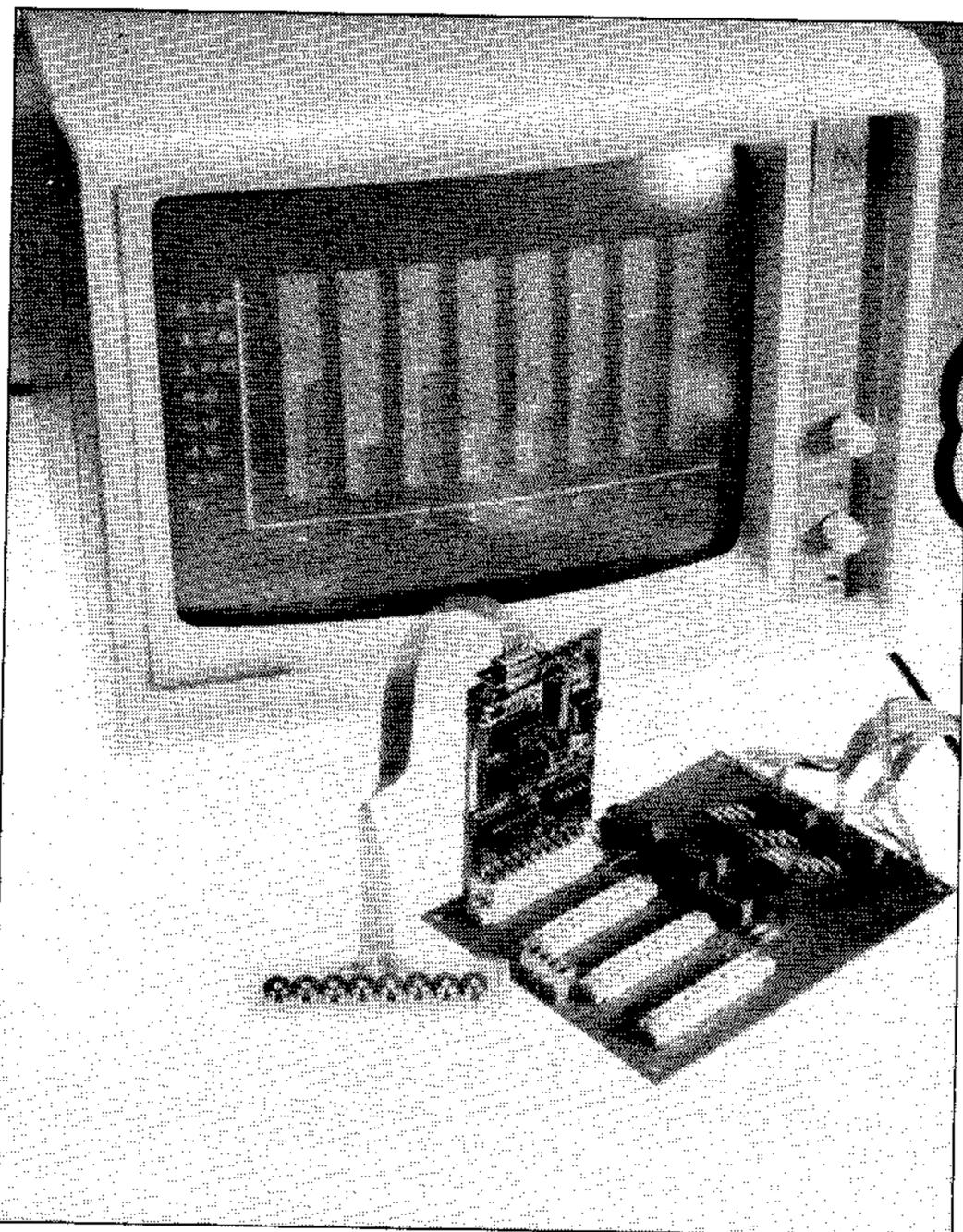


Le principe même de la norme MSX est de garantir la compatibilité des programmes et des modules périphériques pour les ordinateurs qui s'y conforment. Nous entreprenons ici la description de quelques extensions qui permettront aux heureux propriétaires de ce genre de machines d'en élargir le champ d'action.

1 extensions MSX



L'avènement tant attendu de MSX a été un succès au moins en ceci qu'il constitue une louable tentative de développer la compatibilité entre ordinateurs différents, tant pour ce qui concerne le logiciel que pour le matériel. De sorte qu'Elektor est aujourd'hui en mesure de proposer à ces lecteurs MSX-cistes deux modules d'extension: un bus d'entrée/sortie (I/O) MSX et un convertisseur A/N (analogique/numérique) assorti d'un port bidirectionnel à 8 bits (aussi appelé "interface cybernétique").

