

LE MICRO-A ECRAN

**CANON X 07 :
BRANCHEZ VOTRE MICRO-ORDINATEUR
SUR VOTRE TELEVISEUR.**

IMPRESSIONNANT, LE CANON X 07 POUR UN MICRO-PORTABLE ! UNE INTERFACE OPTIONNELLE VOUS PERMET DE LE BRANCHER SUR VOTRE TELEVISEUR ET DE VISUALISER AINSI TOUTES LES OPERATIONS INSCRITES SUR VOTRE X 07.

MAIS LE CANON X 07 N'EST PAS SEULEMENT LE PREMIER MICRO-PORTABLE A ECRAN, IL EST AUSSI LE PREMIER MICRO-MULTICARTES.

SA FORCE ? DES PETITES CARTES EXTRAORDINAIRES POUR REALISER ET CONSERVER VOS PROPRES PROGRAMMES, COMME VOUS L'ENTENDEZ... A LA CARTE.

PRATIQUE, IL PARLE EN BASIC, LE LANGAGE ORDINATEUR FACILE A APPRENDRE.

AVEC SES NOMBREUSES CASSETTES ET CARTES A PROGRAMMES AUSSI ELABORES QUE LA GESTION DE STOCK, LA FACTURATION, LA PAYE, LE TABLEUR,... CANON X 07 A EGALEMENT BIEN D'AUTRES ATOUTS.

GRACE A SES MULTIBRANCHEMENTS : MACHINE A ECRIRE, IMPRIMANTE, ORDINATEUR, MODEM ET MEME VOTRE TELEVISEUR... CE TOUT PETIT ORDINATEUR A TROUVE PLUS D'UN MOYEN POUR DEVENIR GRAND.

JE SOUHAITERAIS RECEVOIR VOTRE DOCUMENTATION COMPLETE SUR LE MICRO-ORDINATEUR X 07.

VOICI MON NOM, MON ADRESSE ET MON TELEPHONE :

NOM _____

SOCIETE _____

N° _____ RUE _____

VILLE _____

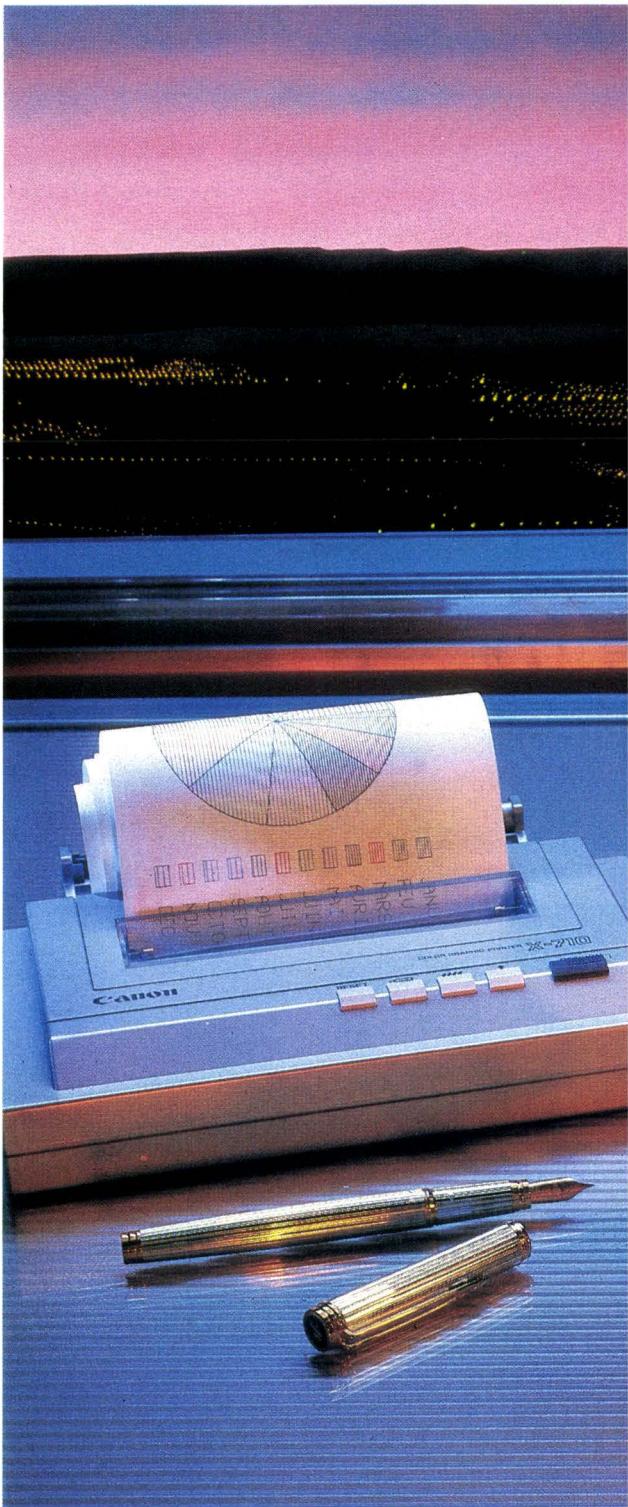
CODE POSTAL _____ TELEPHONE _____

**DEMANDE D'INFORMATION A RENVOYER A CANON FRANCE,
93154 LE BLANC-MESNIL CEDEX, TELEPHONE 865.42.23.**

Canon

CANON, HAUTE TECHNICITE, HAUTE SIMPLICITE

SERVICE-LECTEURS N° 85



PORTABLE RAN.



Synthèse vocale par phonèmes

Nous avons décrit dans Micro-Systèmes n° 45 (septembre 1984) la réalisation d'un synthétiseur de parole pour le micro-ordinateur Oric, basé sur le circuit MEA 8000 et utilisant un vocabulaire limité (57 mots ou expressions).

Afin de donner de nouvelles possibilités à cette réalisation, nous proposons à ceux d'entre vous l'ayant construite (et à ceux qui ne l'ont pas encore fait...) de lui donner maintenant un vocabulaire illimité, en français, par l'utilisation de phonèmes.

Avant de décrire le logiciel et la liste des phonèmes codés pour le MEA 8000, il est nécessaire d'expliquer brièvement ce que l'on entend par « phonèmes » auxquels il est souvent fait référence et dont la signification n'est peut-être pas claire pour tous.

Toute langue parlée utilise un nombre limité de sons élémentaires qui, associés les uns aux autres en séquence forment les mots composant le vocabulaire de cette langue.

On appelle « phonèmes » ces sons élémentaires (au nombre d'une trentaine environ pour la plupart des langues occidentales).

La liste des phonèmes diffère sensiblement d'une langue à l'autre, même si l'on retrouve des parties communes relativement importantes.

Ceci explique que les réalisations utilisant des phonèmes anglo-américains donnent en français des résultats très décevants, et souvent inintelligibles à une personne non habituée.

Phonèmes et logiciels associés

Nous vous proposons ici une liste de 31 phonèmes français augmentée de 7 sons qui, s'ils ne sont pas à proprement parler des phonèmes, permettent d'accroître notamment les performances de la synthèse phonétique proposée ici.

Si l'on ajoute 2 silences (32 et 64 ms), la liste de sons ici en comporte en tout 40.

L'encombrement total (table d'adresses comprise) est de moins d'1 Ko (kilo-octet), ce qui permettra une entrée manuelle au clavier moins fastidieuse.

Dieuse que celle du vocabulaire proposé dans le précédent article.

Le tableau fourni en annexe donne la liste des phonèmes, leur numéro d'ordre afin de les adresser au moyen de la table d'adresses ainsi que le symbole choisi pour les représenter.

Les codes des phonèmes se trouvent implantés en mémoire entre les adresses #A000 et #A3FF.

Ils sont suivis, à l'adresse #A700 d'une routine en langage machine autorisant la prononciation par le synthétiseur d'une expression quelconque. (*)

Cette routine est différente (un peu plus compliquée) de celle utilisée dans l'article précédent, car ici l'expression peut avoir une longueur supérieure à 255 octets, ce qui nécessite la prise en compte des deux octets (poids fort et faible) indiquant sa longueur.

L'ensemble des codes et de la routine en langage machine a été baptisé « Phonespeech » (fig. 1).

En ce qui concerne la mise en œuvre de ces phonèmes, l'idéal serait d'avoir une véritable synthèse à partir du texte écrit qui transcrirait directement le texte en paroles.

Bien que cela soit tout à fait possible (avis aux amateurs...), nous n'avons pas retenu cette voie, pour les principales raisons ci-dessous :

- une véritable synthèse à partir du texte, pour applications professionnelles, nécessite un niveau de performance inaccessible à la synthèse par phonèmes, mais seulement à la synthèse par diphones (association de la fin et du début de deux

APPLICATION : Synthèse vocale par phonèmes d'Hervé BENOIT

Notre réalisation « synthèse vocale sur Oric » vous a beaucoup plu. Toutefois, son vocabulaire apparaît limité (quelques dizaines de mots). Avec ce programme, Oric vous dit tout !

Ordinateurs : Oric 1 ou Atmos

Langages : Basic + langage machine 6502

« ilf& bo ojwrdui »
l'espace entre chaque mot n'est pas obligatoire : il introduit une pause de 32 ms. (le signe « - » produit le même effet)

Une pause plus longue (64 ms) est obtenue au moyen du point. (le signe « = » produit le même effet)

Il est d'autre part possible de prolonger la durée de prononciation d'un phonème en le faisant suivre du signe « + », ce qui augmente d'une trame la durée du phonème. (**)

Enfin, la ponctuation finale permet la création d'une intonation artificielle ou d'une expression chuchotée. (voir encadré 1).

L'expression ainsi définie par l'utilisateur est « fabriquée par le logiciel Phonetoric (fig. 2) dès la détection de l'appui sur RETURN, à partir des phonèmes de la liste, selon le format utilisé par le système de codage de vocabulaire (voir Micro-Systèmes n° 45 ; sa prononciation intervient dès que cette opération est terminée).

L'expression débute à l'adresse #A800 et peut avoir une longueur quelconque mis à part deux limitations propres à l'Oric :

- le nombre de caractères consécutifs ne peut dépasser 80 (2 lignes d'écran) ;
- la mémoire n'est disponible que jusqu'à l'adresse #B3FF.

(*) il est naturellement possible de reloger ce programme à l'adresse #A400 pour une plus grande capacité en mémoire ; #A700 avait été choisi en vue de pouvoir stocker une « vocationne » plus importante...

(**) il est possible d'utiliser plusieurs + à la suite (autant de trames sont ajoutées).

Nu-méro	Son	Exemple ou variantes	Représen-tation
0	A		a
1	E		e
2	I		i
3	ô	(<u>Bateau</u> , ô, oh)	o
4	U		u
5	è(AI, EI, E, ET)		&
6	é		*
7	EU		E
8	OU		w
9	AN (EN)		A
10	IN(UN, AIN, EIN)		I
11	ON		O
12	OI		W
13	B		b
14	D		d
15	F		f
16	G	(<u>Guerre</u>)	g
17	J	(<u>Jatte, gîte</u>)	j
18	K	(<u>Car, quatre</u>)	k
19	L		l
20	M		m
21	N		n
22	P		p
23	R1	(<u>Route</u>)	R
24	R2	(<u>Bonjour</u>)	r
25	S	(<u>Sauce</u>)	s
26	T		t
27	V		v
28	Z	(<u>Zôé, Sosie</u>)	z
29	CH	(<u>Charge</u>)	\$
30	GN	(<u>Montagne</u>)	N
31	AIL	(<u>Travail</u>)	ω
32	EIL	(<u>Vermeil</u>)	#
33	EU1L	(<u>Eil, feuille</u>)	%
34	IEN	(<u>Rien</u>)	<
35	OIN	(<u>Foin</u>)	>
36	o	(<u>Fort</u>)	Ø
37	S(bref)		non utilisé
38	Silence 32 ms		espace ou -
39	Silence 64 ms		point ou =

ponctuation

intonation plate	:
affirmation 1	\
affirmation 2	/
interrogation 1	?
interrogation 2	/
chuchotement	,
prolongation phonème	+

Tableau : liste des phonèmes

Cela permet néanmoins la composition de phrases d'une durée respectable, dont les codes peuvent être sauvegardés sur cassette en vue de leur utilisation dans un programme d'application, tel un jeu, par exemple. On pourra bien sûr les reloger ailleurs dans la mémoire, notamment si l'on désire utiliser le graphique haute résolution simultanément avec la synthèse de parole (***)

Conclusion

Grâce au vocabulaire illimité, le logiciel proposé ici donnera tout son intérêt à la réalisation de la carte synthétiseur que beaucoup d'entre vous ont déjà entreprise.

