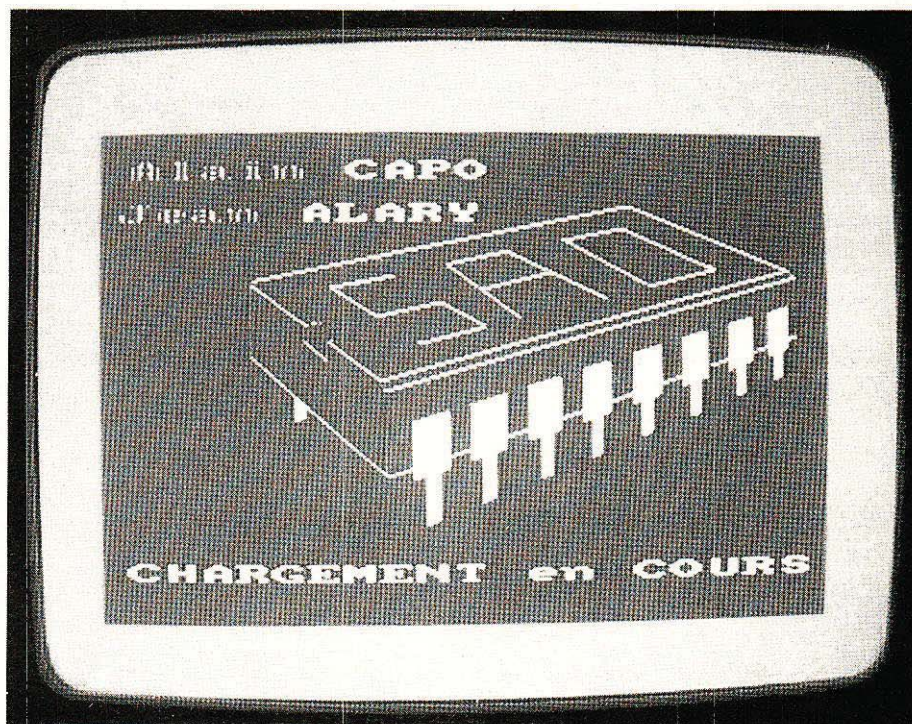


SAO, logiciel de dessin de schémas

SAO fonctionne sur Amstrad CPC 464 avec drive, CPC 664 et COC 6128. Une routine de recopie d'écran intégrée, permettra l'impression sur Amstrad DMP-2000 ou toute autre imprimante compatible (codes Epson). Il permet bien entendu de dessiner des schémas électroniques, mais aussi des diagrammes, des tableaux, et la banque interne de 40 figures sera modifiable à volonté. Plus de 200 dessins de composants sont ici offerts, ainsi que l'écriture alphanumérique comportant les signes et lettres utiles en électronique, plus...



Six mois de travail, d'essais, de modifications, pour vous offrir un logiciel surprenant et simple à adapter ! La rédaction de Radio-Plans et les auteurs sont fiers de vous présenter la version définitive de cet outil de travail qui deviendra vite indispensable.

Sa description sera séparée en deux parties :

1° « la tête », comportant toutes les fonctions et routines utiles. Dans cette première partie, nous vous indiquerons aussi comment modifier votre AMSTRAD pour que son port imprimante soit au standard 8 bits.

2° La banque de dessins (ou POLICES), ainsi que la façon d'en créer vous même, et enfin le

mode d'emploi complet (en français) pour exploiter au plus vite toutes les possibilités offertes.

Voyons donc ce mois-ci la première partie.

Tous les listings présentés ici, doivent être sauvegardés sur une même disquette, au format SYSTÈME (standard sur 464 + CPM/2).

Dans toute série qui se respecte, il est d'usage de présenter les acteurs lors du premier épisode (rassurez-vous, il n'y en a pas autant que dans DALLAS), nous allons donc passer en revue les différents fichiers composant cet univers impitoyable...

SAO, bas

Le premier listing, qui a pour nom « SAO.BAS » figure 1, est court : c'est pour s'échauffer les doigts... En fait, il a aussi d'autres fonctions plus importantes que nous allons détailler.

Tout d'abord une phase interactive où il vous sera demandé :

— Ligne 50 : si vous avez effectué un RESET (CTRL + SHIFT + ESC), ceci est en effet nécessaire pour assurer un bon fonctionnement au logiciel.

— Lignes 60 à 90 : si vous possédez une disquette formatée en DATA pour y sauvegarder vos futurs dessins. Dans le cas contraire, le programme passe en

```

1 REM SAO.BAS
10 REM *** KEY DEF 66,0,0,0,0 ' BREAK interdit
20 MEMORY &SAFD:POKE &SAFE,1:m=0
30 INK 0,1:INK 1,24:BORDER 1:MODE 2:GOSUB 400
40 LOCATE 36,2:PRINT "S.A.O " :LOCATE 36,4:PRINT CHR$(164):"1987"
50 LOCATE 13,9:PRINT "Si vous n'avez pas fait de RESET, il est encore temps !"
60 LOCATE 13,15:PRINT CHR$(24); " Possédez-vous une disquette formatee en DATA :
(O/N) "
70 LOCATE 30,18:PRINT CHR$(24);"(17 Ko par schema)"
80 r$="":WHILE r$<>"0" AND r$<>"N":r$=UPPER$(INKEY$):WEND
90 IF r$="N" THEN 340
100 CLS
110 LOCATE 8,4:PRINT "OPTIONS PAR DEF AUT":LOCATE 8,5:PRINT STRING$(18,"-")
120 LOCATE 6,7:PRINT (" :CHR$(240);") - MONITEUR MONOCHROME"
130 LOCATE 6,9:PRINT (" :CHR$(241);") - 2 LECTEURS DE DISQUETTE (disc systeme e
n "A")"
140 LOCATE 8,14:PRINT "Utilisez les fleches pour changer les options ":LOCATE 35
,16:PRINT "puis ":CHR$(24);" COPY ":CHR$(24);" pour
lancer le programme "
145 LOCATE 5,24:PRINT CHR$(24),"LE DEUXIEME LECTEUR EST-IL ALLUME ?? Sinon RESE
T ":SPACES(11);CHR$(24)
150 GOSUB 260
160 LOAD "VU.SAO" ' ecran presentation
170 BORDER 2:LOAD "ROUTINES.SCH",&A300:BORDER 14
180 REM ***LOAD "COPY-7B.SCH",&A4F4 ' hardcopy 7 bits
190 REM ***LOAD "COPY-8B.SCH",&A4F4 ' hardcopy 8 bits
200 BORDER 11
210 REM *** LOAD "BITS-INT.SCH",&A596:CALL &A596 ' interface 8 bits interne
220 CALL &A4BE:POKE &BC80,&C3:POKE &BC81,&CB:POKE &BC82,&A4 ' merge 484 ***
230 IF m=0 THEN BORDER 9:INK 0,1:INK 1,24
240 IF m=1 THEN BORDER 1:INK 0,13:INK 1,0
250 RUN "SCHEMA.SCH"
260 CALL &BB18
270 IF (INKEY(0)<>-1 AND m=0) THEN m=1:LOCATE 21,7:PRINT "COULEUR " :GOTO 320
280 IF (INKEY(0)<>-1 AND m=1) THEN m=0:LOCATE 21,7:PRINT "MONOCHROME":GOTO 320
290 IF (INKEY(2)<>-1 AND PEEK(&SAFE)=1) THEN POKE &SAFE,0:LOCATE 12,9:PRINT "1 L
ECTEUR DE DISQUETTE " :GOTO 320
300 IF (INKEY(2)<>-1 AND PEEK(&SAFE)=0) THEN POKE &SAFE,1:LOCATE 12,9:PRINT "2 L
ECTEURS DE DISQUETTE":GOTO 320
310 IF INKEY(9)<>-1 THEN 330
320 IF PEEK (&SAFE)=1 THEN LOCATE 5,24:PRINT CHR$(24),"LE DEUXIEME LECTEUR EST-I
L ALLUME ?? Sinon RESET ":SPACE(11);CHR$(24):ELSE
LOCATE 5,24:PRINT SPACE$(71)
325 GOTO 260
330 MODE 2:RETURN
340 MODE 2:LOCATE 19,4:PRINT "POUR PREPARER UNE DISQUETTE AU FORMAT 'DATA'."
350 LOCATE 5,10:PRINT CHR$(24);SPC(26);"INTRODUISEZ LE CP/M":SPC(26);CHR$(24)
360 LOCATE 5,15:PRINT "CPC 464/664 ( CP/M 2.2 ) : Il faudra taper 'FORMAT D'."
370 LOCATE 5,18:PRINT "6128 ( CP/M 3.0 ) : Utiliser 'DISKIT3.COM' et suivre le
s instructions."
380 LOCATE 5,22:PRINT CHR$(24);" APPUYEZ SUR UNE TOUCHE POUR LANCER LE CP/M ":CH
R$(24);SPC(11);"A tout de suite !"
390 CALL &BB18:CPM
400 SYMBOL 253,96,48,120,12,124,204,118,0:' @ a accentue
410 SYMBOL 254,12,24,60,102,126,96,60,0:' 5 e accent aigu
420 SYMBOL 255,48,24,60,102,126,96,60,0:' 2 e accent grave
430 KEY 141,CHR$(191):KEY DEF 15,0,128,128,141:' 0 OHMS
440 KEY 142,CHR$(190):KEY DEF 11,0,136,136,142:' 8 SYGMA
450 KEY 143,CHR$(184):KEY DEF 5,0,131,131,143:' 3 PI
460 KEY 144,CHR$(183):KEY DEF 10,0,135,135,144:' 7 MICRO
470 KEY 145,CHR$(170):KEY DEF 4,0,134,134,145:' 6 3/4
480 KEY 146,CHR$(169):KEY DEF 13,0,129,129,146:' 1 1/2
490 KEY 147,CHR$(168):KEY DEF 20,0,132,132,147:' 1 1/4
500 KEY 148,CHR$(164):KEY DEF 3,0,137,137,148:' 9 COPYRIGHT
510 KEY 149,"FIGURE No " :KEY DEF 7,0,138,138,149:' POINT message
520 KEY 150,CHR$(253):KEY DEF 26,0,64,124,150:' @ a accentue
530 KEY 151,CHR$(254):KEY DEF 12,0,133,133,151:' 5 e accent aigu
540 KEY 152,CHR$(255):KEY DEF 14,0,130,130,152:' 2 e accent grave
550 RETURN