Il existe donc un certain nombre de micro-ordinateurs domestiques équipés du même *Microsoft Extended BASIC* (MSX), qui présentent également quelques autres caractéristiques communes mais non moins intéressantes:

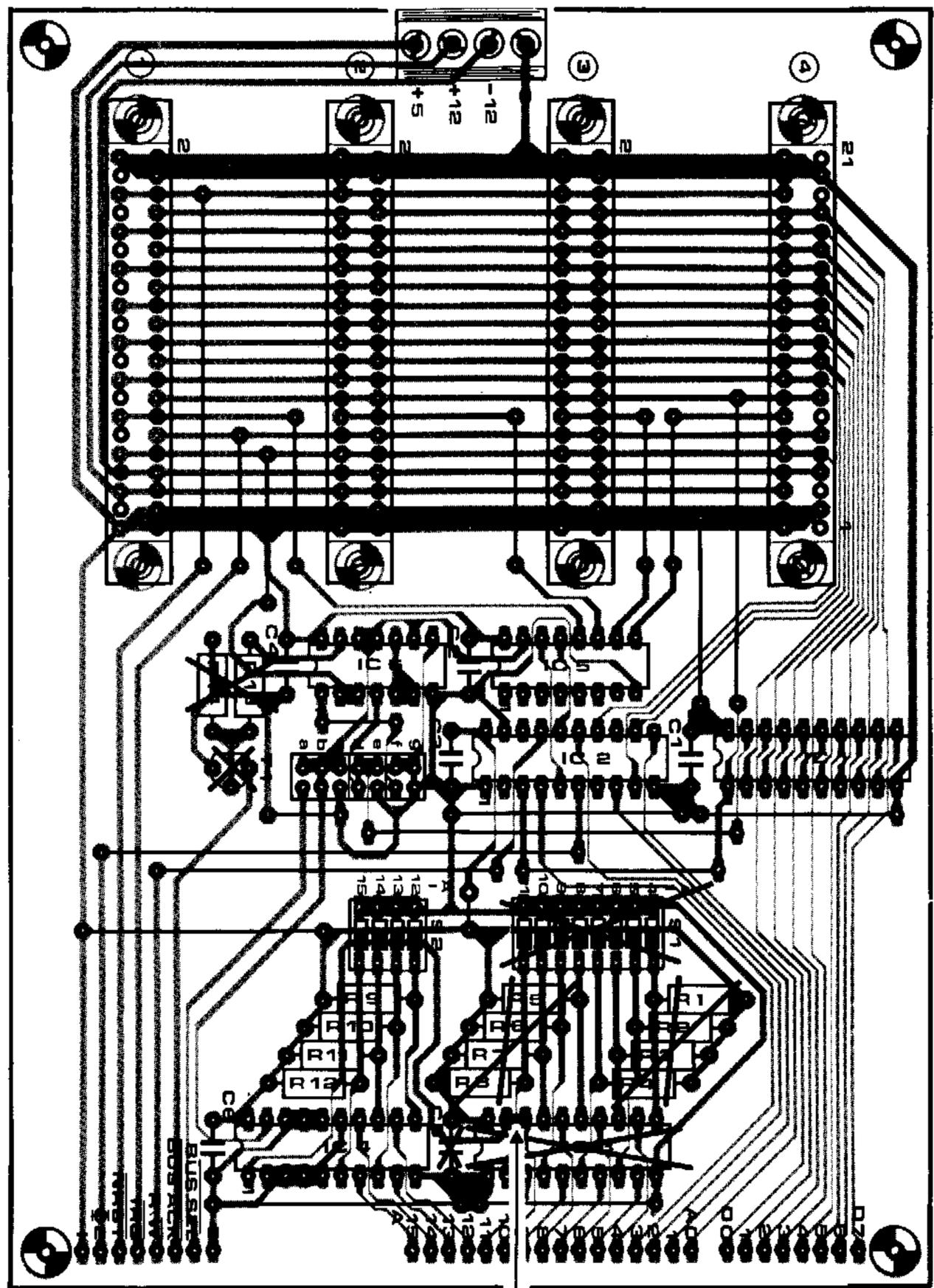
- interface pour imprimante Centronics disponible d'origine
- connecteur pour cartouches ROM ou extensions du circuit
- MS-DOS
- instructions graphiques du BASIC
- ...

