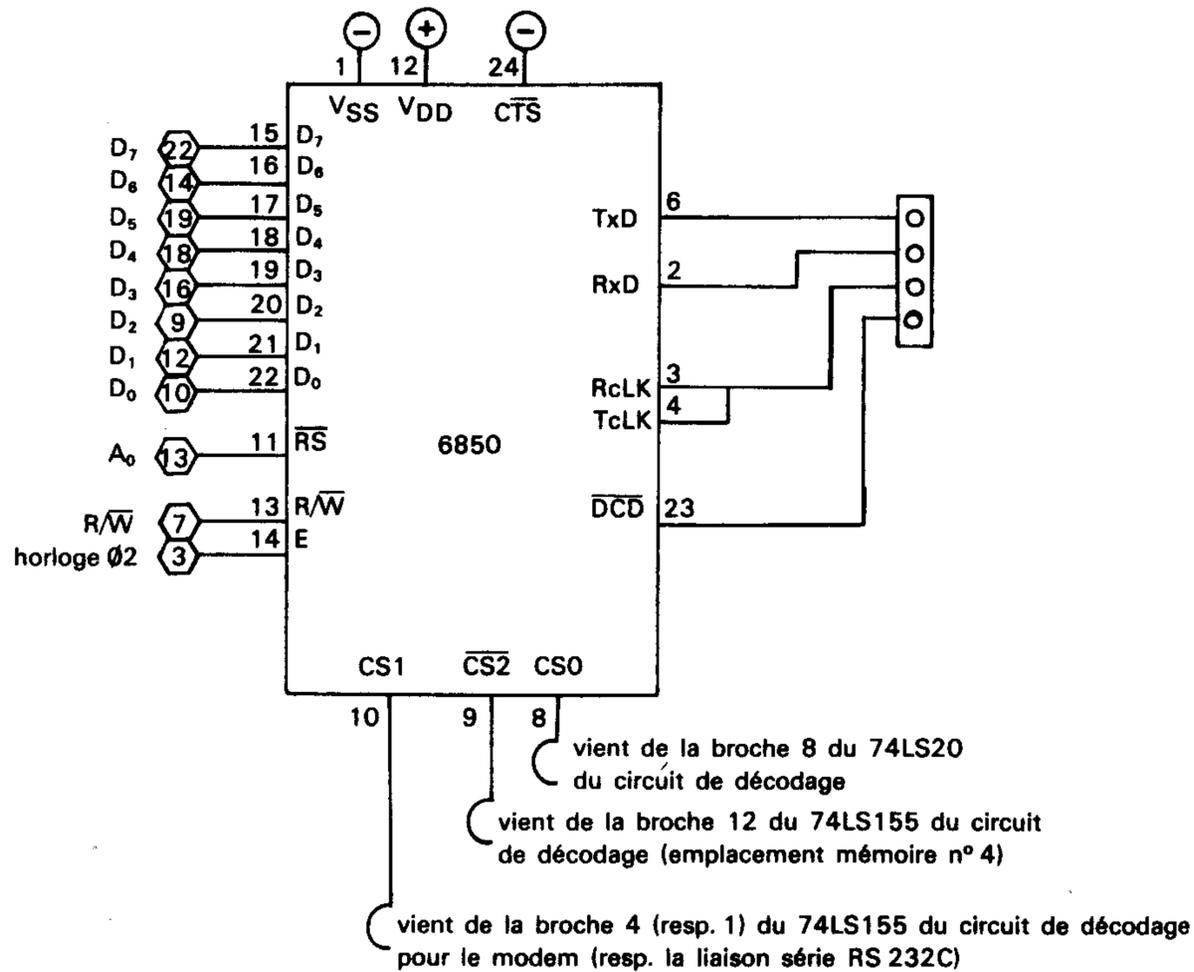


Fig. 3.5. — Schéma de câblage de l'ACIA.

Quelques remarques sur ce schéma :

TxD	représente la sortie série.
RxD	représente l'entrée série.
RcLK	représente l'horloge de réception.
TcLK	représente l'horloge de transmission.
$\overline{\text{DCD}}$	sera utilisé dans le cas de la connexion du modem.
$\overline{\text{CTS}}$	utilisé généralement en liaison avec un modem ne sera pas utilisé dans nos applications.

Dans cet état l'ACIA n'est pas encore opérationnelle. Il lui faut connecter l'horloge qui dans notre cas sera identique pour l'émission et la réception.

Pour cela regardez la figure 3.6 et cablez le circuit générateur de Baud MC 14411 de Motorola. On utilise un commutateur 12 vers 1 pour sélectionner la vitesse de transmission. Le MC 14411 est un circuit oscillateur suivi de plusieurs diviseurs.

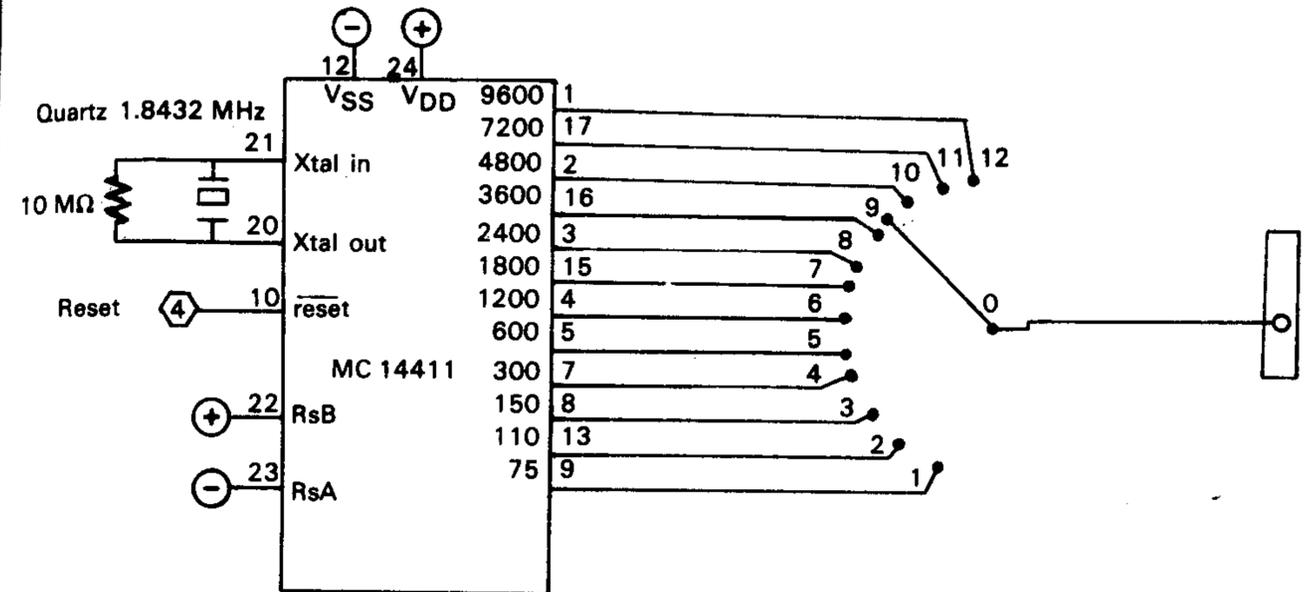


Fig. 3.6. — Schéma de câblage du générateur de bauds MC 14411.

Il vous génère des horloges spécifiques par division de la fréquence de son quartz. Chaque horloge correspond à des vitesses de transmission standards. Ces vitesses s'expriment en bauds, ce qui correspond à 1 bit par seconde.

D'un point de vue pratique, si vous voulez transmettre un message à un périphérique il faut vous renseigner sur sa vitesse de transmission, sa parité, son nombre de stop bits et le nombre de bits par mot.

Pour illustrer un peu cet exposé et aussi pour vous permettre de vérifier si l'ensemble de votre câblage est correct, nous allons vous fournir un petit programme de test de l'ACIA.

```

1 REM" TEST DE L'ACIA"
4 CLS
10 POKE#3F3,0
20 POKE#3F2,255
30 POKE#3F3,255
40 POKE#3F2,255
50 POKE#3FC,3
60 POKE#3FC,1
70 A=PEEK(#3FD)
80 PRINT"TAPEZ"
85 GETA$
87 PRINTA$
90 A=ASC(A$)
100 POKE#3FD,A
110 A=PEEK(#3FC)
130 IFA=2ORA=0THEN110
135 IFA<>3THENPRINT"ERREUR":STOP
140 C=PEEK(#3FD)
150 PRINT"VALEUR";CHR$(C)
170 GOTO70

```

#### Programme de test de l'ACIA.

La philosophie de ce test est de faire boucler l'ACIA sur elle-même. Pour cela il faut relier la sortie TxD à l'entrée RxD de l'ACIA. Les données émises par l'ACIA doivent être reçues par elle-même.

Le registre de données de transmission est différent du registre de données de réception. Le signal lecture-écriture  $R/\overline{W}$  de la patte 13 sert à les différencier. Ce même signal différencie aussi les registres de contrôle et d'état.

```

1 REM "UTILISATION DE L'ACIA"
5 CLS
10 POKE#3F3,0
20 POKE#3F2,255
30 PRINT"EMISSION OU RECEPTION 1/2"
35 INPUTB
40 ON B GOTO 50,200
50 REM EMISSION
60 POKE#3F2,255
70 POKE#3FC,3
80 POKE#3FC,1
90 PRINT"TAPEZ"
100 GETA$
110 PRINTA$
120 A=ASC(A$)
130 POKE#3FD,A
140 A=PEEK(#3FC)
150 IFA<>2 THEN GOTO140
160 GOTO90
200 REM RECEPTION
210 POKE#3F2,0
220 POKE#3FC,3
230 POKE#3FC,1
240 A=PEEK(#3FC)
250 IFA=2ORA=0THENGOTO240
260 C=PEEK(#3FD)
270 PRINT"VALEUR";
280 PRINTCHR$(C)
290 GOTO240

```

#### Programme de réception et de transmission

Les adresses des registres sont donc :

TABLEAU IV

ACIA n° 1	#03FC	lecture : registre d'état écriture : registre de contrôle
	#03FD	lecture : registre de réception des données écriture : registre de transmission des données
ACIA n° 2	#03FE	lecture : registre d'état écriture : registre de contrôle
	#03FF	lecture : registre de réception des données écriture : registre de transmission des données

Pour la plupart des applications une seule ACIA suffit. Si toutefois vous cablez les 2 ACIA, vous pouvez utiliser le même MC 14411.

*Ligne 10:* L'ACIA est initialisée; les bits RC1 et RC0 sont mis à un comme le demande le tableau I.

*Ligne 20:* La division de l'horloge est fixée ici. Il ne faut surtout pas choisir le mode divisé par un, car dans ce cas la réception ne peut plus être asynchrone; vous allez au devant d'ennuis certains. En fait seul le mode divisé par 16 donne les vitesses en baud escomptées avec le MC 14411.

*Ligne 30:* Il faut vider le registre de données de transmission pour remettre le bit 0 du registre d'état à zéro. Si le bit zéro est à 2 cela indique qu'une donnée nouvelle vient d'arriver dans le registre de réception des données. C'est sur ce bit que peut s'effectuer le test d'arrivée de nouvelles données.

*Ligne 40:* Rentre un caractère alphanumérique en A\$.

*Ligne 50:* Convertit ce caractère en ASCII.

*Ligne 60:* Envoie ce nombre ASCII dans le registre transfert de données.

*Ligne 70:* Acquiert le registre d'état.

*Ligne 80:* Isole le bit 0 du registre d'état.

*Ligne 90:* Teste si le registre de réception est plein (bit 0 du registre d'état à 1).

*Ligne 100:* Prélève la valeur du registre de données.

*Ligne 110:* Affiche cette valeur. Elle doit être identique à la valeur tapée.

Si votre montage a passé avec succès cet autotest alors vous avez une ACIA opérationnelle. Il vous est possible maintenant de poursuivre et de cabler votre interface RS 232C.

### L'interface RS 232C

Le boîtier que nous avons choisi pour pouvoir communiquer en norme RS 232C est le SN 75188 de chez TEXAS INSTRUMENT. Il est interchangeable avec le boîtier MC 1488L de chez MOTOROLA.

Le schéma du brochage du boîtier est celui de la figure 3.7.

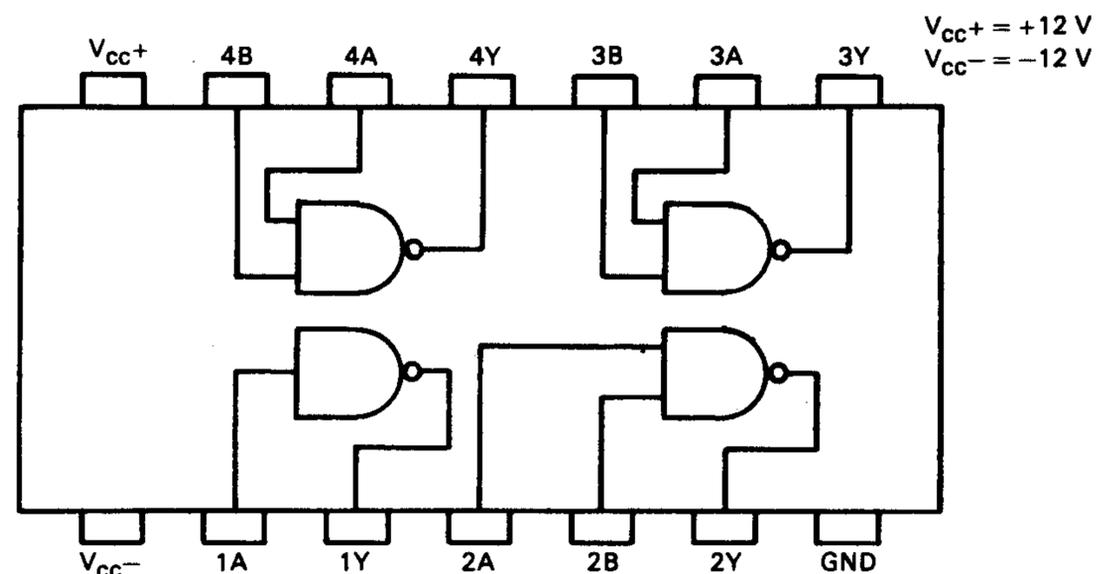


Fig. 3.7. — Boîtier d'interface RS 232C: émission SN 75188.

Les quatre émetteurs de lignes sont spécialisés pour interfacer les terminaux en accord avec la norme IEA RS 232C. Cela nécessite des tensions d'alimentation élevées ! du +12 V et -12 V. Le chapitre 6 vous donne un exemple de réalisation d'alimentation fournissant ces tensions.

En réception, le boîtier qui effectue une réception correcte suivant la norme RS 232C est le boîtier SN 75189 de chez TEXAS INSTRUMENT. Le boîtier équivalent chez MOTOROLA est le MC 1489.

Le schéma du brochage du boîtier est donné par la figure 3.8.

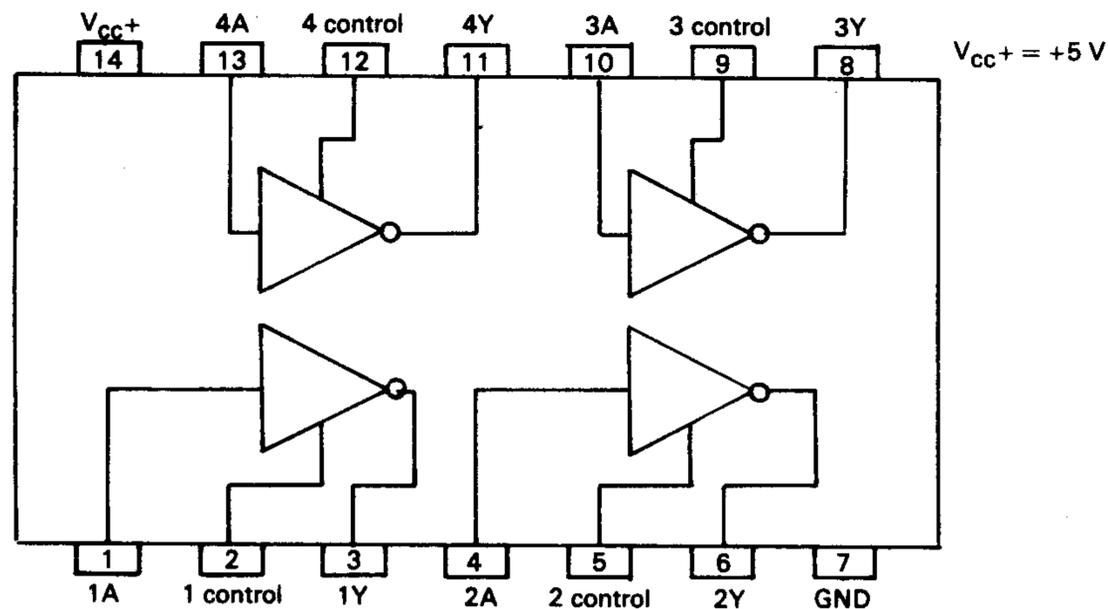


Fig. 3.8 — Boîtier d'interface RS 232C: réception SN 75189.

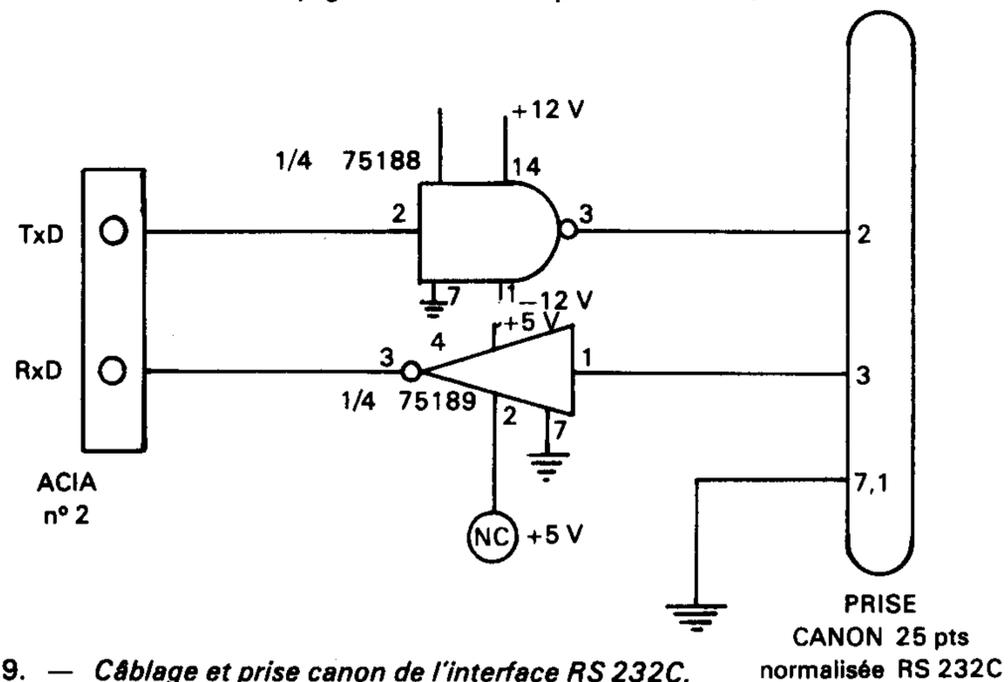


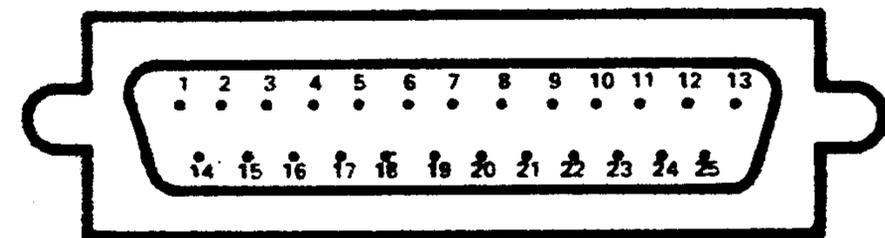
Fig. 3.9. — Câblage et prise canon de l'interface RS 232C.

La broche de contrôle du récepteur sera mise à  $V_{cc}$ .

La figure 3.9 montre le câblage à effectuer.

TABLEAU V

Broche	Signal
1	→ MASSE
7	→ MASSE
3	→ Récepteur des données
2	→ Transmission des données



Connecteur normalisé CANON 25 points.

La norme RS 232C est la plus courante, cette norme est très employée bien que désuète. La majorité des imprimantes est à la norme RS 232C ! Le gros inconvénient de ce type de montage est la diversité des alimentations. Tous les constructeurs essayent de produire des circuits en monotension et pour cette normalisation il est nécessaire d'avoir du +12 V et du -12 V !

De plus l'immunité au bruit n'est pas très bonne ; le départ d'informations par ce procédé ne peut pas excéder quelques dizaines de mètres. Toutes ces raisons nous ont poussé à étudier d'autres normes de transmissions, des transmissions différentielles normalisées sous le nom RS 422.

### 3.3. LA TRANSMISSION BIFILAIRE SYMÉTRIQUE LE 75119

Les circuits que nous allons vous proposer servent à interfacer la technologie TTL avec un système de transmission différentiel sur ligne bifilaire.

Les avantages de la transmission différentielle résident dans une propagation sans erreur sur une distance beaucoup plus importante que la transmission de type RS 232. Une seule tension de 5 V est employée ; cela simplifie les problèmes d'alimentation. L'immunité au bruit est vraiment bonne ; c'est l'avantage principal de ce type de transmission. En fait la donnée D est expédiée sur la ligne ainsi que la donnée barre :  $\bar{D}$ . S'il y a un parasite sur la paire bifilaire, ce parasite affectera la ligne  $\bar{D}$  et la ligne  $\bar{D}$  de la même valeur en amplitude et en signe donc la différence entre D et  $\bar{D}$  restera inchangée.

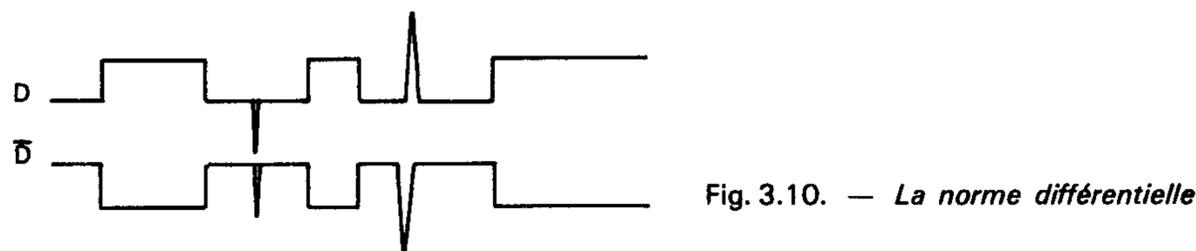


Fig. 3.10. — La norme différentielle

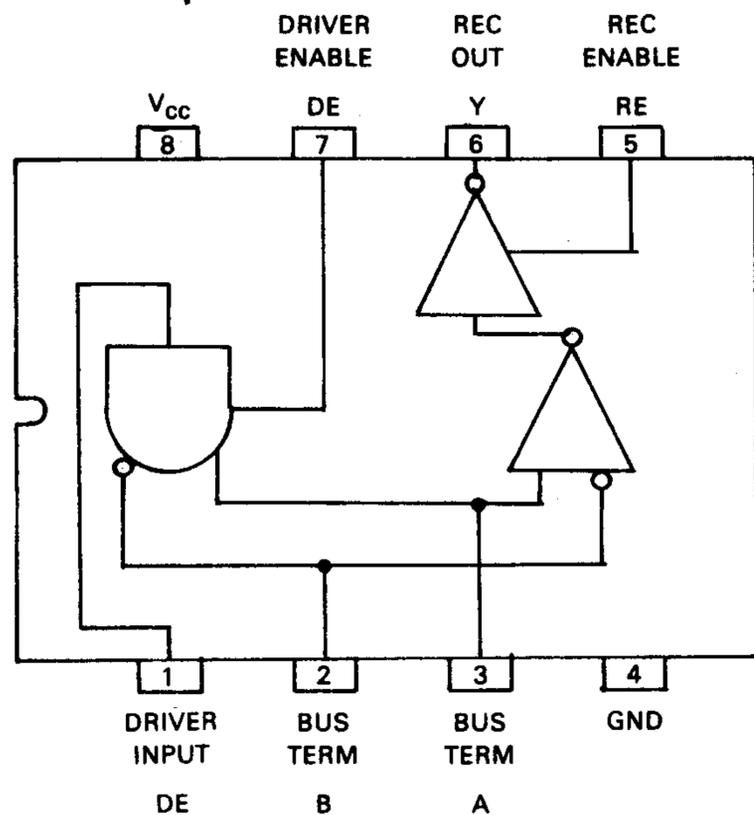


Fig. 3.11. — Le boîtier SN 75119 : émission-réception différentielle.

Le 75119 contient un émetteur différentiel et un récepteur différentiel dans le même boîtier.

En figeant la tension sur la patte 5 à un, la réception est toujours validée et la table de vérité se simplifie :

TABLEAU VI

En émission					En réception				
Entrées		Sorties			Entrées			Sorties	
DE	DI	A	B	Y	DE	A	B	DI	Y
H	H	H	L	H	L	H	L	X	H
H	L	L	H	L	L	L	H	X	L
L	X	Z	Z	?					

La broche 7 correspondant à DE (driver enable) sert à positionner le composant en émission ou en réception. Si DE est à l'état bas (zéro) le bus est en trois états et le composant se comporte comme un récepteur différentiel.

Il suffit donc de connecter le port  $PB_0$  de la PIA n° 1 sur la broche 7 pour pouvoir à sa guise émettre ou recevoir.