Il est toutefois certain que la qualité de parole obtenue ici ne sera pas la même que celle obtenue à partir de mots ou de phrases préenregistrées, surtout en ce qui concerne le naturel de l'élocution (prosodie et intonation).

Cependant, le fait de disposer d'un vocabulaire illimité permet, pour des applications micro-informatiques domestiques, d'accepter cet inconvénient.

(***) Le logiciel en BASIC « Phonetoric » peut d'autre part être utilisé (après quelques modifications mineures) avec le vocabulaire « VOCASPEECH » pour créer et sauvegarder des phrases à partir des mots du vocabulaire publié dans Micro Systèmes n° 45.

Détails de fonctionnement du logiciel « Phonetoric »

- lignes 2 à 15 = initialisation/présentation

- ligne 16 = entrée de la phrase par l'utilisateur

- lignes 17 à 19 = préparation de l'en-tête et initialisation expression.

- lignes 20 à 89 = lecture et transfert d'un phonème

 lignes 20 à 40 = évolution du phonème

 lignes 60 à 72 = détermination de sa position en mémoire

 lignes 84 à 89 = concaténation des phonèmes

- lignes 90 à 92 = longueur de l'expression résultante

- lignes 93 à 97 = évaluation de la ponctuation finale, et appel du sous programme d'intonation correspondant

- ligne 99 = prononciation de l'expression créée

- lignes 100 à 130 = choix de la suite donnée par l'utilisateur

- lignes 140 à 160 = correspondance caractère/n° du phonème

- lignes 200 à 485 = sous-programmes d'intonation

- lignes 500 à 540 = sauvegarde sur cassette de l'expression résultante

- lignes 600 à 670 = sous programme d'allongement de la durée d'un phonème.

En ce qui concerne l'intonation, nous rappelons qu'elle

est définie (en première approximation) par l'allure de la courbe d'évolution du « pitch ».

Le signe « » crée une courbe descendante (avec un palier plat pour les expressions longues).

Le signe « / » crée une courbe montante, inverse de la précédente.

Le signe « ! » crée une courbe montante, puis descendante (avec un palier plat pour les expressions longues).

Le signe « ? » crée la courbe inverse de la précédente.

Le signe « ' » provoque le non-voisement de toutes les trames (chuchotement).

Le signe « . » ne donne pas d'intonation au message en laissant le pitch constant tout au long du message (voix de robot), ce qui est le cas pour chaque phonème codé ici.

Ces « manipulations » sur l'intonation se font par modification des bits définissant la variation du « pitch » (voir article précédent), qui sont normalement à zéro pour les phonèmes utilisés ici.

Il est naturellement possible de faire des manipulations plus « fines », mais cela demande du temps et des essais successifs pour optimiser un message particulier.

ERRATUM

Lors de la publication de notre réalisation « synthèse vocale sur Oric », certaines erreurs (mineures) se sont glissées. Nous vous proposons ici les corrections à apporter pour remettre les choses en ordre :

1 - Programme Basic « Oric Bavarde » p. 255. Ligne 10 : PRINT « composez votre phrase (Nbr de chiffres séparés par espace) » : PRINT.

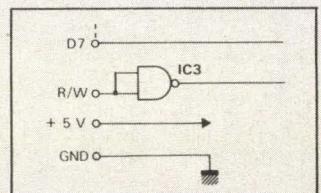
2 - Programme Basic « Chargeur Hexa » p. 257. Ligne 20 : FOR R = #A000 TO #B040 STEP 8.

3 - Liste hexadécimale pp. 258.259. En ce qui concerne la remarque au bas de cette page, il faut ajouter qu'une erreur dans l'en-tête de chaque expression (4 premières trames) peut être aussi « dramatique » que sur la table d'adresses, surtout si elle concerne l'octet faible de la longueur de l'expression.

4 - Liste de vocabulaire p. 256. Le mot n° 01 est UNE et non pas UN.

Enfin, sur certains Oric, il peut se faire qu'en raison de tolérances sur le timing (particulièrement sur les tous premiers matériels) des signaux 02 et R/W, le schéma proposé ne donne pas toute satisfaction.

La solution à ce problème est très simple : elle consiste à ne plus appliquer au 7400 le signal 02 en entrée mais à réunir les deux entrées de la porte au signal R/W (schéma)



```

2 REM***** (C) H.BENOIT 1984 *****
4 PRINTCHR$(20)
5 D$="#A000"
6 D=VAL(LEFT$(D$, 3))
7 DB=VAL(D$)
8 CLS:INK2:PAPER0
9 PRINT:PRINTCHR$(4);CHR$(27);J
  PHONETRIC";CHR$(4):PRINT
:PRINT
10 PRINT"COMPOSEZ VOTRE EXPRESSION AU MO
YEN DESSYMBOLES PHONETIQUES
":PRINT
11 PRINT"æ,e,i,o,u,&ai],*[eh],E[eu],w[œ
œ]" :PRINT
12 PRINT"A[an],I[in],O[on],W[œi],<[ien],
>[œin]" :PRINT
13 PRINT"b,d,g,j,k[c,q],l,m,n,p,R,r,s,t,
v,z" :PRINT
14 PRINT"@[ai],#[eil],x[œuil],-[SIL1];=
(SIL2)" :PRINT
15 PRINT"PONCTUATION OBLIGATOIRE . , ! ?
~ / " :PRINT:PRINT
16 INPUTL$:L=LEN(L$)
17 POKE#A800,0:REM-EN TETE MOT RESUL.
18 POKE#A803,80
19 C=4
20 FOR I=1TO1-1
25 A$ =MID$(L$, I, 1)
27 IFA$=". "THEN A$="="
28 IFA$=" " THEN A$="-"
29 IFA$="+" THEN 600
30 READB$,M
40 IF B$<>A$ THEN 30
60 A=D+(PEEK(DB+2*M)):REM-HH ADR.SON
70 B=PEEK(DB+2*M+1):REM-LL ADR.SON
72 K=PEEK(256*A+B+1):REM-LONG.SON
84 FORJ=4TOK:REM-TRANSFERT CODE VOCAL
86 N=PEEK(256*A+B+J):REM-PHONEME->
87 POKE#A800+C+J-4,N:REM-->EXPRESSION
88 NEXTJ :RESTORE:C=C+K-4
89 NEXTI :PRINT
90 T=C+4:REM-T=LÖNG.EXPRESSION
91 POKE#A800,INT(T/256)
92 POKE#A801,T-256*INT(T/256)
93 IFRIGHT$(L$, 1)="\" THEN GOSUB220
94 IFRIGHT$(L$, 1)="/" THEN GOSUB220
95 IFRIGHT$(L$, 1)="!" THEN GOSUB200
96 IFRIGHT$(L$, 1)="?" THEN GOSUB250
97 IFRIGHT$(L$, 1)=";" THEN GOSUB300
99 POKE#65,168:POKE#64,0:CALL#A700
100 PRINT"PRESSER 0 POUR TERMINER"
102 PRINT"PRESSER 1 POUR SAUVEGARDER"
105 PRINT"ESPACE POUR REPETER"
106 PRINT"AUTRES TOUCHES POUR RECOMMENCE
R"
110 GETE$:IFE$=" " THEN 99
115 IFE$="1" THEN 500
120 IFE$="0" THEN PRINTCHR$(20):END
130 PRINT:RESTORE:GOTO8
140 DATAa,0,e,1,i,2,o,3,u,4,&,5,*,6,E,7,
w,8,A,9,I,10,O,11,W,12,b,13
,d,14,f,15
150 DATAg,16,J,17,k,18,l,19,m,20,n,21,p,
22,R,23,t,24,s,25,t,26,y,27

```

```

,z,28,$,29
160 DATA N,30,0,31,#,32,z,33,<,34,>,35,0
,36,-,38,=,39
170 END
200 POKE#A803,64:FORQ=#A800+T-1TO#A800+T
/2STEP-4*INT(T/48)
201 GOSUB405:NEXTQ
202 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)

203 GOSUB455:NEXTQ:RETURN
220 POKE#A803,64:FORQ=#A800+T-1TO#A800+T
/2STEP-4*INT(T/48)
221 GOSUB405:NEXTQ
222 r .Q=#A807TO#A800+T/2STEP4*INT(T/24)
223 JSUB405:NEXTQ:RETURN
250 .QKE#A803,56:FORQ=#A800+T-1TO#A800+T
/2STEP-4*INT(T/48)
251 GOSUB455:NEXTQ
252 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)
253 GOSUB405:NEXTQ:RETURN
270 POKE#A803,64:FORQ=#A800+T-1TO#A800+T
/2STEP-4*INT(T/48)
271 GOSUB455:NEXTQ
272 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)
273 GOSUB455:NEXTQ:RETURN
300 FORQ=#A807TO#A800+T-1STEP4
305 E$=HEX$(PEEK(Q))
307 G=VAL(LEFT$(E$,2))
310 R=VAL(RIGHT$(E$,1))
315 IFR=0ANDG>2*INT(G/2)THEN330
318 G=2*INT(.r/2)+1:E=16*G
325 POKEQ,E
330 NEXTQ
335 RETURN
405 E$=HEX$(PEEK(Q))
407 G=VAL(LEFT$(E$,2))
410 R=VAL(RIGHT$(E$,1))
415 IFR=0ANDG>2*INT(G/2)THEN435
418 G=2*INT(G/2)+1:E=16*G+15
425 POKEQ,E
435 RET
55 E$=HEX$(PEEK(Q))
457 G=VAL(LEFT$(E$,2))
460 R=VAL(RIGHT$(E$,1))
465 IFR=0ANDG>2*INT(G/2)THEN485
468 G=2*INT(G/2) :E=16*G+1
475 POKEQ,E
485 RETURN
500 CLS:PRINT:PRINTCHR$(20)
510 PRINT"ENREGISTRER PRET?":PRINT
520 INPUT"NOM DU FICHIER";NS
530 CSAUVENS,A#A800,E#A800+T
540 END
600 FORJ=0T03
610 N=PEEK(#A800+C-J)
620 POKE(#A800+C+4-J),N
630 N=PEEK(#A800+C-J-4)
640 POKE(#A800+C-J),N
650 NEXTJ
660 C=C+4
670 GOT089
1000 CALL#F960
1010 LLIST
1020 CALL#F804

```

Fig. 1. – Listing des codes hexadécimaux de la table des phonèmes et de la routine de manipulation en langage machine.