Pour l'heure, nous allons nous consacrer plus précisément au bus I/O universel décrit dans Elektor n° 83, page 5-20. On se souvient que ce bus avait été conçu pour le C64 de Commodore, qui utilise un processeur 6502; les machines MSX par contre font appel au Z80: il faut donc modifier certains signaux. Il en va de même pour les circuits à enficher sur ce bus; le convertisseur A/N et N/A publié dans Elektor n° 83, page 5-68 et le port d'entrée/sortie à 8 bits (ou "interface cybernétique") publié dans Elektor n° 90, page 12-64 subiront de légères modifications.

Modifications du bus

Les signaux importants sur le bus du Z80 sont \overline{IORQ} , \overline{RD} et \overline{WR} qui deviennent actifs (niveau logique bas) respectivement lors d'un accès aux circuits d'entrée/sortie (*input/output request*) lors d'une opération de lecture (*read*) et enfin lors d'une opération d'écriture (*write*). On substitue le signal \overline{IORQ} inversé au signal $\phi 2$ du 6502. C'est pourquoi sur le circuit de la **figure 1** on trouve IC2 (74LS244) remplacé par un circuit intégré équivalent, mais inverseur (LS240). Du coup, tous les autres signaux qui transistent par IC2 sont inversés aussi, ce qui tombe bien puisque l'on pourra utiliser le signal \overline{RD} du Z80 inversé pour commander le bus de données. L'inversion de A0...A3 entraîne une

2



liaison entre les broches 18 et 19 d'IC3

85129-2

Figure 2. Le dessin de circuit imprimé du bus modifié pour MSX. N'oubliez pas la liaison entre les broches 18 et 19 d'IC3.

faut couper la broche 6 d'IC4 à ras du boîtier, et établir une liaison entre la broche 7 d'IC2 et la broche 6 du connecteur (voir **figure 4**).

Voici maintenant les modifications pour le port d'entrée/sortie à 8 bits: couper la broche 8 d'IC1 et relier la broche 8 du support (ou la pastille) à la broche 6 du connecteur sur le bus ($\overline{IRQ} = \overline{WR}$) et rajouter la liaison câblée représentée sur la **figure 6** en gras.

Le chaînon manquant

Pour la liaison entre le bus

d'entrées/sorties et le micro-ordinateur MSX, il y a deux possibilités: le connecteur spécial pour cartouches ou, pour certaines marques comme Spectravideo ou Goldstar, le connecteur d'extension I/O à 50 broches. Pour accéder au connecteur pour cartouches, il nous faut un connecteur spécial comme on n'en trouve nulle part... sauf dans les cartouches elles-mêmes. Nous avons pensé que c'était là une ruse efficace, bon marché et finalement sans risque. Nous vous proposons donc de démonter (sans rien casser) une cartouche de fabrication industrielle (en choisir une qui soit de toute façon parmi les moins chères)

et de brancher le bus universel en parallèle sur le circuit de la cartouche, de telle sorte que celui-ci puisse continuer de fonctionner normalement. A l'inverse, il faudra empêcher le circuit de la cartouche de fonctionner lorsque l'on utilisera l'une ou l'autre carte d'extension branchée au bus d'extension. On trouvera la fonction des différents signaux disponibles sur le connecteur pour cartouches sur le **tableau 1**. Mais avant d'en arriver là, il faut arriver à ouvrir une cartouche. Il nous faut avouer qu'avec certaines d'entre elles, ce n'est pas évident du tout...

Les pistes paires se trouvent à l'avant

Tableau 1.

SLOT ←		SIGNAL DESCRIPTION	
PIN NO.	NAME	I/O	DESCRIPTION
1	CS1	O	ROM 4000 ~ 7FFF select signal (128k)
2	CS2	O	ROM 8000 ~ BFFF select signal (128k)
3	CS12	O	ROM 4000 ~ BFFF select signal (256k)
4	SLTSL	O	Slot selected signal. ← Fixed select signal for each slot.
5		—	Reserved for future use only
6	RFSH	O	Refresh signal
7	WAIT	I	Wait signal to CPU (wired-OR)
8	INT	I	Interrupt request signal
9	M1	O	Fetch cycle signal of CPU
10	BUSDIR	I	This signal controls the direction of external data bus buffer when the cartridge is selected. It is low level when the data is sent by the cartridge.
11	IORQ	O	I/O request signal
12	MERQ	O	Memory request signal
13	WR	O	Write signal
14	RD	O	Read signal
15	RESET	O	System reset signal
16		—	Reserved for future use only
17	A9	O	} Address bus
18	A15	O	
19	A11	O	
20	A10	O	
21	A7	O	
22	A6	O	
23	A12	O	
24	A8	O	
25	A14	O	
26	A13	O	
27	A1	O	
28	A0	O	
29	A3	O	
30	A2	O	
31	A5	O	
32	A4	O	
33	D1	I/O	
34	D0	I/O	
35	D3	I/O	
36	D2	I/O	
37	D5	I/O	
38	D4	I/O	
39	D7	I/O	
40	D6	I/O	
41	GND	—	Ground
42	CLOCK	O	CPU clock 39.545 MHz
43	GND	—	Ground
44,46	SW1,SW2	—	Insert/remove protection, if fitted
45,47	+5 V	—	+5 V power supply
48	+12 V	—	+12 V power supply
49	SOUND IN	I	Sound input (−5 dbm)
50	−12 V	—	−12 V power supply