```

Figure no : 1

ligne 340 et lance le CP/M après vous avoir expliqué la procédure à suivre suivant le type de CPC dont vous disposez.

— Lignes 100 à 150 : si vous désirez changer les options fixées par défaut. Vous pouvez donc opter pour moniteur monochrome ou couleur, et un ou deux lecteurs de disquettes.

Le sous-programme implanté de 260 à 330 teste l'appui sur les flèches haut et bas et modifie les options en conséquence. Un appui sur la touche COPY permet d'en sortir. A ce sujet, précisons que si les options par défaut réunissent les meilleures conditions d'exploitation, toute autre confi-

guration n'est absolument pas un obstacle à une utilisation aisée du logiciel. C'est ainsi que sur moniteur couleur, les teintes ont été soigneusement choisies afin d'apporter un confort visuel maximum.

Nanti de vos précieuses indications, le programme va maintenant attaquer une phase décisive, pendant laquelle il va charger certains fichiers binaires utiles avant de lancer le programme principal appelé « SCHEMA.SCH ».

Celui-ci doit être adapté à vos conditions de travail, et notamment au port imprimante.

La figure 2 vous présente les

trois possibilités que vous pouvez rencontrer.

C'est à vous de fixer les fichiers qui vous seront nécessaires, en supprimant, lors de la saisie, la REM (et les trois étoiles) interdisant l'exécution de l'instruction de chargement (LOAD) qui la suit :

— Ligne 160 : cette ligne est particulière, car « X.BIN » n'est ni un fichier à caractère pornographique, ni même un fichier existant ! En fait, vous pouvez remplacer ce nom par celui d'une image-écran quelconque destinée à vous faire patienter durant le chargement des autres fichiers.

— Ligne 180 : « COPY-7B.SCH » comme son doux nom l'indique est un programme de copie d'écran sur papier. « 7B » signifie 7 bits et les AMSTRAD CPC utilisent, quelle coïncidence, un port d'imprimante parallèle à 7 bits, c'est-à-dire que les données sont acheminées de l'ordinateur vers l'imprimante via sept fils. Vous utiliserez donc ce fichier pour imprimer vos schémas à moins que...

— Ligne 190 : à moins que vous n'avez modifié votre AMSTRAD de façon à transformer ce port d'imprimante 7 bits en port 8 bits (voir chapitre REALISATION). « COPY-8B.SCH » n'est donc utilisable qu'après modification interne de votre machine (ou adjonction d'une interface externe comme nous le verrons plus loin).

— Ligne 210 : ce fichier rend opérationnelles les quelques soudures effectuées dans le cadre du montage 8 bits exposé dans ce numéro. Il ne doit pas être chargé si vous travaillez en 7 bits ou si vous utilisez un interface 8 bits externe. A ce propos, comme vous pouvez le voir sur la figure 2 et contrairement aux autres situations, « SAO.BAS » doit être lancé par le programme fourni avec votre interface, mais vous devez vous assurer que le code-machine créé par celui-ci s'implante entre A596 (ou 42390 en décimal) et A67B (ou 42619, valeur de HIMEM).

Sept bits or huit bits, that is the question... Toute imprimante fonctionnant sur AMSTRAD exige 8 bits de données pour une exploitation optimum (y compris la DMP-2000), par exemple pour

Les fichiers binaires

Ils sont au nombre de quatre mais il ne sera pas nécessaire de les saisir tous. En effet, comme expliqué au chapitre précédent, cela dépend de votre propre équipement. La **figure 3** vous précise leur adresse d'implantation en mémoire (c'est l'adresse du premier octet de chaque fichier), selon les trois possibilités que nous connaissons déjà.

Tous ces programmes sont des chargeurs BASIC construits selon un même modèle, et dont le nom porte l'extension « .DAT » (pour DATA). Suite à la saisie de chaque listing, il vous faut tout d'abord le sauver sur disque avant de le lancer par « RUN ». Si une erreur est détectée, le programme vous indique dans quel block elle se trouve. Après correction refaites un « RUN ». Si tout va bien, il affiche une ligne qu'il vous suffit de reprendre avec le curseur de copie, pour sauvegarder le fichier binaire sur votre disquette au format SYSTEME. Il porte maintenant l'extension « .SCH ». Nous vous conseillons vivement d'effectuer un RESET après chaque sauvegarde de fichier binaire.

— ROUTINES.DAT (voir **figure 4**) produit donc, après lancement, le fichier binaire « ROUTINES.SCH ». Il s'agit d'un ensemble de routines aux fonctions très diverses. La plus importante (en taille tout du moins) s'occupe de la gestion du déplacement des composants sur votre

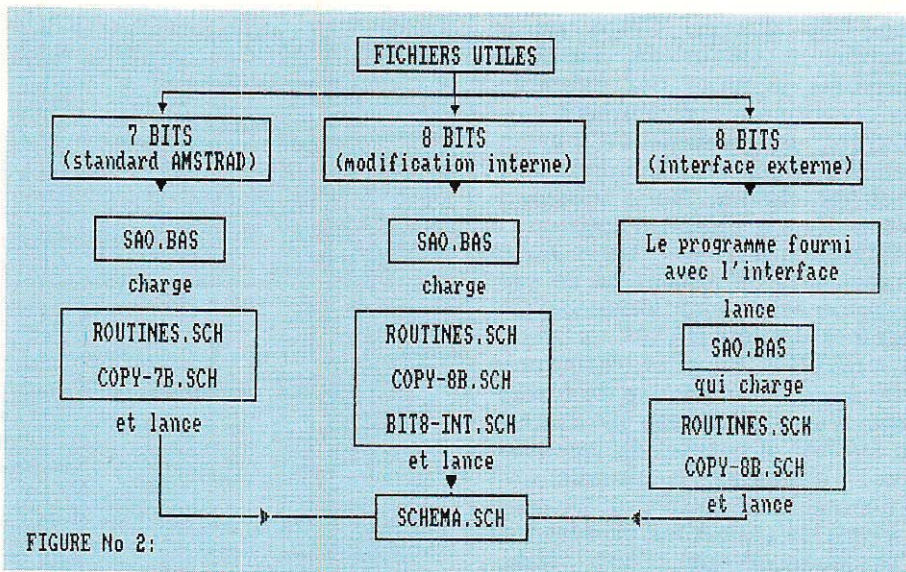


FIGURE No 2:

imprimer les caractères dont le code ASCII est supérieur à 127. Et pourquoi, lors d'une impression graphique, se contenter de 7 aiguilles alors que 8 sont disponibles ? Surtout quand on possède une DMP 2000 qui n'attend que ça, et RADIO-PLANS qui vous explique comment faire une soudure pour adapter l'unité centrale !

Une dernier mot sur « SAO.BAS ». Le sous-programme de redéfinition des touches du pavé numérique (ligne 400 et suivantes) peut être modifié ou complété par vos soins, afin d'y ajouter les caractères que vous jugez utiles de pouvoir écrire dans vos schémas. Ces caractères peuvent être, soit des symboles déjà existants (code ASCII 128 à 255), soit des caractères redéfinis par l'instruction SYMBOL, soit des groupes de caractères entre guillemets. Prenons l'exemple du symbole « Ω » affecté à la séquence CTRL + « 0 ». Il porte le code ASCII 191. Dans un premier temps, il faut associer ce code à une touche virtuelle repérée par un numéro logique (les numéros logiques disponibles vont de 141 à 159). Donc, en ligne 430, nous avons « KEY 141, CHRS (191) » ce qui signifie : la touche 141 écrit le symbole 191. Mais la touche 141 n'existe pas encore physiquement. Il faut donc maintenant la rendre accessible depuis le clavier et c'est le rôle du « KEY DEF 15, 0, 128, 128, 141 » (toujours en ligne 430). Voyons en détail ses cinq paramètres :

— 15 est le numéro logique de la

touche (touche « 0 » du pavé numérique dans le cas qui nous intéresse).