PHONESPEECH-3. 3

_#A000 02 74 02 88 03 80 02 D8 605
 #A008 02 B0 02 C4 03 DC 03 08 610
 #A010 03 68 03 3C 03 54 02 EC 495
 #A018 03 1C 03 AC 03 94 00 7C 481
 #A020 00 8C 00 A4 00 BC 00 D4 704
 #A028 00 E4 00 F4 01 08 01 24 518
 #A030 03 F0 01 3C 03 C4 01 48 576
 #A038 01 5C 01 68 01 74 01 98 468
 #A040 01 C4 01 E8 02 1C 02 44 530
 #A048 02 9C 03 78 02 64 02 6C 493
 #A050 FF FF FF FF FF FF FF FF 2040
 #A058 FF FF FF FF FF FF FF FF 2040
 #A060 FF FF FF FF FF FF FF FF 2040
 #A068 FF FF FF FF FF FF FF FF 2040
 #A070 FF FF FF FF FF FF FF FF 2040
 #A078 FF FF FF 00 00 10 3C 3C 901
 #A080 16 B7 FD D0 16 B7 FE F0 1365
 #A088 16 B7 FE F0 00 18 3C 3C 843
 #A090 FF 97 60 00 FF 97 63 80 1135
 #A098 FF 97 65 00 FA 97 66 80 1138
 #A0A0 A6 97 66 A0 00 18 3C 3C 723
 #A0A8 1E BA 7B 30 1E BA 7D A0 888
 #A0B0 1E BA 76 80 1E BA 76 A0 1004
 #A0B8 1E BA 76 80 00 18 3C 3C 654
 #A0C0 09 97 88 10 0D 97 8F 90 763
 #A0C8 0D 97 8F 10 09 97 8E 90 769
 #A0D0 67 96 86 20 00 10 3C 3C 555
 #A0D8 74 B5 55 40 74 B6 2E 40 854
 #A0E0 28 B5 56 C0 00 10 3C 3C 635
 #A0E8 4C B4 5D 60 4H B4 5E E0 1017
 #A0F0 4A B4 5F A0 00 14 3C 3C 649
 #A0F8 88 B4 55 20 48 B4 5E 20 811
 #A100 4C B4 5F 20 4A B4 5F A0 892
 #A108 00 1C 3C 3C C2 B9 38 00 583
 #A110 11 B6 93 90 11 B6 97 90 984
 #A118 11 B6 97 10 16 B3 8E 18 725
 #A120 5B B2 8D A0 00 18 3C 3C 714
 #A128 3A B3 80 00 3A B3 84 00 734
 #A130 36 B3 8D A0 79 B2 96 20 1015
 #A133 97 B1 CD 20 00 0C 3C 3C 697
 #A140 0E F3 BC 70 0E F3 8D 70 1067
 #A148 00 14 3C 3C 6A B2 85 40 621
 #A150 6A B2 86 C0 6A B2 86 D0 1236
 #A158 6A B2 86 B0 00 0C 3C 3C 726
 #A160 12 F5 77 40 AA B4 76 C0 1106
 #A168 00 0C 3C 3C 29 BA 8E F0 741
 #A170 29 BA 8E F0 00 24 3C 3C 765
 #A178 98 D7 5D C0 98 D7 5D A0 1272
 #A180 99 D6 55 20 8E D6 5C 20 964
 #A188 F8 D8 5E 40 AF D8 56 E0 1218
 #A190 7F D8 5F 40 6B B7 6F C0 1095
 #A198 00 2C 3C 3C 57 B3 DD C0 843
 #A1A0 46 B4 E7 C0 46 B4 DF C0 1338
 #A1A8 AB B4 CF C0 EA B5 C7 40 1428
 #A1B0 FA B6 B6 C0 BA B7 9E 40 1397
 #A1B8 BA D8 7D 40 FA D8 73 40 1236
 #A1C0 FA D8 70 20 00 24 3C 3C 766
 #A1C8 BB B7 B6 40 FB B7 AF 40 1289
 #A1D0 FB B7 AF C0 BB B7 9F E0 1554
 #A1D8 B6 B8 9F 20 BA D8 8F 40 1166
 #A1E0 BA D8 7E C0 FA D8 75 C0 1495
 #A1E8 00 34 3C 3C EB B3 AE 40 824
 #A1F0 97 B4 A7 40 97 B4 A7 C0 1252
 #A1F8 57 B4 A7 C0 67 B6 9F 40 1134
 #A200 77 B6 87 40 BB B7 7E C0 1188
 #A208 EB B7 7E A0 AB B8 7E 20 1217
 #A210 AB D8 76 20 6B D8 76 20 1010
 #A218 BA B7 8C C0 00 28 3C 3C 861
 #A220 7F D8 6D 40 7F D8 6E C0 1161
 #A228 BF D7 87 20 FB D7 8F 20 1214
 #A230 BB D7 9F A0 67 D6 BF E0 1453
 #A238 67 D5 BF 40 61 D5 BE 40 1135

#A240 61 D4 C5 40 00 20 3C 3C 722
 #A248 86 94 BD C3 36 96 C7 A0 1306
 #A250 42 B4 D7 A0 53 B4 CF C0 1283
 #A258 53 B5 CF E0 62 B5 C7 40 1237
 #A260 62 B5 C5 40 00 08 3C 3C 668
 #A268 86 B3 C8 40 00 08 3C 3C 705
 #A270 96 B2 C8 60 00 14 3C 3C 764
 #A278 86 B3 CD C0 86 B2 D6 C0 1428
 #A280 96 B2 CE C0 97 B1 CD C0 1451
 #A288 00 14 3C 3C AF B3 85 40 691
 #A290 AB B3 7E 40 AB B3 86 40 1088
 #A298 FF B3 85 40 00 14 3C 3C 771
 #A2A0 AD AF A5 40 AD AF H6 40 1155
 #A2A8 AD RD 96 40 F9 AD 8D 40 1187
 #A2B0 00 14 3C 3C B7 B7 5D 40 663
 #A2B8 B7 B7 66 C0 B7 B7 66 C0 1320
 #A2C0 B7 B7 5D 40 00 14 3C 3C 663
 #A2C8 A6 B6 B5 40 B6 B6 BF 40 1212
 #A2D0 B6 B6 B7 40 FA B7 A5 40 1273
 #A2D8 00 14 3C 3C C7 AC 84 C0 835
 #A2E0 C6 AB 85 40 C6 AB 85 40 1132
 #A2E8 D6 AB 84 C0 00 1C 3C 3C 857
 #A2F0 82 D1 A5 40 83 CF AE C0 1272
 #A2F8 C3 CE AF 40 C3 D1 9E C0 1394
 #A300 D2 D3 96 40 92 D2 95 40 1204
 #A308 00 14 3C 3C D6 B3 6E 40 707
 #A310 FA B4 67 60 F6 B4 65 C0 1349
 #A318 EA B4 65 C0 00 20 3C 3C 859
 #A320 5A 34 9C 40 5A 84 9D 40 885
 #A328 5A 8A B6 C0 87 AD C6 C0 1300
 #A330 47 B1 CE E0 97 B2 C6 40 1269
 #A338 9B B1 C4 40 00 18 3C 3C 736
 #A340 52 CA BD 40 97 CA BD C0 1271
 #A348 97 CA BD C0 97 CA BD C0 1468
 #A350 83 CA BD 40 00 14 3C 3C 726
 #A358 66 B4 BD C0 66 B5 BF 40 1201
 #A360 65 B5 C7 60 61 B5 D5 C0 1260
 #A368 00 10 3C 3C 9B HD 60 40 637
 #A370 DB AE 6D E0 9B HE 6D 40 1228
 #A378 00 08 3C 3C 0E F3 BC 70 685
 #A380 00 14 3C 3C AF DA 5D 40 690
 #A388 AF DA 5E 40 AF DA 66 40 1110
 #A390 AF DA 65 40 00 18 3C 3C 702
 #A398 80 B5 1B A0 10 D5 04 A0 889
 #A3A0 A0 D5 0C 20 40 D2 2C A0 895
 #A3A8 40 D9 5E 20 00 18 3C 3C 551
 #A3B0 80 D6 1A 20 90 B2 13 20 773
 #A3B8 A4 B2 13 20 A5 B1 24 20 803
 #A3C0 EA 91 75 A0 00 18 3C 3C 800
 #A3C8 B6 F6 90 10 B6 F6 90 10 1176
 #A3D0 B6 F6 96 90 B6 F6 96 10 1316
 #A3D8 9B B4 8D A0 00 14 3C 3C 776
 #A3E0 BB B8 85 C0 BB B8 87 40 1266
 #A3E8 BB B8 87 40 FB B8 85 C0 1330
 #A3F0 00 10 3C 3C F5 B0 A5 40 786
 #A3F8 B5 8F 83 C0 54 B0 39 C0 1156

 #A700 20 2D A7 20 35 A7 A2 00 658
 #A708 A0 03 20 3B A7 20 2D A7 665
 #A710 20 3B A7 20 3B A7 20 3B 607
 #A718 A7 20 3B A7 CC 01 A8 D0 1006
 #A720 EC EC 00 A8 30 E7 20 2D 996
 #A728 A7 20 35 A7 60 A9 80 CD 1017
 #A730 FF 03 D0 F9 60 A9 1A 8D 1147
 #A738 FF 03 60 B1 64 8D FE 03 1029
 #A740 C8 C0 00 D0 03 E8 E6 65 1166
 #A748 60 55 55 55 55 55 55 55 55 691
 #A750 55 55 55 55 55 55 55 55 55 680
 #A758 55 55 55 55 55 55 55 55 55 680
 #A760 55 55 55 55 55 55 55 55 55 680
 #A768 55 55 55 55 55 55 55 55 55 680
 #A770 55 55 55 55 55 55 55 55 55 680
 #A778 55 55 55 55 55 55 55 55 55 680

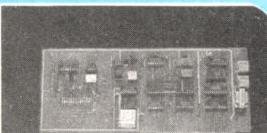
Fig. 2. – Le programme Phonetoric crée l'expression destinée à la routine de synthèse vocale.

MICRO-PÉRIPH — 62, rue Ducouédic - 75014
Paris. Tél. : 321.53.16 - Ouvert du mardi au samedi, de
10 h à 12 h et de 14 h à 18 h. (M° Mouton-Duvernet).

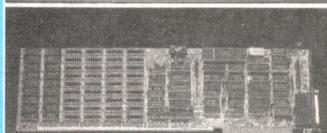
EXTENSIONS POUR IBM PC



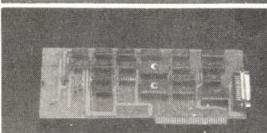
PC-020 CARTE MULTIFONCTION 256K 490F TTC
port série, port parallèle, horloge calendrier.



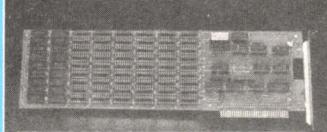
PC-015 CARTE MOPEM CUIIT V21 2690F TTC
300 baud full duplex multifonction



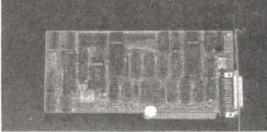
PC-018 CARTE MULTIFONCTION 384K 6650F TTC
port série et parallèle, horloge calendrier, sortie jeu



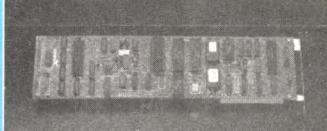
PC-019 CONVERTISSEUR A/D D/A 2090F TTC
D/A 12 bits, 1 voie, A/D 17 bits, 16 voies



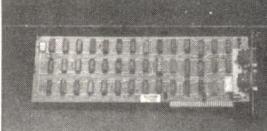
PC-014 CARTE D'EXTENSION MÉMOIRE 512K 7100F TTC



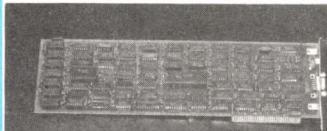
PC-010 I/O PLUS 4290F TTC
port série et parallèle, horloge calendrier, sortie jeu



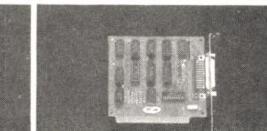
PC-024 ADAPTATEUR DISQUE DUR 500F TTC
adapt. jusqu'à deux Winchester



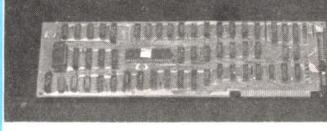
PC-029 X-NET CARTE DE NETWORK 9600F TTC
1MB/voe, 256 PCs



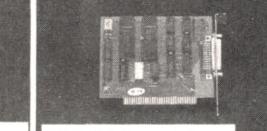
PC-025 CARTE GRAPHIQUE COULEUR 2150F TTC
couleur 320 x 200, NBD 640 x 200



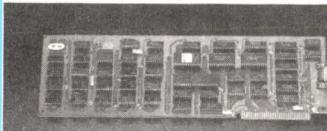
PC-026 CARTE IMPRIMANTE 650F TTC
port parallèle imprimante



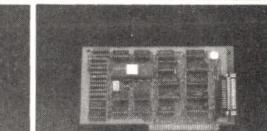
PC-020 CARTE COULEUR GRAPHIQUE AVEC SORTIE PARALLELE
même spécifications que PC-023 avec sortie parallèle



PC-024 DUAL RS-232 1540F TTC
deux ports RS-232C



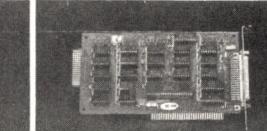
PC-025 CARTE MONOCHROME 1160F TTC
80 colonnes x 25 lignes en mode texte



PC-025 CARTE BUTTER IMPRIMANTS 2600F TTC
BUFFER DE 16 K



PC-16 CARTE MONOCHROME GRAPHIQUE AVEC SORTIE IMPRIMANTE 2660F TTC mode graphique 720 x 348 points

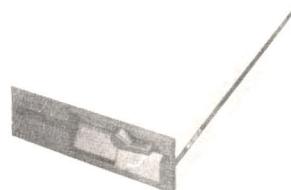


PC-057 CARTE CONTRÔLEUR DE FLOPPY 1420F TTC
pour lecteur de 4 têtes de lecture

NOUVEAU — LECTEUR DE DISQUE MULTITECH :

1 650 F T.T.C.