Input and output is measured with respect to MSX computer

Programme 1.

```

10 FOR A=0 TO 255
20 IF INP(A)<>255 GOTO 30 ELSE 40
30 PRINTA,INP(A)
40 NEXT
    
```

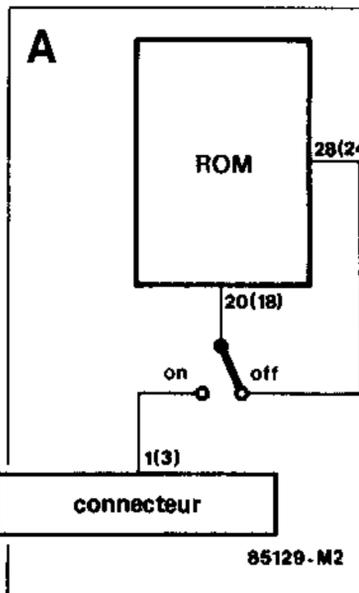
du circuit imprimé, c'est-à-dire du même côté que l'étiquette, et les pistes impaires de l'autre côté. La piste n° 2 se trouve donc devant vous, à l'extrême droite. La piste n° 1 se trouve également à l'extrême droite, mais de l'autre côté du circuit imprimé.

Du fait que le circuit imprimé de la cartouche est le plus souvent à double face, il se trouve que les trous en sont la plupart du temps métallisés, ce qui nous arrange bien. Si par un hasard peu probable vous deviez tomber sur des trous non métallisés, il faudra prélever les signaux correspondants directement sur les broches de la ROM.

Des 50 points de connexion disponibles, nous n'en utiliserons que la moitié. Le câble plat est le plus indiqué pour prélever les signaux et les acheminer vers l'extérieur. Pour les tensions d'alimentation, nous prendrons deux fils supplémentaires, soit 29 en tout.

Les (EP)ROM présentes dans les cartouches sont presque toujours compatibles broche à broche avec les EPROM de la série 27XX. Il n'y aura donc, en règle générale, que très peu de pistes à interrompre et de liaisons nouvelles à établir. Pour une ROM à 28 broches, il faut couper la piste reliée à la broche 20 de ce circuit intégré, tandis que pour une ROM à 24 broches, c'est la piste reliée à la broche 18. Voir également le **tableau 1**. Ces pistes sont reliées par ailleurs aux broches 1 ou 3 du connecteur. Pour invalider la ROM, on en relie la broche 18 (ou 20, selon le cas) au contact central d'un inverseur dont l'un des deux autres contacts est relié au +5 V, et l'autre à la broche 1 ou 3 du connecteur (c'est-à-dire à la piste interrompue auparavant).

Tableau 1.
Brochage d'une
cartouche.



Compléments d'information

Tous les fabricants de cartouches ne donnent pas d'informations détaillées.

Programme 2.

```

10 FOR A=0 TO 7
20 OUT112,A
30 PRINTA,INP(112)
40 NEXT
    
```

Programme 1.
Ce programme vérifie indirectement les bus d'adresses et de données.

Programme 2.
Ce programme vérifie le bon fonctionnement du convertisseur A/N.