$PB_0$	
0	réception
1	émission

L'adresse du port B de la PIA n° 1 est #3F2 : registre de données  
#3F3 : registre de contrôle

Pour nos lecteurs désireux d'économiser un bit de port, il est toujours possible de connecter la broche 7 sur un interrupteur, sans oublier la résistance de rappel !

Grâce à ce montage, vous allez pouvoir communiquer avec un ami à plusieurs centaines de mètres de distance. Cette interface est très

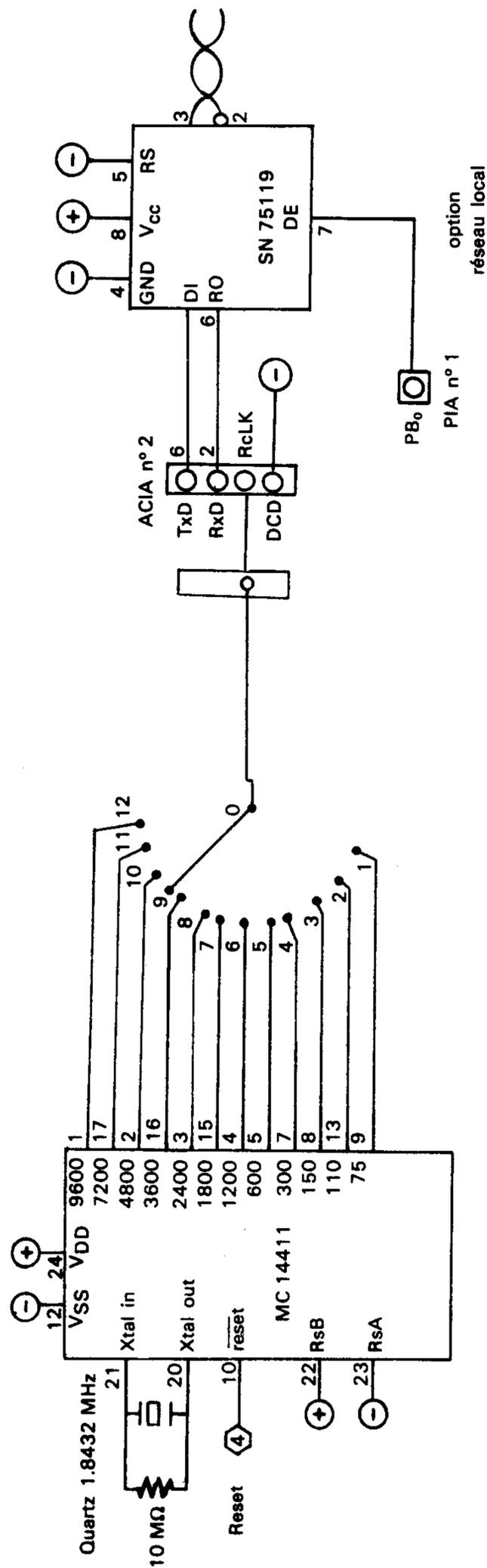


Fig. 3.12. — Câblage de l'interface différentielle.

simple et très performante. Ne vous en privez pas pour votre réseau local !

Le schéma 3.12 vous indique comment câbler cette nouvelle interface. Le programme de transmission-réception est succinct mais suffisant.

Le problème des réseaux locaux réside dans l'obligation de "passer" des fils entre les utilisateurs. La question que l'on peut se poser est "Pourquoi ne pas utiliser le réseau PTT?". Cette question nous a conduit à vous présenter un modem permettant cette utilisation.

### 3.4. LE MODEM

Le mot modem est de plus en plus employé de nos jours, c'est l'abréviation de modulateur démodulateur. De même que les normes RS 232C et RS 422, le modem encode les informations en sortie d'une transmission série, il module les informations par rapport à une porteuse et en réception il démodule par rapport à la même porteuse.

Le but est de faire une élévation du signal en fréquence pour avoir une immunité au bruit plus importante et surtout pour pouvoir transmettre le signal sur les lignes PTT, c'est-à-dire le réseau commuté. La bande passante du réseau PTT est de 300 Hz à 3 000 Hz. Les signaux circulants sur les lignes PTT, encodés par le modem et visualisés sur un oscilloscope sont les suivant :

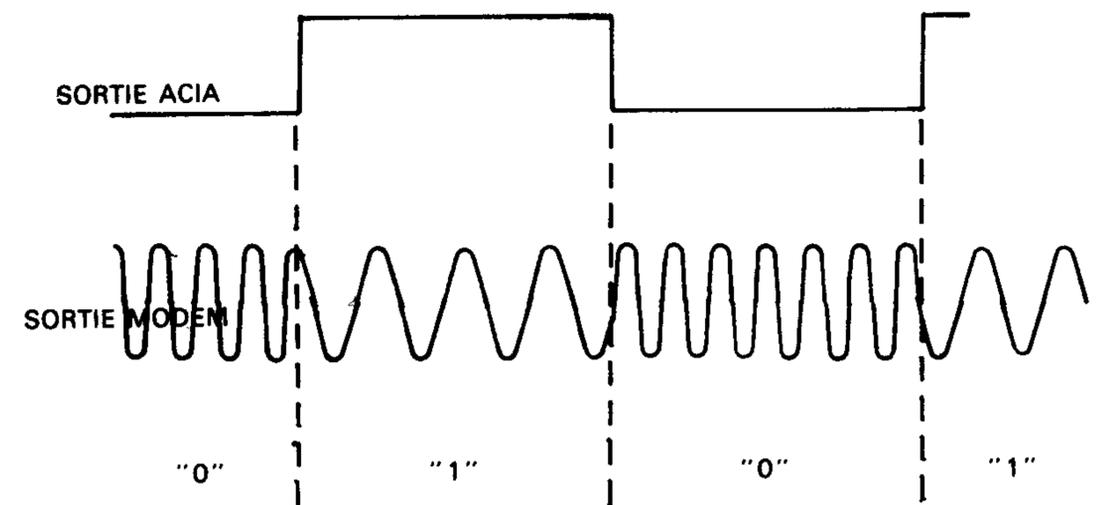


Fig. 3.13. — L'encodage réalisé par le modem.

Les fréquences pour la normalisation CCITT V23 sont celles du tableau VII :

TABLEAU VII

TXR1	TXR2	Vitesse de transmission en bauds (b/s)	TxD	TX fréquences (Hz)
1	0	75	1 0	390 450
0	1	600	1 0	1 300 1 700
0	0	1 200	1 0	1 300 2 100

Le modem que nous avons mis en œuvre est le TCM 3101 de chez TEXAS INSTRUMENT. Ce composant à l'époque où nous écrivons ce livre est seulement en voie de commercialisation.

Ce boîtier TCM 3101 a été choisi car il répondait parfaitement à notre volonté profonde d'intégration et de simplification.

L'avantage qu'il a par rapport à tous les types de boîtier modem existant sur le marché, c'est son alimentation mono tension 5 V. De plus tous les filtres qui gravitaient auparavant autour des modems ont été ici intégrés.

L'intérêt non négligeable offert par l'utilisation du TCM 3101, c'est la possibilité de récupérer l'horloge du modem sur la broche CLOCK et ce faisant ne plus avoir besoin du générateur de bauds pour votre ACIA. Le brochage du boîtier TMC 3101 est représenté sur la figure 3.14.

Le schéma de câblage de l'ACIA est représenté sur la figure 3.15.

Le modem nécessite l'emploi d'un quartz de 4,43361875 MHz. Si votre vendeur de composants local ne peut vous le fournir, ce quartz est disponible à Paris et vous pouvez vous le faire expédier.

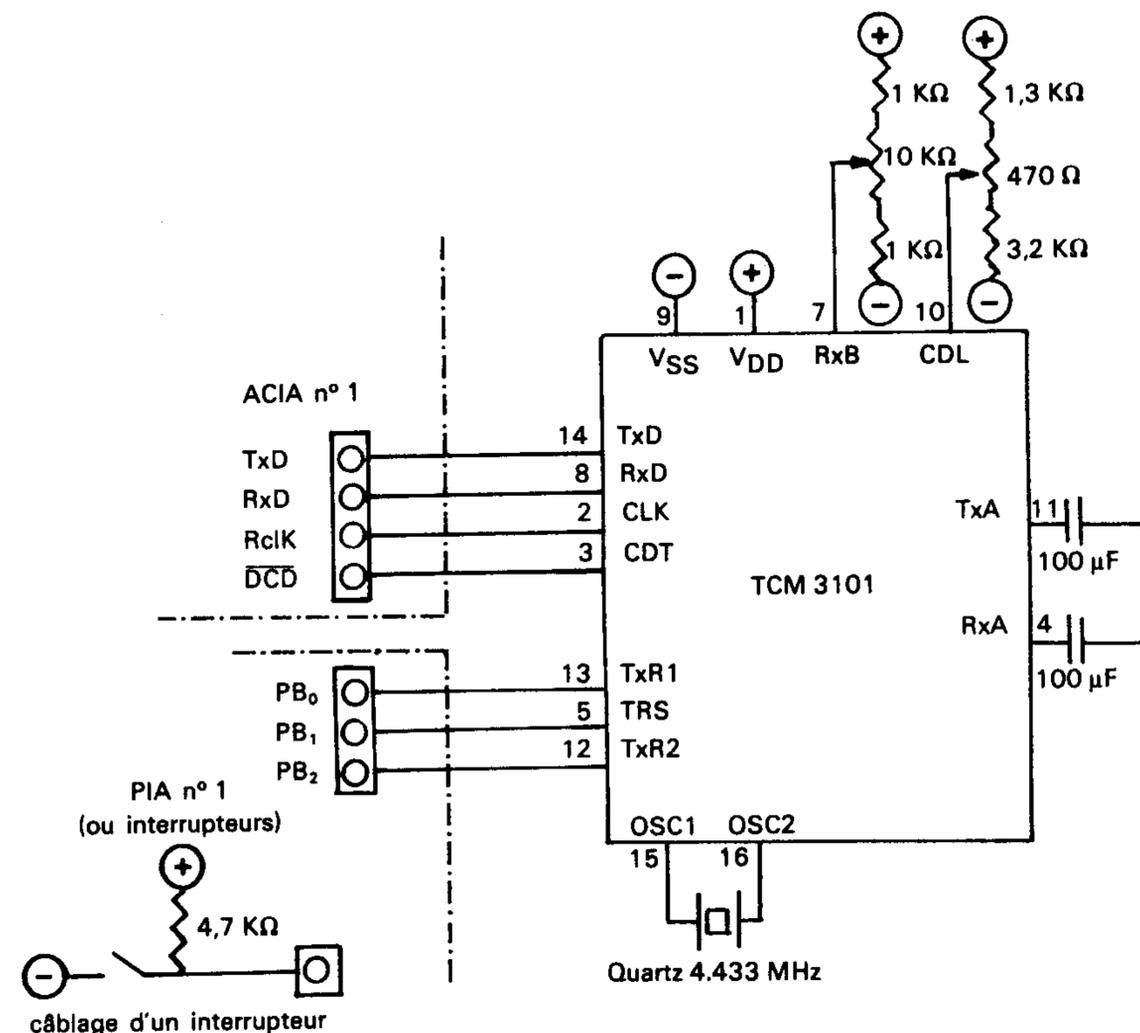


Fig. 3.14. — Schéma de câblage du modem.

Le potentiomètre sur CDL (carrier detect level adjust) sert à ajuster le seuil de détection ; le constructeur préconise une tension d'ajustement telle que  $0,64 V_{DD} \leq CDV \leq 0,74 \times V_{DD}$  d'où le choix de nos résistances.

Le potentiomètre sur RxB (receive BIAS ADJUST) sert à ajuster le seuil de détection final pour minimiser le biais dû à la distorsion.

L'horloge CLK de la broche 2 est égale à 16 fois l'horloge de la vitesse choisie en réception.

TABLEAU VIII

TRS	Vitesse de réception en bauds	Fréquence de l'horloge (KHz) CLK
1	600	9.56 (600 × 16 ≈ 9 560)
0	1 200	19.11 (1 200 × 16 = 19 110)

La signification des fils TRS, TxR1 et TxR2 est expliquée par les tableaux VII et VIII.

La broche CDT (carria detect output) un niveau bas sur cette broche indique un défaut de porteuse. Ce signal une fois inversé peut être envoyé sur l'entrée  $\overline{DCD}$  de l'ACIA ce qui permettra de connaître cette information en scrutant le bit 2 du registre d'état de l'ACIA.

TxD (TRANSMIT DATA) est un registre d'émission

RxD (RECEIVE DATA) est un registre de réception

Le couplage du modem sur la ligne PTT peut se faire de deux façons différentes ; par couplage acoustique ou par isolement par un transformateur de liaison.

● **Couplage acoustique**

Pour effectuer l'amplification nous avons employé le LM 386 de chez National Semiconductor. Le montage à effectuer est celui de la figure 3.15.

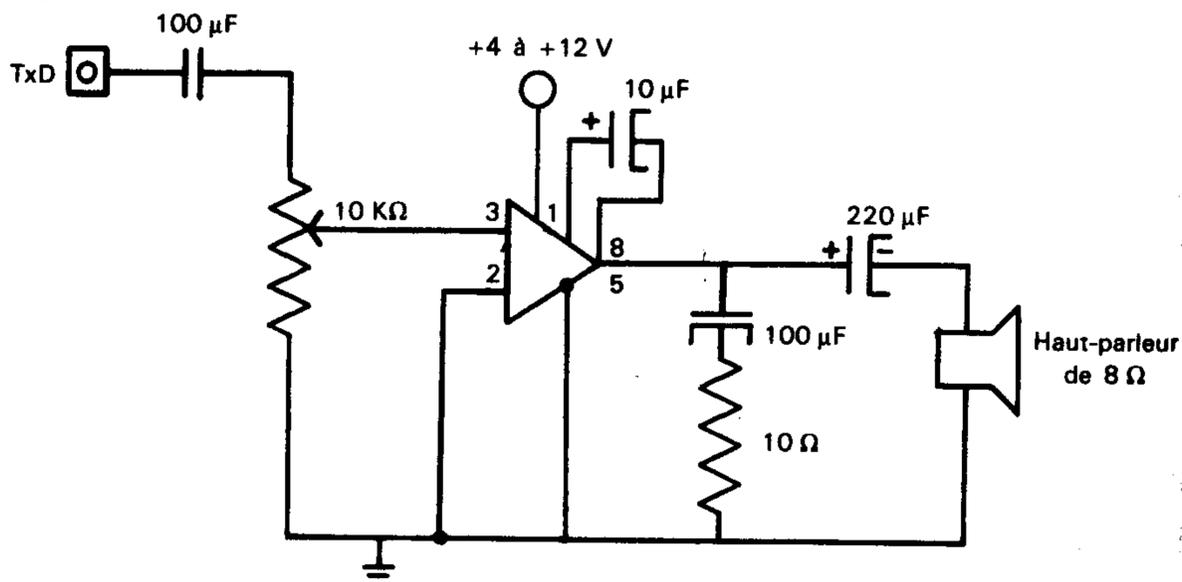


Fig. 3.15. — Couplage acoustique en émission.

Le montage amplifie environ 200 fois.

La réception s'effectue correctement grâce au montage (fig. 3.16).

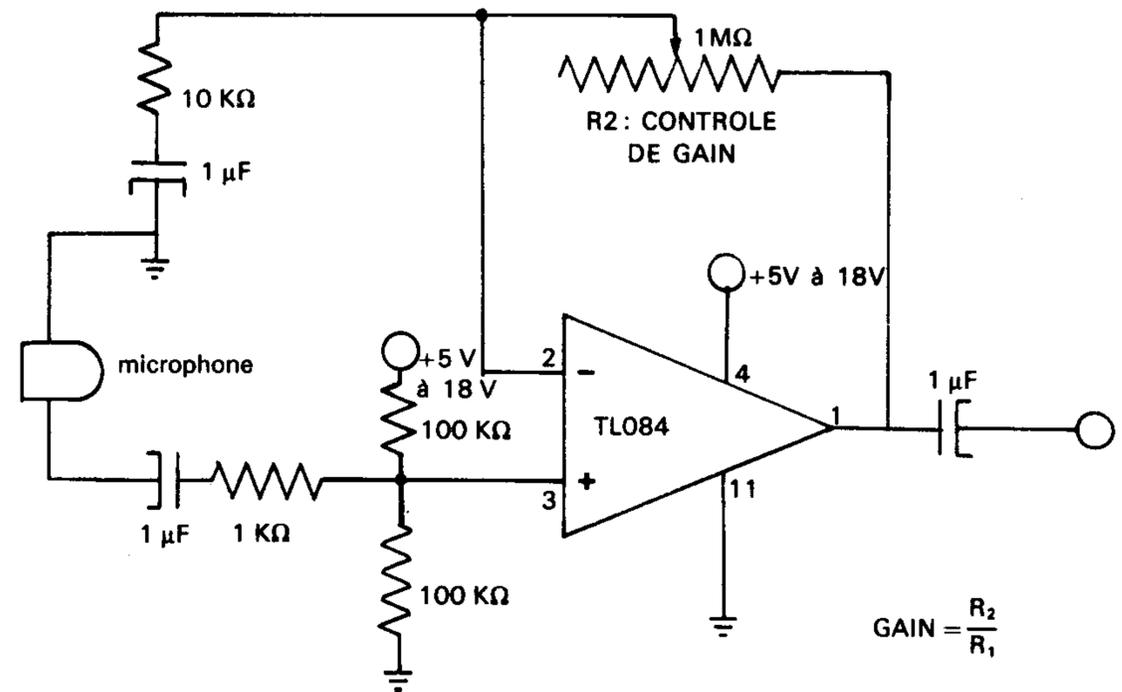


Fig. 3.16. — Couplage acoustique en réception.

Utilisez un microphone dynamique à basse impédance.

En guise de microphone, vous pouvez même employer un haut-parleur 8 Ω !

Il ne reste plus qu'à vous fabriquer un système mécanique vous permettant de connecter votre micro sur l'écouteur téléphonique et votre haut-parleur sur le microphone téléphonique. Cette méthode vous permet d'être isolé électriquement du réseau PTT par contre le réglage de la sensibilité (réglage des gains) est plus délicat.

De nombreux coupleurs acoustiques sont en vente dans le commerce vous pourrez vous en inspirer pour votre réalisation ; si bien sûr vous choisissez de réaliser ce type de coupleur.

● **Couplage par transformateur**

Le couplage peut s'effectuer directement à l'aide d'un transformateur d'isolement. L'administration des PTT exige de tels montages. Nous

avons utilisé le montage préconisé par le constructeur et nous vous le donnons sur la figure 3.17.

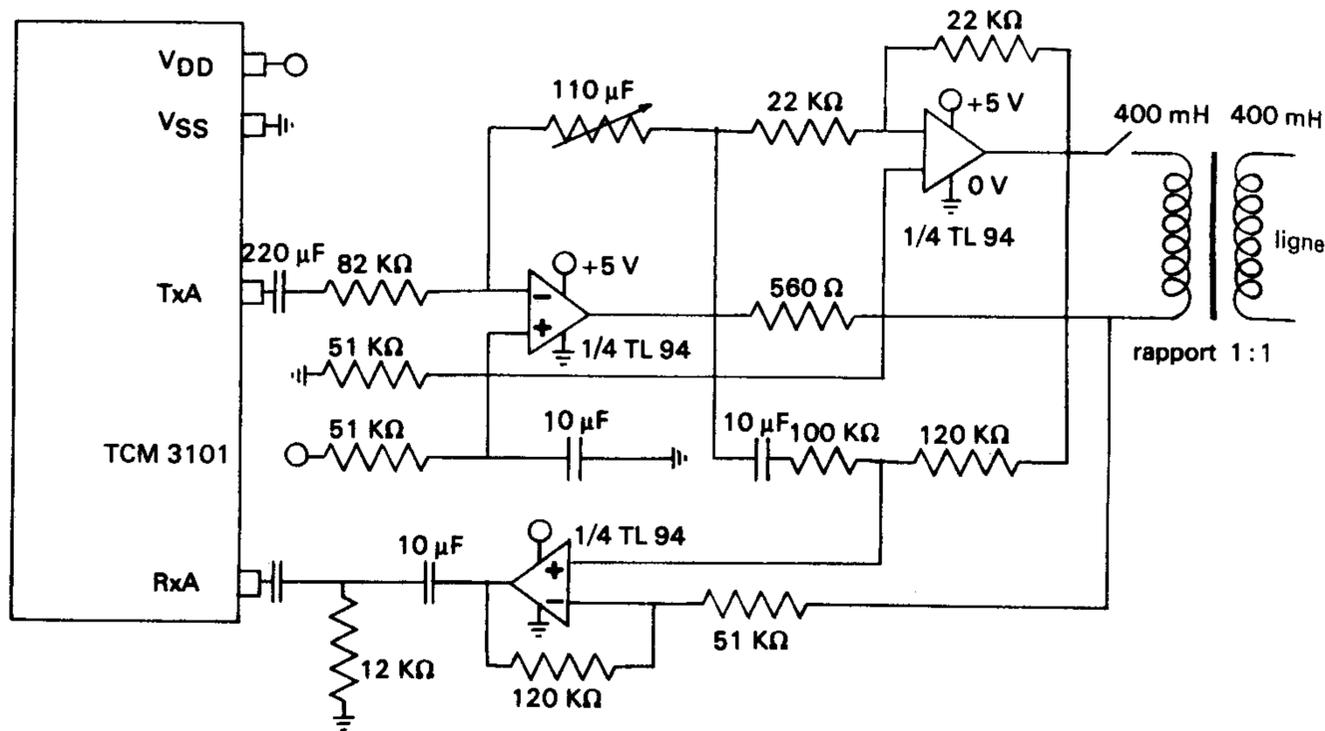


Fig. 3.17. — Couplage du modem par transformateur d'isolement.

Le transformateur est de rapport 1 et a une inductance de 400 mH.

Vous disposez maintenant d'un système complet de transmission :

- en réseau local : \* RS 232C,
- \* différentiel,
- en réseau commuté : \* modem acoustique,
- \* modem isolé par transformateur de liaison.

Il ne vous reste maintenant qu'à pouvoir accéder aux diverses banques de données. Grâce à minitel, qui est distribué gratuitement par les PTT, vous pouvez avoir accès aux services Télétel (normalisation Vidéotex) et aux divers clubs qui vont se développer : Microdial...

REMARQUE: L'intérêt principal du modem que nous venons de vous proposer, outre le nombre restreint de boîtiers, est le prix. Le construc-

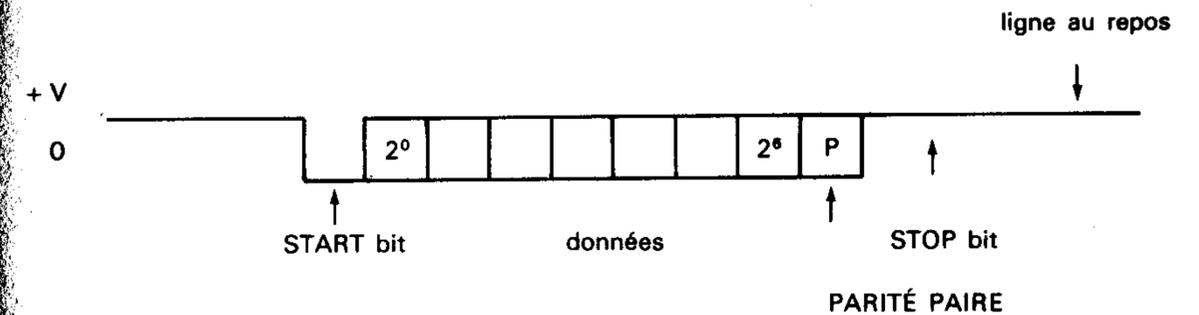
teur nous a annoncé son composant, le TCM 3101 comme étant grand public et peu onéreux.

### 3.5. MINITEL

L'appareil minitel des PTT est aux normes CCITT V23. Il comporte un modem qui émet à 75 bauds et reçoit à 1 200 bauds. La transmission est asynchrone et en FSK.

La modulation/démodulation s'effectue sur 10 bits :

- 1 bit de start,
- 7 bits de données,
- 1 bit de parité paire,
- 1 bit de stop.



Le niveau d'émission est de - 10 dBm. L'impédance de raccordement est de 600 Ω.

TABLEAU IX

	Rapidité de modulation	Porteuse	Bit à 1	Bit à 0
Emission	75 bauds	420 Hz	390 Hz	450 Hz
Réception	1 200 bauds	1 700 Hz	1 300 Hz	2 100 Hz

L'interface périmentel présentée sur le terminal est conforme à la norme NFC 92-250 relative à l'interconnexion entre les dispositifs de périmentel et les récepteurs de télévision grand public.

La prise est représentée sur le schéma 3.18.