- compatible avec II+, //e et //c
- très silencieux, entraînement direct
- détecteur optique de piste zéro
- mécanique TEAC
- peut fonctionner en 40 pistes
- hauteur : 46 mm, poids : 1,95 kg seulement
- pour commander par correspondance, ajouter 35 F pour port et emballage spécial.



MULTITECH clavier pour Apple (1 515 F T.T.C.)

- 90 touches sur un clavier ergonomique et esthétique
- 12 touches de fonction programmables par l'utilisateur
- 10 touches de fonctions définissables par l'utilisateur
- 52 touches pour les commandes en Basic ou DOS
- cordon de 1,60 m
- LED pour "caps lock" et "num lock"
- parfaitement adapté pour l'Apple

CARTES 80 COLONNES POUR //e

80 colonnes pour //e avec supports pour étendre à 64 K

PRIX T.T.C.

850

1 400

EXTENSIONS POUR II+ ET //e

Carte Z80 compatible II+ et //e - PROMOTION

490

Carte 80 colonnes pour II+ avec caractères français minuscules, majuscules

750

Carte contrôleur de disques

420

Interface pour faire copie d'écran graphique Epson/Mannesman/Taxan

465

Carte entrée/sortie avec deux VIA 6522 pour gérer 32 lignes

755

CARTE HORLOGE avec logiciels

1 650

CARTE IEEE-488 PROGRAMMATEUR D'EPROMS (2716, 2732, 2764) avec logiciel incorporé

830

PROGRAMMATEUR DE HUIT EPROMS avec alimentation et interface Apple

3 850

CARTE 128 K avec logiciels de pseudo disk

1 850

CARTE CONVERTISSEUR ANALOGIQUE/NUMÉRIQUE sur 8 bits, 16 voies

1 085

CARTE CONVERTISSEUR A/D sur 12 bits, 8 ou 16 voies, gain programmable

5 880

CARTE INTERFACE IMPRIMANTE bufferisée 64 K avec copie d'écran graphique

1 645

CARTE GRAPPLER fonctions graphiques pour Epson, Mannesman, Taxan, etc.

650

CARTE DE BUFFER ET MULTIFONCTION pour clavier II+

780

CARTE INTERFACE microbuffer 32 K avec copie d'écran graphique

1 400

INTERFACE GRAPHIQUE pour Epson avec polices Script, Gothique, Romain, etc

2 000

INTERFACE SÉRIE avec deux portes de sortie extensibles à huit portes

1 960

CARTE Z80 avec 64 K qui tourne à 6 MHz

2 250

TRACKBALL pour II+, avec démultiplicateur des potentiomètres

390

TESTEUR DE CIRCUITS INTÉGRÉS pour TTL, HMOS, CMOS, etc.

1 150

CARTE MÈRE entièrement montée avec 6502, Z80, 64 K de RAM, sept slots

2 990

ALIMENTATION A DÉCOUPAGE 5 ampères, + 5 V et - 5 V, + 12 V et - 12 V

650

JOYSTICK de luxe avec micro-réglage des axes "x" et "y" pour II+ et //e

275

VENTILATEUR extérieur 220 V qui alimente l'Apple

320

APPAREIL pour encocher les disquettes avec trou carré et repère

90

Moniteur Zenith vert 12" anti-reflets

990

BUFFER microfazer 8 K extensible parallèle/parallèle

2 050

U-NSCRUMP dispositif de protection de logiciel par le hard

560

U-BCD convertisseur en BCD, jusqu'à 8 chiffres

1 120

WILDCARD II+ nouvelle version qui copie 128 K pour II+ et //e

1 400

MODÈNE BUZZ-BOX 300 bauds full duplex liaison RS 232

1 100

MODÈNE WS 2000 norme V 21 et V 23 CCITT et BELL

2 372

NOUVEAU - DEAMON 65C02 CMOS, II+ et //e pour tourner 4 fois plus vite

4 350

FINGER PRINT pour Epson

695

INTERFACE avec 16 K buffer en sortie série et parallèle

1 650

INTERFACE avec 64 K buffer en sortie série et parallèle

2 650

IMPRIMANTES

EPSON RX 80, car. français, 100 cps, impression italique ou romain

3 250

EPSON RX 80 friction/traction, 100 cps, impression italique ou romain

3 950

EPSON FX 80 F/T 160 cps, car. téléchargeables, espacement proportionnel

5 550

EPSON RX 100 F/T grand format, 100 cps

6 480

EPSON FX 100 F/T grand format, 160 cps, car. téléchargeables, etc.

7 200

EPSON LQ 100 parallèle avec matrice 17 x 24, 200 cps

6 450

NOUVEAU - Epson DX 100 marguerite 110 ou 132 colonnes, très silencieuse

14 165

EPSON LQ 100 parallèle avec matrice 17 x 24, 200 cps

1 450

NOUVEAU - Epson P 40 imprimante thermique

8 520

NOUVEAU - Epson JX 80 trois couleurs, 160 cps, car. téléchargeables

5 550

TAXAN KP 810, 140 cps

5 550

MANNESMAN MT 80, aiguilles carrées, ruban en mylar

3 550

LIVRES

CP/M user guide pour Apple

EMBER - Si vous possédez une carte

128 K, Applesoft s'étend à 160 K

MULTIPLAN EXPAND pour utiliser

votre carte 128 K avec Multiplan

SOFTTERM II émulation du terminal

sur

Apple

2 050

ASCII Express Professional

1 200

DATA CAPTURE logiciel de

790

communication pour //e

Z-TERM professional - logiciel de

1 500

communication en CP/M

750

MERLIN assembleur

MUNCH-A-BUG debugger pour

520

programmes en assembleur

ROUTINE MACHINE - Gestion de

590

routines ampersand en Applesoft

SPEED STAR - Compilateur de

programmes en BASIC

STRUCTURED BASIC - Un basic

1 050

amélioré qui ressemble au PASCAL

THE TEAM - Tableur, traitement de

920

texte, graphique et gestion de fichier

FAST DOS

290

FLIGHT SIMULATOR II - Simulateur

490

de vol très élaboré

28, bis rue de l'Est

92100 BOULOGNE

Commodore

rockwell

AIM 65 et AIM 65/40 (prix nous consulter)

Logiciels : Basic, PL/65, Forth, Assembleur, Pascal
Cartes d'extension : mémoire, CRT, RS232C, IEEE 1/0
parallèles, 1/0 Analog. Digit.

Double unité de disque AIM 65

En coffret câblé :

2 x 250 K 9800 F HT 2 X 500 K 10800 F HT

à monter en rack :

2 x 250 K 6800 F HT 2 x 500 K 7800 F HT

Logiciels et utilitaires disponibles sur disquette.

Moniteurs TAXAN

Vert, 12", H.R. 1350 F Couleur Vision I ou ex 3350 F

Ambre 1450 F Vision II 4150 F

Moniteur CIAEGI Vert 1050 F Ambre 1150 F

CIAEGI version OEM coffret métal - ZZMhz

SSV moniteur châssis ou coffret métal 9", 12", 15"

Terminal OEM 12" 3950 F HT - Terminal de table

5600 F HT - Claviers. Alimentations.

Programmateur de mémoires à partir de 8000 F

effaceur.



UC48K + 1 drive + moniteur

9450 F

UC64K + ROM trait text

5350 F

UC64K biprocesseur,

2 floppy intégrable, clavier

+ fonctions séparé, venti-

tilé 6200 F

Drive floppy 2350 F

Toutes cartes d'extension.

Quelques exemples composants (Tarif sur demande)

4116, 22 F - 4164, 78 F - 2114, 24 F - 6116, 98 F

2716, 45 F - 2732, 88 F - 2764, 165 F - 6502, 95 F

6802, 65 F - Z80A, 72 F - WD1771, 225 F - WD1795, 354 F

TTL 74LS - CMOS 4000 - Quartz - Supports - Connec-

teurs.

DISQUETTES 3 M 5" SF DD les dix: 220 F

C64 Pal 3000 F

C64 RVB 3900 F

Interface RVB 600 F

VIC 20 Pal 1690 F

C64 + mono disque + monit. coul. 9400 F

C64 + lecteur K7 + monit. vert 4500 F

VIC 20 + K7 + autoF. Basic 2200 F

SX64 Portable + 1 logiciel 13000 F

Comptabilité 64 1200 F

Traitement de texte 1150 F

Facturation 2200 F

Super Base 64 (base de données) 1190 F

Calc résult (tableau 32 pages) 990 F

Multiplan 1180 F

Interface centronix 560 F

Interface RS 232 C 345 F

Interface IEEE-488 950 F

Assembleur 64 350 F

Tool 64 550 F

Simon's Basic 950 F

Carte 80 colonnes 1650 F

Logo 1300 F

Clavier AZERTY 450 F

Koala Pad (table graphique + log.) 1350 F

Flight Simulator II 695 F

Fort Apocalypse 425 F

Star commando 165 F

Chop lifter 485 F

Jeep 145 F

Attack ou Revenge of the mutant 150 F

Olympic Skier 125 F

Solo Flight 225 F

Caverns of Khafka 190 F

CREDIT CREG ou leasing LOCABAIL

NOS PRIX SONT INDICATIFS ET PEUVENT

ETRE CHANGÉS SANS AVIS.

SERVICE-LECTEURS N° 161

MICROWAY VOUS OUvre LA VOIE
A L'INFORMATIQUE PERSONNELLE

Allcott Electronics

"LE MONT-FLEURI - AV. RIVIERA
06500 MENTON - TEL. (93) 28.39.00 - 35.27.72

SYSTEME COMPLET LIVRE AVEC :

- Boîtier CM Slimline en plastique ABS Spécial renforcé.
- 64 K RAM CM 4164 + 6502 Basic en mémoire.
- Z 80 intégré (CP/M)
- Carte 80 colonnes sur plot
- Lecteurs diskettes slim drive Teac originaux F.D. 55 A (250KB)
- Ventilateur intégré.
- Clavier détachable type IBM PC, avec 10 touches fonctions programmées et reprogrammables - PAD numérique et opérations réelles - Sonorité réglable, voyants tension/Min. Maj./ et appl. numérique.
- Moniteur 12" vert ou ambre anti reflets.
- Socle orientable.

SERVICE-LECTEURS N° 162



BON DE COMMANDE

Nom Prénom

Adresse Ville

Code Postal Tel

Date Signature

Quantité	Désignation	Prix unit. TTC	Prix total TTC

MODE DE REGLEMENT

- Contre-remboursement
 Mandat-letter joint

Port du

TOTAL



« MOTIF » sur Macintosh

Dans le manuel de référence du basic Microsoft, figure à plusieurs reprises dans l'appendice E « accès aux routines de la ROM du Macintosh » la variable **PATTERN%**().

Cette variable est conjointement employée avec l'instruction **VARPTR**. Les différentes valeurs de **PATTERN%(0)** jusqu'à **PATTERN%(3)** définissent le motif du fond de remplissage.