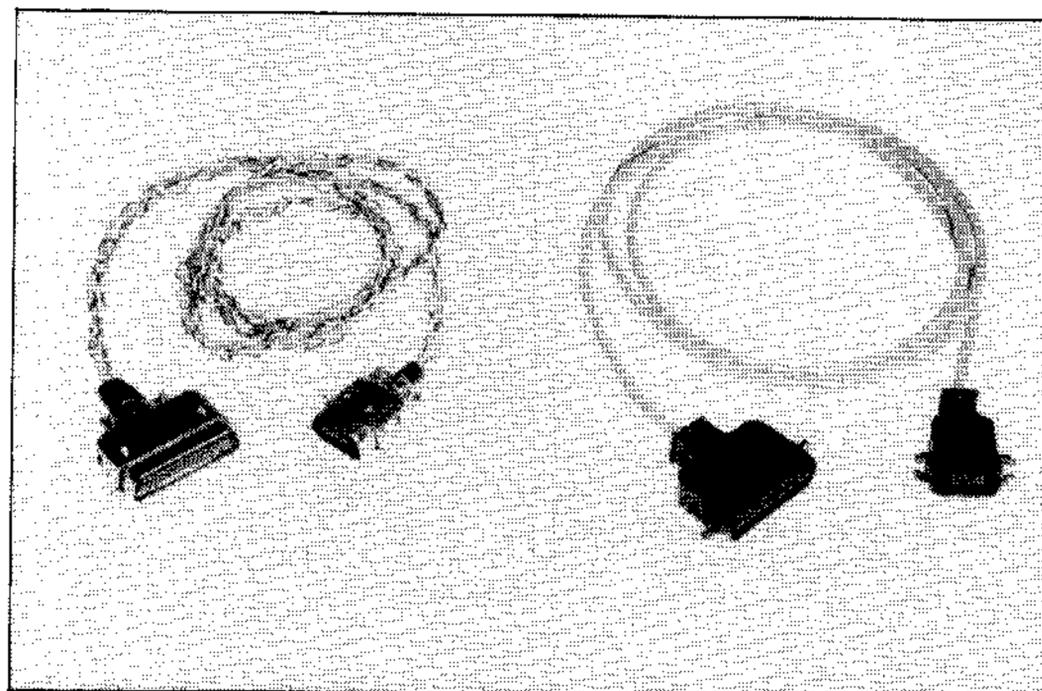
Programme 3.

```

10 OPEN"grp:"FOROUTPUTAS#1
20 ACOL=9:BCOL=8:CCOL=14
30 R1=8:R2=8:U1=4:U2=4
40 SCREEN2,2
50 COLOR15,4,4
60 CLS
70 LINE(32,0)-(32,176)
80 LINE(31,0)-(31,176)
90 LINESTEP(+0,+0)-STEP(+210,+0)
100 LINESTEP(+0,+1)-STEP(-210,+0)
110 FORY=155TO15STEP-20
120 PRESET(0,Y-2)
130 PRINT#1,MID$(STR$(155-Y),1);";";
140 LINE(31,Y)-(36,Y+1)
150 NEXTY
160 GOSUB180
170 Q=0:GOTO160
180 FORA=1TO8
190 OUT0,A-1
200 H(A)=INP(0)/1.6
210 A$(A)=CHR$(A+48)
220 NEXT
230 FORN=56TO232STEP24
240 Y=160:X=N
250 Q=Q+1:H=H(Q)
260 A$=A$(Q)
270 IFNOT(H<=M(Q)-10RH>M(Q)+1)THEN320
280 PRESET(X-5,Y+24)
290 PRINT#1,RIGHT$(A$,2);
300 IFP=1ANDM(Q)=>4THENGOSUB350ELSEGOSUB380
310 M(Q)=H
320 NEXTN
330 P=1
340 RETURN
350 LINE(X-1,Y-U1-H)-(X-R1,Y-U1),ACOL,BF
360 LINE(X,Y-U2-H)-(X-1+R2,Y-U1),BCOL,BF
370 GOTO400
380 LINE(X-1,Y-U1-H)-(X-R1,Y),ACOL,BF
390 LINE(X,Y-U2-H)-(X-1+R2,Y),BCOL,BF
400 LINE(X-R1,0)-(X-1+R2,Y-U1-H),3,BF
410 PRESET(X,Y-H)
420 DRAW" c=ccol;a2m+=r1;,+u1;m-=r2;,+u2;"
430 PRESET(X-1,Y-H-U1-U2)
440 DRAW" c=ccol;a0m+=r1;,+u1;m-=r2;,+u2;"
450 PAINT(X,Y-U1-H),CCOL
460 PRESET(X,Y)
470 DRAW" c=b;l+=r1;u+=u1;m+=r1;,+u1;"
480 PAINTSTEP(-4,-1),0
490 PRESET(X,Y),0
500 DRAW" r+=r2;u+=u2;m-=r2;,+u2;"
510 PAINT STEP(+4,-1),0
520 RETURN

```

Programme 3.
Ce programme assure la visualisation sous forme de colonnes tridimensionnelles des valeurs converties par le circuit analogique/numérique.



lées sur leur produit; c'est pourquoi nous publions ici le tableau 1, dans lequel les termes Entrée et Sortie, c'est-à-dire *Input* et *Output*, sont remplacés respectivement par les lettres "I" et "O". Les broches 44 et 46 du connecteur sont censées activer un circuit de protection de l'ordinateur; malheureusement ce circuit n'est pas toujours présent. De sorte que nous vous recommandons de toujours prendre soin de mettre l'ordinateur hors tension au moment d'insérer ou d'extraire une cartouche du connecteur. Ceci même lorsque vous avez la certitude que la cartouche est dotée d'un circuit de protection!