— 0 signifie pas de répétition si on garde cette touche enfoncée.

— 128 est le code renvoyé si on appuie sur la touche seule.

— le deuxième 128 est le code renvoyé si on appuie simultanément sur SHIFT et sur la touche. En effet, dans ces deux cas, c'est un zéro qui apparaît (128 est donc le numéro logique d'une touche virtuelle écrivant le code ASCII 48 « 0 »).

— enfin, 141 est le numéro logique de la touche virtuelle précédemment définie pour afficher « Ω ».

Vous pouvez déjà taper et sauvegarder « SAO.BAS » ! (en laissant en place la REM de la ligne 10 pour l'instant).

FIGURE No 3: implantation des fichiers binaires

↑	SYSTEME	↑	SYSTEME	↑	SYSTEME
	#A67B		#A67B		#A67B
	libre		libre		code-machine pour l'interface
	#A59B		#A5B7		#A596
	COPY-7B.SCH		BIT8-INT.SCH		COPY-8B.SCH
	#A4F4		#A596		#A4F4
	ROUTINES.SCH		COPY-8B.SCH		ROUTINES.SCH
	#A300		#A4F4		#A300
	SCHEMA.SCH		ROUTINES.SCH		SCHEMA.SCH
	basic		#A300		basic
	#0170		SCHEMA.SCH		#0170
			basic		
			#0170		

```

1 REM ROUTINES.DAT
10 MEMORY &A2FF:ad=&A300
20 WHILE ad<=&A4F3
30 t=0:FOR i=1 TO 56:READ v$:v=VAL
  ("%"+v$)
40 POKE ad,v:t=t+v:ad=ad+1:NEXT
50 READ b,a:IF a<t THEN PRINT "ER
REUR DANS LE BLOC":b:STOP
60 WEND
70 PRINT "SAVE ";CHR$(34);"ROUTINE
S.SCH":CHR$(34);",B,&A300,&01F4"
80 END
100 REM --- BLOCK 1---
110 DATA DD,66,0B,DD,6E,0A,22,27
120 DATA A4,DD,66,09,DD,6E,08,22
130 DATA 29,A4,DD,66,07,DD,6E,06
140 DATA 22,2B,A4,DD,66,05,DD,6E
150 DATA 04,22,2D,A4,DD,66,03,DD
160 DATA 6E,02,22,2F,A4,DD,66,01
170 DATA DD,6E,00,22,31,A4,3E,01
180 DATA 1,5505
190 REM ---BLOCK 2---
200 DATA 32,47,A4,CD,33,A4,CD,1B
210 DATA BB,FE,E0,CA,0B,A4,FE,7F
220 DATA CA,12,A4,FE,0D,CA,19,A4
230 DATA FE,4D,CA,20,A4,FE,6D,CA
240 DATA 20,A4,FE,F1,11,FE,FF,28
250 DATA 6C,FE,F5,11,F0,FF,28,65
260 DATA FE,F9,11,D0,FF,28,5E,FE
270 DATA 2,8584
280 REM ---BLOCK 3---
290 DATA F0,11,02,00,28,57,FE,F4
300 DATA 11,10,00,28,50,FE,F8,11
310 DATA 30,00,28,49,FE,F2,11,FF
320 DATA FF,28,3C,FE,FB,11,FB,FF
330 DATA 28,35,FE,FA,11,EF,FF,28
340 DATA 2E,FE,F3,11,01,00,28,27
350 DATA FE,F7,11,08,00,28,20,FE
360 DATA 3,6493
370 REM ---BLOCK 4---
380 DATA FB,11,18,00,28,19,FE,48
390 DATA 28,2D,FE,68,28,29,FE,56
400 DATA 28,2A,FE,76,28,26,FE,52
410 DATA 28,3A,FE,72,28,36,C9,DD
420 DATA 2A,27,A4,18,04,DD,2A,29
430 DATA A4,DD,6E,00,DD,66,01,19
440 DATA DD,75,00,DD,74,01,C9,2A
450 DATA 4,5737
460 REM ---BLOCK 5---
470 DATA 2B,A4,18,03,2A,2D,A4,7E
480 DATA FE,01,28,08,3E,01,77,23
490 DATA 3E,00,18,06,3E,FF,77,23
500 DATA 3E,FF,77,C9,2A,2F,A4,7E
510 DATA FE,00,28,03,3D,18,02,C6
520 DATA 01,77,C9,2A,31,A4,3E,01
530 DATA 18,13,2A,31,A4,3E,02,18
540 DATA 5,4317
550 REM ---BLOCK 6---
560 DATA 0C,2A,31,A4,3E,03,18,05
570 DATA 2A,31,A4,3E,04,77,C9,00
580 DATA 00,00,00,00,00,00,00,00
590 DATA 00,00,00,21,00,C0,11,FE
600 DATA 5A,3A,47,A4,FE,00,28,01
610 DATA EB,01,00,40,ED,B0,C9,00
620 DATA F5,3A,51,A4,CD,DE,BB,F1
630 DATA 6,4493
640 REM ---BLOCK 7---
650 DATA C9,00,3E,00,32,81,A4,0E
660 DATA 07,CD,0F,B9,C5,3E,FF,CD
670 DATA 72,CA,21,FF,9A,16,00,1E
680 DATA 00,0E,41,CD,66,C6,30,0A
690 DATA 3E,00,CD,72,CA,C1,CD,18
700 DATA B9,C9,3E,FF,32,81,A4,18
710 DATA EF,00,DD,6E,00,DD,66,01
720 DATA 7,6172
730 REM ---BLOCK 8---
740 DATA 46,23,5E,23,56,EB,11,FF
750 DATA 9A,DD,7E,02,FE,01,28,0F
760 DATA CD,77,BC,30,06,21,FE,5A
770 DATA CD,83,BC,CD,7A,BC,C9,CD
780 DATA 8C,BC,30,0E,21,FE,5A,11
790 DATA 00,40,01,00,00,3E,02,CD
800 DATA 9B,BC,CD,8F,BC,C9,2A,80
810 DATA 8,6393
820 REM ---BLOCK 9---
830 DATA BC,22,F1,A4,3A,82,BC,32
840 DATA F3,A4,C9,E5,2A,F1,A4,22
850 DATA 80,BC,3A,F3,A4,32,82,BC
860 DATA CD,80,BC,21,CB,A4,22,81
870 DATA BC,21,80,BC,36,C3,E1,D8
880 DATA C8,FE,1A,37,3F,C0,B7,37
890 DATA C9,00,00,00,00,00,00,00
900 DATA 9,7060
910 REM ---TERMINE---

```

Figure no : 4

page de dessin. Cette partie modifie la valeur de certaines variables (comme par exemple, les coordonnées du curseur graphique), en testant en permanence l'appui sur certaines touches du clavier. Nous y reviendrons dans le chapitre consacré à « SCHEMA.SCH ». Un autre aspect très important de ce programme est la simulation binaire des instructions BASIC « LOAD » et « SAVE ». En effet, conjuguée avec la reconnaissance du formatage d'une disquette (SYSTÈME ou DATA), il permet de remédier aux problèmes habituellement rencontrés sur le CPC 464 quand il s'agit de traiter les erreurs lors d'un chargement, d'une sauvegarde, ou de tout accès au disque en général. La fonction ERR ne reconnaît pas ce type d'erreurs sur 464. Pour poursuivre dans cette voie — en bravant les foudres de Mister SUGAR — citons le cas du MERGE, qui s'obstine à répondre « EOF MET » (en français : fin de fichier rencontré), lorsqu'on utilise cette commande pour merger un programme sur disquette. Notons au passage que la solution qui consiste à sauvegarder les fichiers à merger sous forme ASCII ne fonctionne pas si deux drives sont branchés. Mais, n'ayez crainte, cette situation a été prise en compte par « ROUTINES.SCH ». Parmi ses autres fonctions, citons encore une simulation du « GRAPHICS PEN », instruction réservée aux CPC 664 et 6128 (encore un « trou » dans la ROM du 464), et permettant d'attribuer au graphisme une couleur différente de celle du texte. Comme vous venez de le voir, ces quelques octets (500 très exactement) sont d'une utilité certaine !

— COPY-7B.DAT (figure 5) qui crée « COPY-7B.SCH » est la copie d'écran pour port d'imprimante 7 bits. Seul AMSTRAD a oublié le huitième !

Cette hardcopy est en fait une adaptation pour « SAO » d'une routine que les fidèles lecteurs de RADIO-PLANS ont déjà pu apprécier dans le N° 469, page 35.