Le motif de remplissage est donc défini par quatre valeurs de la variable entière dimensionnée **PATTERN%()**, qui chacune est représentée en mémoire sur deux octets. Chaque bit étant une représentation de l'état de l'écran, la valeur de l'octet peut être définie dans une matrice comme l'indique la **figure 1**.

Le programme

La **figure 2** nous fournit l'organigramme du programme de la **figure 3** et sa simplicité montre la richesse des routines pré-programmées. En fait, le programme permet de dessiner un motif sur l'écran (au moyen de la souris). Au fur et à mesure que le dessin se forme, les valeurs à fournir pour les différents indices de **PATTERN%(0)** sont actualisées. Ainsi, lorsque le motif obtenu est correct, il suffit de reporter les valeurs affichées sur l'écran dans le programme l'exploitant.

Bien entendu, ce logiciel est adaptable à tout matériel autorisant la redéfinition de caractères (à l'utilisation de la souris près, toutefois). Pour utiliser le

programme, après avoir tapé le classique **RUN**, il suffit de se servir de la souris.

Le curseur, matérialisé par la flèche noire (**fig. 4 et 5**) est déplacé vers la zone d'action. Le rectangle blanc dans la zone gauche de l'écran d'accueil représente une matrice 8×8 points, dont chaque pixel sera allumé par un « click » lorsque le curseur le pointera.

Notons que la zone gauche de l'écran a pour fond une répétition du motif de base.

Lorsque le résultat est satisfaisant, l'utilisateur a plusieurs possibilités.

Tout d'abord, les nombres affichés à gauche donnent les valeurs en bases décimales ou hexadécimales à fournir à la fonction **PATTERN%** (**fig. 6 et 7**).

Si l'on désire imprimer l'écran affiché, il suffit, à l'aide de la souris, de placer le curseur sur le rectangle « recopie d'écran » et de « cliquer » deux fois. Enfin, lorsque la session est terminée, on place le curseur dans le rectangle « quitter » et on clique deux fois, ce qui ramène l'utilisateur au niveau programmation. ■

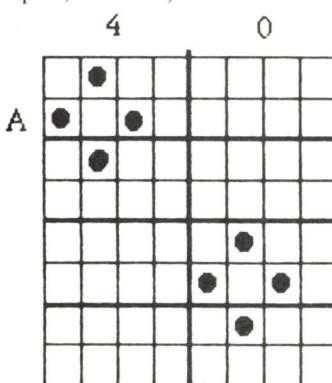


Fig. 1. – La fonction **PATTERN%**() permet de définir deux octets de fond dans une matrice de 8×8 points.

Ainsi, les indices de cette fonction vont-ils de 0 pour les deux octets du haut de la matrice jusqu'à trois pour les deux du bas, pour un caractère donné.

UTILITAIRE :
Un logiciel de création de motifs
de R. MAU

Le Macintosh se présente comme l'appareil le plus riche du marché au niveau graphique. Mais pour tirer au maximum profit de ses performances, il est nécessaire de bien connaître les routines de la ROM. Ce programme est une application de ses capacités.

**Langage : Basic Microsoft
Ordinateur : Macintosh d'Apple**

LES ROUTINES UTILISÉES DE LA ROM du MACINTOSH

• CALL TEXTFACE(n)

n = 1 caractères gras

n = 0 caractères normaux

• CALL TEXTSIZE(n)

n = taille des caractères

• CALL MOVETO(x,y)

positionne à un point déterminé de l'écran (x = abscisse, y = ordonnée)

• CALL FRAMEROUNIRECT(VARPTR(Cadre%(0)),X,Y)

dessine un rectangle arrondi aux angles, les valeurs de cadre() déterminent, la position du tracé, (haut, gauche, bas, droit) les valeurs de X et de Y, la forme de la courbe (**fig. a**)

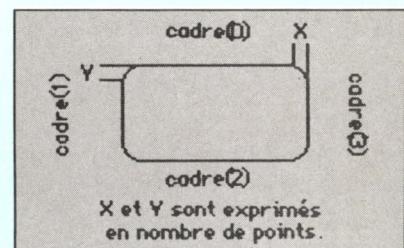


Fig. a. – Dans la fonction **FRADEROUNIRECT**, chaque indice du paramètre « cadre » correspond à un des côtés du rectangle à dessiner.

• CALL

PENSIZE(n,m)

n = largeur du trait

m = hauteur du trait

• CALL INVERTROUNIRECT(VARPTR(Cadre%(0)),X,Y)

inverse le contenu du rectangle arrondi aux angles.

• CALL FILLARC(VARPTR(Cadre%(0)),STARTANGLE,ARCANGLE VARPTR(MOTIF(0))) (fig. b)

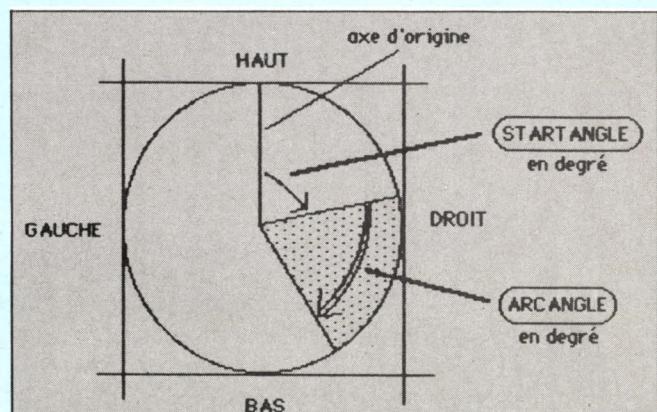


Fig. b. – Indication du rôle de chaque paramètre de la fonction **FILLARC**

```

8 REM ----- INITIALISATIONS -----
9  DEFINT A-Z toutes les variables sont entières
: CALL TEXTFACE(0) caractères normaux
10 DIM MOTIF(3), MASQUE(7,1), CADRE(15)
11 DATA -32768, 128, 16384, 64, 8192, 32, 4096, 16
   valeurs des différents masques en décimal
12 DATA 2048, 8, 1024, 4, 512, 2, 256, 1
13 FOR I=0 TO 7 lecture des data masques
: READ MASQUE(I,0)
: READ MASQUE(I,1)
: NEXT I

14 DATA 169, 329, 206, 445, 229, 329, 266, 445
   valeurs des limites des 2 cadres du menu normaux

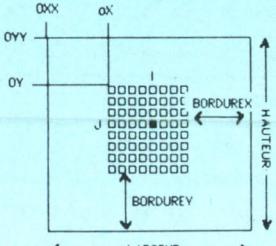
15 FOR I=0 TO 7
: READ CADRE(I) lecture des data cadres normaux
: NEXT I

16 DATA 168, 328, 207, 446, 228, 328, 267, 446
   valeurs des limites des 2 cadres du menu inversés

17 FOR I=8 TO 15
: READ CADRE(I) lecture des data cadres inversés
: NEXT I

55 REM _____ calcul des coordonnées du décor

60 OXX=10 : OYY=10
70 LARGEUR=280 : HAUTEUR=267
80 BORDUREX=(LARGEUR-81)/2 : BORDUREY=(HAUTEUR-81)/2
90 OX=BORDUREX+OXX : OY=BORDUREY+OYY



```

```

95 REM _____ dessin du décor
100 CLS
105 REM *** tracé de la grille
110 LINE (OXX,OYY)-STEP(LARGEUR,HAUTEUR),B
120 LINE (OX,OY)-STEP(81,81),B
125 REM *** tracé du tableau d'affichage
130 CALL TEXTSIZE(20)
140 CALL MOVETO(358,30) : PRINT "Valeurs"
150 CALL TEXTSIZE(9)
160 CALL MOVETO(350,46) : PRINT "décimal"
170 CALL MOVETO(412,46) : PRINT "hexadécimal"
180 CALL TEXTSIZE(9)
190 FOR I=0 TO 3
: CALL MOVETO(300, 70+(I*20))
: PRINT "MOTIF#",I,""
: NEXT I
   tracé du rectangle et de la ligne autour de "Valeurs"

200 LINE(300,10)-STEP(180,40),B
210 LINE(300,46)-STEP(180,0)
220 CALL TEXTSIZE(12)
230 CALL TEXTFACE(1) caractères gras
235 REM *** tracé du menu
240 CALL MOVETO(340,190) : PRINT "copie d'écran"
   tracé du cadre autour de "copie d'écran"
250 CALL FRAMEROUNDRECT(VARPTR(CADRE(0)),15,15)
260 CALL MOVETO(362,250) : PRINT "quitter"
   tracé du cadre autour de "quitter"
270 CALL FRAMEROUNDRECT(VARPTR(CADRE(4)),15,15)

500 REM ----- MENU -----
   (initialisations)
510 LASTI=-1:LASTJ=-1:LASTCOLOR=0

```

```

520 IF simple clic on va au sous programme "deccin du motif à l'aide de la souris"
   si la souris est dans la zone du menu "copie d'écran" et double clic alors sous programme de copie
530 IF ABS(MOUSE(1)-406)<66 AND ABS(MOUSE(2)-187)<17 AND
   MOUSE(0)=-2 THEN GOSUB 2000 : REM (double clic sur copie
   d'écran)

   si la souris est dans la zone du menu "quitter" et double clic alors fin du programme
540 IF ABS(MOUSE(1)-406)<66 AND ABS(MOUSE(2)-247)<17 AND
   MOUSE(0)=-2 THEN
      CALL INVERTROUNDDIRECT(VARPTR(CADRE(12)),15,15)
: END
550 GOTO 510 boucle

999 REM ----- DESSIN DU MOTIF A L'AIDE DE LA SOURIS -----
tant que la souris se trouve dans la zone de travail et que le bouton est enfoncé
1000 WHILE ABS( MOUSE(1)-(OX+40))<40 AND ABS( MOUSE(2)-(OY+40))<40
AND MOUSE(0)=-1

1005 REM _____ calcul des indices de la grille
1010 X=MOUSE(1)-OX : Y=MOUSE(2)-OY
1020 I=INT(X / 10) : J=INT(Y / 10)
   la souris pouvant sortir entre temps de la zone de travail il faut rectifier les indices
1030 IF J>8 THEN J=7
1040 IF I>8 THEN I=7
1050 IF I<0 THEN I=0
1060 IF J<0 THEN J=0
   calcul des coordonnées de référence
1070 X=OX+1+(10*I) : Y=OY+1+(10*)J

1075 REM _____ inversion de la couleur de la zone sélectionnée
   si le dernier carré sélectionné est le même on saute
1080 IF LASTI=I AND LASTJ=J THEN 1250 : REM( TEST PREALABLE )
1090 COLOR=POINT(X,Y) : REM( COLOR=33=>NOIR COLOR=30=>BLANC)
1100 IF LASTCOLOR=33 AND COLOR=30 THEN 1250
1110 IF COLOR=33 THEN COLOR=30 ELSE COLOR=33
1115 REM _____ affichage de la zone sélectionnée
1120 CALL PENSIZE(10,10) le point affiché aura la taille d'un carré
1130 PSET(X,Y),COLOR
1140 LASTI=I : LASTJ=J on mémorise la dernière position
1145 REM _____ calcul des valeurs de MOTIF( )
1150 V=J MOD 2
1160 MASK=MASQUE(I,V)
1170 U=INT(J/2)
1180 IF COLOR=33 THEN MOTIF(U)=MOTIF(U) OR MASK : ELSE
   MOTIF(U)=MOTIF(U) XOR MASK

1185 REM _____ affichage du motif autour de la grille
   ici le crayon ne sera pas noir mais comme le motif
1190 CALL PENPAT (VARPTR(MOTIF(0)))
   on détermine la largeur et l'épaisseur du crayon
1200 CALL PENSIZE(BORDUREX-2,BORDUREY-2)
   on trace un rectangle, ainsi son cadre sera constitué des motifs
1210 LINE(OXX+1,OYY+1)-STEP(LARGEUR-2,HAUTEUR-2),B

1215 REM _____ affichage des valeurs de motif( )
1220 FOR H=0 TO 3
: CALL MOVETO(366, 70+(H*20))
: PRINT MOTIF(H) valeur décimal de MOTIF()
: CALL MOVETO(444, 70+(H*20))
: PRINT HEX$(MOTIF(H)) valeur hexadécimal de MOTIF()
: NEXT H
: CALL PENNORMAL on rétablit le crayon normal
1230 LASTCOLOR=COLOR
1240 IF LASTCOLOR=33 THEN LASTCOLOR=30 : ELSE LASTCOLOR=33
1250 WEND
1260 RETURN

1999 REM ----- COPIE D'ECRAN -----
2000 CALL INVERTROUNDDIRECT(VARPTR (CADRE(8)),15,15)

```

copie d'écran

```

2010 LCOPY
2020 CALL INVERTROUNDDIRECT(VARPTR (CADRE(8)),15,15)

```

copie d'écran

2030 RETURN

Fig. 3. – Le listing du programme Basic. Notez la richesse des commentaires obtenus par le traitement de texte Mac Write.