Comme on a pu le déduire de ce qui précède, les cartouches ne contiennent pas grand-chose: une mémoire morte programmée, un petit circuit imprimé, un connecteur... et c'est tout. Certaines cartouches contiennent cependant un peu plus qu'une simple ROM: on y trouve aussi une interface, comme l'interface pour disquettes de Sony, ou encore le synthétiseur FM de Yamaha. Mais ce sont là des exceptions. Mais ce qui nous intéresse, c'est le petit circuit imprimé, et rien d'autre.

Voici à présent comment nous vérifierons le bon fonctionnement de nos circuits. Le microprocesseur Z80 distingue 256 adresses d'entrée/sortie; chaque connecteur en mobilise quatre. Lorsque tous les interrupteurs de S2 sont fermés, le processeur doit trouver le bus d'entrée/sortie aux adresses 0 à 15. Pour s'assurer que l'ordinateur trouve bien le connecteur d'entrée/sortie que nous souhaitons qu'il adresse, il suffit de renvoyer une information précisément sur le connecteur concerné, et de vérifier que cette information a été reçue par l'ordinateur. Pour ce faire, nous polarisons certaines lignes de donnée de façon à obtenir une configuration reconnaissable. Au cours de cette manipulation, il faut s'assurer que l'ordinateur ne fait jamais que des opérations de lecture sur les lignes concernées! Les liaisons que nous allons établir concernent toujours deux bits de donnée et la ligne de sélection du connecteur (**slot select = SS**). Pour le premier, on reliera les broches 18 et 17 à la broche 7; pour le deuxième, ce seront les broches 15 et 16 et 7; sur le troisième connecteur, la broche 7 est reliée aux broches 13 et 14, tandis que sur le quatrième, elle est reliée aux broches 11 et 12. Veillez à bien respecter le nouvel ordre des connecteurs, et non pas l'ancien... Tant qu'un connecteur n'est pas adressé, toutes ses lignes de donnée sont au niveau logique haut ("1"). Deux d'entre elles passent au niveau

logique bas lorsque la ligne \overline{SS} (broche 7) à laquelle elles sont reliées passe elle-même au niveau logique bas, c'est-à-dire lorsque le connecteur en question est adressé. De sorte que l'ordinateur lira les valeurs suivantes décimales sur les différents connecteurs:

- connecteur 1 = 63
- connecteur 2 = 207
- connecteur 3 = 243
- connecteur 4 = 252