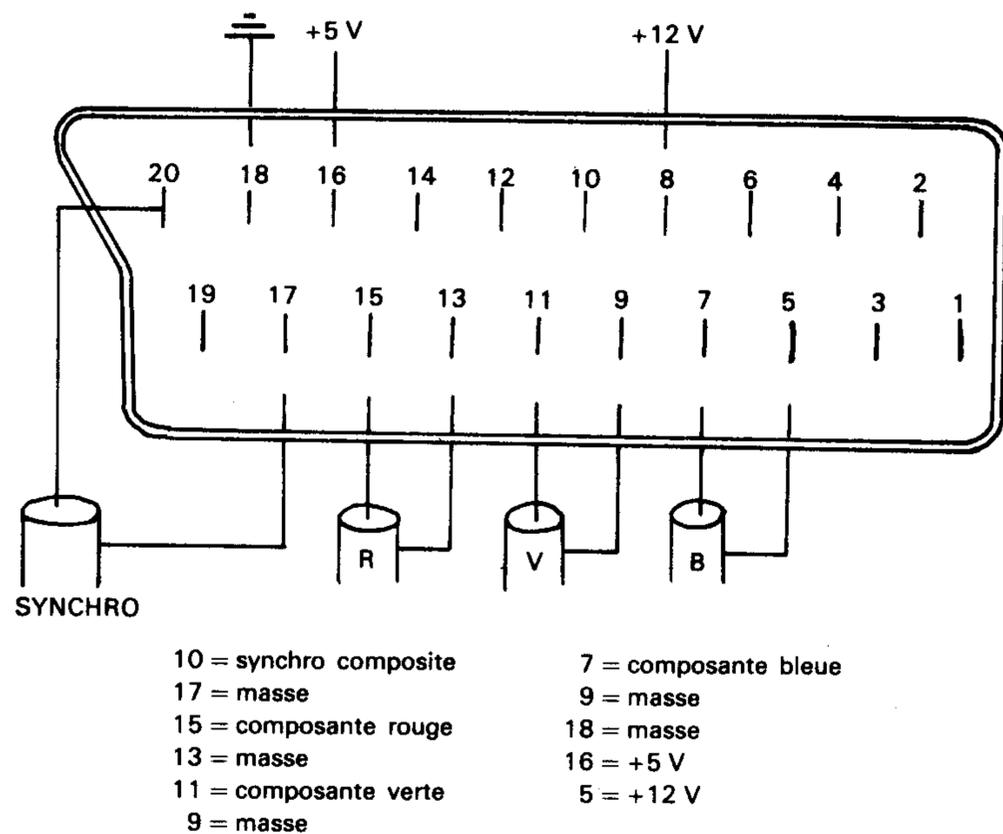


Fig. 3.18. — L'interface PERITELEVISION.

La prise DIN 5 broches à l'arrière de l'appareil sert à la connexion d'ordinateur. Les signaux sur cette prise sont :

- Rx : Réception des données
- Tx : Transmission des données
- M : Masse de signalisation
- PT : Périphérique prêt à émettre
- TP : Minitel prêt à émettre

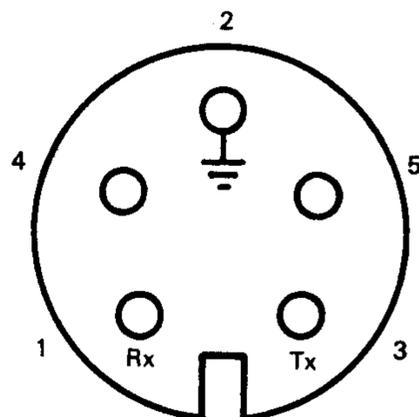


Fig. 3.19. — Prise arrière de MINITEL.

A la mise sous tension du minitel, la vitesse des échanges sur la prise est initialisée à 1 200 bauds dans les deux sens. Au repos la ligne est au +5 V comme montré sur la figure 3.19.

Les codes ASCII disponibles sur cette prise de sortie du minitel sont standards. Les 11 touches spécifiques qui sont sur le clavier du minitel sont envoyées sur la prise comme il suit :

Touche	Code hexadécimal
envoi	#13,#41
retour	#13,#42
répétition	#13,#43
guide	#13,#44
annulation	#13,#45
sommaire	#13,#46
correction	#13,#47
suite	#13,#48
shift + envoi	#0D
retour	#19,#42
répétition	#19,#4A
guide	#19,#48

annulation	#19,#5C
sommaire	#19,#53
correction	néant
suite	#19,#41

Il faudra donc filtrer ces codes pour ne pas les prendre en compte dans l'ORIC.

De plus un certain nombre de commandes peut être envoyé au minitel à l'aide de la prise pour forcer le passage à des configurations particulières.

La liste des commandes ci-dessous n'est pas exhaustive :

Programmation	Codes à envoyer
vitesse	#1B,#39,octet status vitesse P,E <sub>2</sub> ,E <sub>1</sub> ,E <sub>0</sub> ,R <sub>2</sub> ,R <sub>1</sub> ,R <sub>0</sub> E <sub>i</sub> = vitesse émission R <sub>i</sub> = vitesse réception valeurs de E <sub>i</sub> , R <sub>i</sub> : 1 = 75 bauds 2 = 300 bauds 3 = 1 200 bauds
mode rouleau Arrêt mode rouleau 80 colonnes par page loupe haut Arrêt loupe haut loupe bas Affichage curseur Arrêt affichage curseur Effacement écran Vidéo inverse Vidéo normale	#1B,#3A,#69,#43 #1B,#3A,#6A,#43 #1B,#3A,#69,#42 #1B,#3A,#69,#46 #1B,#3A,#6A,#46 #1B,#3A,#6A,#47 #11 #14 #0C #1B,#5D #1B,#5C

A la mise sous tension et en l'absence de réception de codes spéciaux de commande, les informations transitent dans le minitel de la façon suivante :

- clavier vers modem vers écran,
- clavier vers prise vers écran.

Il est possible de modifier ces aiguillages et par exemple de relier la prise vers le modem de façon à utiliser le modem du minitel et expédier les informations qui arrivent sur la prise directement vers la ligne PTT.

Pour faire cet aiguillage il faut envoyer les commandes suivantes :

- mise en route #1B,#3B,#61,code récepteur,code émetteur
- Arrêt #1B,#3B,#60,code récepteur,code émetteur.

Le code des différents modules est le suivant :

Modules	Code émission	Code réception
Prise	#53	#5B
Clavier	#51	#59
Écran	#50	#58
Modem	#52	#5A

REMARQUE: Si le code récepteur et le code émetteur sont choisis pour le même module alors celui-ci est bloqué.

#### ● Filtrage transcodage

Il va falloir écrire une handler qui va effectuer le traitement suivant de minitel vers l'ORIC :

- Touche envoi = Retour chariot (#13).
- Touche shift + annulation = Déconnection de la liaison
- Touche correction = Back space (#39).

Les codes envoyés par les touches spécifiques du minitel seront filtrés, ces touches ne sont pas nécessaires pour utiliser minitel en guise

de terminal informatique. Il est possible toutefois de les configurer pour des fonctions spécifiques telles que annulation de lignes, passage en mode rouleau, ou autres.

Dans le sens ORIC vers minitel, les codes à filtrer et à transcoder sont :

— les codes de l'ORIC allant de 80 à 9F (voir tableau des codes ASCII de l'ORIC).

Le programme à réaliser est constitué de la sorte :

- \* initialiser l'ACIA pour la liaison avec minitel (1 200 bauds, 1 bit start, 1 bit stop, parité paire),
- \* envoi des commandes au minitel (passage au mode rouleau, affichage du curseur, effacement de l'écran),
- \* prise en compte et filtrage des caractères envoyés par minitel,
- \* encodage des caractères envoyés par ORIC vers minitel.

Un Handler complet serait très long à écrire — et pourrait même faire l'objet d'un autre ouvrage. Les éléments que nous vous avons proposés vous donnent les bases nécessaires. De plus ce genre de logiciel à déjà été écrit pour d'autres ordinateurs (Goupil, Apple, ...) et vous pourrez obtenir beaucoup de renseignements auprès de Microdial (Club d'utilisateurs d'ordinateurs et de Minitel).

## 4

# Interface de puissance pour sortie Centronics

### MISE EN GARDE

Ce chapitre décrit une interface pour la commande de lampes, moteurs,... fonctionnant en 220 V. Bien que les schémas fournis soient sains et sans difficultés de réalisations, une mise en œuvre approximative conduirait à de nombreux déboires.

Nous vous conseillons vivement de bien lire et de réaliser dans l'ordre décrit par le présent chapitre, et surtout de prendre les plus grandes précautions quant à la partie essais des montages utilisant le 220 V : ne pas faire du "bricolage", isoler le 220 V et agir comme pour un appareil électroménager ; débrancher avant d'ouvrir, refermer avant de rebrancher.

Avec un minimum de sérieux cette interface vous offre la télécommande complète de votre maison.

## 4.1. PRÉSENTATION DE L'INTERFACE CENTRONICS

### Généralités

L'interface Centronics a été développée pour répondre aux besoins spécifiques des liaisons entre processeurs et imprimantes. Pendant longtemps les interfaces pour imprimantes se sont faites au moyen de liaison série (voir chapitre sur les ACIA's). Néanmoins la mise en œuvre de liaisons série n'est pas évidente et la vitesse est limitée. L'interface Centronics est une simplification du célèbre bus IEEE 488 (qui définit un protocole de commandes et d'échanges d'informations entre 15 appareils).

### Les points marquants de l'interface Centronics

- Transmission parallèle.
- Liaison unidirectionnelle pour les données.
- Protocole de Handshake pour le contrôle du flux d'informations.
- Validation des données.

Reprenons ces points un par un :

- La transmission des données est parallèle ; la plupart des imprimantes travaillent avec le code ASCII : ce code est sur 7 bits plus éventuellement 1 bit de parité. Il est donc plus simple, autant pour le processeur que pour le périphérique, de laisser les données en parallèle. Physiquement les 8 bits de données seront véhiculés sur 8 fils différents.
- Un signal de validation vient avertir le périphérique que les signaux présents sur les 8 fils de données sont bien des données utilisables.
- Une fois que le code ASCII a été mémorisé puis traité par l'imprimante, celle-ci prévient le processeur qu'elle est de nouveau prête à recevoir un autre code. Cette technique qui consiste à émettre un signal de retour pour valider une bonne communication et demander de continuer est connue sous le nom de "poignée de mains" (Handshake en anglais).

- On peut remarquer que le flux d'informations ne va que du processeur vers le périphérique. Ces 10 signaux (8 de données plus 2 de Handshake) suffisent pour la gestion d'un seul périphérique. Pour le cas où l'on veut connecter plusieurs périphériques il est nécessaire d'ajouter quelques signaux pour tenir compte :

- de l'adressage sélectif de chaque périphériques,
- de leurs vitesses de traitements qui ne sont pas toujours les mêmes.

### Chronogramme des signaux

Prenons un exemple simple.

L'ordinateur veut transmettre à l'imprimante le signe "#" qui correspond en ASCII à 35 ce qui en binaire donne :

0011	1001
3	5

Ceci se traduira physiquement sur les fils de sorties (données) par la figure 4.1.

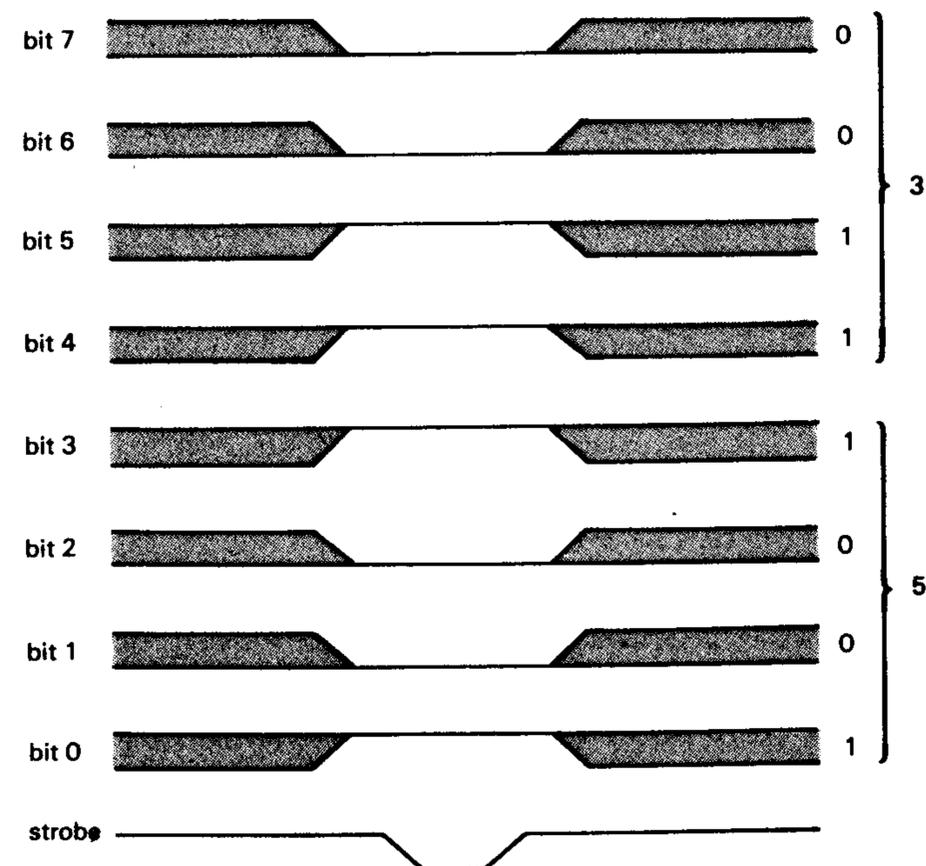


Fig. 4.1. — Transmission Centronics.

Les parties grisées correspondent à des états non connus

Le signal strobe sert à valider les données présentes : quand cette ligne est à l'état bas, l'imprimante sait que les données présentes sur le bus Centronics sont utilisables.

Au bout d'un certain temps (ou d'un temps certain pour des périphériques vraiment lents) l'imprimante répond par une impulsion négative sur le fil acknowledge (ACK) (fig. 4.2).

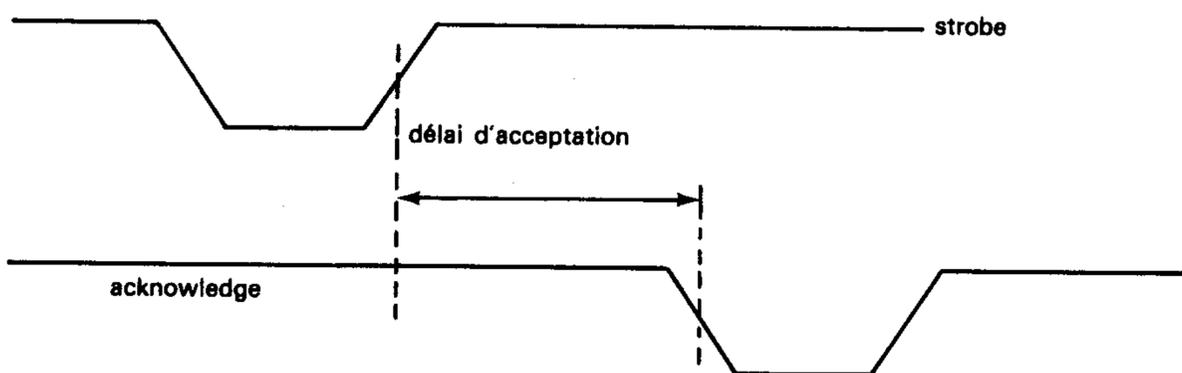


Fig. 4.2. — Diagramme des temps ACK et STB.

Le processeur peut continuer à émettre les données dès la réception du signal d'acceptation. Il peut aussi s'arrêter sans signal préalable, le périphérique étant en attente constante des ordres du processeur.

#### 4.2. LA MISE EN ŒUVRE SUR ORIC-ATMOS

ORIC offre de série l'interface Centronics minimum : c'est-à-dire 10 fils plus la masse électrique. Le connecteur n'est pas normalisé "Centronics" mais reste d'un emploi courant. Pour l'adapter sur une imprimante du commerce il faudra réaliser soi-même son cordon de connection.

Maintenant regardons un peu le plan de câblage de l'interface Centronics dans l'ORIC (fig. 4.3).

Le processeur d'interface VIA 6522 est câblé sur le connecteur de l'imprimante et sur le processeur sonore pour les huit fils de données. Par contre les deux fils utilisés pour protocole (strobe et acknowledge) sont reliés uniquement entre la VIA et le connecteur. Donc grâce à ces deux signaux il est possible de faire la différence entre les données destinées au processeur sonore et celles pour l'imprimante.

Comme on le voit la conception du matériel ("Hardware") est simple ce qui laisse supposer que tout le protocole de "Handshake" est géré par le microprocesseur 6502.

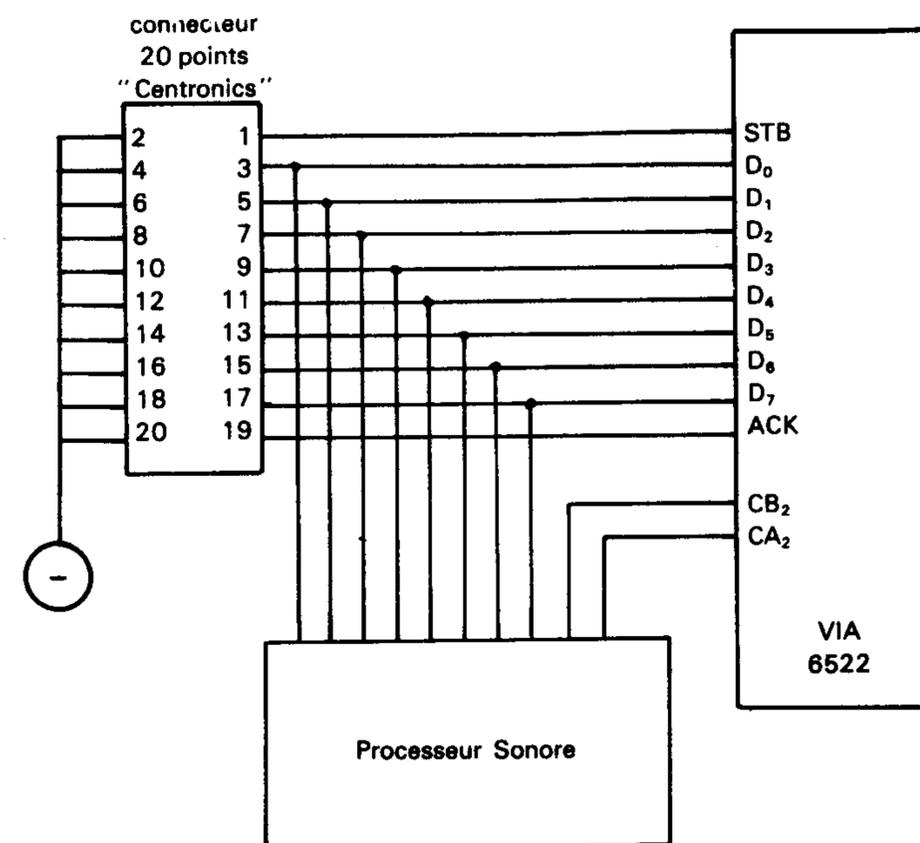


Fig. 4.3. — L'interconnexion du bus Centronics.

Nous n'allons pas entrer dans les détails de programmation des registre du VIA 6522 car l'intérêt de l'interface de puissance est de n'utiliser que des ordres BASIC. Ce qu'il faut retenir cependant :

- Le port PA<sub>0</sub> à PA<sub>7</sub> fonctionne de façon identique à celui-ci de la PIA 6821 décrite dans le chapitre 2.
- De même que la sortie PB<sub>4</sub>.
- Quant à CA<sub>1</sub> c'est une entrée qui est similaire du point de vue de l'utilisateur à l'entrée CA<sub>1</sub> de la PIA (et qui est utilisée dans la réalisation de l'interface analogique-digitale).

### 4.3. LE LOGICIEL POUR L'INTERFACE ET LES ORDRES "BASIC" ASSOCIÉS

#### Fonctionnement

Au vu de ce qui vient d'être schématiquement décrit il est aisé de comprendre comment fonctionne le logiciel.

La sortie de PB<sub>4</sub> ayant été initialisée à 1, le processeur 6502 envoie sur le port A la donnée à transmettre par l'intermédiaire du 6522. Après cela il met la broche PB<sub>4</sub> à zéro puis à 1 (impulsion de validation).

Enfin il attend la "réponse" de l'imprimante, sous forme d'une transition de 1 à 0 sur CA<sub>1</sub>. A cet instant tout est prêt pour un nouvel échange.

#### Utilisation

Pour l'utilisation en BASIC de l'interface Centronics on dispose de deux ordres BASIC :

```
LPRINT  
LLIST
```

- LLIST sert à sortir sur une imprimante le listage d'un programme. Nous ne l'utiliserons pas pour la suite.
- LPRINT sert à imprimer une valeur.

*Exemple:* Pour le possesseur d'une imprimante le programme :

```
10 A=5  
20 LPRINT A
```

imprime la valeur 5, et en plus fait faire un retour chariot: la tête d'impression revient en début de ligne et le papier avance d'un cran.

Pour que la routine BASIC n'envoie pas de retour chariot la syntaxe devient :

```
LPRINT A;
```

De façon plus précise, pour que la routine envoie sur le bus Centronics une valeur hexadécimale il faut taper :

```
LPRINT CHR$(N);
```

N entier variant de 0 à 255.

Pour N = 239 :

$$239 = 14 * 16 + 15$$

ce qui correspond en hexa à: #EF.

En fait :

```
LPRINT CHR$(239);
```

correspond sur l'écran à :

```
PRINT HEX$(239);
```

Pourquoi ?

Tout simplement parce que la plupart des imprimantes travaillent en ASCII. Ainsi quand l'imprimante reçoit un code hexa elle l'interprète comme un code ASCII, donc l'ordre LPRINT CHR\$(N) envoie sur le bus la valeur hexa de N sachant que cette valeur sera interprétée comme CHR\$(N) par l'imprimante.

Pour reprendre le premier exemple où l'on désirait envoyer "35" sur le bus Centronics, il faut faire :

```
LPRINT CHR$(35);
```

ce qui donne le chronogramme du premier exemple.

Après cette présentation nécessairement longue de l'interface Centronics de l'ORIC-ATMOS, nous pouvons passer sans tarder à la présentation de l'interface.

### 4.4. INTERFACE DE PUISSANCE POUR BUS CENTRONICS

Fig. 4.4. : le synopsis général de l'interface, le tout épaulé par un programme BASIC.

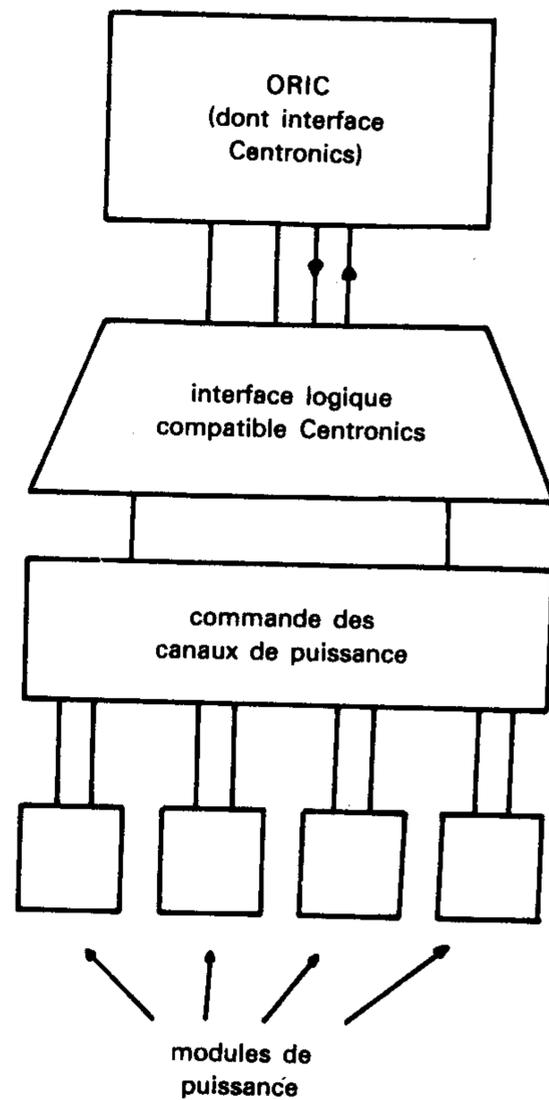


Fig. 4.4. — Synopsis général de l'interface de puissance.

### Interface logique compatible centronics

Cette partie doit répondre à deux exigences :

- participer au protocole de Handshake,
- mémoriser les données utiles.

Regardons ensemble le schéma de la partie logique (fig. 4.5).

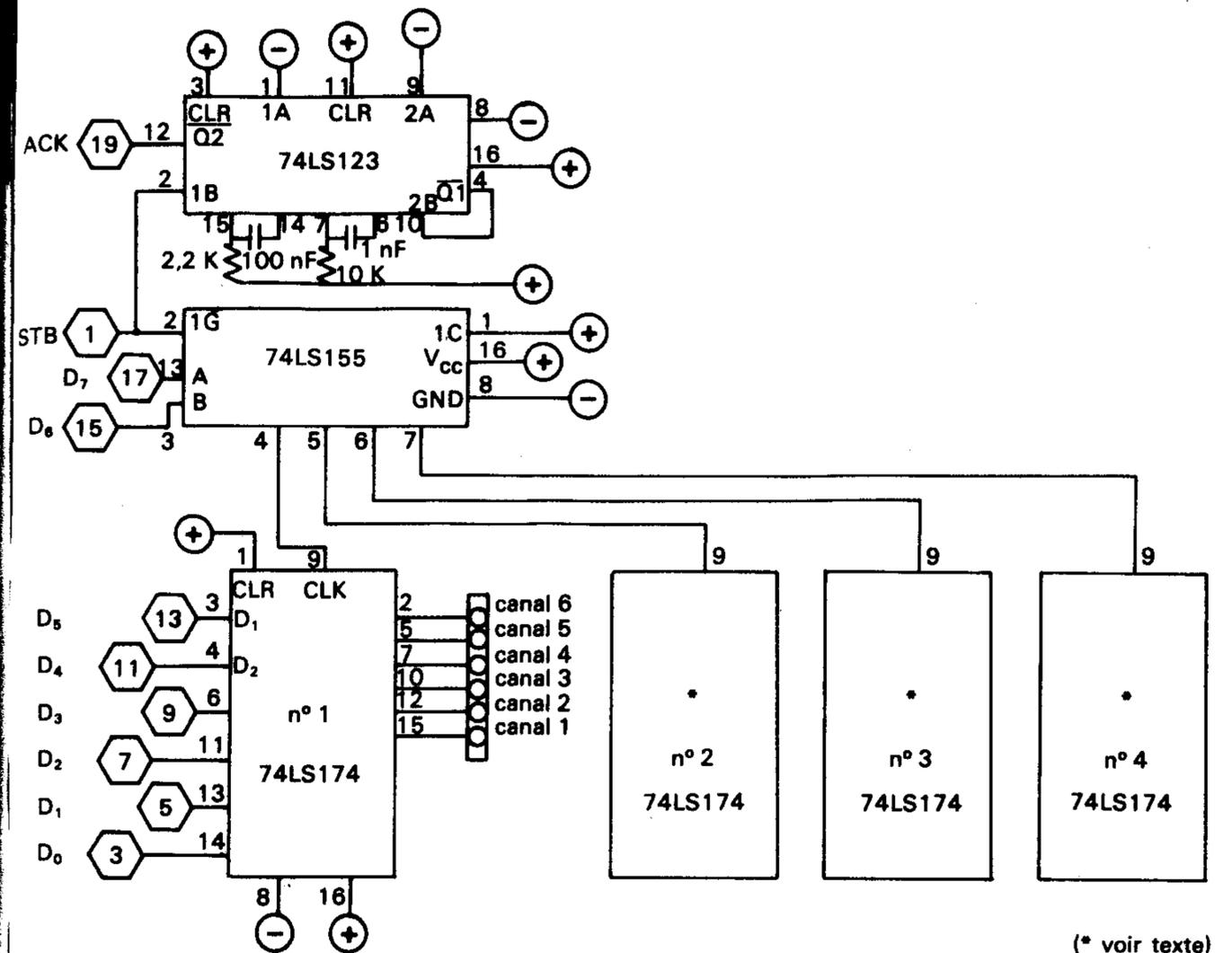


Fig. 4.5. — Interface de puissance. Partie logique.