218 – MICRO-SYSTEMES

Novembre 1984

Organisation générale du programme

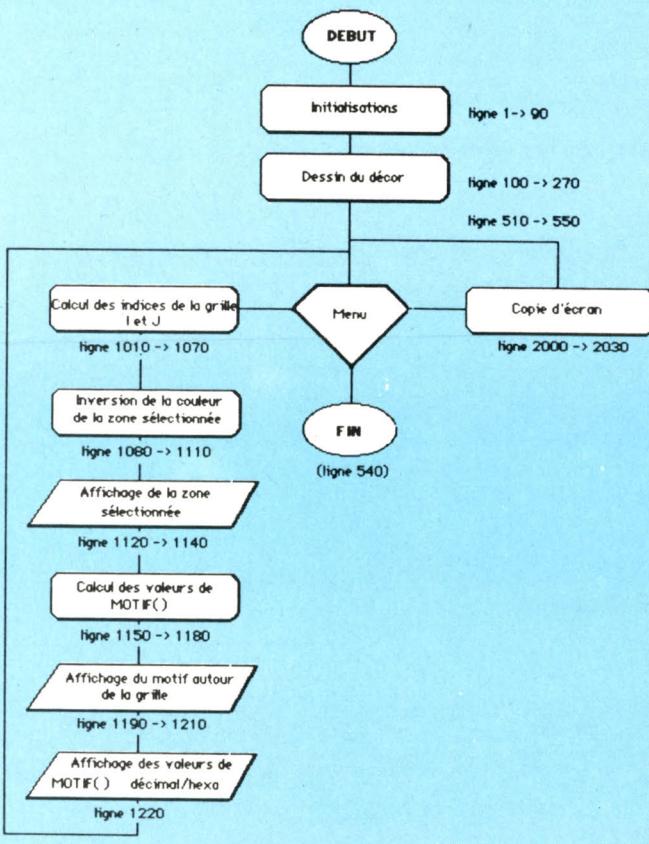


Fig. 2. – L'organigramme du programme MOTIF a été obtenu à partir d'un Macintosh.

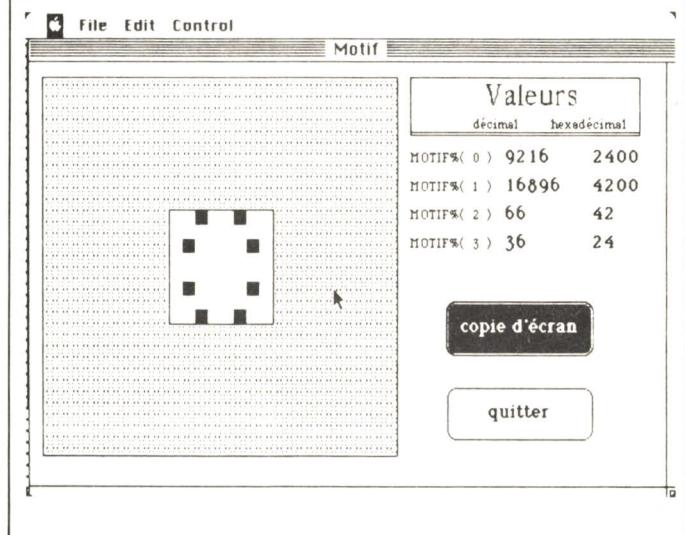
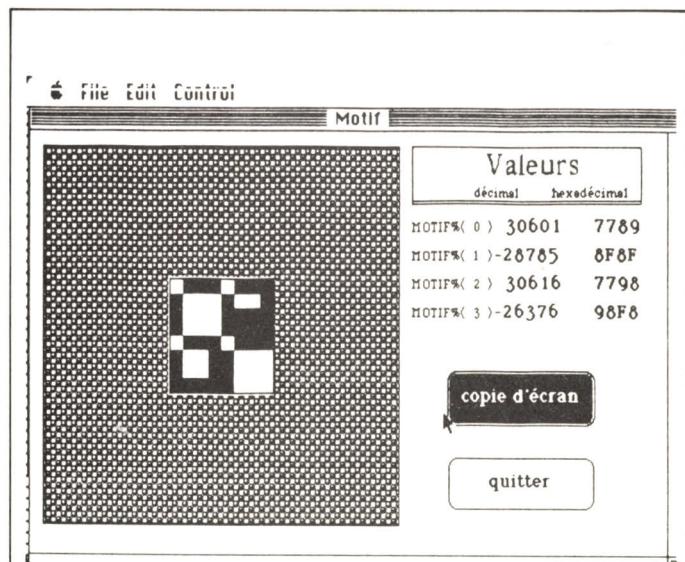


Fig. 4. – Un exemple d'affichage d'un motif sur le fond de travail.

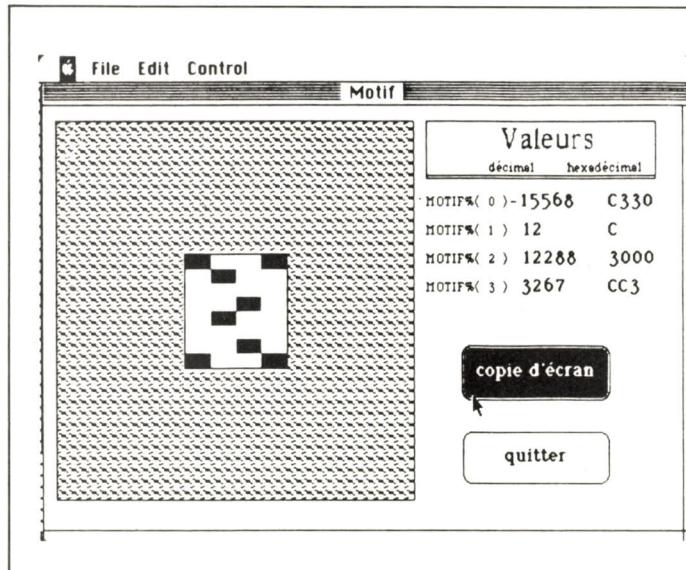


Fig. 5. – A l'aide de la souris, il est possible de gérer les menus : ici deux exemples de sortie de l'écran sur imprimante.

```

1 REM ****
2 REM           exemple d'application du programme MOTIF
3 REM                           creation d'un camembert
4 REM                               juillet 1984
5 REM                           par mau richard
6 REM ****
10 REM     valeurs hexa des motifs de fond
20 DIM PATTERN%(33)
30 DATA &H7789,&H8F8F,&H7798,&HF8F8,&H40A0,&H4000,&H040A,&H0400
40 DATA &h081C,&h3663,&hC183,&H60C,&h8,&h1422,&h4122,&h1408,&h210A,&hA401,&h5005,&h2844,&hC060,&
h3010,&h180C,&h603
50 DATA &hFFAE,&hFFAB,&hFFAB,&hFFFF,&hC330,&hC,&h3000,&hCC3
60 FOR J=0 TO 31
70   READ PATTERN%(J)
80 NEXT J
85 REM
90 REM phase d'initialisations
100 TOTALDEPENSE=0
110 DATA 2214.56,1080.52,867,1200,450,375,946.75,200
120 FOR I=0 TO 7:READ CUMULPOSTE(I)
130 TOTALDEPENSE=TOTALDEPENSE+CUMULPOSTE(I)
140 NEXT
145 REM
150 REM calcul des pourcentages
160 FOR I=0 TO 7
170 POURCENTAGE(I)=(CUMULPOSTE(I)/TOTALDEPENSE)*100
180 NEXT
185 REM
190 REM calcul de l'angle correspondant
200 FOR I=0 TO 7
210 ARCANGLE(I)=(360*POURCENTAGE(I))/100
220 NEXT
225 REM
230 REM dessin d'un camembert 8 portions
240 CLS
250 CIRCLE (190,140),100
260 RECTANGLE%(0)=40:RECTANGLE%(1)=90:RECTANGLE%(2)=240:RECTANGLE%(3)=290
270 STARTANGLE=0
280 FOR I=0 TO 7
290 CALL FILLARC(VARPTR(RECTANGLE%(0)),STARTANGLE,ARCANGLE(I),VARPTR(PATTERN%(I*4)))
300 STARTANGLE=STARTANGLE+ARCANGLE(I)
310 NEXT
320 END

```

Fig. 6. – Listing d'un programme utilisant les paramètres générés par le logiciel MOTIF.

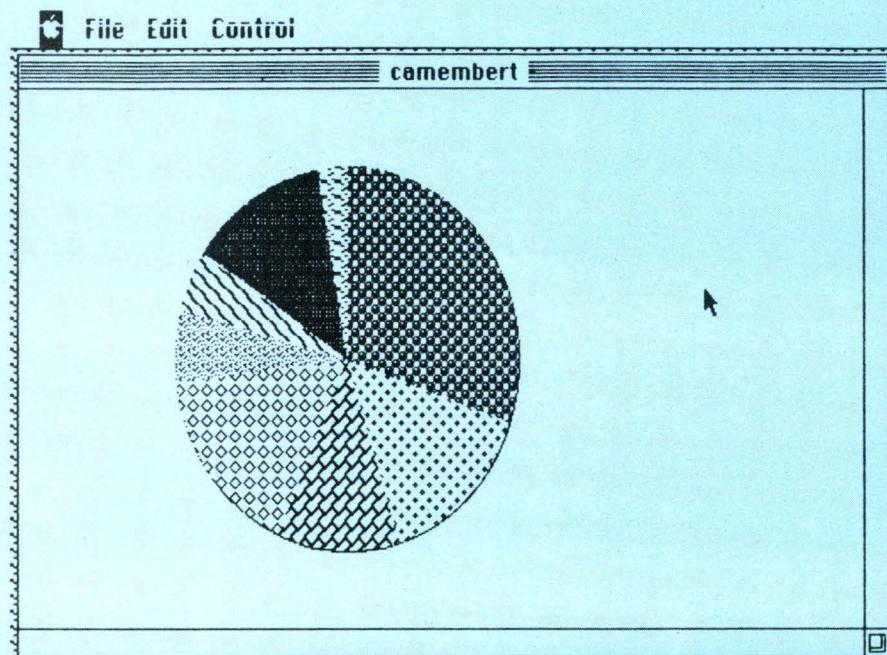


Fig. 7. – Dessin d'un camembert par le programme d'application.

Basic étendu pour Canon X-07

Il est très agréable d'avoir accès à une bibliothèque sonore préprogrammée. Dans l'élaboration et la pratique des jeux d'arcades, le son est, en effet, le complément indispensable du graphisme. Si le Canon X-07 est bien pourvu sur ce dernier point (instructions PSET, PRESET, LINE, CIRCLE...), il n'en est pas de même en ce qui concerne l'aspect sonore. L'utilisation de la seule instruction BEEP est loin de permettre l'exploitation complète des possibilités sonores de cet ordinateur. Pour pallier cette carence, nous vous proposons plusieurs instructions sonores, qui simulent les différents bruits dont peut avoir besoin un programmeur.

(Ce type d'instructions existe déjà sur certains micro-ordinateurs. On pensera par exemple aux ordres ZAP, PING, EXPLODE du Basic de l'Oric.)