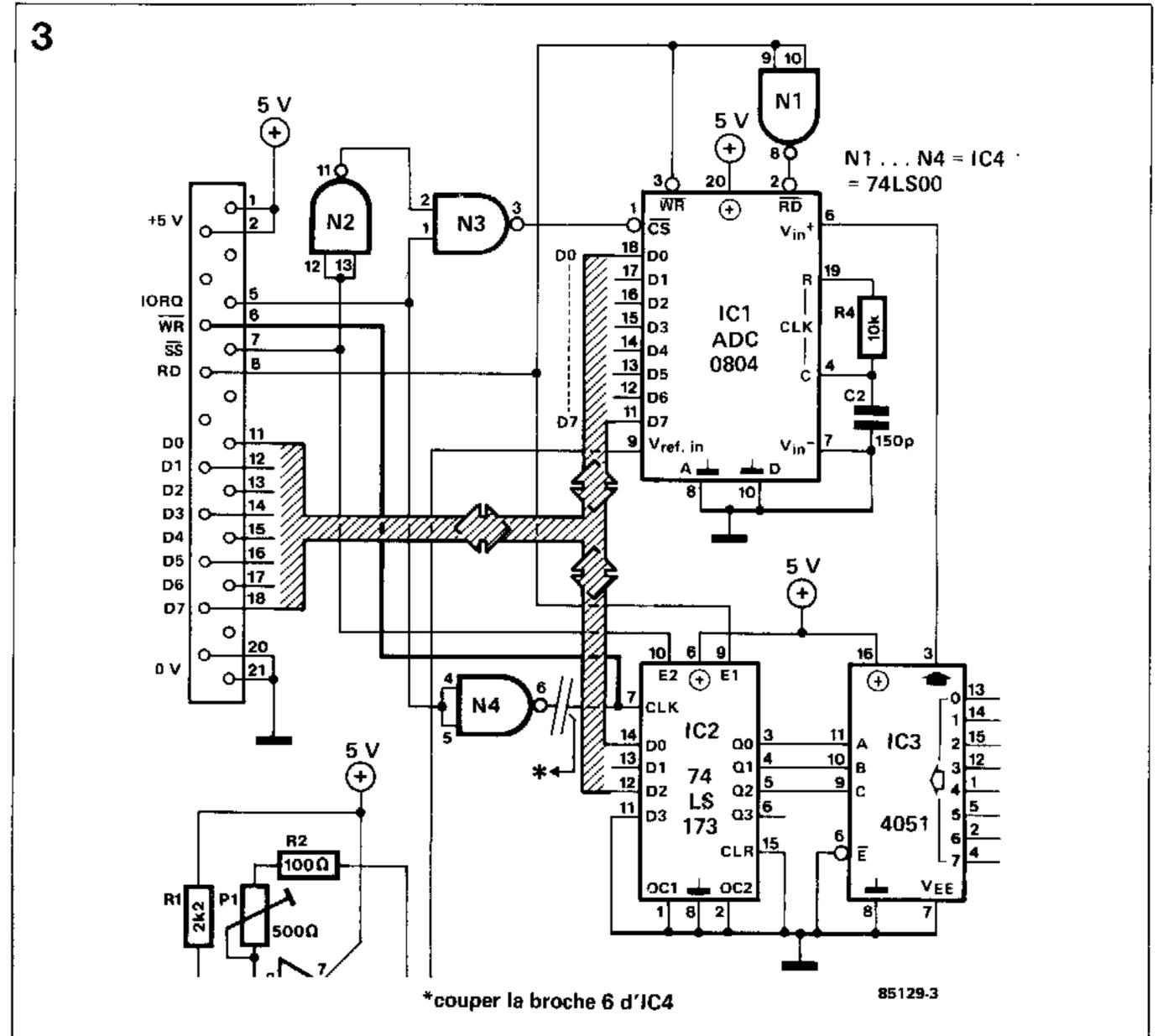
Le premier programme donné en exemple ici passe en revue toutes les adresses d'entrée/sortie de 0 à 255. De 0 à 127, le processeur ne peut trouver que le bus I/O. A partir de 128, il peut trouver aussi des adresses utilisées par MSX comme par exemple 152 (VDP) et 162 (PSC). Le programme considère qu'à aucune de ces adresses il ne trouvera une donnée dont tous les bits sont au niveau haut. Si ce programme n'arrivait pas à lire les données "câblées" sur les connecteurs, c'est précisément le câblage du circuit qu'il conviendra de révéifier. Ou peut-être avez-vous oublié les ponts de câblage, b, d et g? Et IC2, est-ce bien un 74LS240?

Si tout s'est bien passé jusqu'ici, on va pouvoir mettre en place le convertisseur analogique/numérique sur le connecteur 1, dont on aura supprimé au préalable la liaison entre broches 18, 17 et 7! Pour en tester le bon fonctionnement, nous vous proposons un deuxième programme. Le signal d'entrée du convertisseur A/N est simulé à l'aide d'un potentiomètre pour chacune des huit entrées qui reçoit donc une tension comprise entre 0 et 5 V.

L'adresse du connecteur utilisée dans ce deuxième programme est différente; ce qui implique qu'il faut ouvrir certains des interrupteurs de S2, à savoir ceux qui portent les numéros 12, 13 et 14 sur le circuit imprimé. Ce même programme permet d'ailleurs de tester aussi le bus d'entrée/sortie à 8 canaux. On relie pour cela toutes les entrées aux sorties correspondantes. Les valeurs lues par le processeur sont alors comprises entre 255 et 248. Une dernière remarque concernant le bus à 8 canaux: remplacez donc, dans la ligne 20, la variable A par NOT A AND 255...

Pratique

Les signaux reçus par le convertisseur A/N peuvent être visualisés de manière spectaculaire sur l'écran. Le troisième programme proposé ici donne un bel exemple de mise en forme: chaque valeur convertie est rendue visible par une colonne ver-



ticale tridimensionnelle dont la hauteur est proportionnelle à la valeur. Si cela vous chante, vous pouvez facilement personnaliser ce programme: la couleur est traitée par la ligne 20, le format des colonnes par la ligne 30. N'oubliez pas cependant que les possibilités graphiques des ordinateurs MSX sont relativement limitées...