Le format et l'interligne ont été fixés de façon à respecter le mieux possible les exigences d'impression d'un schéma électrique (proportions, clarté...). Un petit défaut, toutefois, peut être

```

1 REM COPY-7B.DAT
10 MEMORY &A4F3:ad=&A4F4
20 WHILE ad<=&A59B
30 t=0:FOR i=1 TO 56:READ v$:v=VAL
  ("%"+v$)
40 POKE ad,v:t=t+v:ad=ad+1:NEXT
50 READ b,a:IF a<t THEN PRINT "ER
REUR DANS LE BLOC":b
60 WEND:POKE &A59C,0:POKE &A59D,0
70 PRINT "SAVE ";CHR$(34);"COPY-7B
.SCH":CHR$(34);",B,&A4F4,&00AA"
80 END
100 REM ---BLOCK 1---
110 DATA CD,90,A5,3E,1B,CD,87,A5
120 DATA 3E,33,CD,87,A5,3E,13,CD
130 DATA 87,A5,CD,BA,BB,CD,E7,BB
140 DATA 32,2D,A5,11,00,00,21,8F
150 DATA 01,22,9B,A5,3E,0A,CD,87
160 DATA A5,3E,0D,CD,87,A5,3E,1B
170 DATA CD,87,A5,3E,2A,CD,87,A5
180 DATA 1,6793
190 REM ---BLOCK 2---
200 DATA 3E,04,CD,87,A5,3E,7F,CD
210 DATA 87,A5,3E,02,CD,87,A5,0E
220 DATA 00,06,07,E5,C5,D5,CD,F0
230 DATA BB,D1,C1,21,9D,A5,BE,E1
240 DATA 37,20,01,A7,CB,11,2B,10
250 DATA EA,79,CD,87,A5,13,E5,21
260 DATA 7F,02,37,ED,52,E1,36,05
270 DATA 2,6768
280 REM ---BLOCK 3---
290 DATA 2A,9B,A5,18,D2,CD,1B,BB
300 DATA FE,51,20,02,18,1E,7C,B4
310 DATA 20,04,7D,B5,28,16,7C,FE
320 DATA FF,28,11,11,00,00,22,9B
330 DATA A5,18,91,CD,2E,BD,38,FB
340 DATA CD,2B,BD,C9,3E,1B,CD,87
350 DATA A5,3E,40,CD,87,A5,C9,00
360 DATA 3,6187
370 REM ---TERMINE---

```

Figure no : 5

reproché à cette routine car elle n'imprime pas la dernière colonne à droite de l'écran, soit 1 pixel de moins à l'extrémité de chaque ligne). La raison en est simple : l'écran est composé de 640 points horizontaux, mais il est impossible de commander à l'imprimante de tracer 640 points si l'on travaille en 7 bits ! Lors du passage en mode graphique, la machine demande un paramètre lui indiquant le nombre de colonnes à imprimer. Ce paramètre N est décomposé en deux nombres n1 (le poids faible) et n2 (le poids fort), et est calculé selon la formule « N = n1 + (n2*256) ». Or, en 7 bits, n1 et n2 ne peuvent varier qu'entre 0 et 127, il est donc impossible de les définir pour N = 640. (639 = 127 + 2*256 étant la valeur la plus proche). Voilà pour les Grandeur et Décadence de la copie d'écran pour port 7 bits.

— COPY-8B.DAT, présenté en figure 6, est une adaptation au port d'imprimante à 8 bits, du programme décrit plus haut. Il conserve ses qualités en s'affranchissant de ses défauts, à condition — bien entendu — de modifier votre AMSTRAD. Vous pourriez imprimer toutes les colonnes de votre écran et bien d'autres choses...

— BIT8-INT.DAT est un tout petit fichier que vous pouvez voir dans la toute petite **figure 7**. Ces quelques octets suffisent à faire fonctionner la réalisation pratique présentée plus loin. Les trois dernières lignes (140 à 160) comportent des REMs qu'il vous faut supprimer en accord avec votre AMSTRAD préféré. De plus amples informations sur le rôle de ce fichier vous sont fournies quelques lignes plus bas. Les lecteurs souhaitent utiliser la copie d'écran 8 bits, ou le MERGE pour 464 (transparent pour 664 et 6128), à d'autres usages que « SAO », seront comblés dès le mois prochain : ils trouveront une routine remplissant ces fonctions.

Il ne vous reste plus qu'à éteindre votre précieuse machine et à faire chauffer votre fer à souder...

Réalisation

Au risque de nous répéter, précisons qu'il n'est pas du tout impératif d'effectuer cette modification pour exploiter SAO.

Pourtant, si vous ne risquez plus de perdre une garantie, les auteurs vous conseillent très vivement ce passage à 8 bits du port d'imprimante. Pour leur part, ils en exploitent les avantages — l'un sur 464 l'autre sur 6128 —, depuis un an environ, sans problème.

La solution retenue est la moins coûteuse qu'il soit possible d'envisager, tant sur le plan de l'énergie que du portefeuille. Les auteurs ne revendiquent nullement la paternité de l'idée, mais seraient bien en peine de citer le réel géniteur, car de multiples signatures se sont posées sous des articles quasiment identiques !

Le « plus » que nous proposons, tient dans la routine d'analyse, totalement transparente : il n'est plus nécessaire ici de faire des commutations logicielles, pour ne passer en huit bits que pour les modes graphiques.

Voyons le principe :

Quand il s'agit d'imprimer, tout passe par une routine système d'adresse hexa BD2B. Celle-ci reçoit en entrée, le code ASCII du caractère ou l'image binaire du point à imprimer. En sortie, elle envoie les bits 6 à 6 de ce code sur le port d'imprimante.

```

1 REM COPY-8B.DAT
10 MEMORY &A4F3:ad=&A4F4
20 WHILE ad<=&A595
30 t=0:FOR i=1 TO 56:READ v$:v=VAL
("&"+v$)
40 POKE ad,v:t=t+v:ad=ad+1:NEXT
50 READ b,a:IF a<>t THEN PRINT "ER
REUR DANS LE BLOCK":b:STOP
60 WEND
70 PRINT "SAVE ";CHR$(34);"COPY-8B
.SCH";CHR$(34);",B,&A4F4,&00A2"
80 END
100 REM ---BLOCK 1---
110 DATA CD,88,A5,3E,1B,CD,7F,A5
120 DATA 3E,33,CD,7F,A5,3E,16,CD
130 DATA 7F,A5,CD,BA,BB,CD,E7,BB
140 DATA 32,95,A5,11,00,00,21,8F
150 DATA 01,22,93,A5,3E,0A,CD,7F
160 DATA A5,3E,0D,CD,7F,A5,3E,1B
170 DATA CD,7F,A5,3E,2A,CD,7F,A5
180 DATA 1,6716
190 REM ---BLOCK 2---
200 DATA 3E,04,CD,7F,A5,3E,80,CD
210 DATA 7F,A5,3E,02,CD,7F,A5,0E
220 DATA 00,06,08,E5,C5,D5,CD,F0
230 DATA BB,D1,C1,21,95,A5,BE,E1
240 DATA 37,20,01,A7,CB,11,2B,10
250 DATA EA,79,CD,7F,A5,13,E5,21
260 DATA 80,02,37,ED,52,E1,38,05
270 DATA 2,6731
280 REM ---BLOCK 3---
290 DATA 2A,93,A5,18,D2,CD,1B,BB
300 DATA FE,51,20,02,18,16,7C,FE
310 DATA FF,28,11,11,00,00,22,93
320 DATA A5,18,99,CD,2E,BD,38,FB
330 DATA CD,2B,BD,C9,3E,1B,CD,7F
340 DATA A5,3E,40,CD,7F,A5,C9,00
350 DATA 00,00,00,00,00,00,00
360 DATA 3,5447
370 REM ---TERMINE---

```

Figure no : 6

```

1 REM BIT8-INT.DAT
10 MEMORY &A595:ad=&A596
20 WHILE ad<=&A5B6
30 t=0:FOR i=1 TO 33:READ v$:v=VAL
("&"+v$)
40 POKE ad,v:t=t+v:ad=ad+1:NEXT
50 READ a:IF a<>t THEN PRINT "ERRE
UR EN DATA":STOP
60 WEND
70 PRINT "SAVE ";CHR$(34);"BIT8-IN
T.SCH";CHR$(34);",B,&A596,&0021"
80 END
100 DATA 3E,C3,32,2B,BD,21,A2,A5
110 DATA 22,2C,BD,C9,F5,C5,CB,7F
120 DATA 20,05,01,00,FE,18,03,01
130 DATA 20,FE,ED,49,C1,F1,CF
140 REM CPC 6128: DATA 1B,88,3843
150 REM CPC 664 : DATA 0B,88,3827
160 REM CPC 464 : DATA F2,87,4057

```

Figure no : 7

Le « SOFT » a deux rôles. La première partie détourne toutes les données arrivant en BD2B sur la seconde partie. Là, chaque code est analysé, et on isole son 8^e bit (ou bit N° 7). Ce bit est alors envoyé au 8255 où il apparaît sur la broche N° 12. Cette sortie est une bascule qui va donc positionner la piste correspondante du port d'imprimante à 1 ou 0, par l'intermédiaire du fil soudé entre ces deux points. Ensuite, la main est rendue à BD2B qui remplit sa fonction habituelle. Donc, à l'instant où cette routine libère les 7 premiers bits, le 8^e est présent à l'appel et l'imprimante reçoit bien les 8 bits attendus.