### Protocole de Handshake

Le signal strobe (STB) indique par une impulsion que les données sont valides. Le 74LS123 est un double monostable. Le premier monostable est déclenché par le front montant de STB, et il génère une impulsion de 99 µs sur  $\overline{Q_1}$  (fig. 4.6).

$\overline{Q_1}$  déclenche le deuxième monostable câblé comme le premier.  $\overline{Q_2}$  étant câblé sur ACK (acknowledge).

On a bien généré sur ACK une impulsion négative  $99 + 4,5 \mu s$  après la strobe =  $103,5 \mu s$  la procédure de Handshake est bien simulée. (Simulée car ce montage ne s'occupe pas de savoir si les données ont bien été prises en compte ou non.)

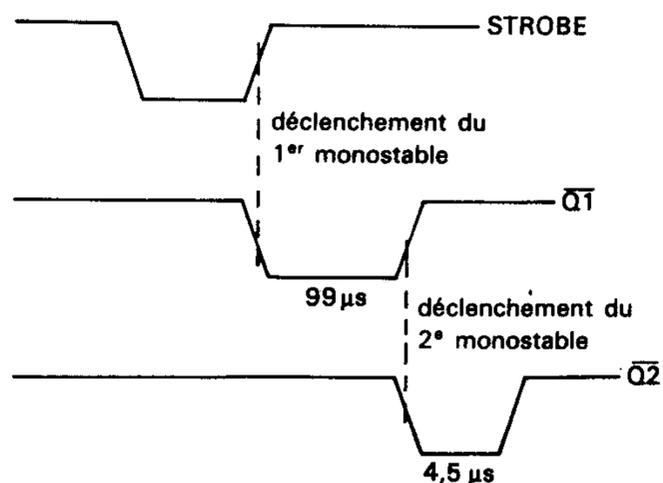


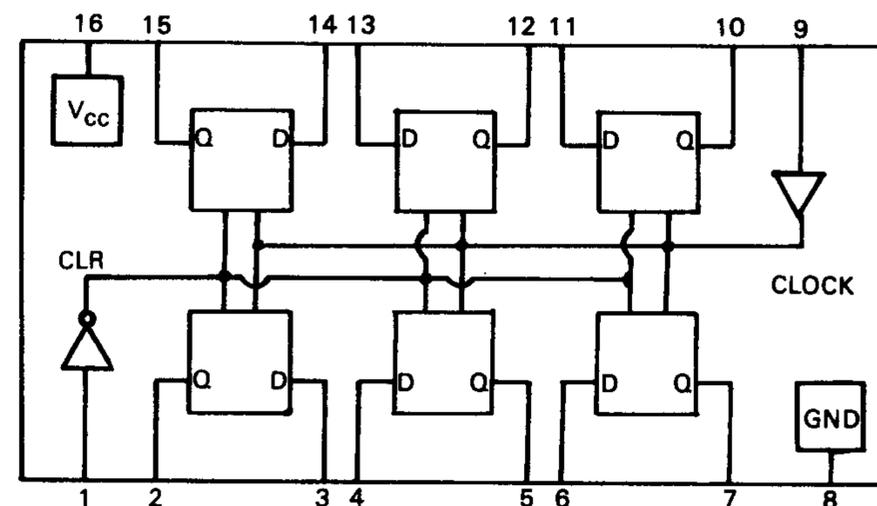
Fig. 4.6. — Diagramme des temps du Handshake.

### Mémorisation des données

Passons maintenant à la mémorisation des données. Sur le schéma une seule cellule 74LS174 a été représentée: le câblage des autres est identique. Les données  $D_5$  à  $D_0$  sont câblées en parallèles sur chaque circuit. Les sorties  $Q_1$  à  $Q_6$  sont les commandes des circuits de puissance. (Six par 74LS174 et quatre 74LS174 donne 24 canaux commandables.)

En regardant la table fonctionnelle du 74LS174 (fig. 4.7) on constate que la mémorisation des données — sur l'entrée D — se fait sur le front montant de l'horloge. Voyons maintenant d'où vient l'horloge. Le 74LS155, déjà utilisé, est un démultiplexeur: suivant les états présents sur les entrées A et B l'entrée 1G sera recopiée sur une des sorties. Par exemple si  $A = 1$  et  $B = 0$  on a  $Y_1 = 1G$ .

Les chronogrammes sont données en figure 4.8.



Entrées			Sortie
CLR	CLOCK	D	$Q_n$
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	$Q_{n-1}$

Fig. 4.7. — Fonctionnement et logique du 74LS174.

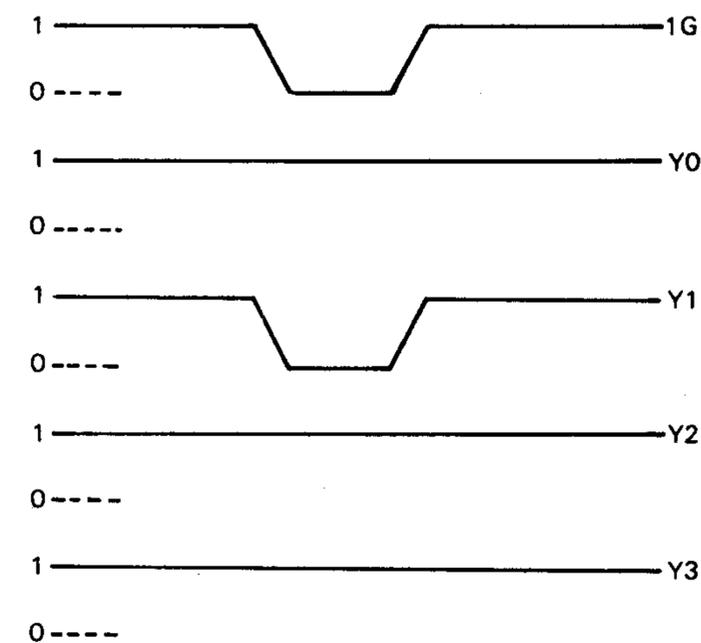


Fig. 4.8. — Chronogramme du 74LS155.

Le signal STB de validation des données est propagé sur la sortie sélectionnée par A et B. Or ces sorties attaquent les différentes "entrées horloges" des 74LS174: la boucle est bouclée.

### Ce qu'il faut retenir

L'interface logique est modulaire: il n'est pas indispensable de câbler les 24 canaux. Avec trois circuits vous avez une interface simple (6 canaux) facilement extensible (par groupe de 6 canaux). Le tout accessible directement par des ordres BASIC.

## 4.5. COMMANDE DES CANAUX DE PUISSANCE

Comme toujours, lorsque l'on désire commander des tensions et des intensités importantes, il est nécessaire de se prémunir d'un retour éventuel dans les circuits logiques: si par malheur un des circuits de l'ORIC recevait du 220 V, il serait temps pour vous... d'aller jardiner (c'est moins dangereux!).

Conscient du problème les électroniciens d'hier et d'aujourd'hui utilisent plusieurs techniques, dont les transformateurs et les optocoupleurs. Les transformateurs sont très utilisés pour les alimentations.

L'optocoupleur (fig. 4.9) utilise la lumière pour véhiculer un signal. L'isolation entre la partie émettrice et réceptrice du circuit peut atteindre plusieurs milliers de volts.

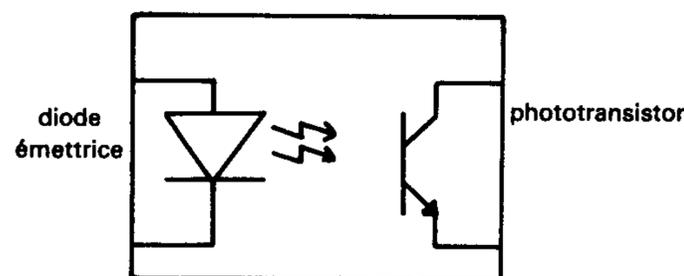


Fig. 4.9. — Principe d'un optocoupleur.

Nous allons utiliser un composant du type optocoupleur créé tout spécialement pour le grand public et la commande d'appareils en 220 V:

Le MCP 3022 de General Instrument, dont la structure interne est donnée figure 4.10, c'est un optotriac.

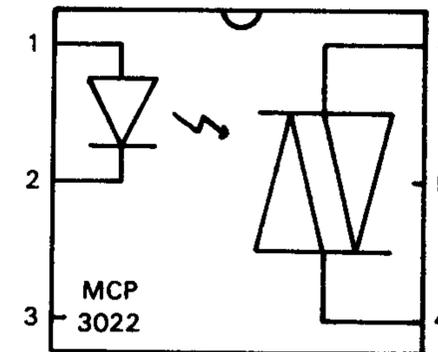


Fig. 4.10. — Schéma du MCP 3022.

Entre la broche 1 et 2 on dispose d'une diode émettrice de lumière. Cette diode sera directement commandée par les sorties du 74LS174 (fig. 4.11).

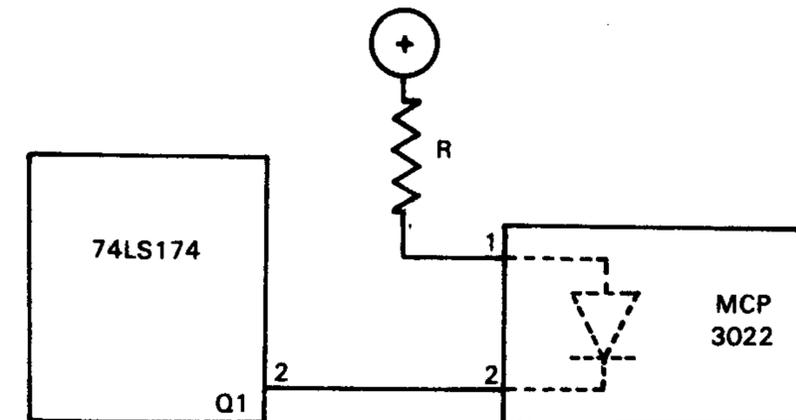


Fig. 4.11. — Connexion du MCP 3022.

La résistance R est définie par le courant I qui doit passer dans la diode  $I = 5 \text{ mA}$  donc  $R = 390 \Omega$ .

Maintenant nous allons revenir un peu sur la réalisation pratique de cette interface, car quelques précautions doivent être prises.

Le câblage des circuits est réalisé conformément au schéma (fig. 4.5) sachant que + représente le +5 volts et - la masse. Les hexagones sont représentatifs du connecteur relié à l'ORIC.

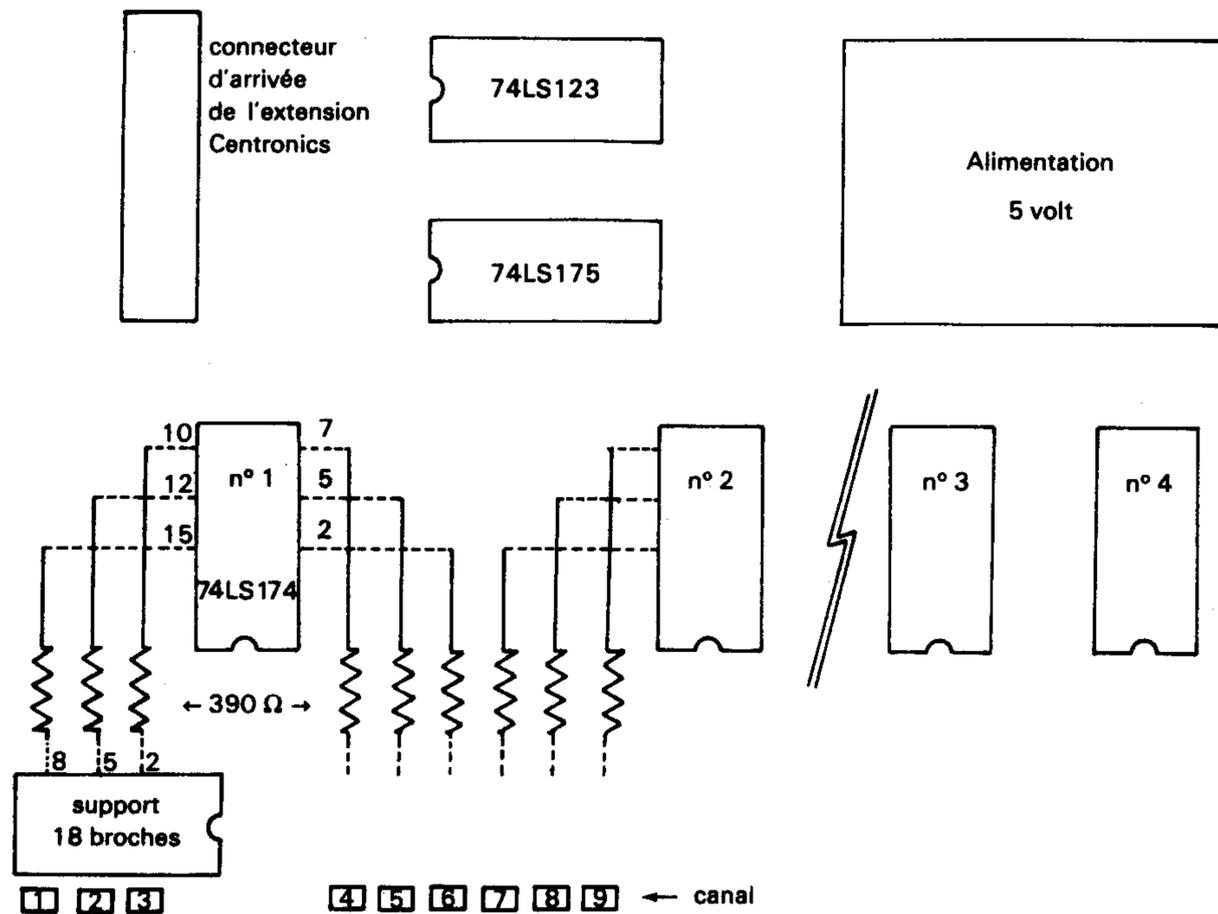


Fig. 4.12. — Implantation de l'interface de puissance (partie logique).

La figure 4.12 donne l'implantation ainsi que le câblage des sorties Q des 74LS174 et les liaisons avec les MCP 3022.

En trait fort la résistance vue de dessus et en trait pointillé la patte de la résistance qui continue sur le dessous de la plaquette de câblage. Les pattes des résistances sont suffisamment grandes pour ne pas à avoir

à utiliser du fil pour ce câblage. On utilise un support 18 broches qui recevra ultérieurement les optotriacs. Passons au câblage complet du support 18 broches.

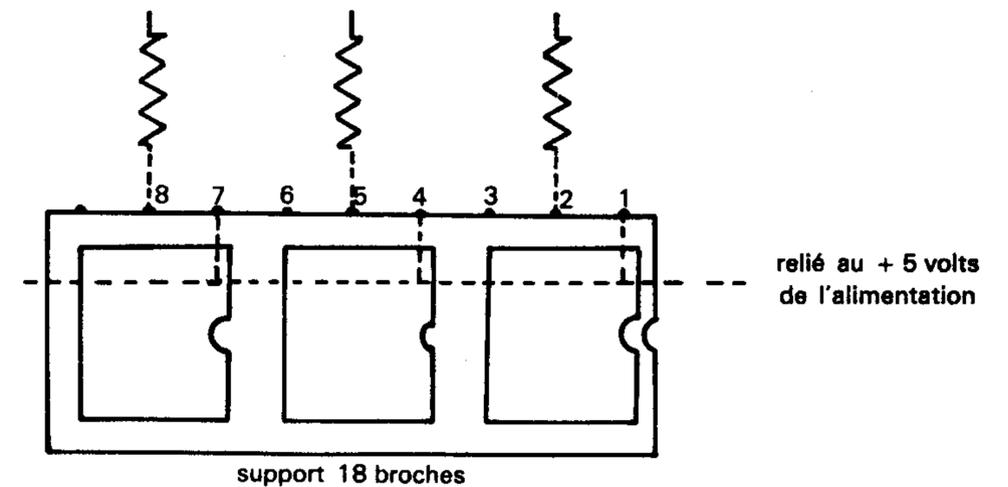


Fig. 4.13. — Détail de l'alimentation des MCP 3022.

La figure 4.13 donne le détail du câblage du MCP 3022. Les traits en pointillés représentent une équipotentielle sous la plaque à trous. Il ne restera plus qu'à placer les optotriacs pour terminer la 2<sup>e</sup> phase du montage. Avant cela on va contrôler si le câblage est bon en utilisant une diode LED enfoncée dans le support 18 broches entre les pattes 1-2, 4-5 et 7-8.

La patte la plus longue est à mettre au +5 volts (c'est-à-dire en 1 (ou 4 ou 8)).



Diode émettrice de lumière

## 4.6. PROGRAMME DE MISE EN ŒUVRE

Pour se remettre un peu de tout ce câblage, on va programmer un peu. On reviendra plus tard sur l'électronique de commande des lampes et des moteurs.

Si vous avez bien suivi le circuit décrit, vous avez en figure 4.14 le récapitulatif utile pour la programmation des canaux.

D <sub>7</sub>	1	0	1	0
D <sub>6</sub>	1	1	0	0
D <sub>5</sub>	canal 6	canal 12	canal 18	canal 24
D <sub>4</sub>	canal 5	canal 11	canal 17	canal 23
D <sub>3</sub>	canal 4	canal 10	canal 16	canal 22
D <sub>2</sub>	canal 3	canal 9	canal 15	canal 21
D <sub>1</sub>	canal 2	canal 8	canal 14	canal 20
D <sub>0</sub>	canal 1	canal 7	canal 13	canal 19
74LS174	n° 1	n° 2	n° 3	n° 4

Fig. 4.14. — Récapitulatif de l'adressage des canaux.

Sachant que pour sélectionner un canal il faut mettre la sortie à 0.

Pour sélectionner le canal 15, il faut envoyer le nombre hexadécimal 10111011 = #BB d'où LPRINT CHR\$(187). Et si vous avez positionné une LED sur le canal 15, elle s'allume.

Tapez ce programme destiné au canal 1 :

```

5 CLS
10 N=1
20 LPRINT CHR$(255);
30 VS=key$+"2"
40 N=ABS(N-51+ASC(V$))+1
50 WAIT N
60 LPRINT CHR$(254);
70 GOTO 20

```

Ce programme génère sur la LED une impulsion lumineuse dont la fréquence est réglable en appuyant sur la touche 1 ou 3.

Et pour les plus courageux qui ont câblé les 24 voies, voici le programme complet de mise en œuvre de l'interface

```

1 REM" INTERFACE DE PUISSANCE POUR CENTRONICS"
5 DIMY(24)
6 CLS
7 GOTO470
10 INPUT"NO DU CANAL";A
15 I=0
16 CLS
17 IFA=0THENGOTO470
20 IFA>24 ORA<1 THENGOTO210
30 B=A+6
40 REPEAT
45 I=I+1
50 B=B-6
60 UNTILB<=6
70 C=INT(2^(B-1))
80 IFY(A)=0THENX(I)=X(I)-C:Y(A)=1:GOTO200
90 IFY(A)=1THENX(I)=X(I)+C:Y(A)=0:GOTO200

```

```

200 LPRINTCHR$(X(I));
210 FORA=1TO24
220 PRINT"CANAL NO  "A;
230 IFY(A)=1THENPRINTTAB(20)"ALLUME"
240 IFY(A)=0THENPRINTTAB(27)"ETEINT"
250 NEXTA
260 PRINT:PRINT" 0 POUR TOUT ETEINDRE"
450 GOTO10
470 REM INITIALISATION ET EXTINCTION
480 RESTORE
490 FORI=1TO4
500 READN
510 LPRINTCHR$(N);
515 X(I)=N
520 NEXTI
530 DATA255,127,191,63
540 FORA=0TO24
550 Y(A)=0
560 NEXTA
570 GOTO210

```

Le mode d'emploi de ce programme est très simple : après l'avoir tapé faites, simplement RUN (return).

L'état des sorties de l'interface de puissance sera listé sur l'écran, sachant que l'initialisation du programme met toutes les sorties à l'état inactif.

Pour allumer un canal il suffit de taper le n° du canal (entre 1 et 24) puis de faire RETURN (si ce canal est déjà allumé cette procédure l'éteindra).

Il ne vous reste plus qu'à tester votre montage avec des diodes LED et ce programme.

Tout va pour le mieux, vous réglez en maître absolu sur votre population de LED ! Parfait, alors passons à la partie "Puissance".

## 4.7. RÉALISATION DE LA PARTIE " PUISSANCE "

Nous en étions resté à ce support 18 broches (un par groupe de trois canaux) qui jusqu'à présent est utilisé pour les led de contrôle (fig. 4.15).

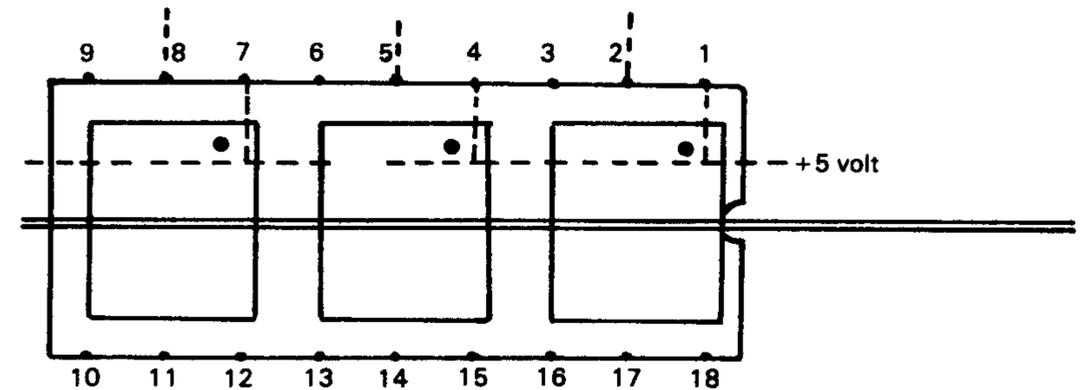


Fig. 4.15. — Implantation des MCP 3022.

**ATTENTION :** Les 2 parties (demi-plans pour les matheux) délimitées par le double traits, doivent être totalement indépendantes électriquement : les seuls ponts devant exister entre ces deux zones sont les optotriacs MCP 3022.

Comme on l'a vu lors de la présentation du MCP 3022 la broche 5 n'est pas utilisée : pire elle ne doit surtout pas être connectée à quoi que ce soit (qu'on se le dise !), donc seules les broches 10, 12, 13, 15, 16 et 18 du support seront utilisées.

Il n'est malheureusement pas possible d'utiliser l'optotriac directement : le courant maximum qu'il puisse supporter reste faible, de l'ordre de 100 milliampères. Il faut donc utiliser un triac plus puissant pour la commande d'appareils classique. Par exemple pour une lampe de 100 watts, il faut au minimum un triac passant 0,5 A et plus de 5 A pour un appareil de 1 000 W.

Avant de donner le schéma utilisant un triac puissant, on va tester le MCP 3022. (En contrôlant à chaque étape la mise au point est plus rapide!). Ce montage de test est donné en figure 4.16.