Votre Canon dispose d'une instruction dont vous n'avez probablement jamais eu connaissance : PAINT. Son existence n'est en effet mentionnée dans aucun des trois manuels qui accompagnent l'ordinateur, et, toute tentative d'utilisation de celle-ci se solde par un échec (apparition systématique du message « SN Error »). L'explication de ce mystère est simple : chaque fois que l'interpréteur rencontre le mot clé « PAINT », il se branche à la routine d'erreur par l'intermédiaire de vecteurs situés en RAM. On peut se demander quel est l'intérêt d'une telle instruction. La réponse est fort simple : si le mot clé « PAINT » est inopérant dans la version de base du Canon X-07, il n'en est pas de même lorsque l'extension permettant l'utilisation d'un téléviseur est connectée. En effet, dans ce dernier cas, le contenu des vecteurs situés en RAM correspondant à l'instruction « PAINT » est modifié, et cette dernière retrouve alors son rôle classique, à savoir le « coloriage » d'une zone limitée. Toutefois, ce n'est pas dans ce but que nous allons l'exploiter.

Une liaison avec l'interpréteur

Pour pouvoir utiliser de nouvelles instructions, il faut bien sûr créer les routines ! Cependant, cela n'est pas suffisant : il est nécessaire d'élaborer une

structure autorisant l'insertion de ces nouvelles instructions dans un programme en Basic. Pour cela, les vecteurs situés en RAM correspondant au mot clé « PAINT » ne doivent plus pointer sur la routine d'erreur, mais contenir l'adresse de la routine en codes machines. Ainsi, l'utilisation de « PAINT » ne provoquera plus de message d'erreur mais l'exécution d'une routine en langage machine.

Par exemple, POKE 154, 158 : POKE 155, 206 provoque l'effacement de l'écran à chaque utilisation de « PAINT » (pour ce faire, l'adresse de l'instruction « CLS » a été « pokée » en 154 et 155, qui sont les vecteurs situés en RAM correspondant à PAINT). Il devient ainsi possible, à condition de respecter les servitudes liées à l'interpréteur, de créer une nouvelle instruction. C'est cette technique qui a été utilisée dans le programme que nous vous présentons.

La nécessité d'un « mini-interpréteur »

Les instructions du type de PAINT, c'est-à-dire non fonctionnelles dans la version de base du Canon, sont peu nombreuses ; or, une réelle amélioration de l'exploitation des possibilités sonores du Canon ne peut se faire que par le biais de l'ajout de plusieurs instructions nouvelles. Il est donc nécessaire de permettre plusieurs choix à

UTILITAIRE :
Une extension à votre Basic
d'A. RITOUX et E. SANDER
Ce programme, en mettant à votre disposition plusieurs instructions sonores, sera un aide précieux à la réalisation de jeux d'arcades. Outre cette application de vous permettra de personnaliser votre Basic en créant vos propres instructions.

Ordinateur : Canon X-07

Langage : Assembleur NSC-800
(compatible Z 80)

partir du même mot clé. PAINT n'est plus alors considéré comme une nouvelle instruction mais comme un préfixe indiquant la présence de celle-ci : son exécution provoque l'appel d'une routine qui a pour rôle d'interpréter l'instruction préfixée par PAINT et de se brancher en fonction d'une table d'adresses à la routine correspondante de votre choix.

Cette sélection s'effectue ici par une reconnaissance numérique : à chaque ordre correspond un numéro. Ainsi PAINT 1 désigne une fonction SOUND, PAINT 2 : H/P... Cette présentation, certes un peu moins lisible qu'une écriture en toutes lettres, présente plusieurs avantages. Il est beaucoup plus aisément de faire correspondre une adresse à un nom qu'à une chaîne alphanumérique. Ce programme ayant aussi pour

objet la création de nouvelles instructions, cette caractéristique n'est pas négligeable, car elle autorise diverses évolutions de la routine. De plus, une frappe des mots clés en toutes lettres peut se révéler astreignante à cause du préfixe : on conçoit assez aisément la lourdeur d'une expression du type « PAINTEXPLODE » par exemple si l'on désire générer le son d'une explosion.

Le programme

Les différents éléments nécessaires à une bonne compréhension de ce programme vous sont fournis ici. Tout d'abord un élément indispensable est mis à votre disposition : le dé-sassemblage du programme (fig. 1). Les adresses des principales routines et des tables de données vous sont indiquées dans la figure 2.

STRUCTURE INTERNE DU PROGRAMME

\$1CB0	Table d'adresses
\$1C00	Mini-interpréteur
\$1C26	DOKE
\$1C35	SOUND
\$1C46	H/P
\$1C5B	ZAP
\$1C86	PHONE
\$1CCD	MOTOR
\$1CF5	ZIP
\$1DA9	Initialisation
\$1DC0	STORM

Fig. 2. – Structure interne du programme de décodage de l'instruction PAINT.

1C00 CALL FE5E	1C5A RET	1CAA XOR a	1CFE POP bc
1C03 CP 20	1C5B CALL FE5E	1CAB OUT (F4),a	1cff LD c,a
1C05 JR C,1C0C 05	1C5E LD c,a	1CAD LD a,05	1D00 PUSH bc
1C07 LD e,05	1C5F PUSH bc	1CAF LD b,00	1D01 RST 8 2C ,
1C09 JP F1C7	1C60 RST 8 2C ,	1CB1 DJNZ 1CB1 FE	1D03 CALL FE5E
1C0C LD b,a	1C62 CALL FE5E	1CB3 DEC a	1D06 OUT (F3),a
1C0D RST 8 2C ,	1C65 POP bc	1CB4 JR NZ,1CAF F9	1D08 POP bc
1C0F SLA b	1C66 LD b,a	1CB6 DEC e	1D09 PUSH bc
1C11 LD d,00	1C67 PUSH bc	1CB7 JR NZ,1C9D E4	1D0A XOR a
1C13 LD e,b	1C68 RST 8 2C ,	1CB9 POP de	1D0B DEC a
1C14 PUSH IX	1C6A CALL FE5E	1CBA PUSH de	1D0C OUT (F4),a
1C16 LD IX,1BC0	1C6D OUT (F3),a	1CBB LD a,10	1D0E XOR a
1C1A ADD IX,de	1C6F POP bc	1CBD LD b,00	1D0F OUT (F2),a
1C1C LD e,(IX+0)	1C70 PUSH bc	1CBF DJNZ 1CBF FE	1D11 POP bc
1C1F LD d,(IX+1)	1C71 XOR a	1CC1 DEC a	1D12 PUSH bc
1C22 POP IX	1C72 CPL	1CC2 JR NZ,1CBD F9	1D13 DEC c
1C24 PUSH de	1C73 OUT (F4),a	1CC4 DEC e	1D14 JR NZ,1D13 FD
1C25 RET	1C75 XOR a	1CC5 JR NZ,1CBB F4	1D16 DEC a
1C26 CALL FFCC	1C76 OUT (F2),a	1CC7 POP de	1D17 JR NZ,1D0F F6
1C29 PUSH de	1C78 POP bc	1CC8 POP bc	1D19 POP bc
1C2A RST 8 2C ,	1C79 PUSH bc	1CC9 DJNZ 1C93 C8	1D1A DJNZ 1D09 ED
1C2C CALL FFCC	1C7A DJNZ 1C7A FE	1CCB POP hl	1D1C OUT (F4),a
1C2F EX (sp),hl	1C7C INC a	1CCC RET	1D1E RET
1C30 LD (hl),e	1C7D JR NZ,1C76 F7	1CCD CALL FE5E	1DA9 LD hl,1D9A
1C31 INC hl	1C7F POP bc	1CD0 LD d,a	1DAC LD (0045),hl
1C32 LD (hl),d	1C80 DEC c	1CD1 LD a,FF	1DAF RST 38 D8 "
1C33 POP hl	1C81 JR NZ,1C70 ED	1CD3 OUT (F4),a	1DB1 CALL 00A2
1C34 RET	1C83 OUT (F4),a	1CD5 XOR a	1DB4 CALL CE9E
1C35 CALL FFCC	1C85 RET	1CD6 OUT (F3),a	1DB7 XOR a
1C38 XOR a	1C86 CALL FE5E	1CD8 LD c,d	1DB8 LD (002B),a
1C39 DEC a	1C89 PUSH af	1CD9 LD a,80	1DBB CPL
1C3A OUT (F4),a	1C8A RST 8 2C ,	1CDB OUT (F2),a	1DBC INC a
1C3C PUSH bc	1C8C CALL FE5E	1CDD LD a,FF	1DBD OUT (F4),a
1C3D LD c,F2	1C8F LD e,a	1CDF OUT (F4),a	1DBF RET
1C3F OUT(c),e	1C90 POP af	1CE1 LD b,40 @	1DC0 CALL FE5E
1C41 INC c	1C91 LD b,a	1CE3 DJNZ 1CE3 FE	1DC3 LD b,a
1C42 OUT(c),d	1C92 PUSH hl	1CE5 XOR a	1DC4 PUSH bc
1C44 POP bc	1C93 PUSH bc	1CE6 OUT (F4),a	1DC5 LD a,FF
1C45 RET	1C94 PUSH de	1CE8 LD a,0F	1DC7 OUT (F4),a
1C46 CALL FE5E	1C95 LD a,01	1CEA LD b,FF	1DC9 DEC a
1C49 AND a	1C97 OUT (F3),a	1CEC DJNZ 1CEC FE	1DCA JR NZ,1DC9 FD
1C4A JR NZ,1C4F 03	1C99 LD a,B0 -	1CEE DEC a	1DCC OUT (F4),a
1C4C OUT (F4),a	1C9B OUT (F2),a	1CEF JR NZ,1CEA F9	1DCE LD a,20
1C4E RET	1C9D LD a,FF	1CF1 DEC c	1DD0 LD b,FF
1C4F DEC a	1C9F OUT (F4),a	1CF2 JR NZ,1CDD E9	1DD2 DJNZ 1DD2 FE
1C50 JR Z,1C57 05	1CA1 LD a,05	1CF4 RET	1DD4 DEC a
1C52 LD e,05	1CA3 LD b,00	1CF5 CALL FE5E	1DD5 JR NZ,1DD0 F9
1C54 JP F1C7	1CA5 DJNZ 1CA5 FE	1CF8 PUSH af	1DD7 POP bc
1C57 DEC a	1CA7 DEC a	1CF9 RST 8 2C ,	1DD8 DJNZ 1DC4 EA
1C58 OUT (F4),a	1CA8 JR NZ,1CA3 F9	1CFB CALL FE5E	1DDA RET
			1DBB NOP

Fig. 1. – Listing du programme en mnémoniques Z 80 et des implantations de chaque instruction.

Une bonne assimilation des thèmes développés dans les lignes précédentes ainsi que les documents présentés devraient rendre la compréhension de ce logiciel aisée pour les fanatiques du Z 80.

Sans être d'une longueur excessive, celui-ci occupe tout de même plusieurs centaines d'octets. Cette taille est suffisante pour rendre une faute de frappe très délicate à déceler. Pour éviter ce type de mésaventures,

nous vous proposons figure 3 un programme de chargement. Celui-ci, une fois démarré par la commande « RUN » effectue l'entrée de la routine par groupe de 8 octets. Il faut introduire ligne par ligne les octets

de la figure 4 (sans les séparer par un espace). Chaque fin de ligne doit être validée par un appui sur la touche RETURN. Il faut alors introduire la somme de contrôle située à droite de cette ligne. Si celle-ci

correspond au calcul de l'ordinateur, vous pouvez continuer votre chargement. Sinon, il est nécessaire de réintroduire la ligne incriminée.

Cette méthode de saisie a déjà fait ses preuves et assure une fiabilité quasi totale du chargement.