Outre les paramètres de couleur et de format déjà mentionnés, nous signalons encore que les lignes 70 et 80 commandent le dessin de l'axe vertical, tandis que les lignes 90 et 100 s'occupent de l'axe horizontal. Le

réticule est l'oeuvre des lignes 110 à 150. C'est autour de la ligne 170 qu'est construite la boucle dans laquelle le programme tourne indéfiniment. Le reste du programme consiste en deux routines. Les lignes 180 à 220 chargent les données en provenance des canaux A/N. Les lignes 230 à 340 assurent l'affichage de ces données sur l'écran sous forme de colonnes verticales. La routine à partir de la ligne 350 opère la conversion proprement dite de la donnée en une colonne de hauteur proportionnelle. La ligne 400 a pour fonction d'effacer ce qui resterait à cet

Figure 3. Le convertisseur A/N publié en mai 1985 modifié pour le standard MSX. Remarquez entre autres la suppression de la broche 6 d'IC4.

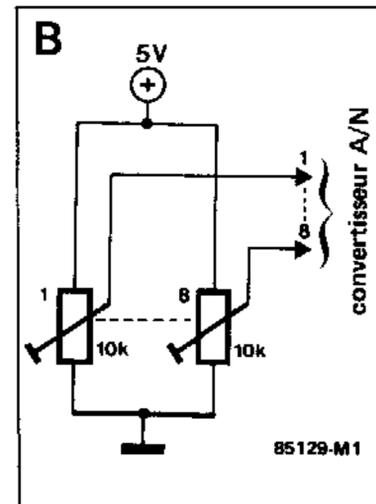
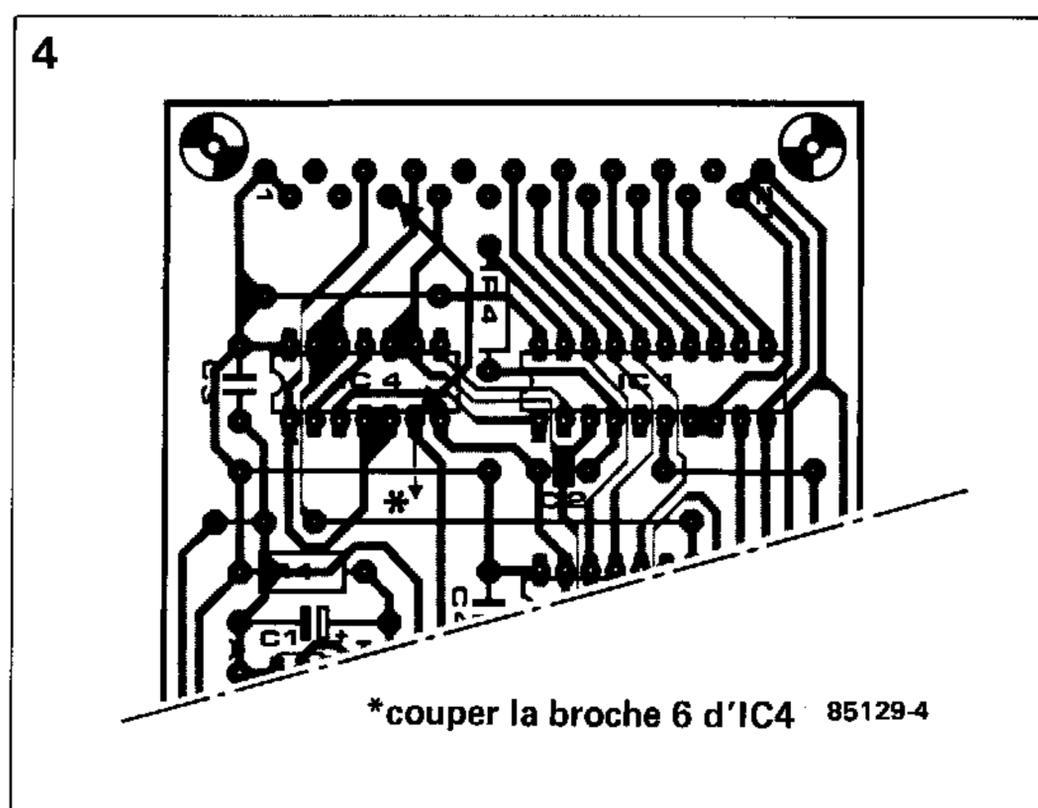


Figure 4. Le circuit imprimé du convertisseur A/N comporte un nouveau pont de câblage.