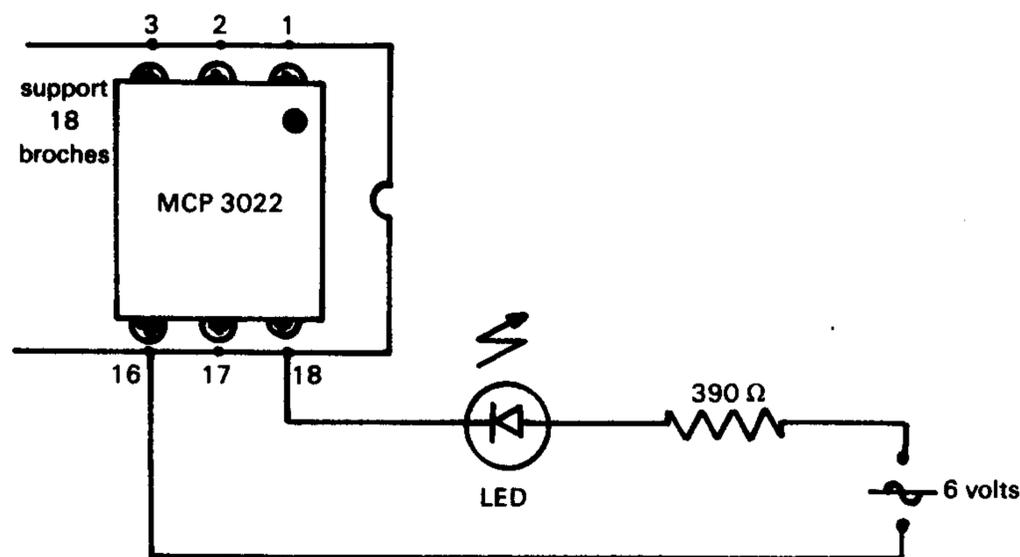


Fig. 4.16. — Montage de test des MCP 3022.

(Les numéros de broches sont donnés à titre indicatif car il y a d'autres possibilités.)

Il est important d'alimenter le montage (diode LED + résistance de  $390 \Omega$ ) en alternatif. Dans le cas d'une alimentation en continu le triac interne au MCP 3022 ne pourrait pas couper.

Une fois ce montage réalisé, utilisez le programme pour les 24 canaux et pilotez le canal que vous avez câblé: la Led doit s'allumer et s'éteindre à la demande.

Si cela ne se produisait pas, revenez à la dernière étape, vérifiez à l'ohmmètre votre câblage. Pour avoir facilement une tension faible alternative, vous pouvez utiliser le transformateur de l'alimentation avant le pont de diode.

*Nota:* Si le transformateur délivre plus de 6 V, mettre une résistance un peu plus forte ( $470 \Omega$  pour 9 V alternatif par exemple).

Maintenant passons à l'utilisation d'un triac plus puissant. Le TIC 236D de Texas Instruments (fig. 4.17).

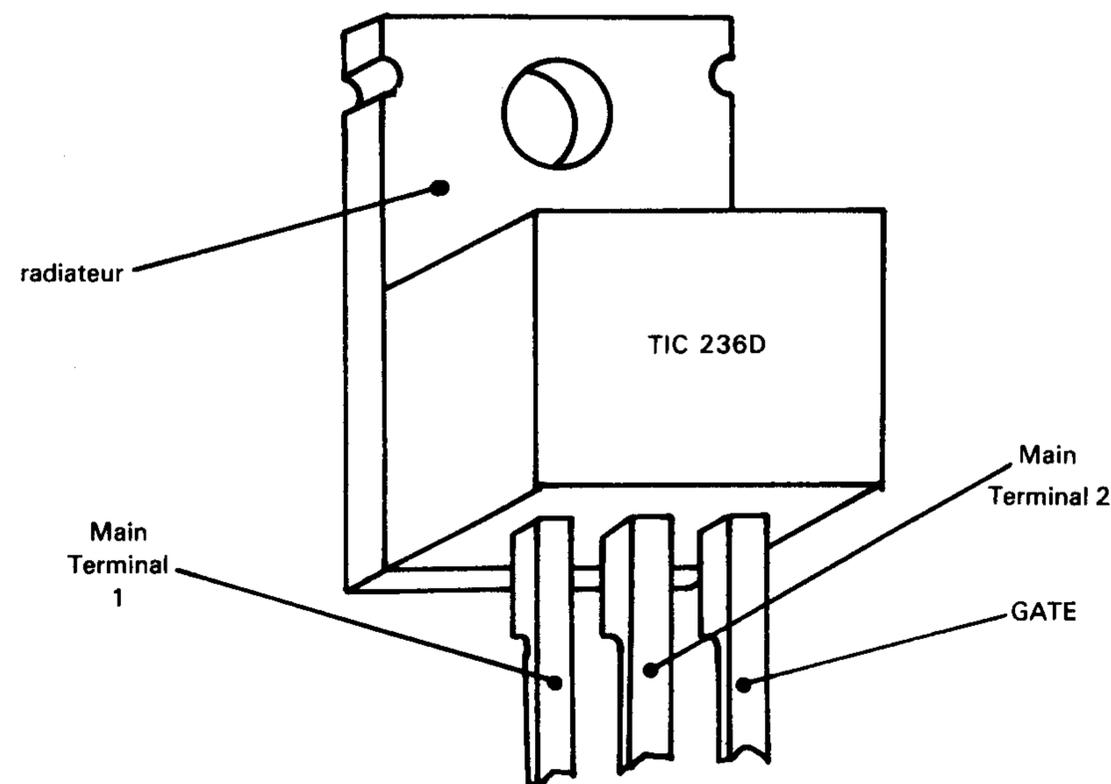


Fig. 4.17. — Dessin du TIC 236D.

Attention au suffixe: le "D" indique que le TIC supporte 400 V. Même si cela peut paraître surdimensionné pour une utilisation en 220 V, il est préférable d'avoir un "pied de pilote" assez important.

Il faut savoir que le radiateur est connecté à la broche MAIN TERMINAL 2.

Le schéma équivalent de TIC 236D est donné par la figure 4.18. On retrouve en fait exactement la même fonction que pour le MCP 3022.

L'intérêt d'utiliser le TIC 236D en plus du MCP 3022 réside dans le courant que supporte ce circuit: 125 A en crête et 12 A en régime continu.

Ce qui donne une puissance maximum en régime continu de 2 500 W. Mais attention pour des puissances aussi conséquentes il est nécessaire de boulonner un radiateur sur le TIC 236D.

Et maintenant comment brancher le TIC 236D. C'est simple et représenté sur la figure 4.19.

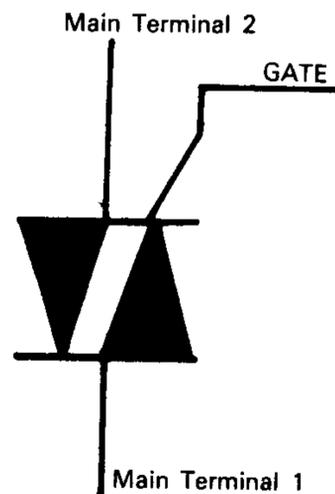


Fig. 4.18. — Synopsis du TIC 236D.

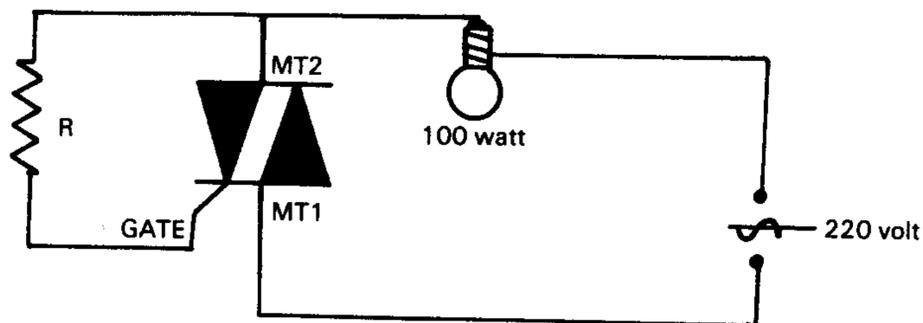


Fig. 4.19. — Fonctionnement du TRIAC.

Suivant la valeur de la résistance le courant qui passe dans la broche G est suffisant pour déclencher le triac : une fois déclenché, le triac laisse passer le courant et la lampe s'allume.

Faites ce petit montage avec une résistance  $R = 47 \text{ k}\Omega$  (1/4 Watt suffit pour la résistance) : la lampe ne s'allume pas. Débrancher le secteur et changer la résistance :  $R = 10 \text{ k}\Omega$  : une lampe de 100 W s'éclaire.

Un peu de théorie : cela vous permettra de mieux mettre au point votre montage représenté en figure 4.20.

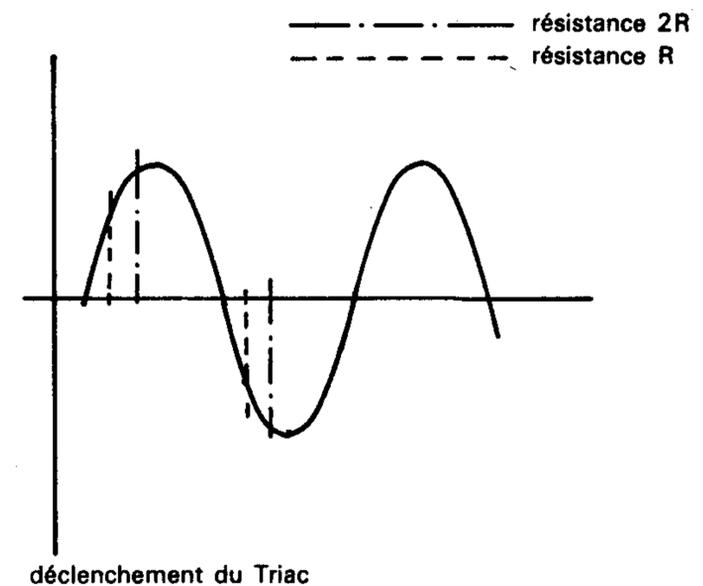


Fig. 4.20. — Le découpage de la tension.

Tension du secteur : plus la résistance est forte plus la tension doit être grande pour que le courant de la gachette (GATE) soit suffisant pour déclencher le TIC.

Si la résistance est trop grande le courant reste toujours inférieur au courant critique de déclenchement et donc la lampe ne s'allume jamais. Si la résistance est trop faible le courant devient trop fort et la résistance chauffe exagérément. Alors ne soyez pas tenté de baisser trop la valeur de la résistance.

Un moyen simple de voir si tout est correcte : au bout de quelques minutes débranchez le secteur et faites apprécier "à la main" par votre meilleur ennemi la température de la résistance et celle du TIC :

— si le TIC est très chaud, montez un radiateur (attention à ne pas toucher ce radiateur sous tension !),

— si la résistance est trop chaude, augmentez sa valeur.

Vous êtes un peu plus familiarisé avec l'utilisation d'un TRIAC.

Alors il suffit de le raccorder au montage déjà réalisé. On a vu qu'en changeant la résistance R, la lampe s'allume ou s'éteint. Or le MCP 3022 marche soit en passant (résistance nulle) soit en bloqué (résistance infini), d'où le schéma équivalent figure 4.21.

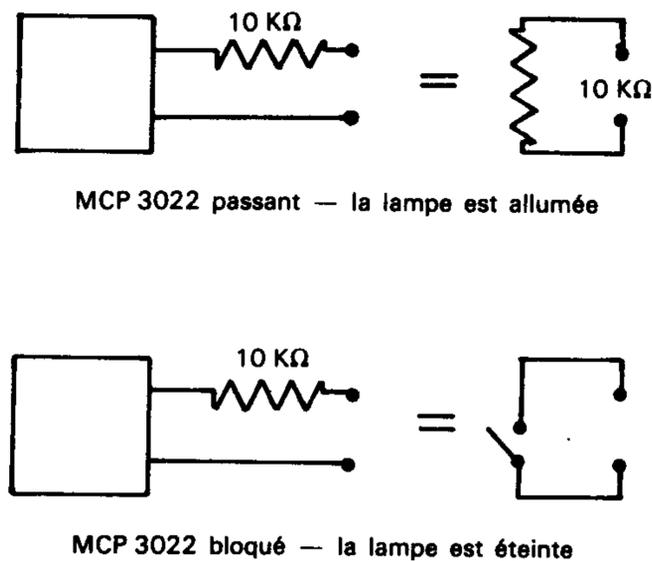


Fig. 4.21. — Extinction et allumage de la lampe.

Les plus rapides ont déjà dessiné le schéma final; le voici pour les autres (fig. 4.22).

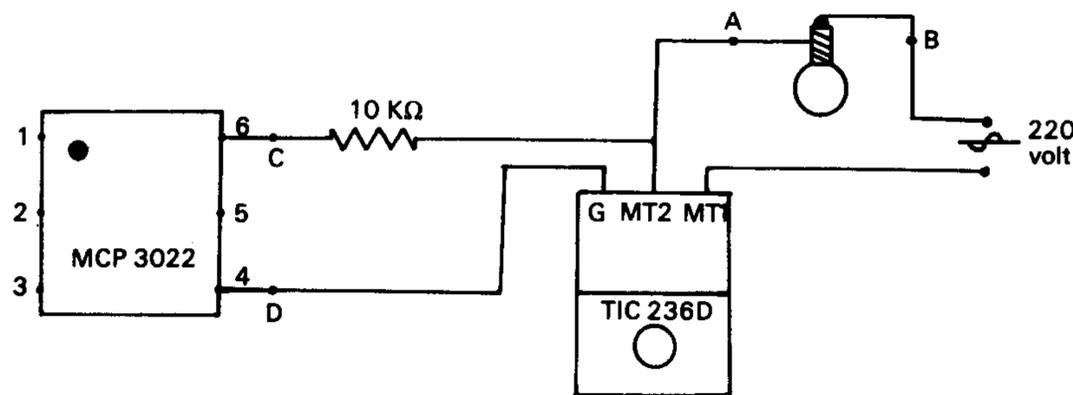


Fig. 4.22. — Schéma définitif de la partie puissance.

Attention toute cette partie ne doit pas avoir de contact électrique avec la partie logique.

## 4.8. CONCLUSIONS ET CONSEILS

Quelques petits conseils de réalisation.

### Un petit exemple: vous désirez commander un radiateur électrique

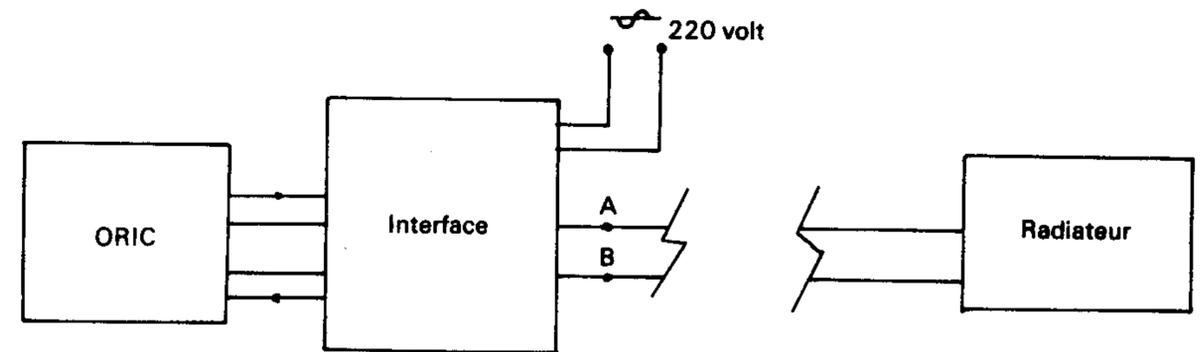


Fig. 4.23. — Montage d'exemple 1.

Le double trait indique que la commande est en 220 V. Alors deux possibilités :

— Soit tout votre montage de puissance est dans la même boîte, et les deux fils pour commander le radiateur partent de A et de B (sur la figure 4.22); cela est maladroit car il y a une grosse intensité qui passe dans ces fils d'où une perte importante d'énergie (il faut utiliser une grosse section pour les fils) et du danger si ces fils sont apparents.

— Soit le triac et la résistance sont connectés dans une boîte à part près du radiateur (ou éventuellement suivant les possibilités dans le radiateur). Il faut donc une prise secteur pour le radiateur, mais les deux fils de commande — s'ils restent en 220 V — sont à très faible intensité: 5 mA, à peu près 1 W, ce qui est faible. La section des fils peut être faible... à vous de choisir suivant ce que vous pilotez!

Il est conseillé de n'utiliser des résistances de valeur inférieur à 10 K qu'avec précaution; faites vos propres essais, suivant les conditions il faut savoir aménager le schéma.

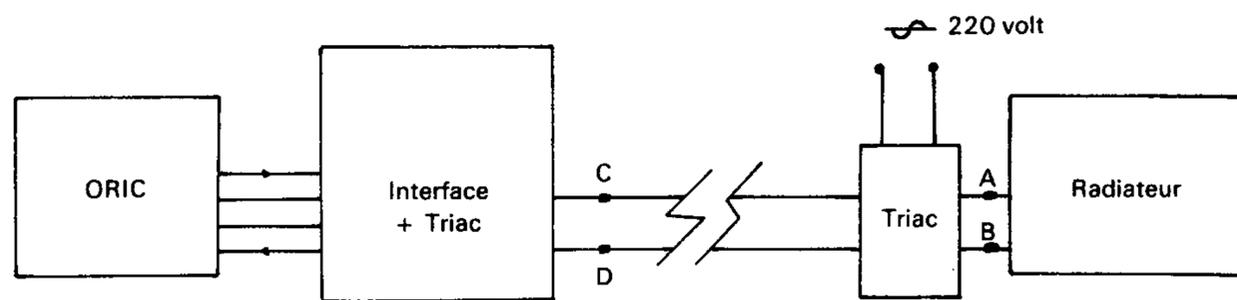


Fig. 4.24. — Montage d'exemple 2.

### Parasitage

Comme la plupart des montages à bases de TRIAC il se produit un certain parasitage, surtout à l'allumage et à l'extinction. L'utilisation que nous avons de notre interface étant relativement "calme" la seule "nuisance" que nous avons constaté est un léger "Ploc" sur une radio en AM à l'allumage de la lampe. Néanmoins et surtout pour les lecteurs habitants en appartement, vérifiez que vous ne gênez pas autrui.

Il est possible de compliquer le schéma pour diminuer ces nuisances — très faibles répétons-le : l'ORIC n'est pas affecté par l'allumage ou l'extinction — mais ce n'est pas le but de l'ouvrage que de faire de l'électronique.

Il est, bien sûr, possible d'utiliser d'autres sortes de TRIAC. A vous de bien vérifier son comportement et son utilisation. Le principe du schéma reste bon mais il vous faudra sûrement l'adapter.

Un dernier conseil, prenez bien soin d'isoler les parties supportants du 220 V.

### Applications

Il vous est désormais possible de commander les différents appareils ménagers qui composent habituellement une maison. Si cette interface n'est pas perfectionnée au point de passer l'aspirateur ou de donner le biberon au petit dernier, elle peut être facilement utilisée pour réguler

la température des pièces ou d'une serre, en utilisation conjointe avec le capteur de température.

Une des applications la plus classique reste la simulation de présence lorsque vous vous absentez. En annexe 2, quelques informations ont été donné pour acquérir l'heure : suivant celle-ci vous pouvez commander l'extinction ou l'allumage de lampe qui feront croire, vu de l'extérieur, à une présence dans la maison. Cela ne vous dispensant pas de fermer votre porte à clé.

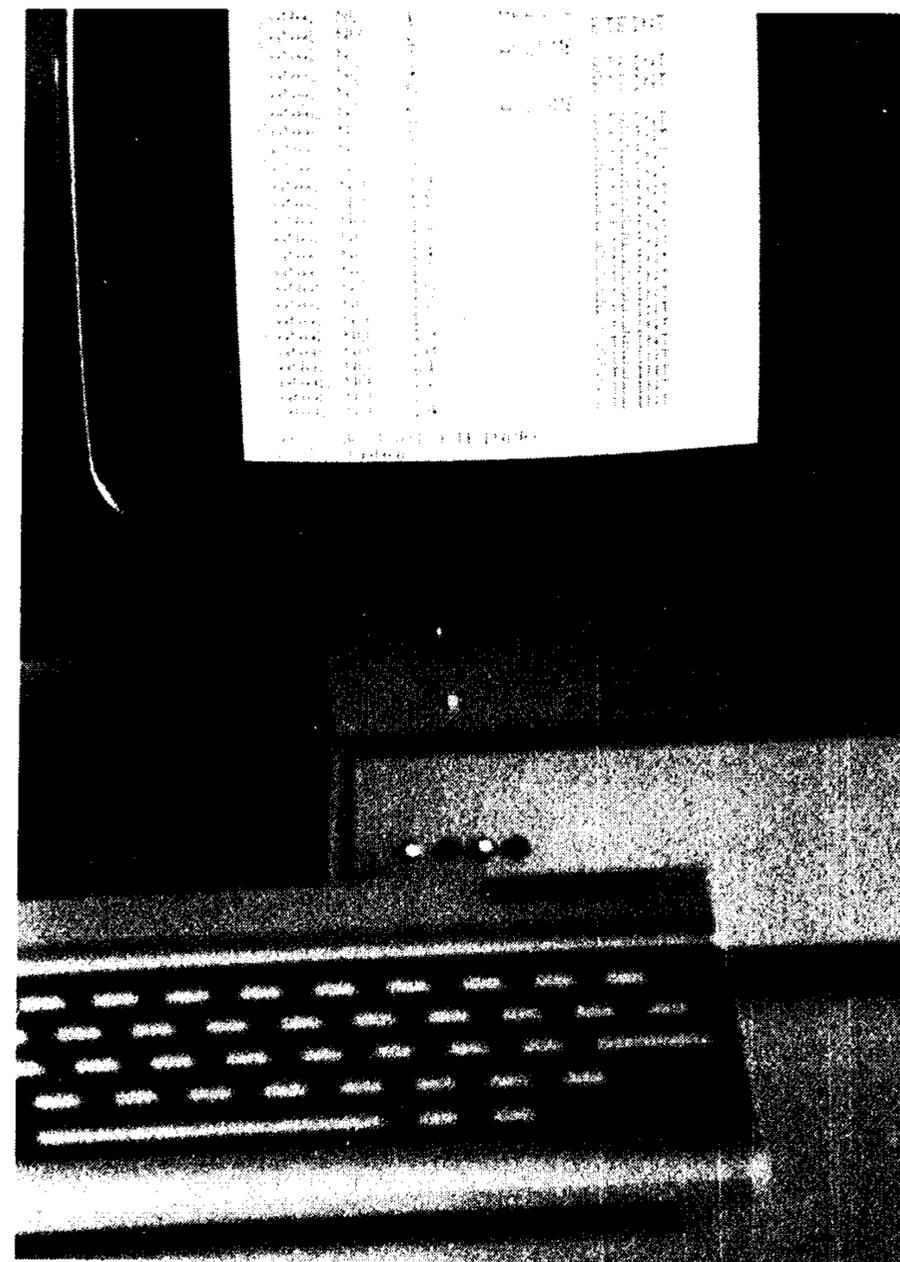


Fig. 4.25. — Photo de la réalisation des auteurs.

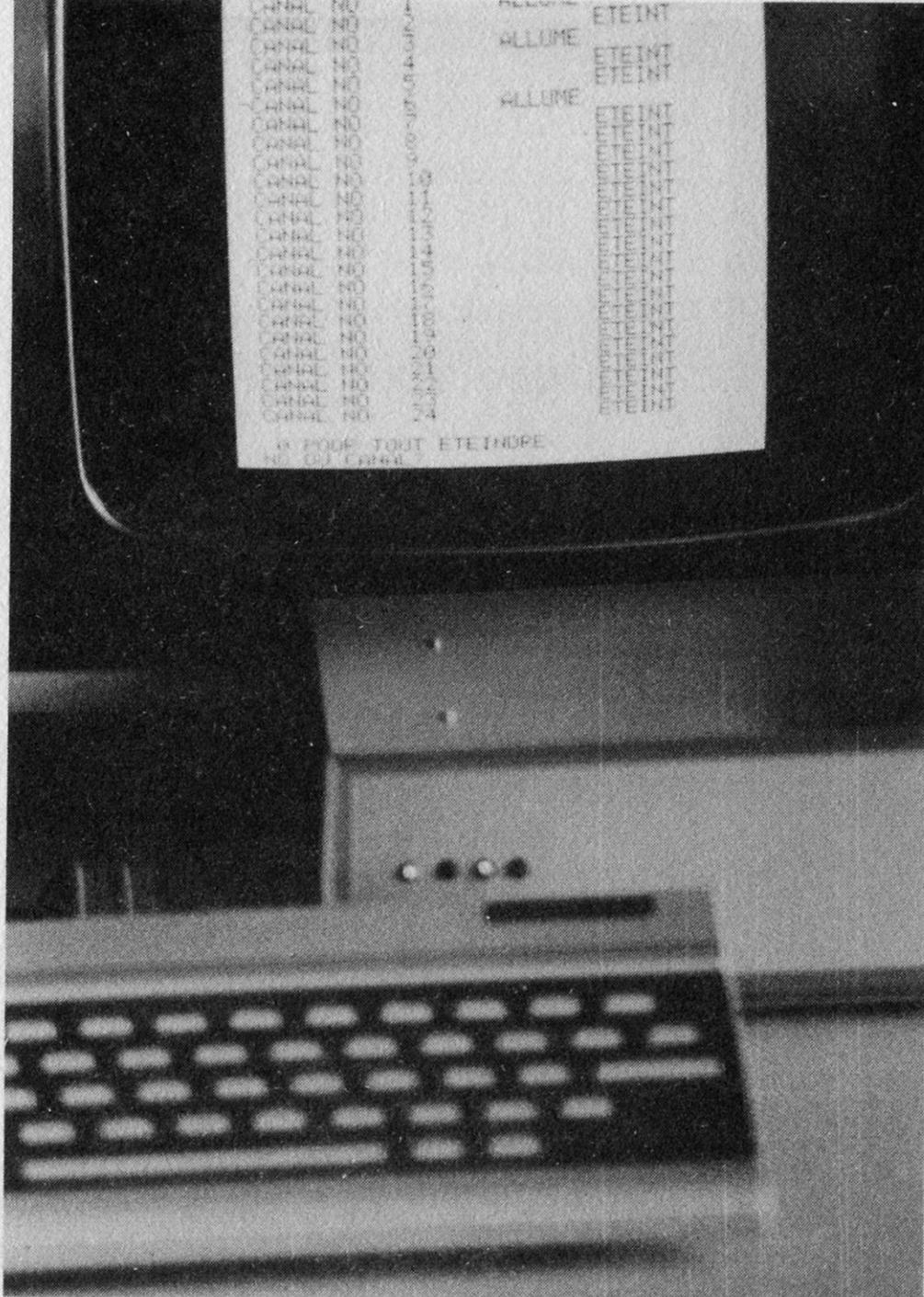


Fig. 4.25. — Photo de la réalisation des auteurs.