En fait, le 8^e bit est reconstitué à l'aide de circuits détournés. A

l'origine, la broche 12 du 8255 est réservée à l'écriture des données dans l'interface cassette, fonction qu'elle continue de remplir.

Si l'on part du principe que l'on n'a pas besoin des envois de données sur cassette quand l'impression est demandée, on ne se rend compte d'aucune gêne (si l'on n'a pas oublié de lancer le logiciel !).

La **figure 8** est coupée en deux : la partie « a » montre la modification de principe, la partie « b », celle de « terrain ».

La broche 9 du PRINTER PORT, au lieu d'être en permanence portée à zéro volt, est reliée à la broche 12 du 8255. C'est tout !

Le choix vous est laissé de mettre en place ou non, un inverseur permettant de revenir à l'état originel. Les photos vous montrent un 464 avec inter + led (visualisant la position 8 bits), ainsi qu'une cinch offrant la possibilité d'accéder au 5 V interne. C'est celui de l'auteur cobaye... Pour le 6128, un simple inverseur (que son propriétaire laisse en permanence sur la position « 8B »), a suffit.

Pour mener à bien cette opération, deux observations s'imposent :

1° Couper soigneusement l'actuelle alimentation de la broche 9 du PRINTER PORT.

2° Soudez le fil sur la broche 12 du 8255 (ou sur une piste équivalente), avec un fer de faible puissance, débranché du réseau EDF.

Naturellement, nous n'insistons pas sur les précautions à prendre au moment d'ouvrir la machine, ou de déconnecter le clavier de la carte principale.

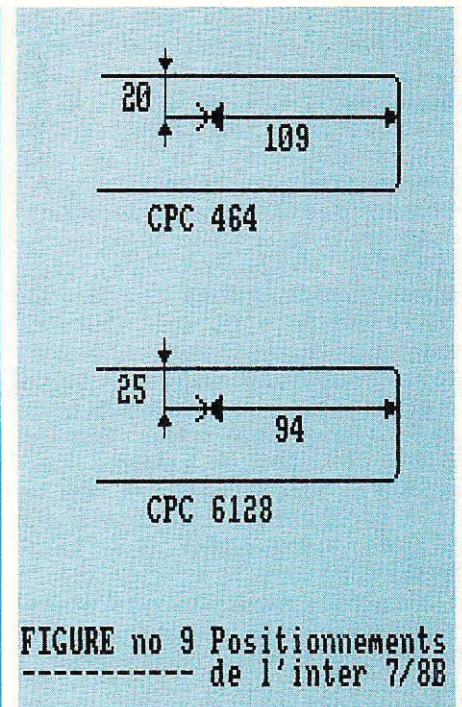
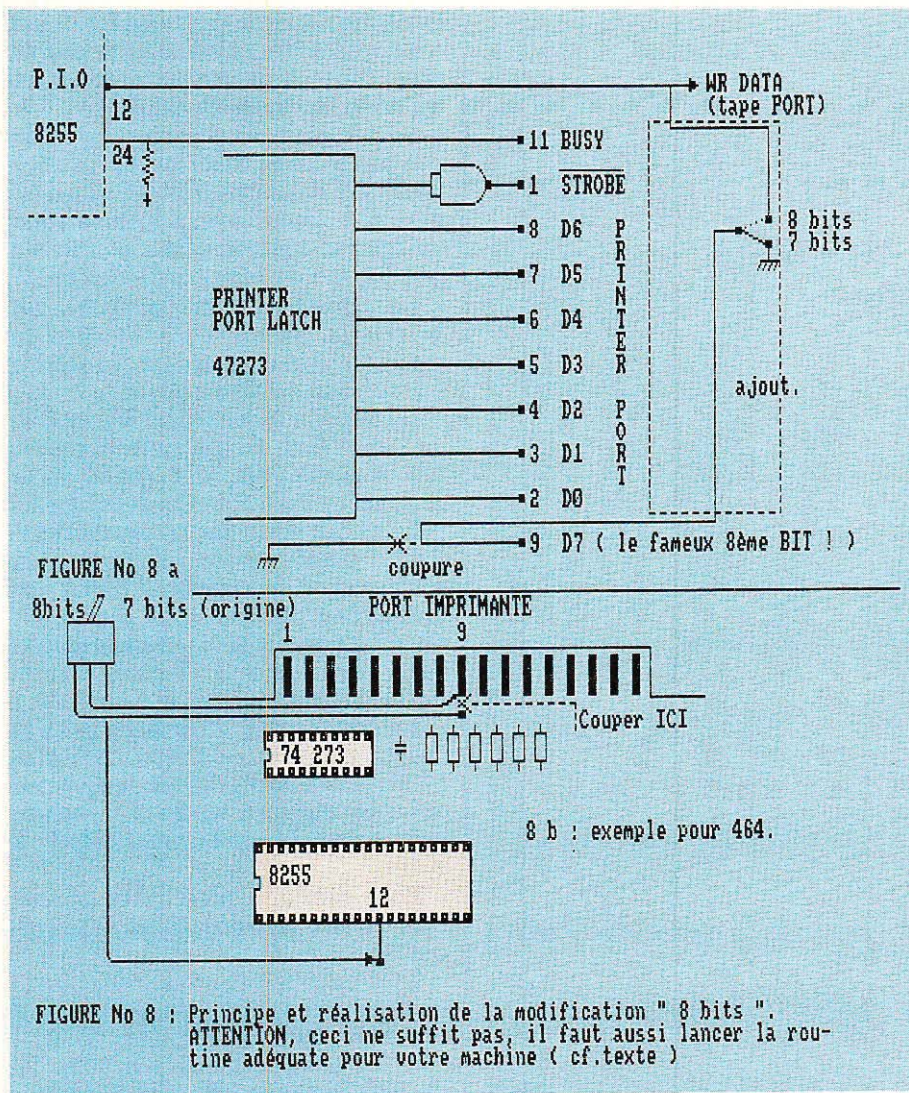
Sur le 6128, il est plus simple de souder le fil directement sur la patte 12 du 8255. Pour le 464, un trou métallisé est très facilement accessible. C'est lui qui sert de relais.

La **figure 9** vous donne les repères utiles pour positionner l'inverseur sur 464 et 6128.

NOTA : Pas de perceuse dans le plastique S.V.P. ! Le forêt monté dans un mandrin à main, est beaucoup plus simple et rassurant...

Tests

Vérifiez l'absence de « corps étrangers » à l'intérieur de votre machine. Refermez-là, et passez aux essais : l'inverseur doit être en position 8 BITS. La première



Vous pouvez y lire « Développement du nombre de codes imprimables, pour sortie à 8 BITS seulement ». Suivez les instructions et vous aurez une preuve tangible de la présence des 8 BITS. Pour sélectionner cette fonction, tapez CHR\$(27) ; « 6 » et non pas CHR\$(27) ; CHR\$(6).

Si votre imprimante n'est pas une DMP-2000, nous vous invitons à vous reporter à son mode d'emploi afin de profiter des fonc-

vérification consiste à faire fonctionner la machine sans le « SOFT », c'est-à-dire en 7 BITS. Il suffit donc de faire imprimer les codes habituels 32 à 126 par une boucle du type : « FOR i = 32 TO 126 : PRINT#8, CHR\$(i) ; NEXT », ou même d'effectuer un simple « LIST#8 ». Ça marche ! Nous voyons donc que le nouveau port 8 BITS est compatible avec l'ancien : vous pouvez laisser l'inverseur sur 8 BITS, en permanence, avec ou sans logiciel.