La sauvegarde et le chargement du programme peuvent se faire à l'aide des options « S » et « L » du Moniteur-Déassembleur paru dans le numéro 42 de « Micro-Systèmes » ou, si vous ne l'avez pas entré, en utilisant les programmes des **figures 5 et 6**.

```

10 REM***** CHARGEUR HEXADECIMAL *****
20 CLS
30 X=&H1BC0
40 PRINT HEX$(X);
50 INPUT A$
60 IF LEN(A$)<>16 THEN CLS:GOTO 40
70 FOR T=0 TO 7
80 U=VAL("&H"+MID$(A$,T*2+1,2))
90 S=S+U
100 POKE X+T,U
110 NEXT
120 INPUT "SOMME ";SO
130 IFSO=STHENS=0:SO=S:X=X+8:IFX=&H1DE0
THEN PRINT"TERMINÉ":ENDELSECLS:GOTO40
140 S=0
150 PRINT" ERREUR !!":CLS:GOTO40

```

Fig. 3. – Exemple de programme Basic effectuant le chargement hexadécimal de la routine en langage machine.

1BC0 :26 1C 35 1C 46 1C 5B 1C = 364
1BC8 :86 1C CD 1C F5 1C C0 1D = 889
1BD0 :AA F1 AA F1 AA F1 AA F1 = 1644
1BD8 :AA F1 AA F1 AA F1 AA F1 = 1644
1BE0 :AA F1 AA F1 AA F1 AA F1 = 1644
1BE8 :AA F1 AA F1 AA F1 AA F1 = 1644
1BF0 :AA F1 AA F1 AA F1 AA F1 = 1644
1BF8 :AA F1 AA F1 A9 1D 9E 1D = 1207
1C00 :CD 5E FE FE 20 38 05 1E = 930
1C08 :05 C3 C7 F1 47 CF 2C CB = 1165
1C10 :20 16 00 58 DD E5 DD 21 = 846
1C18 :C0 1B DD 19 DD 5E 00 DD = 1001
1C20 :56 01 DD E1 D5 C9 CD CC = 1356
1C28 :FF D5 CF 2C CD CC FF E3 = 1610
1C30 :73 23 72 E1 C9 CD CC FF = 1354
1C38 :AF 3D D3 F4 C5 0E F2 ED = 1381
1C40 :59 0C ED 51 C1 C9 CD 5E = 1112
1C48 :FE A7 20 03 D3 F4 C9 3D = 1173
1C50 :28 05 1E 05 C3 C7 F1 3D = 776
1C58 :D3 F4 C9 CD 5E FE 4F C5 = 1485
1C60 :CF 2C CD 5E FE C1 47 C5 = 1265
1C68 :CF 2C CD 5E FE D3 F3 C1 = 1451
1C70 :C5 AF 2F D3 F4 AF D3 F2 = 1502
1C78 :C1 C5 10 FE 3C 20 F7 C1 = 1192
1C80 :0D 20 ED D3 F4 C9 CD 5E = 1237
1C88 :FE F5 CF 2C CD 5E FE 5F = 1398
1C90 :F1 47 E5 C5 D5 3E 01 D3 = 1225
1C98 :F3 3E B0 D3 F2 3E FF D3 = 1462
1CA0 :F4 3E 05 06 00 10 FE 3D = 648
1CA8 :20 F9 AF D3 F4 3E 05 06 = 984
1CB0 :00 10 FE 3D 20 F9 1D 20 = 673
1CB8 :E4 D1 D5 3E 10 06 00 10 = 750
1CC0 :FE 3D 20 F9 1D 20 F4 D1 = 1110
1CC8 :C1 10 C8 E1 C9 CD 5E FE = 1388

Fig. 4. – Liste des codes hexadécimaux du programme de décodage.

L'utilisation

A chaque mise sous tension de l'appareil, les premiers octets de la RAM sont réactualisés. Il est donc nécessaire de modifier à chaque fois le vecteur contenant l'adresse de la routine à exécuter à chaque utilisation de l'ordre PAINT. Le programme se charge de cette fonction. Il faut tout de même modifier le START\$ de façon à ce que la routine d'initialisation soit exécutée à chaque mise sous ten-

sion ; pour cela, il convient de faire :

```
START$ = « exec &h1DA9 »
+ chr$(13):0FF1
```

Comme nous l'avons déjà mentionné, chaque instruction est associée à un nombre. Les différents paramètres inhérents à chaque mot clé doivent suivre celui-ci, séparés entre eux par une virgule. Par exemple, PAINT 1, 14 provoque l'émission continue d'un son très aigu. Le rôle de chaque instruction

```

0 REM **** SAUVEGARDE ****
10 CLS
20 INPUT "COMBIEN DE SAUVEGARDES DESIRE
Z-VOUS";A
30 INIT#1,"CASO:"
40 FOR T=1 TO A
50 PRINT#1,"EXTSON"
60 FOR I=0 TO 150 :NEXT
70 FOR I=&H1BC0 TO &H1DDA
80 OUT#1,PEEK(I)
90 NEXT I,T
100 PRINT" SAUVEGARDES EFFEC- TUEES ."
110 END

```

Fig. 5. – Programme de sauvegarde de la routine entrée.

1CD0 :57 3E FF D3 F4 AF D3 F3 = 1488
1CD8 :4A 3E 80 D3 F2 3E FF D3 = 1245
1CE0 :F4 06 40 10 FE AF D3 F4 = 1214
1CE8 :3E 0F 06 FF 10 FE 3D 20 = 701
1CF0 :F9 0D 20 E9 C9 CD 5E FE = 1281
1CF8 :F5 CF 2C CD 5E FE C1 4F = 1321
1D00 :C5 CF 2C CD 5E FE D3 F3 = 1455
1D08 :C1 C5 AF 3D D3 F4 AF D3 = 1467
1D10 :F2 C1 C5 0D 20 FD 3D 20 = 1023
1D18 :F6 C1 10 ED D3 F4 C9 43 = 1415
1D20 :2C 1F 09 0B 1B 0E 0F 1C = 179
1D28 :5C 4B 4A 09 11 08 01 0C = 288
1D30 :14 42 1B 3C 6B 7D 12 0F = 438
1D38 :0A 01 17 72 65 11 74 61 = 479
1D40 :6F 7C 1B 1D 1B 1A 0D 78 = 477
1D48 :62 03 12 1A 0A 63 65 11 = 372
1D50 :11 0B 0A 11 75 76 76 11 = 425
1D58 :1F 1E 10 00 00 00 00 00 = 77
1D60 :63 22 0F 01 01 4E 78 68 = 452
1D68 :00 CD 9E CE AF 21 1F 1D = 837
1D70 :06 49 AE EF 23 10 FB 3E = 856
1D78 :37 EF 06 03 21 00 00 11 = 353
1D80 :01 00 19 30 FD 10 F5 CD = 793
1D88 :9E CE 2F 21 00 1C 22 9A = 660
1D90 :00 21 0C E8 22 45 00 33 = 431
1D98 :33 C9 94 1D 69 1D 21 B4 = 776
1DA0 :00 36 84 3E A3 CD 28 E4 = 884
1DA8 :C9 21 9A 1D 22 45 00 FF = 775
1DB0 :D8 CD A2 00 CD 9E CE AF = 1327
1DB8 :32 2B 00 2F 3C D3 F4 C9 = 856
1DC0 :CD 5E FE 47 C5 3E FF D3 = 1349
1DC8 :F4 3D 20 FD D3 F4 3E 20 = 1139
1DD0 :06 FF 10 FE 3D 20 F9 C1 = 1066
1DD8 :10 EA C9 00 00 00 00 00 = 451

```

0 REM **** CHARGEMENT ****
10 CLS
20 INIT#1,"CASI:"
30 INPUT#1,I$:IF I$<>"EXTSON" THEN GO
30
40 FOR I=&H1BBF TO &H1DDA
50 POKEI,INP(#1)
60 NEXT
70 PRINT "CHARGEMENT EFFECTUE"
80 FOR I=0 TO 1000:NEXT
90 EXEC&H1DA9

```

ENCADRE B : création d'une instruction DOKE

ROLE :	chargement en mémoire sur 2 octets d'une valeur codée sur 16 bits.
ROUTINE COMMENTEE EN LANGAGE D'ASSEMBLAGE	
1C26 CALL FFCC	récupère l'adresse à charger dans DE
1C29 PUSH DE	place la valeur de DE en haut de la pile
1C2A RST 8 2C	est-ce suivi d'une virgule ? Sinon erreur.
1C2C CALL FFCC	récupère la valeur à DOKer dans DE
1C2F EX (SP), HL	
1C30 LD (HL), E	
1C31 INC HL	
1C32 LD (HL), D	
1C33 POP HL	{ procède à l'implantation récupère la valeur de HL
1C34 RET	retour à l'interpréteur
LIAISON AVEC L'INTERPRETEUR :	
POKE &H1BC0 + 0*2, &H26 : POKE &H1BC1 + 0*2, &H1C	
SYNTAXE : PAINT 0, adresse à charger, valeur à planter	
NOTA : il ne vous est pas nécessaire d'entrer l'instruction DOKE ; celle-ci est déjà intégrée dans le listing de la figure 4.	

Fig. 6. – Programme de chargement de la routine de décodage de l'instruction PAINT.

ENCADRE A

N°	NOM	ROLE	1 ^e PARAMETRE	2 ^e PARAMETRE	3 ^e PARAMETRE	EXEMPLE
1	SOUND	Production d'un son ininterrompu d'une hauteur donnée	Hauteur du son produit $0 \leq x \leq \$FFFF$			1,14
2	H/P	Contrôle du haut-parleur permettant l'interception du son produit par SOUND	0 fermeture H/P, 1 ouverture H/P et émission du dernier son produit			2,1
3	ZAP	Bruit d'objet qui tombe. Plusieurs à la suite produisent un bruit de laser	Nombre de fois $\$0 \leq x \leq \FF	Longueur d'un Zap $\$1 \leq x \leq \FF	Hauteur du son $\$0 \leq x \leq \FF	3, 5, 20, 0
4	PHONE	Produit un bruit similaire à une sonnerie de téléphone (occupé)	Nombre de sonnerie $\$1 \leq x \leq \FF	Longueur d'une sonnerie $\$1 \leq x \leq FF$		4, 10, 20
5	MOTOR	Produit une sorte de discret bruit de moteur. Le volume est assez bas	Durée du bruit $\$1 \leq x \leq \FF			5, 10
6	ZIP	Seul : bruit de tir sidéral ; plusieurs consécutifs : autre bruit de laser	Nombre de ZAP $\$1 \leq x \leq \FF	Longueur d'un ZAP $\$1 \leq x \leq \FF	Hauteur du son produit $\$0 \leq x \leq \FF	6, 4, 40, 1
7	STORM	Produit un crépitement semblable à une rafale de mitraillette	Longueur de la rafale $\$1 \leq x \leq \FF			7, 40

Nota : Dans les colonnes « paramètre » sont indiqués le rôle des paramètres ainsi que les intervalles dans lesquels ils doivent être compris. Ces paramètres peuvent être des données numériques fixes, mais aussi des variables ou même les résultats de tout calcul pouvant être interprété par le Basic du Canon X-07 (\times , $+$, $-$, LOG, EXP, SIN, COS...).

(elles sont au nombre de 8) et des différents paramètres qui l'accompagnent sont développés dans l'**encadré A**.

Création de nouvelles instructions

Comme nous en avons déjà fait la remarque, le but de ce

programme est aussi de vous permettre de créer vos propres mots clés. Plus que de longues explications, une référence à l'**encadré B** devrait vous aider à bien comprendre la marche à suivre. On y crée pas à pas l'instruction DOKE : celle-ci existe déjà sur certains micro-ordinateurs. En plus de cet exemple, certaines bases sont nécessaires

à la création de nouveaux ordres.

- Le numéro de l'instruction doit être compris entre 0 et 31. Les 8 premiers sont déjà utilisés mais la possibilité d'extension est encore grande.

- La liaison de la routine avec l'interpréteur se fait par : PAINT 0, &H1BC0 + N * 2,A

où N représente le numéro affecté au nouvel ordre et A l'adresse (codée sur 16 bits) de la routine correspondante.

- Il est nécessaire, pour faciliter l'insertion de la routine dans le logiciel de base, que le registre HL pointe au retour sur le caractère suivant le dernier paramètre de l'instruction. ■