Cette interface nécessitant une alimentation spécifique, le plus simple est de reprendre le schéma de l'alimentation donné au chapitre 6.

Notre réalisation est présentée sur les photos suivantes (fig. 4.25, 4.26). Les quatre leds au-dessus de l'ORIC sont les recopies des canaux 1, 2, 3 et 4. On remarque bien la conformité d'allumage avec le programme et la clarté de la présentation de celui-ci.

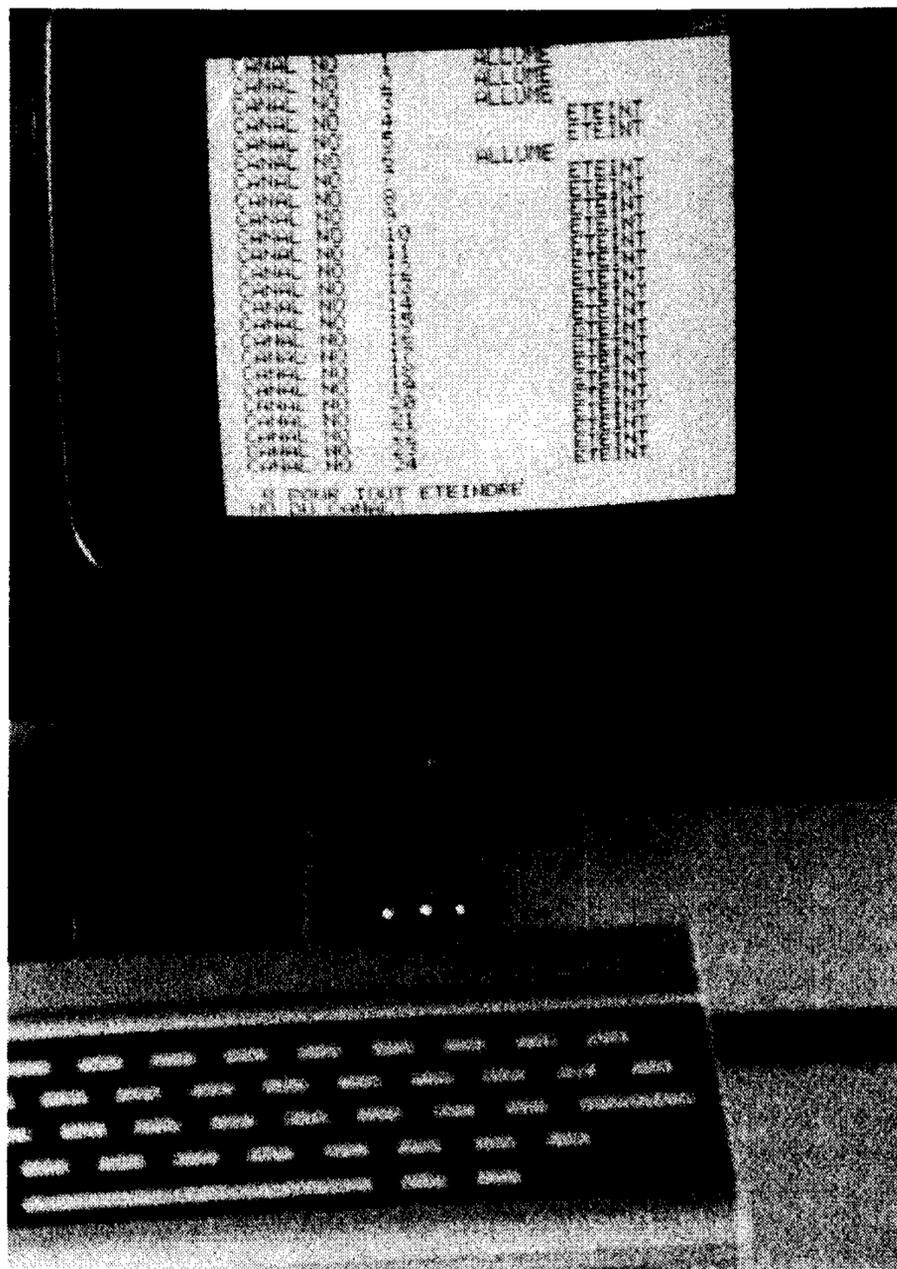


Fig. 4.26. — Photo de la réalisation des auteurs.

## 5

# L'ORIC à cœur ouvert

### 5.1. ET L'ORIC DEVINT ATMOS

- Nous avons déjà parlé de la différence qui existait entre l'ORIC-1 et l'ATMOS. Celle-ci résidait essentiellement dans la ROM : c'est-à-dire la mémoire morte qui contient le Basic et les routines d'exploitations du matériel.

Physiquement cette ROM peut se présenter sous deux aspects :

- en un seul circuit : ROM 27128 d'une capacité de 16 kilo-octets,
- en deux circuits : deux ROM 2764 d'une capacité chacune de 8 kilo-octets.

- La plupart des ATMOS ainsi que des ORIC sont livrés avec la première version. Néanmoins, nous proposons dans ce chapitre de rendre votre ORIC compatible des deux aspects.

Pourquoi ? Tout simplement parce qu'à l'heure où nous écrivons ce livre il est assez facile de se procurer des 2764 chez les revendeurs de composants et que notre programmeur de PROM est conçu pour ce genre de composants.



Ainsi une fois compatible il vous sera possible de changer la mémoire morte, soit pour y mettre un logiciel entièrement de votre cru soit pour la remplacer par le logiciel de la ROM de l'ATMOS.

- Voici le programme qui permet de dupliquer le logiciel de l'ATMOS à partir de l'interface programmeur d'EPROM 2764.

Ce programme est dans son principe similaire à celui qui est proposé au chapitre 2. Sa spécificité réside dans la ligne 260 qui lit la ROM de l'ORIC pour l'écrire sur le programmeur :

```

1 REM"DUPLICATEUR DE LA ROM INTERNE"
5 CLS
10 POKE#3F8,0
20 POKE#3FA,255
30 POKE#3F8,255
32 REM
34 REM
40 POKE#3F9,0
50 POKE#3F8,255
60 POKE#3F9,255
62 REM
64 REM
70 POKE#3F7,0
80 POKE#3F6,252
90 POKE#3F7,255
91 POKE#3F6,255
92 REM
94 REM
100 PRINT"DUPLICATEUR DE LA ROM "
110 PRINT"  ATMOS SUR 2764  "
120 PRINT
130 PRINT"EPROM NO 1 : 1 "
140 PRINT"EPROM NO 2 : 2 "
150 GETA$
160 IFA$="1" THEN INIT=0:GOTO190
170 IFA$="2" THEN INIT=8192:GOTO190
180 CLS:GOTO100
190 PRINT"PLACER L'EPROM"
192 PRINT:PRINT"INTERRUPTEUR SUR MODE PROGRAMMATION"

```

```

194 PRINT:PRINT"COPIAGE DE LA ROM "A$
195 PRINT"TAPER SUR P"
196 GETB$
197 CALL#E76A
200 FOR N=0 TO 124 STEP4
210 POKE#3F6,N+128
220 REM
230 FOR M=0 TO 255
240 POKE#3FA,M
250 REM
260 POKE#3F8,PEEK(#C000+M+INIT+N*64)
270 REM
280 POKE#3F6,N
290 REM
300 FOR I=1 TO 25
310 NEXT I
320 REM
330 POKE#3F6,N+128
340 NEXT M
350 NEXT N
360 CLS:PRINT:PRINT:PRINT
370 PRINT"FIN DU COPIAGE DE LA ROM "A$
380 PRINT:PRINT:PRINT
390 PRINT"PLACER L'INTERRUPTEUR EN MODE LECTURE"
400 PRINT:PRINT"SORTIR L'EPROM"
410 PRINT:PRINT"POUR CONTINUER FAIRE RUN"
420 CALL#E93D

```

#### Programme duplicateur de la ROM interne

- D'abord, il vous faut connecter l'interface à un ORIC-ATMOS puis charger le programme. Ensuite faire RUN et laissez-vous guider par le Menu. Une fois les deux EPROM 2764 programmées, vous avez une copie fidèle en deux circuits du logiciel de l'ATMOS.

A ce stade deux remarques : Il n'a pas été utile d'ouvrir l'ATMOS pour réaliser cette duplication. Par contre, il faut disposer d'un ATMOS et il faut savoir que la duplication de logiciel — à juste titre dénommée "Piratage" — est interdite sauf à usage strictement privé du copiste.

Ce sont d'ailleurs les mêmes lois qui s'appliquent au présent ouvrage. Nous donnons donc ce programme sans garantie d'utilisation.

## Réalisation

Maintenant vous avez vos 2 EPROM's 2764 avec le logiciel de l'ATMOS. Il suffit de vous munir d'un tournevis cruciforme, un fer à souder, de la tresse à dessouder, un support de circuit 28 broches et un autre de 14 broches.

- Pour commencer déconnectez entièrement votre ORIC. Retournez-le : sortir délicatement la languette métallique donnant le n° de série (à partir de ce moment votre ORIC n'est plus sous garantie !). A l'aide du tournevis, sortir les 6 vis et dégagez le capot. La photographie 5.1 vous donne les principaux points importants.

- Dans le cas où il n'y a qu'un circuit 27128 il est sur support à l'emplacement n° 2. Le n° 1 est repéré à côté. Pour placer le support il faut d'abord sortir la soudure qui bouche les trous de l'emplacement n° 1. Utilisez la tresse à dessouder et essayez de chauffer le moins possible les pistes. Prenez votre temps pour cette opération, elle doit être réalisée avec soins. Une fois ceci achevé, placez le support 28 broches et soudez-le sur le circuit imprimé. (Les trois vis en haut du circuit imprimé permettent de désolidariser celui-ci du clavier.)

Répétez la même opération pour l'emplacement n° 3 sous le haut-parleur.

- A présent il suffit de mettre un 74LS00 sur le support IC11, et l'EPROM 2764 n° 1 (resp. n° 2) dans l'emplacement n° 1 (resp. n° 2). Attention à ne pas vous tromper !

Puis prenez une pince coupante et sectionnez le pontet entre b et a de LK2. Voilà c'est fini, il vous suffit de remonter le circuit imprimé, refermer la boîte et mettre en marche.

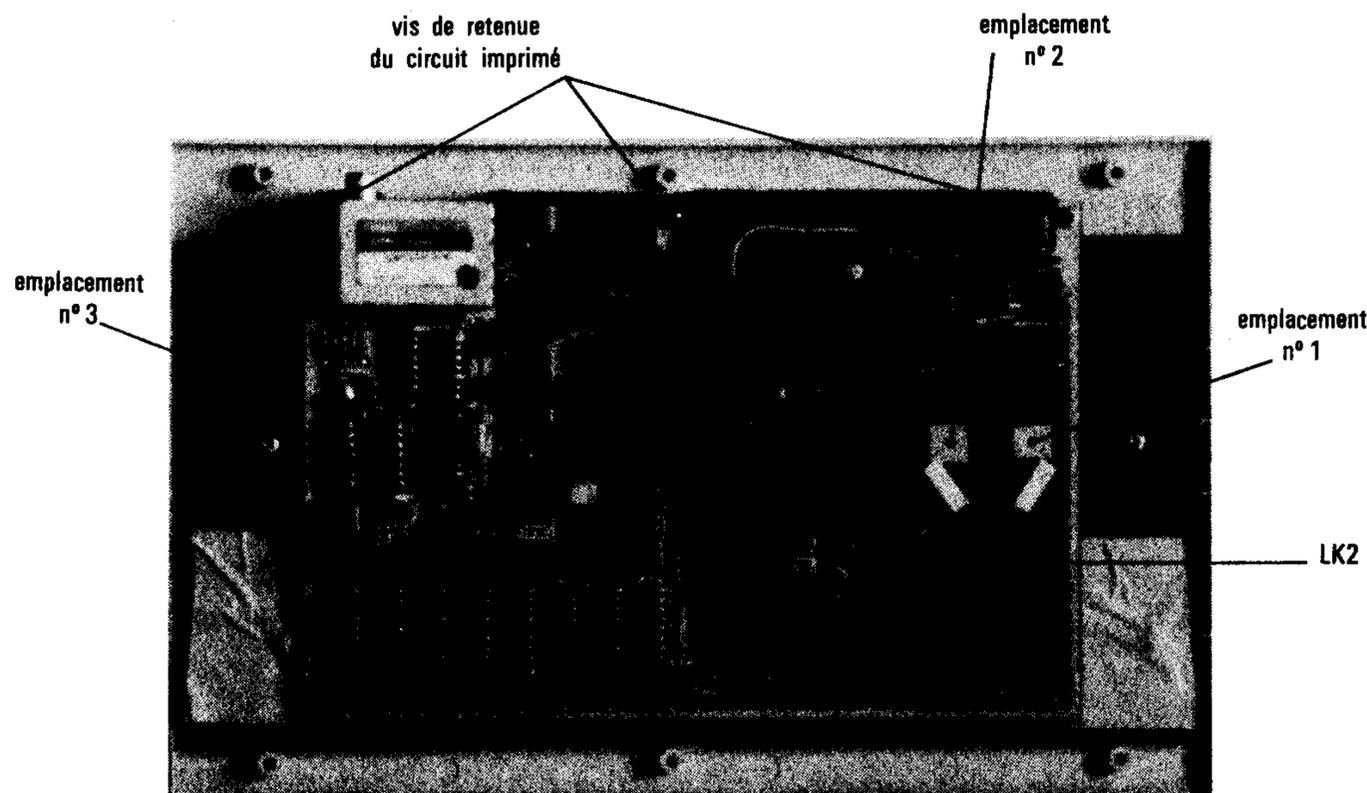


Fig. 5.1. — Photo de la carte d'ORIC.

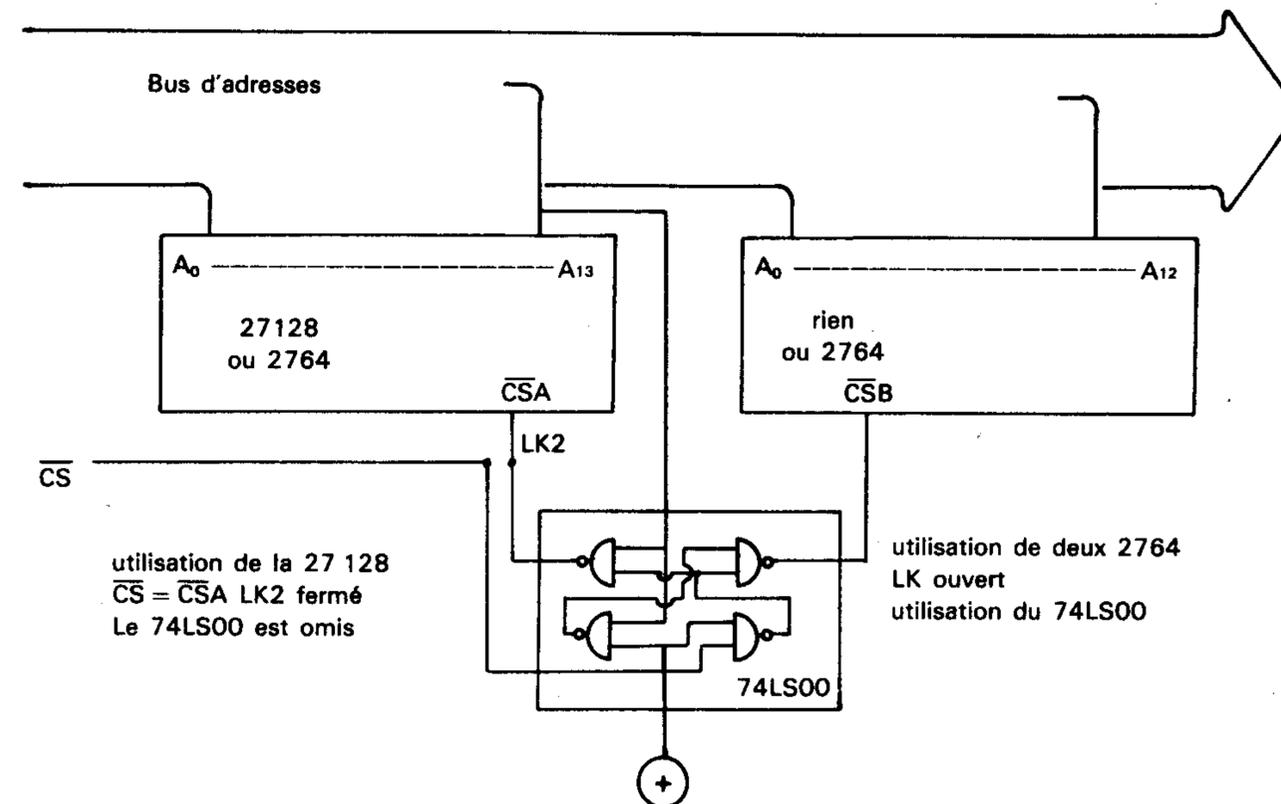


Fig. 5.2. — Schéma d'adressage de la mémoire morte.

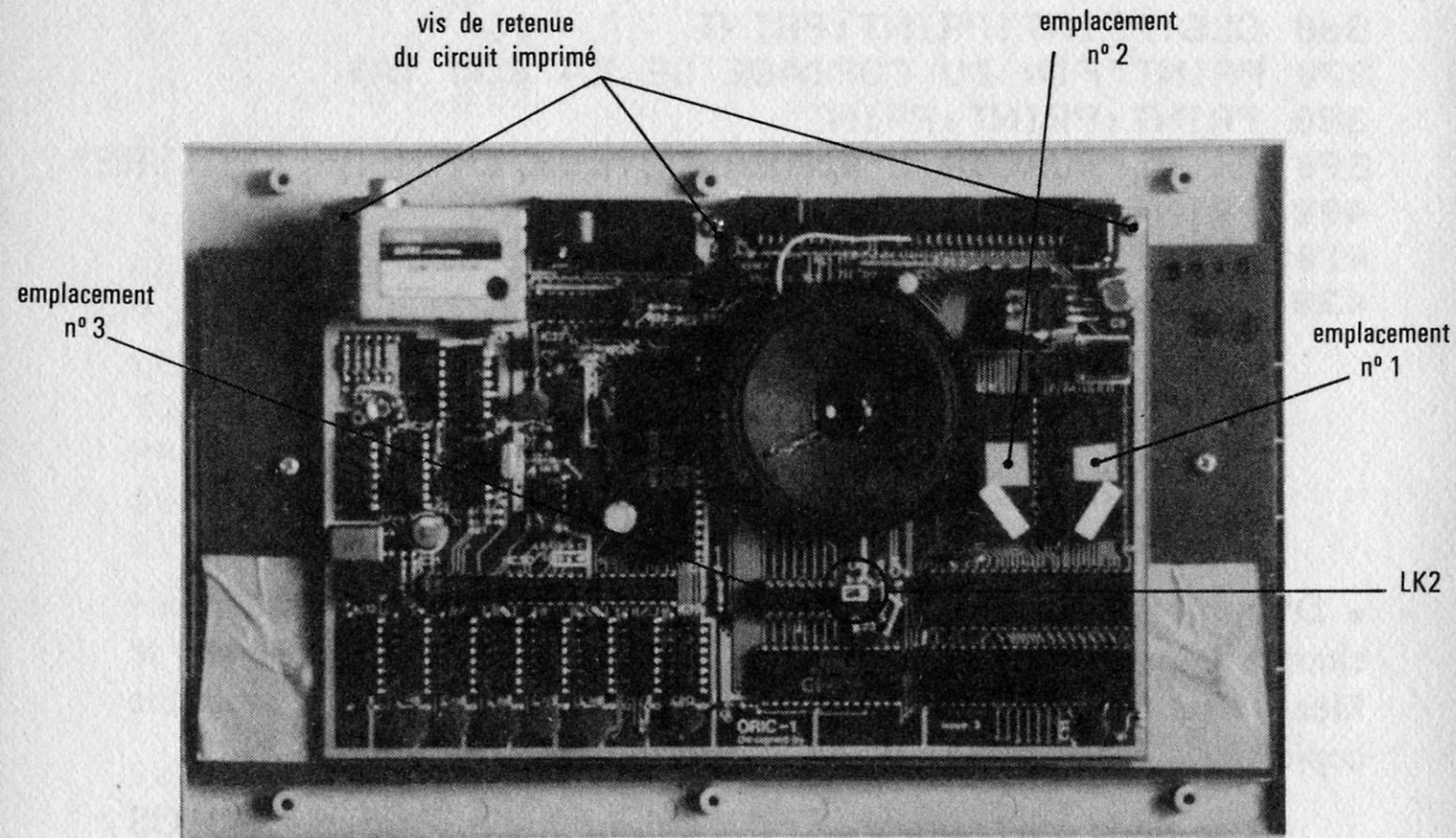


Fig. 5.1. — *Photo de la carte d'ORIC.*

Notre réalisation a marché au premier essai à la mise sous tension. Vous disposez à présent du BASIC complet de l'ATMOS. Alors bonne programmation !

Pour ceux qui sont intéressés par les fondements théoriques de cette mutation, voire la figure 5.2 qui donne l'explication de l'adressage de la ROM dans l'ORIC.

## 5.2. REPHASAGE DE Ø2

Nous allons décrire un léger problème dans la conception des entrées-sorties de l'ORIC et comment y remédier.

### Description

- Notre interface utilise l'horloge Ø2 du microprocesseur. Ceci est bien compréhensible : tous les circuits périphériques modernes utilisent cette phase de l'horloge pour se synchroniser.

Voici en figure 5.3 le câblage interne de Ø2.

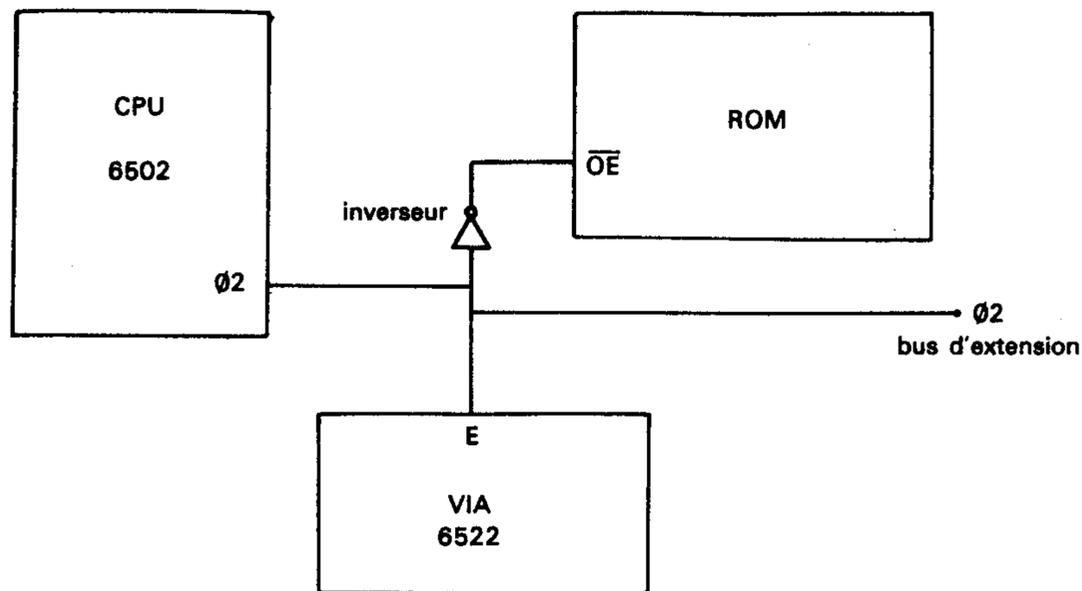


Fig. 5.3. — Ø2 dans l'ORIC.

C'est le 6502 qui délivre Ø2 aux autres périphériques, or il existe un phénomène en électronique qui veut qu'il ne soit pas possible avec un seul circuit (du type du 6502) d'alimenter beaucoup d'autres circuits. Par comparaison il n'est pas possible avec un seul radiateur de chauffer toute une maison. En électronique, on utilise des circuits relais comme cet inverseur en aval de la ROM pour redonner au signal une certaine vigueur, et surtout ne pas l'affaiblir en amont.

En effet, si on "demande trop" à Ø2 en sortie, la VIA risque d'être mal synchronisée. Ce défaut entraîne — entre autre — un empêchement pour relire les programmes K7.

A titre d'expérience, nous avons relié un fil unique à la broche Ø2 du bus d'extension. 30 cm de fil empêche la relecture de K7, et 20 cm n'influe pas. Alors, même si l'interface marche avec les 3 PIA's et les 2 ACIA's l'horloge Ø2 n'est pas assez bien découpée pour permettre la relecture des programmes.

Pour palier ce défaut d'ORIC, deux solutions possibles : réaliser un câblage de l'interface où la liaison avec Ø2 est inférieure à 10 cm.

- Personnellement nous avons réaliser la deuxième solution. (Notre câblage faisait dans les 30 cm!) Elle consiste à utiliser un inverseur laissé libre pour redécouper Ø2 (fig. 5.4).