Et le 8^e BIT ? Si vous possédez une DMP-2000, assurez-vous que le DIP DSI-7 (7 ou 8 BITS) est en position haute. Puis charger « BIT8-INT.DAT » et faites un « RUN ». Effectuez la sauvegarde du fichier binaire si ce n'est pas encore fait. Tapez « CALL A596 ». Et voilà, le huitième BIT existe ! Toujours pour la DMP-2000, reportez-vous à la page 1, chapitre 6 du manuel de l'imprimante (AUTRES FONCTIONS...).

```

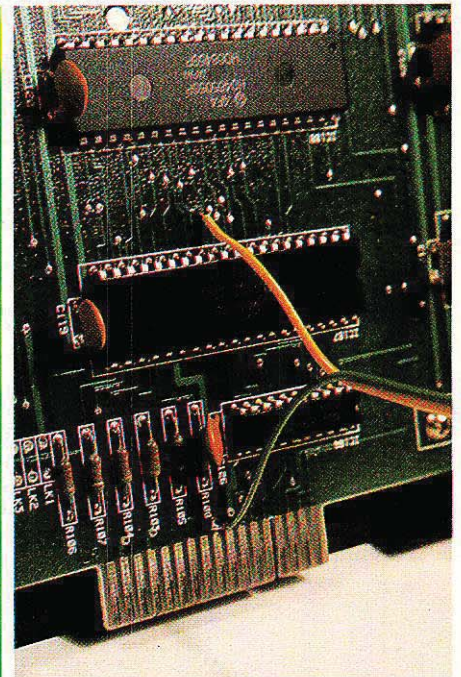
1 REM SCHEMA.SCH
2 nlec!=PEEK(&5AFE):IF nlec!=1 THEN d$="B:" ELSE d$="A:"
30 DEFINT a-z:e=3:m=3:m$="":g=0:dc=0
40 trans!=&A433:comut!=&A447:gpen!=&A448:flag!=&A451:tdisc!=&A452:numlec!=&A468:
50 sysdat!=&A46A:errdisc!=&A481:inout!=&A482:hardcopy!=&A4F4
60 MODE 2:POKE comut!,0:CALL trans!
70 GOSUB 230 'menu
80 DEFINT a-z
90 CALL &A300,0v,0w,0x,0y,0z,0t
100 IF (v<0 OR v>639) OR (w<0 OR w>399) THEN SOUND 1,600,15
110 IF INKEY(62)<>-1 THEN IF dc=0 THEN dc=1 ELSE dc=0
120 GOSUB 1040
130 IF INKEY(52)<>-1 THEN GOSUB 870
140 ON t GOSUB 180,200,230,230
150 GOSUB 150:GOTO 70
160 ' BRANCHEMENTS
170 ORIGIN v,w ON a GOSUB 3010,3110,3210,3310,3410,3510,3610,3710,3810,3910,4010
180 ,4110,4210,4310,4410,4510,5020,5110,5210,5310,5410,5510,5610,5710,3730,3520,3150
190 RETURN
200 ' COPY
210 GOSUB 150:POKE comut!,0:CALL trans!:t=0:RETURN
220 ' DEL
230 POKE flag!,0:CALL gpen!:GOSUB 180:POKE flag!,1:CALL gpen!:t=0:RETURN
240 ' MENU
250 ns="":v=320:w=200:x=1:y=1:z=0:POKE comut!,0:CALL trans!:IF t=3 THEN 450
260 rs="":ns="":t=0:MODE 2:RESTORE 290
270 LOCATE 1,1:PRINT CHR$(24);" (R)epertoire (C)harger (S)auver (E)raser (I)
280 )primer (N)ouveau (Q)uitter ";CHR$(24)
290 FOR i=1 TO 69 STEP 17:FOR j=4 TO 18 STEP 2:READ comp$
300 IF i=35 THEN LOCATE i+2,j ELSE IF i=52 THEN LOCATE i+1,j ELSE IF i=69 THEN L
310 OCATE i-2,j ELSE LOCATE i,j
320 PRINT comp$:NEXT:NEXT
330 DATA 1 POINTS,2 RECTANGLES,3 CARRÉS,4 POINTILLES, 5 COIN/CROIX,6 CONNEXIONS,
340 7 FLECHE,8 ALPHA,9 RESISTANCE,10 AJUSTABLE,11 POTENTIOMETRE,12 CONDENSATEUR,13 C
350 OND POLARISE, 14 IC, 15 MASSE/TERRE,16 CERCLES
    
```

```

300 DATA 17,DIODE,18,PONT,19,LED,20,INTERS,21,INVERSEUR,22,POUSSOIR,23,RELAIS,24
FUSIBLE,25,ZENER,26,SELF,27,VARICAP,28,QUARTZ,29,F.1.,30,NPN,31,PNP,32,FET,33
AND,34,NAND,35,OR,36,NOR,37,EXOR,38,EXNOR,39,NO,40,ACCU
310 LOCATE 4,22:PRINT "TAPEZ une LETTRE du MENU ou le NUMERO du COMPOSANT : ";
320 LINE INPUT " ",r$:r$=UPPER$(r$):IF r$="" THEN a=0:RETURN
330 ok$="RCSEINQ":IF INSTR(ok$,r$)=0 THEN 350
340 ON INSTR(ok$,r$) GOTO 490,490,490,490,490,660,730,770
350 a=VAL(LEFT$(r$,2)):IF a<1 OR a>40 THEN 310
360 IF (a>16 AND a<25) THEN m=3
370 IF (a>24 AND a<33) THEN a=a-8:m=4
380 IF (a>32 AND a<41) THEN a=a-16:m=5
390 IF e=m THEN RETURN ELSE e=m
400 POKE numlec!,0:POKE sysdat!,&41:CALL tdisc!
410 IF PEEK(errdisc!)=255 THEN LOCATE 4,22:PRINT CHR$(24);"INSERREZ DISQUETTE SY
STEME EN 'A' PUIS 'TAPEZ UNE TOUCHE':CHR$(24):CALL &BB18:LOCATE 4,22:PRINT SPACE$
(55):GOTO 400
420 CHAIN MERGE "COL-"+RIGHT$(STR$(e),1),60,DELETE 5000-
430 RETURN
440 ' Fenetre choix composant
450 t=0:WINDOW#1,35,45,12,15:CLS#1:PRINT#1,"COMPOSANT":PRINT#1," No.":
460 LINE INPUT#1," ",r$:IF r$="" THEN 240 ELSE a=VAL(LEFT$(r$,2))
470 IF a<1 OR a>40 THEN 450 ELSE GOTO 360
480 ' repertoire/sauvegarde/chargement/eraser
490 CLS:LOCATE 4,4:PRINT CHR$(7);"INSERREZ LA DISQUETTE DATA DANS LE LECTEUR ";
LEFT$(a,1);" ET APPUYEZ SUR UNE TOUCHE"
500 CALL &BB18
510 POKE numlec!,nlec!:POKE sysdat!,&C1:CALL tdisc!
520 IF PEEK(errdisc!)=255 THEN GOTO 490
530 POKE &A700,nlec!:CAT:POKE &A700,0
540 ON INSTR(ok$,r$) GOTO 640,590,560,620
550 ' sauvegarde
560 LINE INPUT "NOM DU SCHEMA A SAUVEGARDER (SANS extension) : ",n$:IF n$="" THE
N 640
570 n$=d$+n$+".SAO":PRINT:PRINT "SAUVEGARDE EN COURS...":CALL inout!,1,&n$:GOTO
640
580 ' chargement
590 LINE INPUT "NOM DU SCHEMA A CHARGER (extension par default = .SAO) : ",n$:IF
n$="" THEN 640
600 n$=d$+n$:IF INSTR(n$,".")=0 THEN n$=n$+".SAO"
605 PRINT:PRINT "CHARGEMENT EN COURS...":CALL inout!,0,&n$:GOTO 640
610 ' eraser
620 LINE INPUT "NOM DU SCHEMA A ERASER (AVEC extension) : ",n$:IF n$="" THEN 640
630 n$=d$+n$:PRINT:PRINT "EFFACEMENT...":ERA,&n$:POKE &A700,nlec!:CAT:POKE &A70
0,0
640 PRINT:PRINT " TAPEZ UNE TOUCHE":CALL &BB18:GOTO 240
650 ' impression
660 WINDOW#0,33,47,8,15:PAPER 1:PEN 0:CLS
670 PRINT:PRINT "APPUYEZ SUR 'O':PRINT "POUR CONFIRMER":PRINT
680 PRINT " AUTRE TOUCHE":PRINT:PRINT " ANNULATION"
690 CALL &BB18:IF INKEY(34)=-1 THEN 710
700 MODE 2:POKE comut!,1:CALL trans!:CALL hardcopy!
710 PAPER 0:PEN 1:GOTO 240
720 ' effacer
730 mo$=" NOUVEAU":lo$="N":o=46:GOSUB 810
740 IF co<>1 THEN 240
750 MODE 2:POKE comut!,0:CALL trans!:GOTO 240
760 ' quitter
770 mo$=" QUITTER":lo$="Q":o=67:GOSUB 810
780 IF co<>1 THEN 240
790 CALL 0
800 ' fenetre impression/nouveau/quitter
810 WINDOW#0,33,47,8,15:PAPER 1:PEN 0:CLS
820 PRINT:PRINT " CTRL/SHIFT ";lo$:PRINT:PRINT mo$:PRINT
830 PRINT " AUTRE TOUCHE":PRINT:PRINT " ANNULATION"
840 co=0:CALL &BB18:IF INKEY(o)=160 THEN co=1
850 PAPER 0:PEN 1:RETURN
860 ' graduation on/off
870 IF g=1 THEN g=0 ELSE g=1
880 ORIGIN 0,0:POKE flag!,g:CALL gpen!
890 g1=2:FOR i=8 TO 632 STEP 24
900 IF g1=2 THEN g1=4 ELSE g1=2
910 MOVE i,399:DRAWR 0,-g1
920 MOVE i,0:DRAWR 0,g1
930 NEXT
940 FOR i=1 TO 2:PLOT 320,4+i:PLOT 320,395-i:NEXT
950 g1=2:FOR i=8 TO 399 STEP 16
960 IF g1=2 THEN g1=4 ELSE g1=2
970 MOVE 0,i:DRAWR g1,0
980 MOVE 639,i:DRAWR -g1,0
990 NEXT
1000 FOR i=1 TO 2:PLOT 4+i,200:PLOT 636-i,200:NEXT
1010 POKE flag!,1:CALL gpen!
1020 POKE comut!,0:CALL trans!:RETURN
1030 ' coordonnees on/off
1040 IF dc=0 THEN RETURN
1050 LOCATE 1,1:PRINT CHR$(24);v;CHR$(24);
1060 FOR i=0 TO 100:NEXT ' temporisation
1070 LOCATE 1,1:PRINT SPACE$(12):RETURN

```

Figure no : 10



tions supplémentaires offertes par le port à 8 BITS.