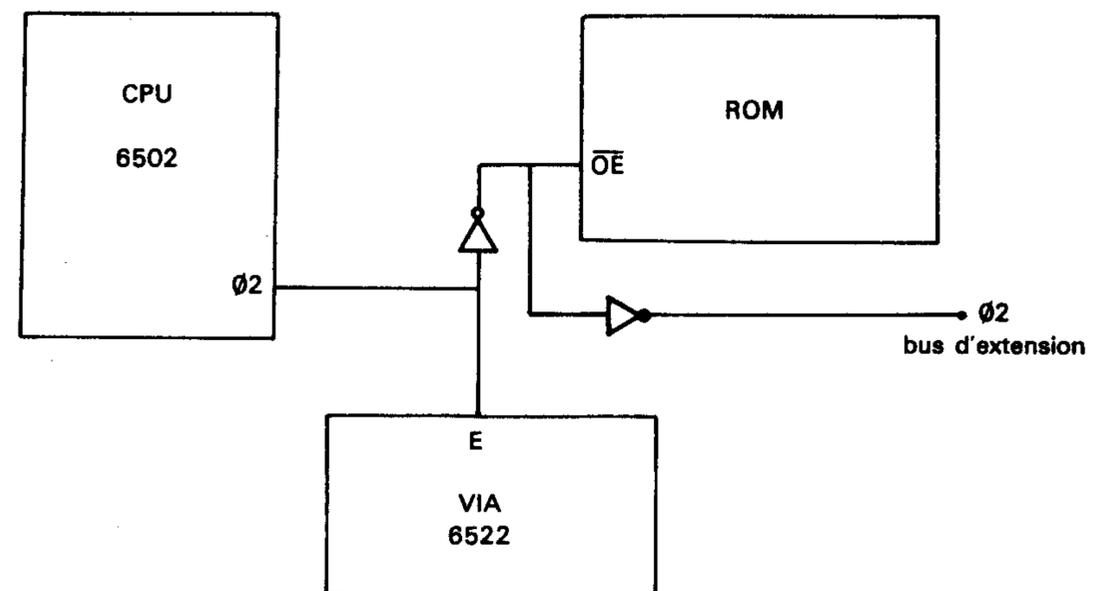


Fig. 5.4. — Ø2 après évolution.

Après cette modification, nous n'avons plus eu aucun problème de chargement K7 en présence de notre interface. Ce qui est dommage, c'est l'oubli par les concepteurs de l'ORIC de ce problème, alors que la solution tient en trois fils.

Sachez que cette solution entraîne l'ouverture de l'ORIC et la perte de la garantie. A vous de voir (et si vous avez déjà fait la transformation ATMOS alors n'hésitez plus!).

### Réalisation pratique

Une fois la boîte ouverte et le circuit déposé, repérez-vous à l'aide de la photo 5.5.

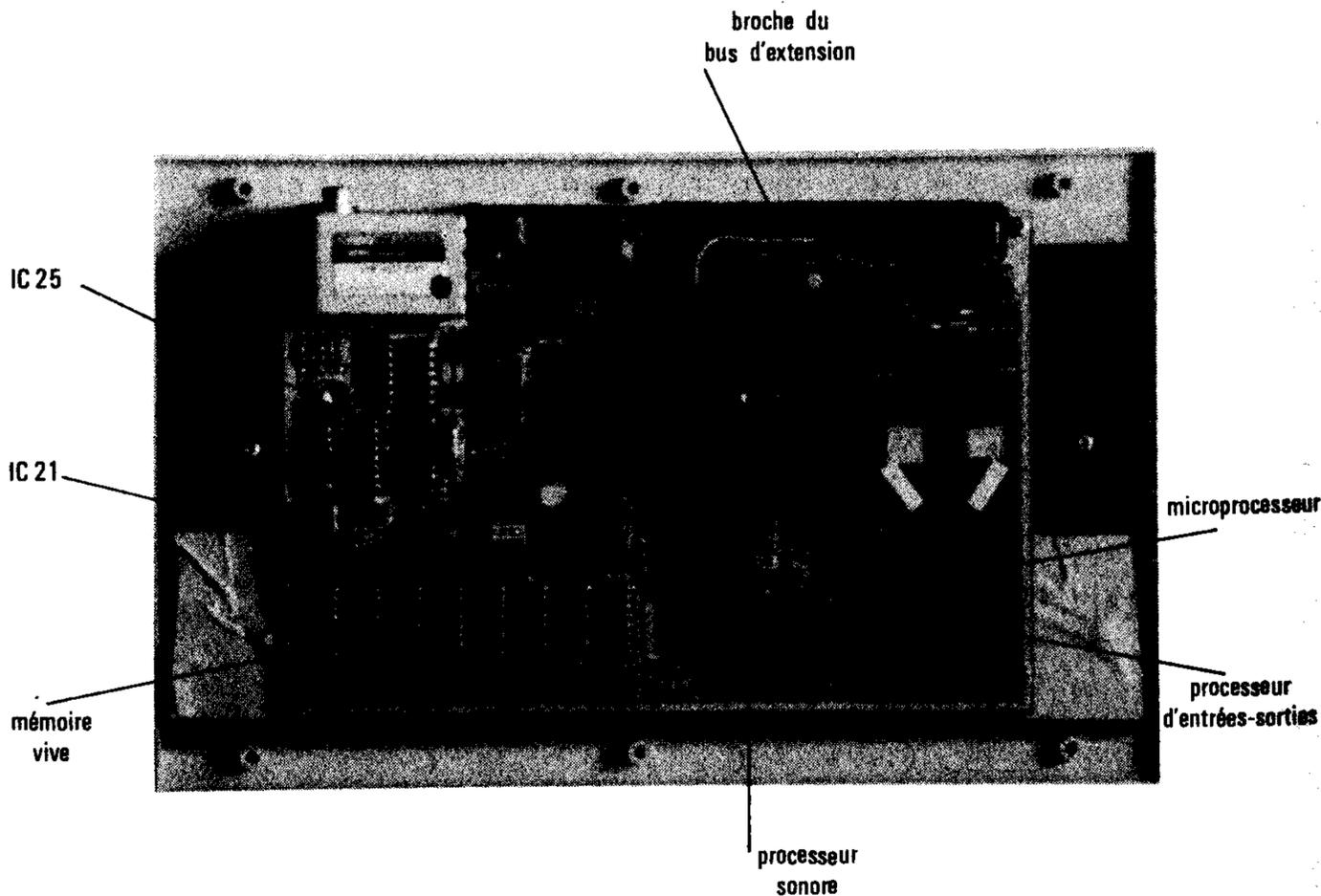


Fig. 5.5. — Photo explicative Ø2.

Le circuit IC21 contient entre les broches 5 et 6 l'inverseur en aval de la ROM (revoir fig. 5.3).

Le circuit IC25 possède un inverseur non utilisé entre les broches 8 et 9.

Le principe de la modification est de relier la broche 6 de IC21 à la broche 9 de IC25 et de remplacer le signal disponible sur la broche 3 du bus d'extension par celui de la broche 8 de IC25 (fig. 5.6).

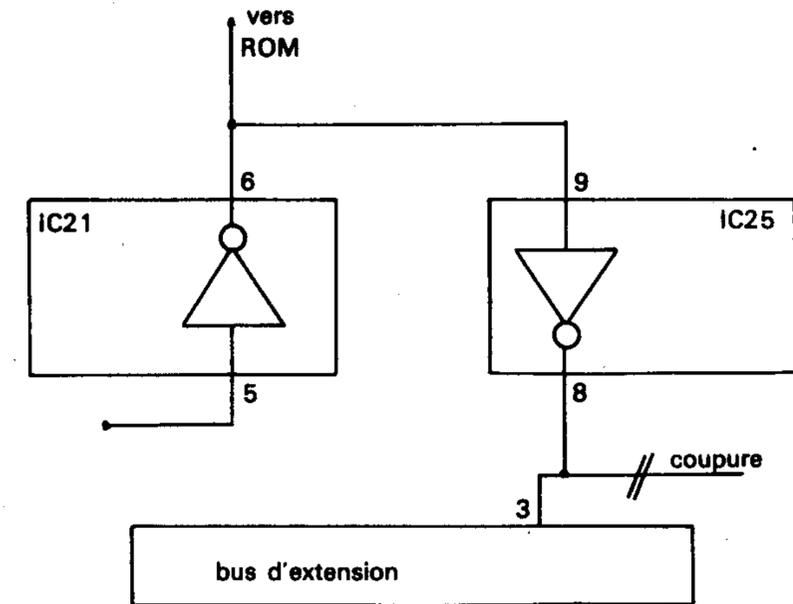


Fig. 5.6. — Schéma de modification.

La photo 5.7 vous donne la réalisation pratique. Pour plus de clarté, l'explication est reprise à l'aide du dessin 5.8.

La broche 9 de IC25 est reliée à la masse, pour des raisons de passages de pistes. Il faut donc couper de chaque côté de la broche 9 pour l'isoler (double trait), puis relier à nouveau la piste (triple trait). Puis on relie la broche 9 — isolée de la piste désormais — à la broche 6 de IC21 (trait en croix). Enfin un fil part de la broche 8 pour aller sur la patte 3 du connecteur 34 points (trait en triangle). N'oubliez pas de couper la broche du connecteur pour la déconnecter de l'ancienne piste.

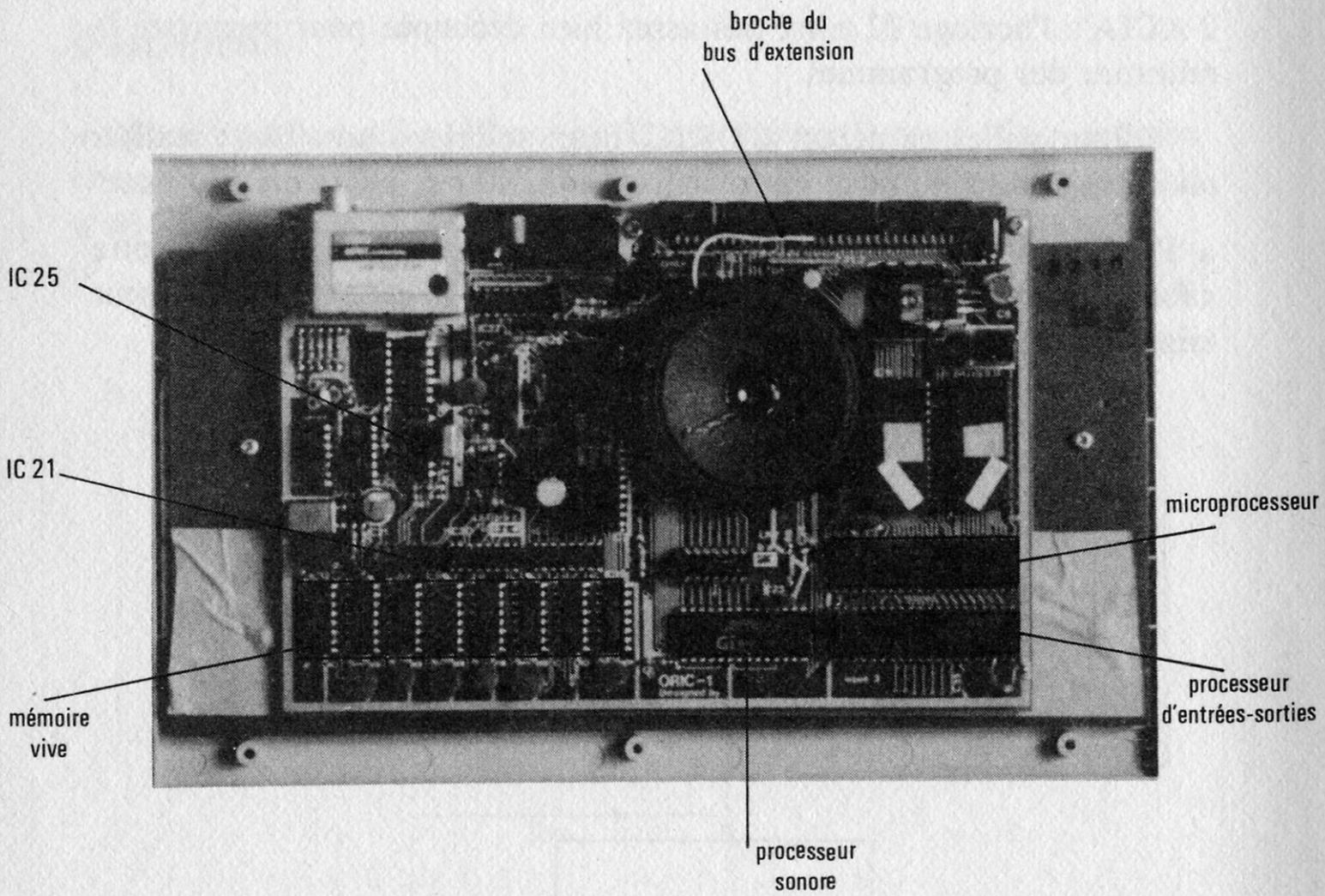


Fig. 5.5. — Photo explicative Ø2.

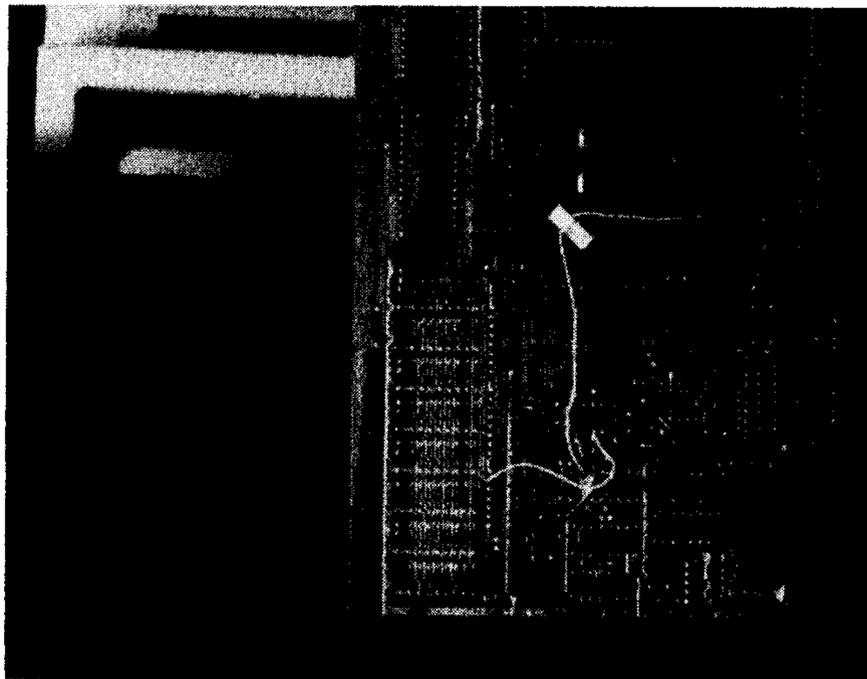


Fig. 5.7. — Photo de la modification.

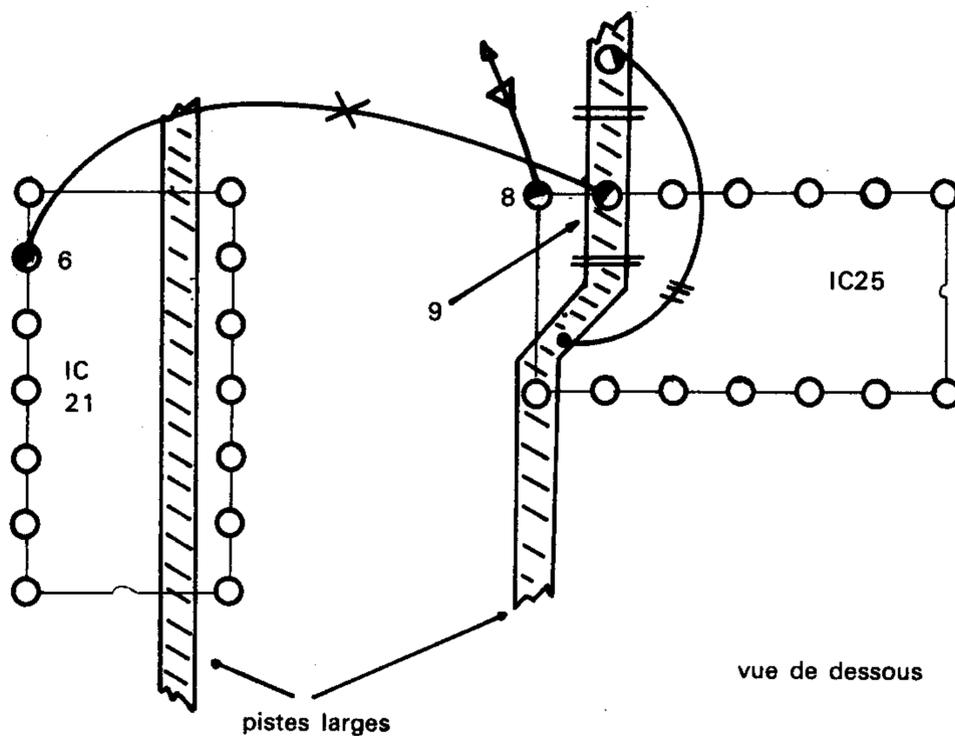


Fig. 5.8. — Dessin du câblage.

Pour couper la piste qui relie la broche 9 de IC25, on peut utiliser avec profit une perceuse de modéliste plutôt qu'un cutter.

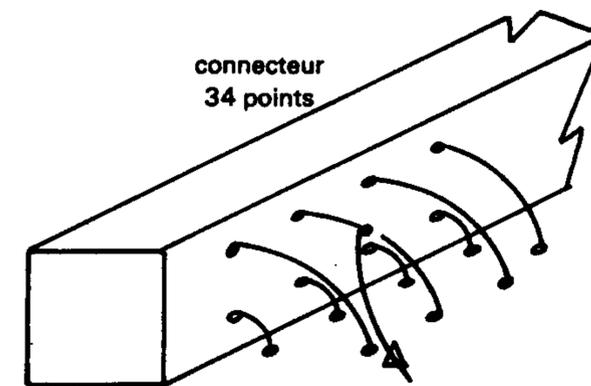


Fig. 5.9. — Dessin du connecteur d'extension.

### 5.3. PANNES EN VRAC

Dans ce paragraphe, nous allons donner quelques dépannages possibles, dans le cas bien sûr où votre garantie n'est plus utilisable. Dans le cas contraire utilisez les bons offices de votre revendeur.

#### L'encodeur de clavier

Si vous constatez que certaines touches sont inopérantes, il y a de fortes présomptions envers l'encodeur du clavier : c'est le circuit intégré CD 4051 qui se trouve sur la carte clavier qu'il faut changer.

#### Encore le clavier

Cette fois-ci il se bloque au bout de quelques minutes. En plus le son est perturbé.

La modification consiste premièrement à souder une résistance de  $22\text{ K}\Omega$  entre les broches 6 et 18 du circuit IC4 (c'est le processeur sonore 8912).

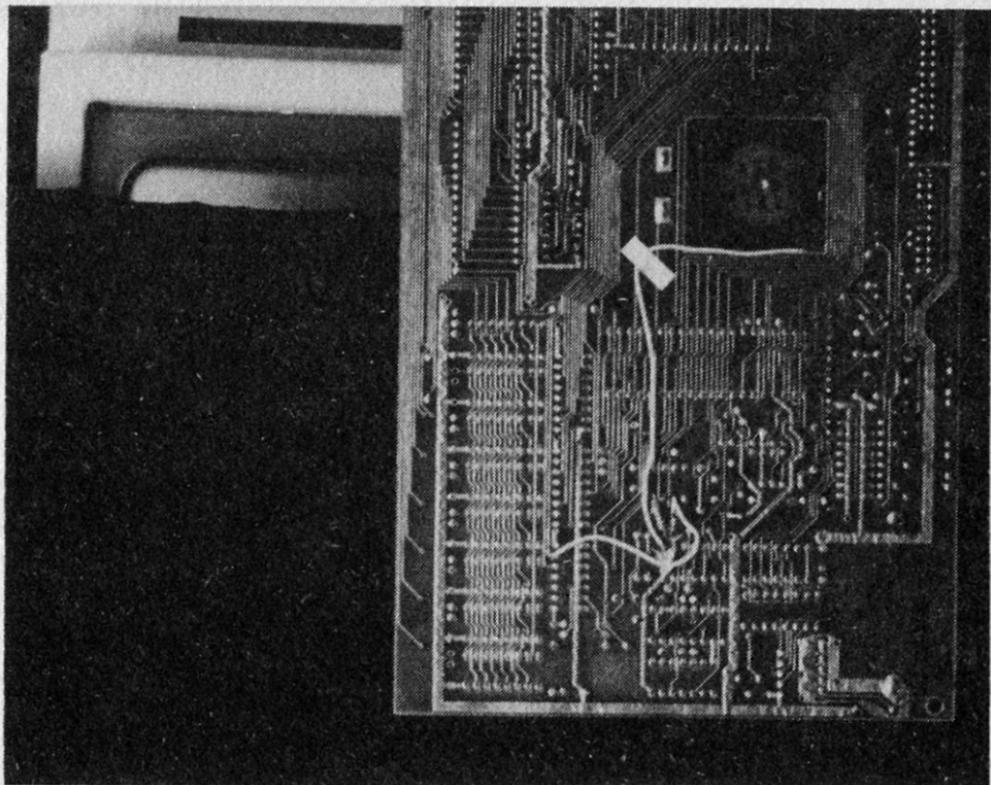


Fig. 5.7. — *Photo de la modification.*

Deuxièmement il reste nécessaire d'intercaler un condensateur de 1 nF entre la broche 19 de la VIA 6522 (IC6) et la broche 18 du processeur de son 8912 (IC4) (fig. 5.10).

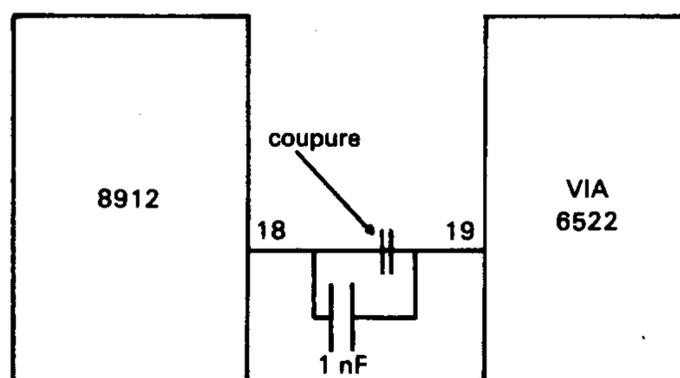


Fig. 5.10. — Dessin de la 2<sup>e</sup> modification.

La dernière évolution est utile pour ceux qui ont des problèmes de chargement (si c'est en utilisation avec l'interface voir aussi le paragraphe sur le rephasage de Ø2).

Le principe est de rajouter un filtre sur l'entrée K7 de la VIA. Ce filtre est réalisé à l'aide d'une résistance de 1 KΩ et un condensateur de 2,2 nF (fig. 5.11).

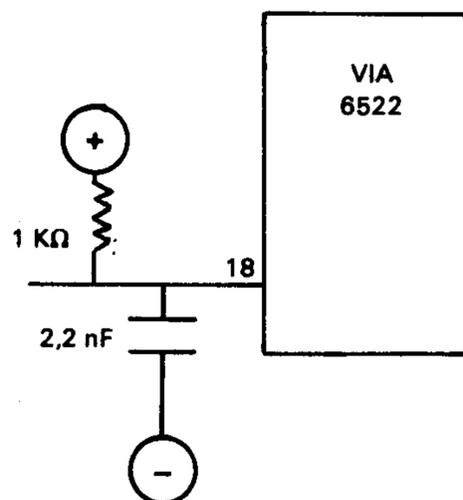


Fig. 5.11. — Le filtre pour le chargement K7.

### Précautions

Ces pannes ont été, à la naissance des ORIC, assez courantes. Beaucoup d'appareils sont vendus avec ces modifications déjà réalisées. Par contre celles-ci ne sont pas des solutions miracles: il existe des pannes beaucoup plus sérieuses et non réparables par des amateurs. Heureusement, elles sont fort rares et l'ORIC est un appareil d'une excellente fiabilité.

# 6

## Conseils de réalisation pratique

### 6.1. COMMENT UTILISER LES SCHÉMAS

Pour illustrer notre propos nous allons prendre la figure 4.5 comme exemple.

- Les circuits intégrés sont représentés par des rectangles et le nom est donné en grandes lettres.

Les indications à l'intérieur du rectangle (CLR, GND, ...) sont les appellations mnémoniques des entrées ou sorties du circuit. Elles ne sont pas utiles pour le câblage, mais donnent une meilleure (voire bonne!) compréhension du schéma.

Le numéro des broches figure à l'extérieur du rectangle. En figure 6.1 le principe de numérotation d'un circuit intégré 16 broches.

Faites très attention, l'erreur qui consiste à se tromper de broche est courante: surtout quand on câble le circuit, on se trouve alors à l'envers par rapport au dessin 6.1.

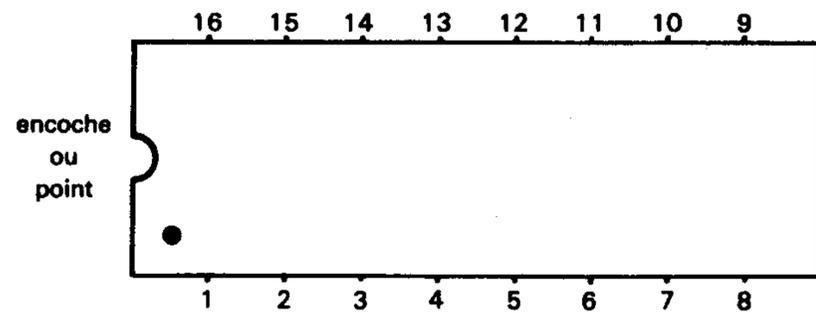


Fig. 6.1. — Principe de numération des IC.

### Les alimentations

● En continu, elles sont représentées dans des cercles. Le  $\ominus$  représente la masse et est connecté à la broche 34 du bus d'extensions (même si ce n'est pas explicite sur les schémas) ou les broches paires du connecteur Centronics.

Le cercle  $\oplus$  ou bien  $\oplus 5$  représente le 5 V positif. Attention, il n'est pas relié au 5 V de l'ORIC (broche 33) sous peine de graves ennuis. En fait la broche 33 n'est jamais utilisée.

Lorsque des cercles ont un nombre en plus du signe il représente la tension à connecter  $\ominus 15$  : -15 V.

● En alternatif, le sigle est  $\sim$  suivi de la valeur en volt.

### Les bus de l'ORIC

Les broches des bus — d'extensions ou Centronics — sont données par des hexagones ayant leur numéro à l'intérieur. La signification de la broche est écrite à côté et correspond aux appellations de la figure 1.4.

### Les composants discrets

Les résistances et condensateurs sont dessinés de façon classique : leurs valeurs avec l'unité sont données à côté.

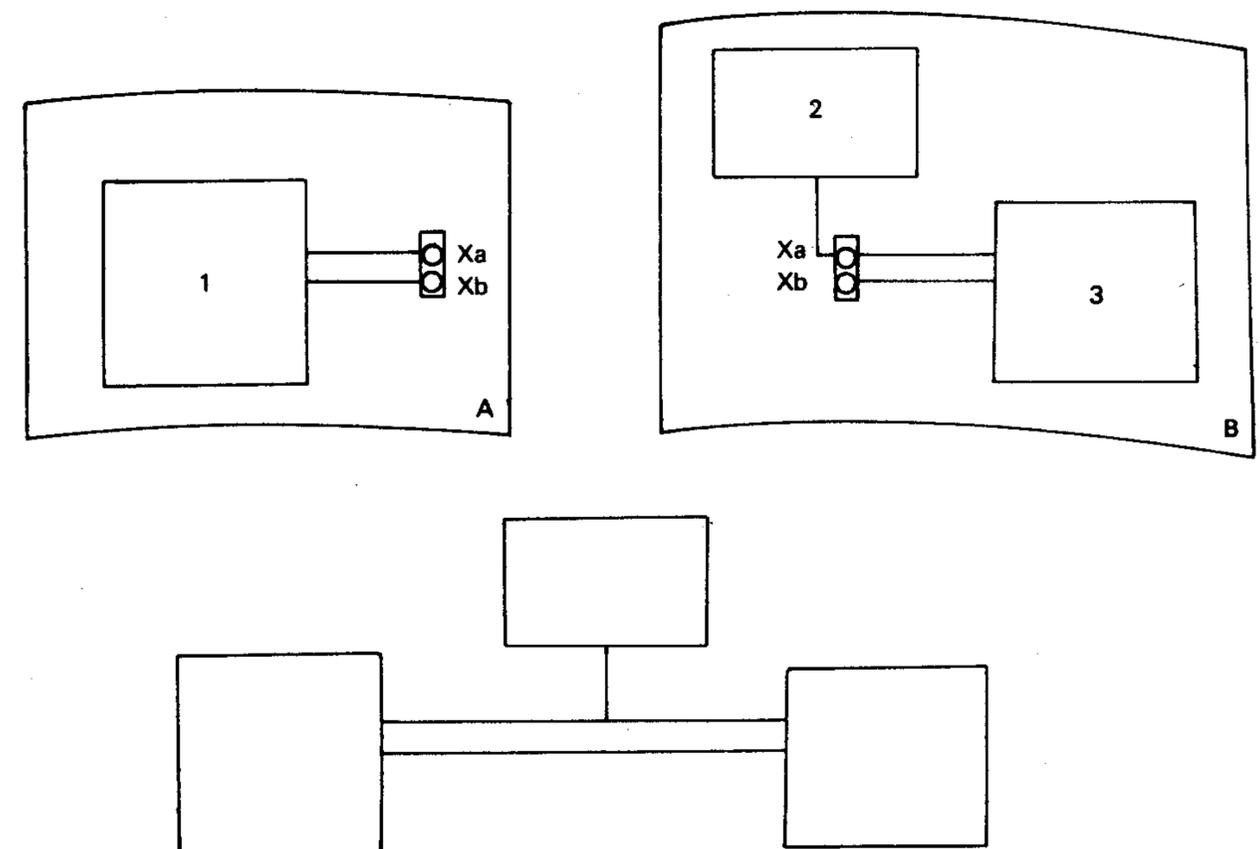
### Les équipotentielles

Ce terme désigne tout simplement les fils du câblage. Ils sont représentés en trait fort. Par contre il arrive que les schémas ne soient pas donnés en un seul dessin.

Les raccords inter-schémas sont symbolisés de deux façons :

- 1)  suivi de l'explication en clair du raccordement.
- 2)  un rectangle avec un ou plusieurs cercles à l'intérieur.

En figure 6.2 voici l'interprétation :



Les planches A et B sont équivalentes au schéma ci-dessus.

Fig. 6.2. — Symbolisme.

L'avantage, pour nous, de cette présentation est la modularité : en effet notre interface est constitué de blocs qui peuvent intervenir dans plusieurs montages différents — ainsi il n'est pas utile de reprendre le schéma en entier.

Maintenant que vous avez compris les schémas, digéré le texte, restez avec nous pour quelques conseils sur l'outillage.

## 6.2. UN PEU D'OUTILLAGE N'A JAMAIS MANQUÉ

Les 2 principales méthodes de câblage de ces maquettes sont le wrapping et le soudage de fils.

Le wrapping est une solution élégante, pratique qui demande un peu de matériel spécifique mais dont l'inconvénient majeur est le prix des supports.

Pour des questions de disponibilités de matériel et de solidité dans le temps nous avons utilisé un fer à souder et du fil.

La photo 6.3 donne une idée d'implantation possible. Il manque sur cette carte une ACIA.

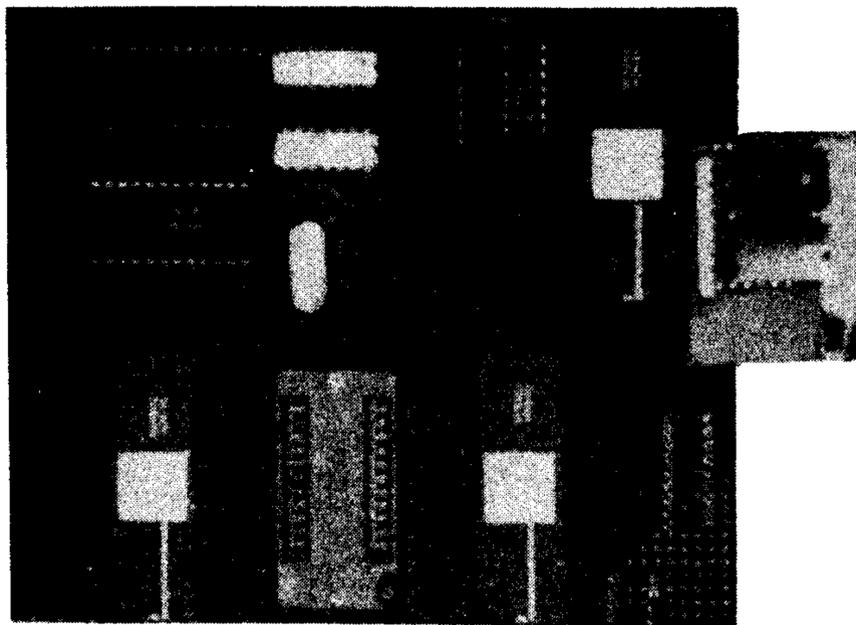


Fig. 6.3. — Photo de l'implantation.

Nous avons pris une plaque à trous munis de pastilles étamées sur un des côtés. Après nous avons soudé des supports sur cette plaque et relié avec le fil 8/10 (gaine comprise). Dans certains cas (câblage des bus de données) il est très pratique d'utiliser du fil en nappe.

Pour ce faire les seuls outils que nous ayons utilisés sont : un fer à souder 25 W, une petite pince coupante, une petite pince plate, et une pince à dénuder (et de la soudure, un peu!).

A propos du dénudage des fils, certains se servent de leurs dents (à déconseiller) ou bien il existe du fil à gaine rétractable (le fil se dénude lorsque l'on approche le fer à souder).

Nous n'allons pas donner ici un cours de soudure. Le mieux pour les débutants complets est de demander une démonstration (5 min maximum) à un ami.

Et puis les trois ultimes accessoires indispensables :

- beaucoup de soins,
- encore plus de patience,
- une crème pharmaceutique contre les brûlures du fer à souder.

## 6.3. RÉALISATION DES ALIMENTATIONS

Au cours des différents montages, il nous a fallu, outre l'alimentation 5 V destinée aux circuits classiques, -15 V pour le convertisseur analogique/digital et -12 V +12 V pour les émetteurs de ligne compatible RS 232C.

### L'alimentation de base 5 volts

L'ORIC possède une alimentation 5 V, mais il ne nous est pas possible de l'utiliser. Bien que présente sur le bus d'extension, elle n'est pas assez puissante pour notre interface. Le régulateur utilisé à l'intérieur de l'ORIC chauffe beaucoup en mode normal.

N'étant pas et ne voulant pas devenir électronicien, le schéma que nous avons utilisé pour ces alimentations reste très classique (fig. 6.4).

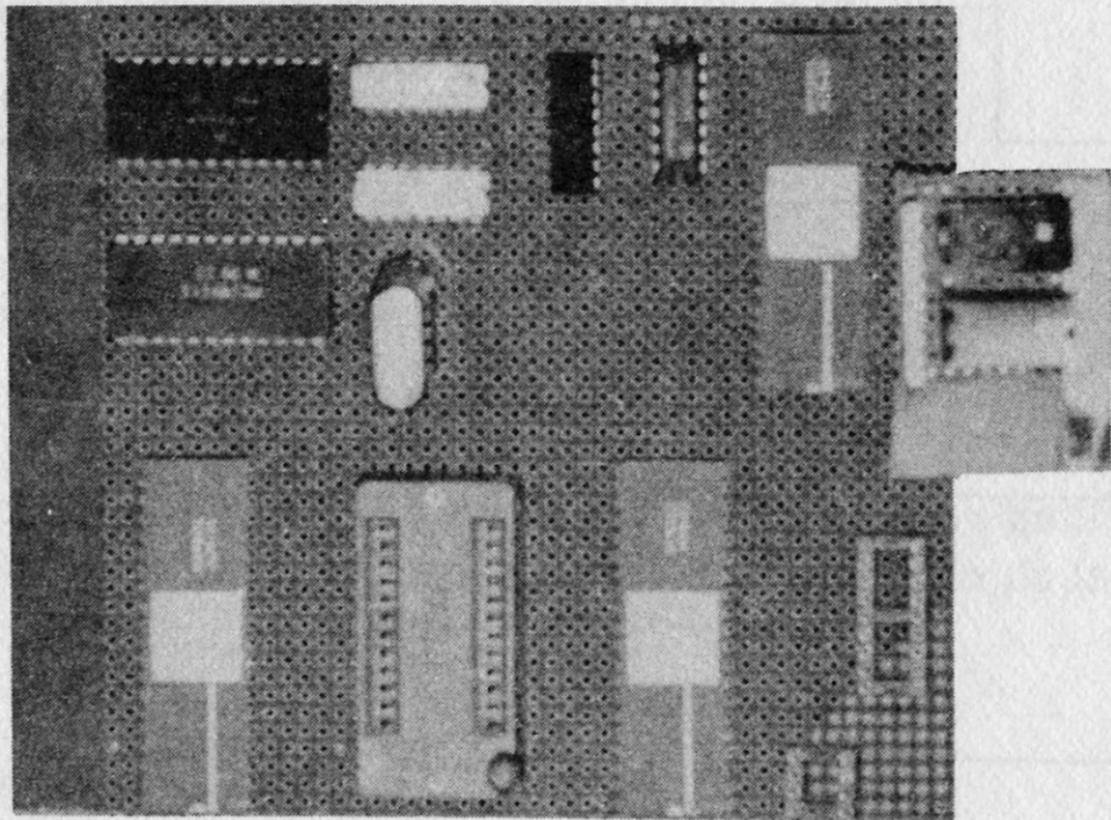


Fig. 6.3. — *Photo de l'implantation.*

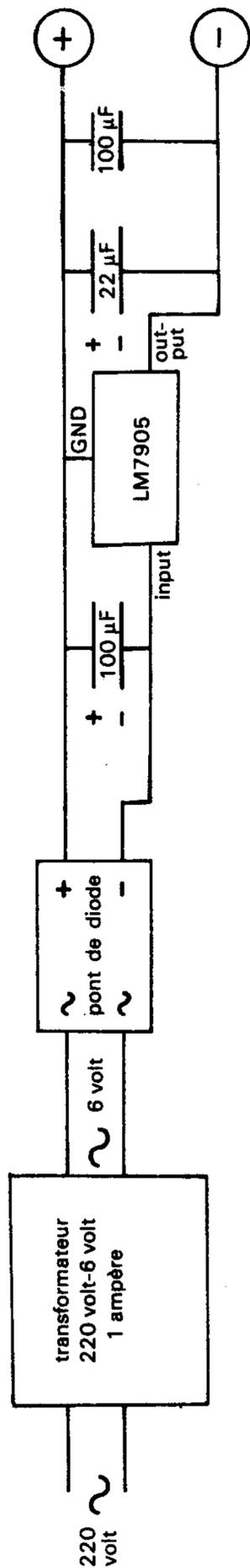


Fig. 6.4. — Schéma alimentation 5 volts.

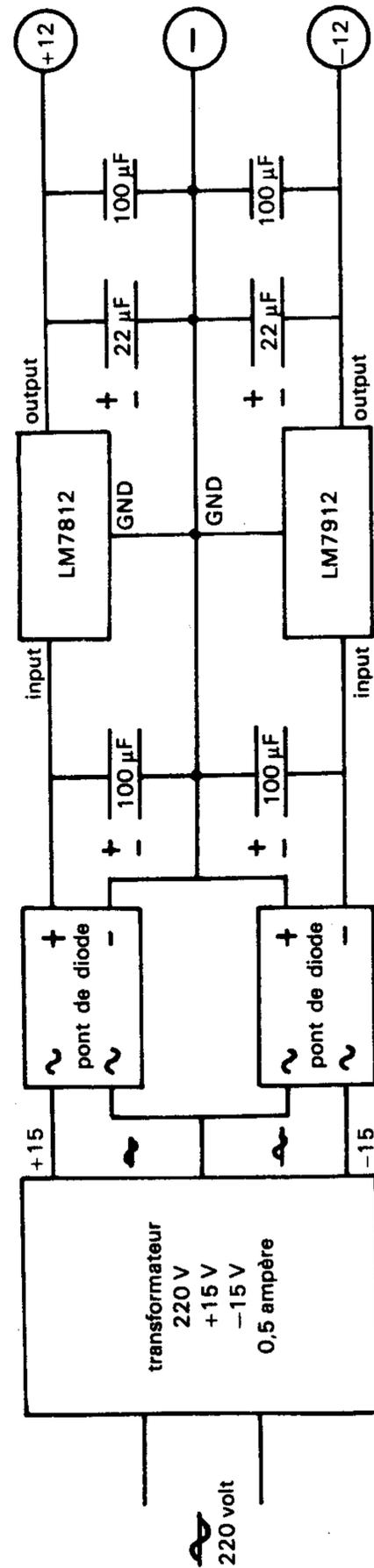


Fig. 6.5. — Schéma alimentation -12 et +12 volts.

On trouve dans le commerce des circuits qui ne demandent qu'un minimum de composants externes pour régler à tension fixe.

Le schéma est directement extrapolé des données du constructeur. Les composants se présentent sous l'aspect d'un transistor (boîtier avec deux ou trois broches). Il ne serait pas raisonnable de donner ici la position des broches à souder vu la diversité dans le choix des composants et des boîtiers. En effet, il est très possible d'utiliser des circuits équivalents. Il est néanmoins nécessaire de choisir un régulateur passant plus de 1 ampère et de le monter sur un radiateur, surtout si vous avez réalisé tous les montages proposés.

● La réalisation du -12 V +12 V ne pose pas plus de problème (fig. 6.5). Le schéma reste similaire aux inconvénients près dû à la symétrie de l'alimentation. Le transformateur est plus complexe (donc plus onéreux). Les remarques faites aux paragraphes précédents restent valables si ce n'est l'intensité utile qui peut être beaucoup plus faible.

Pour finir le chapitre des alimentations, deux mots à propos du -15 V du convertisseur analogique digital.

Nous utilisons à nouveau le transformateur 2 × 15 V et un régulateur intégré de -15 V suivant le schéma 6.6. Cette figure montre les éléments à ajouter en plus du schéma 6.4.

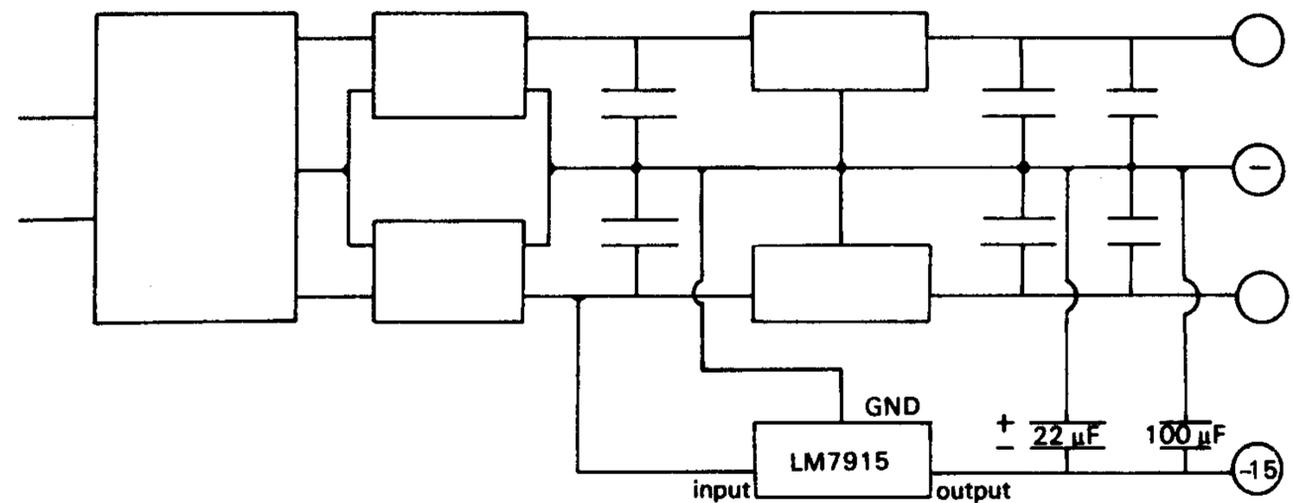


Fig. 6.6. — Schéma alimentation -15 volts.

- La tension de 21 V pour programmer les mémoires mortes 2764 a été obtenue à partir d'un régulateur variable LM 117.

Le montage est réalisé suivant la figure 6.7. L'ajustage de la tension choisie se fait à l'aide d'un voltmètre.

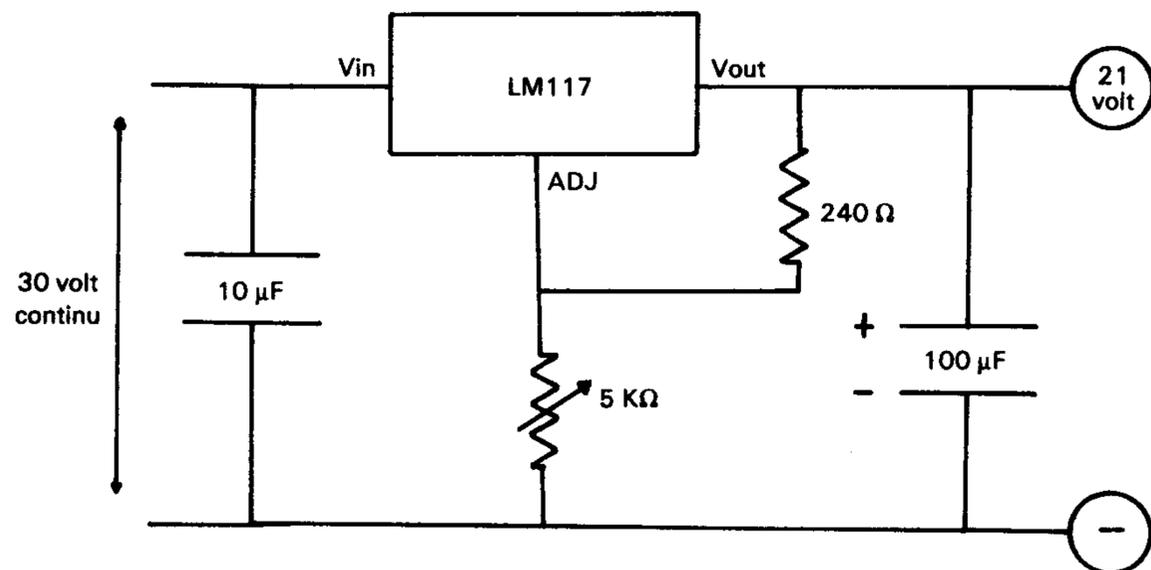


Fig. 6.7. — Schéma régulation 21 volts.

Ces schémas d'alimentation se retrouvent dans la plupart des livres sur l'électronique, nous ne nous y attarderons pas plus longtemps.

#### 6.4. CONSEILS PRATIQUES

Nous conseillons vivement de mettre tous les composants actifs sur supports: cela évitera de les abîmer au montage et permettra de contrôler le câblage avant la mise sous tension.

- N'essayez pas trop de changer les schémas — dans un premier temps — faites les fonctionner, comprenez-les et ensuite viendra le temps des évolutions.

- Pour le choix des composants, achetez une revue d'électronique (genre ELEKTOR, Radio-Plans, etc...). Les publicités insérées dans ces périodiques donnent des adresses de revendeurs et souvent les prix des composants. Il n'est pas possible de donner — même à titre indicatif — des prix dans ce livre: en effet dans moins de 6 mois ces prix seront périmés.

- Pensez vite à mettre vos œuvres en boîte: elles seront mieux protégées; un fil abîmé peut causer de graves dommages.

Avant de mettre vos montages sous tensions vérifiez votre câblage — en sortant les composants actifs — avec un ohmètre. Assurez-vous une dernière fois que vous n'inversez pas le plus et la masse (certains circuits sont très susceptibles et n'apprécient guère ce genre d'humour!).

- Le premier temps est un temps d'initiation: les programmes et les schémas ont tous été testés. Une bonne recopie doit donc assurer une mise au point immédiate. Si par malchance il reste des problèmes, alors vient le temps de la réflexion. Vous avez, avec ce livre, tous les éléments pour "débuger" vos montages. Alors bon courage!

## Annexe 1

### Adresses des périphériques et de leurs registres

#### PIA N° 1

— Manette de jeu	registre donnée A	#3F0
— Afficheurs	registre contrôle A	#3F1
	registre donnée B	#3F2
	registre contrôle B	#3F3

#### PIA N° 2

— Convertisseur analogique	registre donnée A	#3F4
— Programmeur	registre contrôle A	#3F5
	registre donnée B	#3F6
	registre contrôle B	#3F7

#### PIA N° 3

— Lecteur-programmeur	registre donnée A	#3F8
	registre contrôle A	#3F9
	registre donnée B	#3FA
	registre contrôle B	#3FB

#### ACIA N° 1

— Modem	registre contrôle	#3FC
	registre donnée	#3FD

#### ACIA N° 2

— RS 232C	registre contrôle	#3FE
	registre donnée	#3FF

## Annexe 2

### La fonction TIME de l'ORIC et de l'ATMOS

L'ORIC possède un décompteur interne dont le pas est le centième de seconde, ce qui est d'une très bonne précision dans la plupart des applications. Ce décompteur est accessible par deux octets à l'adresse #276.

Il vous suffit de faire DEEK(#276) pour acquérir la valeur du décompteur.

Pourquoi décompteur: tout simplement parce que la valeur se décrémente tous les centième de seconde. Pour initialiser ce compteur rien de plus simple:

```
DOKE #276,0
```

Et voici un petit programme pour compter les secondes:

```
10 DOKE #276,0
20 A=INT((#FFFF-DEEK(#276))/100)
30 FOR I=0 TO 10
40 NEXT I
50 CLS
60 PRINT A: GOTO 20
```

La boucle en ligne 30 et 40 permet une lecture moins désagréable.

Maintenant au lieu de mettre une boucle, tapez 30 WAIT 5. La fonction WAIT réinitialise le décompteur alors attention...

## Bibliographie

*ORIC-1, Manuel de Programmation Basic*, (ASN diffusion).

*Manuel de ORIC-ATMOS*, (ASN diffusion).

*Visa pour ORIC*, (éditions SORACOM).

*Microprocesseur et mémoires*, (Thomson-EFCIS).

*The TTL Data Book*, (Texas Instruments).

*The Power Semiconductor Data Book*, (Texas Instruments).

*Linear Data Book*, (National Semiconductor).

Imprimerie de la Manutention à Mayenne

Dépôt légal : novembre 1984

N° d'Éditeur : 4183

Avec «DES EXTENSIONS A CONSTRUIRE POUR VOTRE ORIC/ATMOS» vous aurez tous les éléments pour réaliser vous même des modems, des liaisons RS 232 C, des lecteurs et programmeurs de PROM, des manettes de jeux, des dispositifs de pilotage à distance d'appareils domestiques (chauffage, protection de la maison). Tous les montages, décrits **pas à pas**, ont été intégralement **réalisés** et **testés** par les auteurs, tous deux ingénieurs.