Voilà pour la partie pratique. Comme vous pouvez le constater, aucune alimentation n'est nécessaire, et la modification est ridiculement simple.

Il faudra penser à charger le logiciel adapté à votre machine : 464, 664, 6128. Entre les trois machines, le Hard est identique, seul le Soft doit être adapté.

Schéma.SCH

Nous avons divisé ce programme en deux parties bien différentes. La deuxième, composée exclusivement des sous-programmes de dessin des composants, sera publiée le mois prochain. La figure 10 nous montre donc la première partie du listing de « SCHEMA.SCH ».

Il ne sera pas fait ici d'analyse ligne par ligne. Nous tenterons simplement d'expliquer le rôle de



chacun des modules indépendants composant ce programme :

— LA BOUCLE PRINCIPALE (lignes 70 à 130). Son instruction centrale est « CALL A300 » : on fait appel à une routine binaire contenue dans « ROUTINES.SCH ». Elle a pour but d'afficher le composant choisi au menu et de tester les touches du clavier. Suivant les touches appuyées, le composant sera retourné, déplacé, fixé, ou effacé. Le principe d'affichage devait répondre à trois critères :

- Rapidité du déplacement.
- Possibilité de déplacer un composant par-dessus un autre, déjà placé.
- Possibilité de supprimer un composant à n'importe-quel moment.

1) Le composant apparaît dans la ZONE 1.

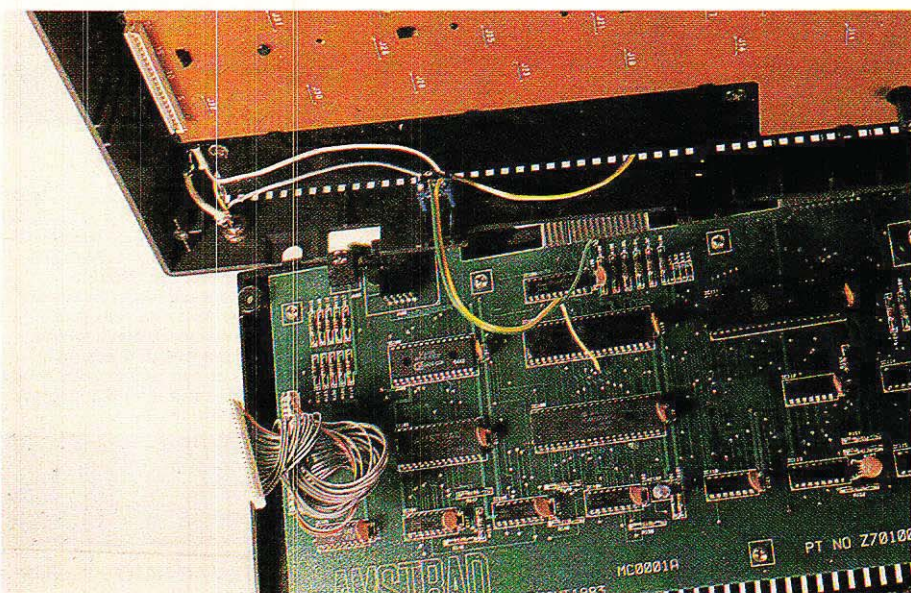
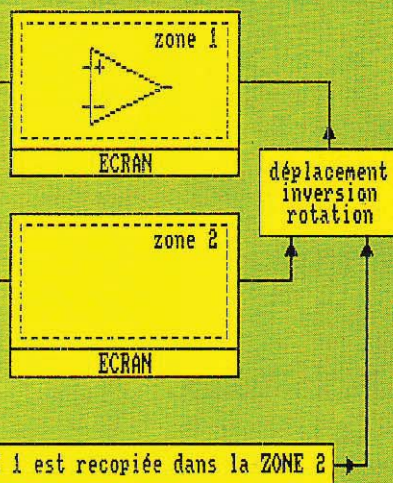
2) Il disparaît quand la ZONE 2 est appelée. (*)

3) Le composant réapparaît en ZONE 1. Sa nouvelle position a été prise en compte.

4) Retour en 2.

* Si COPY a été appuyé, le composant est fixé dans la ZONE 2.

FIGURE No 11: le cycle d'affichage.



ZONE 2 et, tout composant « COPYier » apparaîtra dorénavant, quelle que soit la ZONE présente à l'écran.

La touche « DEL » fait de même, à la différence que le composant est redessiné avec la couleur du fond avant la recopie. Ainsi, il disparaît.

Dans cette BOUCLE, nous trouvons également un test — en BASIC — de l'appui sur les touches « C » et « G ». Elles commandent respectivement l'affichage des coordonnées du curseur graphique (GOSUB 1040), et celui d'une graduation sur le pourtour de l'écran (GOSUB 870). La ligne 80 provoque un « bip » dès que l'on sort des limites de cet écran.

La figure 11 décrit le cycle d'affichage. La ZONE 1 correspond à la mémoire-écran. La ZONE 2 est un tampon de 16 Ko, pouvant être rempli avec cette mémoire-écran. Au lancement de « SAO », tous les octets de ce tampon sont à zéro, il affiche donc une page blanche. Le cycle consiste à faire apparaître alternativement à l'écran la ZONE 1 et la ZONE 2. Entre ces deux apparitions, le composant choisi peut subir toutes les transformations souhaitées.

La fonction d'échange ZONE 1 / ZONE 2 est assurée par une routine binaire : sa rapidité est telle que le transfert est indiscernable.

Un appui sur « COPY » a pour effet de recopier ZONE 1 dans

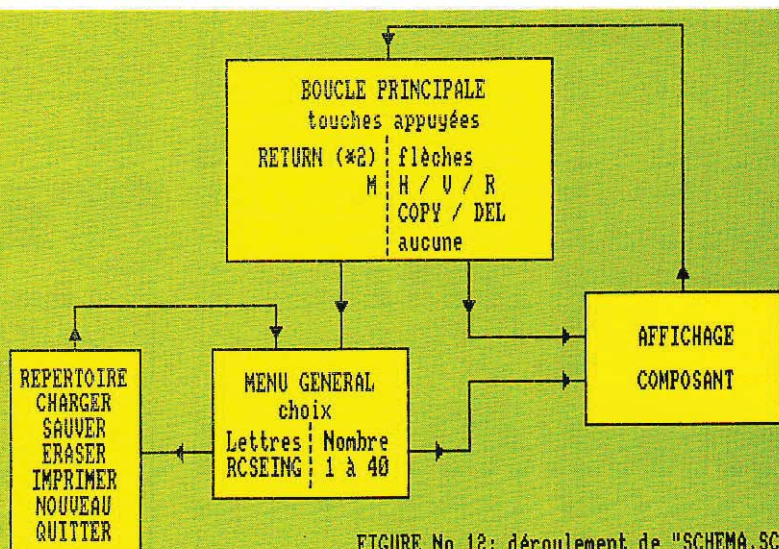


FIGURE No 12: déroulement de "SCHEMA.SCH".

OPTIONS PAR DEFAUT

(*) - NOMBRE MAXIMUM
 (*) - 2 LIGNES DE DIMENSION (dis. système en "M")

Utilisez les flèches pour changer les options
 puis **ENTER** pour lancer le programme

Clavier (Clear) (Shift) (Shift) (Shift) (Shift) (Shift) (Shift)

1 POINTS	9 RESISTANCE	17 DIODE	25 COND	33 IND
2 RECTANGLES	10 AJUSTABLE	18 PONT	26 DEL	34 NAND
3 CARRÉS	11 POTENTIOMETRE	19 LED	27 VARICAP	35 OR
4 POINTILLES	12 CONDENSATEUR	20 INTERR	28 ORGATE	36 NOR
5 COIN/COIN	13 COND POLARISE	21 INVERSEUR	29 F.L.	37 DIOR
6 COMPLEXES	14 IC	22 POUSSOIR	30 NON	38 DIAND
7 FLECHE	15 MASSE/TERRA	23 RELAIS	31 PNE	39 NO
8 ALPHA	16 CERCLES	24 THERIE	32 PET	40 ACCF

APPUIZ sur LETTRE du NOM ou le NUMERO de COMPOSANT : M

HUIT possibilités pour le IC :

18000
 1801250
 I

INSERER LA DISQUETTE 5.25" DANS LE LECTEUR "F" ET APPUYEZ SUR UNE TOUCHE

Drive B: user 0

CELE	.500	17K	MONTEST.500	17K	TALK	.500	17K
DIAGRAM	.500	17K	EDMTEST.500	17K	TALK2	.500	17K
MONFREQ	.500	17K	ZERON	.500	17K	TALK3	.500

25K free

NON EN SCHEMA & CHARGER (SANS extension) : DIAGRAM

CHARGEMENT EN COURS...

SAO by SAC 0

PARTICULARITES DU POIN DES FLECHES :

ATTENTION: P + CTRL = CLR
 M + CTRL = ENTER

WINGO: 1 + SHIFT
 CTRL

TAB	0	1	2	3	4	5	6	7	8	9	CLR/DEL	N	1	2	3	4	5	6	7	8	9	ENTER	P	1	2	3	4	5	6	7	8	9	ENTER	W	1	2	3	4	5	6	7	8	9	ENTER
SHIFT	0	1	2	3	4	5	6	7	8	9	SHIFT	N	1	2	3	4	5	6	7	8	9	SHIFT	W	1	2	3	4	5	6	7	8	9	SHIFT	W	1	2	3	4	5	6	7	8	9	SHIFT

Possibilités ALPHA avec CTRL

FIGURE No

ONLY PARTS OF SAO

...and OTHER (no limit)

MICAS, une extension de SAO 0

Et en changeant seulement les PILES..... à SUITE....

Et pourquoi pas

FIGURE No 1 : Diagramme des niveaux de la chaîne principale.

MICAS

MATRIEL du bus SOLA

FIGURE No 1

EMPLI DE PILESSANCE 2 x 13000000 . AC POWER PL

Compteurs

Boards de données

COMPTESSEUR PARALLELE 8 x 8 BITS

Sortie binaire

MSB

BIT 2

LSB

Entrée analogique

1ère Partie

Cassée vers l'extérieur

15 V POWER

0 V POWER

15 V POWER

OUT INTERCOM

TS.1

12P

ots. 800 + 8000 (sans TERT)

IC

Insert. dans STRIP 10.

15 volts

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

verrouillage

effacement

point decimal gauche

point decimal droit

ATTACHEZ TIL 111.

— LE MENU GÉNÉRAL (lignes 220 à 430 de la figure 10). Il affiche cinq colonnes de huit composants, et un bandeau supérieur donnant accès aux autres fonctions du programme.

A chaque demande d'un nouveau composant (nombre entre 1 et 40), on regarde si ce composant est présent en mémoire. Si ce n'est pas le cas, on « MERGE » la colonne dont il fait partie (voir lignes 350 à 420). Avant un « MERGE », on vérifie si la disquette placée dans le lecteur « A » est bien formatée en SYSTÈME. Cela évite dans une certaine mesure, les « COL-X.BAS not found ».

La sélection d'une lettre du bandeau en vidéo-inverse renvoie aux différents sous-programmes concernés. Vous pouvez consulter à ce sujet, la figure 12. Elle montre un organigramme très simplifié de « SCHEMA.SCH ».

— REPERTOIRE / CHARGER / SAUVER / ERASER. Ces quatre modules s'étendent de la ligne 480 à la ligne 640. Ils sont traités ensemble puisque leurs caractéristiques sont communes. On effectue un test de présence et de formatage de la disquette (DATA), puis on affiche un catalogue, avant de se brancher sur le sous-programme demandé.

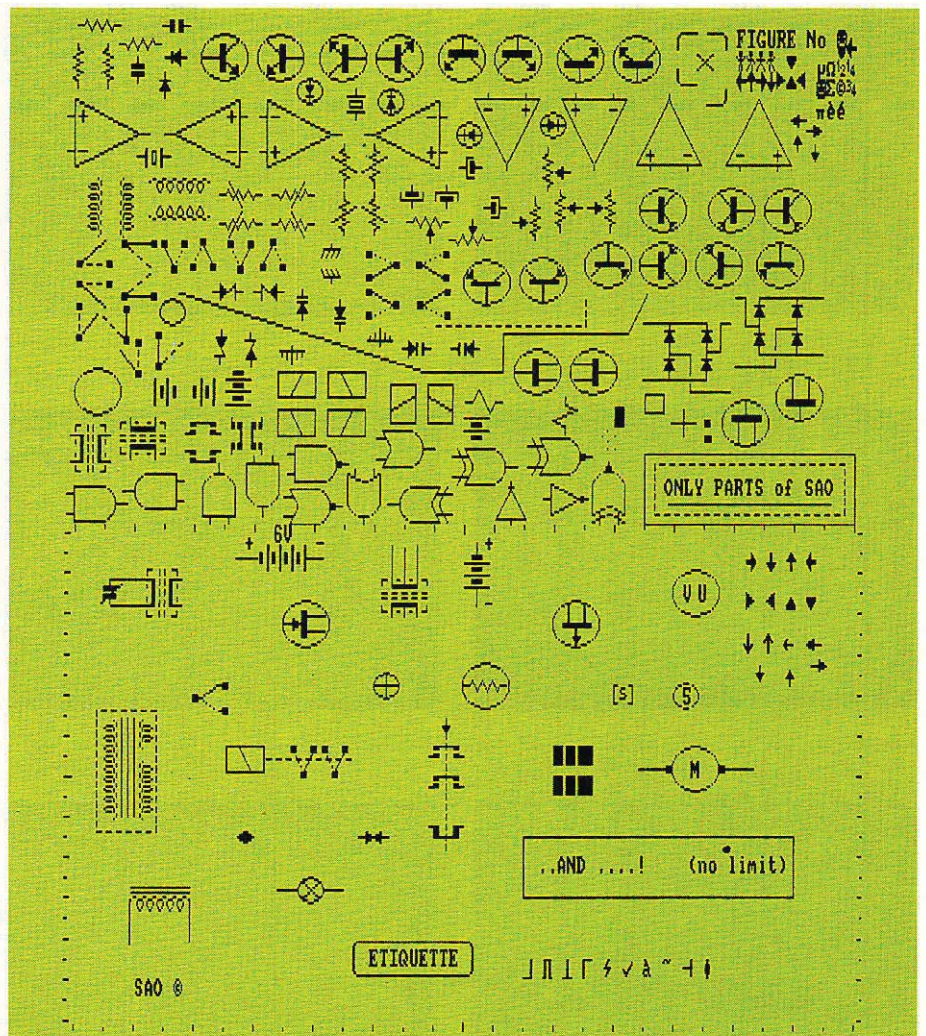
Les fonctions « CHARGER » et « SAUVER » sont remplies par une routine binaire (le « CALL inout ! » en lignes 570 et 600) et « ERASER » s'accommode de toutes les fantaisies. Autrement dit, le programme ne plantera pas lors des opérations de sauvegarde, chargement et effacement.

Dans tous les cas, ces quatre modules aboutissent à la ligne 640 qui nous renvoie au MENU général.

— IMPRESSION / NOUVEAU / QUITTER (lignes 650 à 790). Là encore, beaucoup de points communs à ces trois sous-programmes : affichage d'une fenêtre (ligne 660 et GOSUB 810) demandant une confirmation de la commande. Action, puis retour au MENU général.

L'impression utilise le fichier binaire « COPY-7 ou 8 B.SCH » par le « CALL hardcopy ! » de la ligne 700.

L'option « NOUVEAU » efface le dessin en cours, en remettant à zéro les octets de ZONE 1 (la



mémoire-écran) par un « MODE 2 ». Ensuite, ceux-ci sont recopiés dans ZONE 2 (la mémoire-tampon) par la routine « trans ! » (voir ligne 750).

Enfin, « QUITTER » sort du programme en « laissant les lieux dans l'état où il les a trouvés en entrant ». Le « CALL 0 » en ligne 790 équivaut en effet à un RESET.

Voici enfin quelques recommandations pour la saisie :

Certaines variables se terminent par un point d'exclamation (!). Celui-ci sert à les définir comme réelles, les autres étant entières. Ne l'oubliez surtout pas !

Accordez une grande attention aux noms des variables en général, les erreurs pouvant en résulter étant très difficiles à localiser.

N'hésitez pas à prendre votre temps pour recopier les fichiers publiés ce mois-ci : de fréquentes sauvegardes temporaires valent mieux que de multiples erreurs de saisie, dues aux yeux rouges

par la psychose du « syntax error ». Nous déclinons toute responsabilité en cas d'internement prolongé !

Conclusion

Vous devrez attendre le mois prochain pour essayer, et voir enfin fonctionner « SAO ». Croyez-le, nous le déplorons, mais cela aurait nécessité d'inclure ici, des pages destinées au prochain article : dessin de composants, phase d'essai, et surtout, le mode d'emploi. Vous comprendrez sans doute qu'il eût été injuste d'envahir RADIO-PLANS avec un sujet unique, si intéressant soit-il...

Nous espérons sincèrement que cette frustration insoutenable ne vous poussera pas à la dernière extrémité, et souhaitons vivement vous retrouver le mois prochain, pour le second et dernier épisode de cette palpitante série...

Jean ALARY et Alain CAPO