

---

# Partie 6

---

**Son**

# 6/0

## Table des matières

---

- 6/1 Définitions**
- 6/2 Introduction au circuit sonore AY3-8912**
- 6/3 Programmation du circuit sonore en Basic**
  - 6/3.1 Exemples d'utilisation du générateur sonore en Basic**
- 6/4 Programmation du circuit sonore en Assembleur**
  - 6/4.1 De nouvelles instructions sonores**
- 6/5 Logiciels**
  - 6/5.1 Fichiers musicaux sous interruptions**
  - 6/5.2 Emission d'un Beep sonore en utilisant la macro du firmware MC SOUND REGISTER**
    - 6/5.2.1 Emission d'un beep sonore paramétrable
- 6/6 La synthèse vocale**
  - 6/6.1 Le synthétiseur vocal TECHNI-MUSIQUE**
    - I. Le synthétiseur vocal TECHNI-MUSIQUE et le non-initié
    - II. Utilisation sous Basic
    - III. Utilitaire de conversion
    - IV. Le synthétiseur pédagogique
    - V. Un répondeur téléphonique
    - VI. Le synthétiseur vocal et langage machine



# 6/1

## Définitions

---

Avant d'entrer dans les détails du circuit sonore des CPC, nous allons donner quelques définitions.

Un *son* est une vibration issue d'un point particulier de l'espace, et qui se propage à vitesse constante dans toutes les directions. Cette vibration est produite par un choc (caillou jeté dans l'eau, objet qui tombe sur le sol, etc.) ou par une réaction (bruit que fait un ressort comprimé qui se détend, son produit par une membrane de haut parleur sur laquelle on applique un courant électrique, etc.).

L'oreille humaine peut différencier deux sons grâce à trois grandeurs appelées *hauteur*, *volume* et *timbre*.

*Hauteur d'un son (aussi appelée « fréquence ») :*

Les sons audibles ont une fréquence comprise entre 16 et 15 000 périodes par seconde (ou Hertz). Les sons ayant des fréquences inférieures à 16 Hertz sont appelés « infra sons » ; les sons ayant des fréquences supérieures à 15 000 Hertz sont appelés « ultra sons ».

*Volume d'un son :*

C'est la puissance sonore issue du mouvement vibratoire.

*Timbre :*

Une note est composée d'une fondamentale et de plusieurs harmoniques. Si nous arrivons à distinguer une même note issue de deux instruments différents, c'est grâce à la différence d'harmoniques entre les deux instruments.



# 6/2

## Introduction au circuit sonore AY3-8912

---

Le circuit AY3-8912 de General Instruments est souvent utilisé sur les micro-ordinateurs familiaux. La raison en est simple : ce générateur sonore allie facilité de programmation et diversité des sons générés. Il est entièrement programmable au moyen de 16 registres internes dont la description sera faite ci-après. Une fois ses registres programmés, AY3 n'a plus besoin du micro-processeur pour générer un son, même si celui-ci dure plusieurs secondes. Ceci est un atout supplémentaire non négligeable, car la programmation des registres internes est ultra rapide, et le micro-processeur disposera de beaucoup de temps pour s'occuper d'autres choses, ce qui est important dans les jeux d'arcades par exemple.

### STRUCTURE INTERNE

AY3-8912 peut gérer trois voies sonores simultanément et une voie de bruit blanc. Le schéma interne ci-dessous nous montre que ce circuit est divisé en huit parties :

#### 1°) Générateurs sonores

Au nombre de trois, un pour chaque canal. Ils produisent des signaux carrés de fréquence programmable par les registres internes.

#### 2°) Générateur de bruit blanc

Il produit des signaux rectangulaires de fréquence variable et quelconque. Le bruit obtenu est un souffle plus ou moins aigu.

#### 3°) Mélangeur de canaux

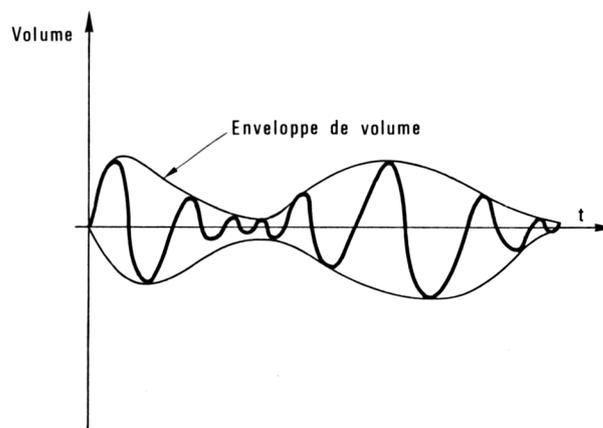
Permet de mixer le bruit blanc et les trois canaux sonores.

## 4°) Contrôleur d'amplitude

Fournit au convertisseur digital analogique (CDA) le volume sonore de la sortie analogique. Ce volume peut être fixé ou contrôlé par le générateur d'enveloppe.

## 5°) Générateur d'enveloppe

On appelle enveloppe la forme qui « entoure » un son dans le temps.



Le générateur d'enveloppe agit sur le contrôleur d'amplitude pour contrôler la variation d'amplitude d'un son dans le temps, sur les trois canaux.

## 6°) Convertisseur digital analogique

Un par voie ; ils permettent de convertir les signaux numériques issus des générateurs sonores en un signal analogique qui sera envoyé sur un haut parleur.

## 7°) Port d'entrée/sortie

Permet de sélectionner un des 16 registres internes du AY3 par l'intermédiaire de deux signaux : BDIR et BC1.

Si BDIR = 0 et BC1 = 0    AY3 est inactif.

Si BDIR = 0 et BC1 = 1    AY3 est en mode lecture.

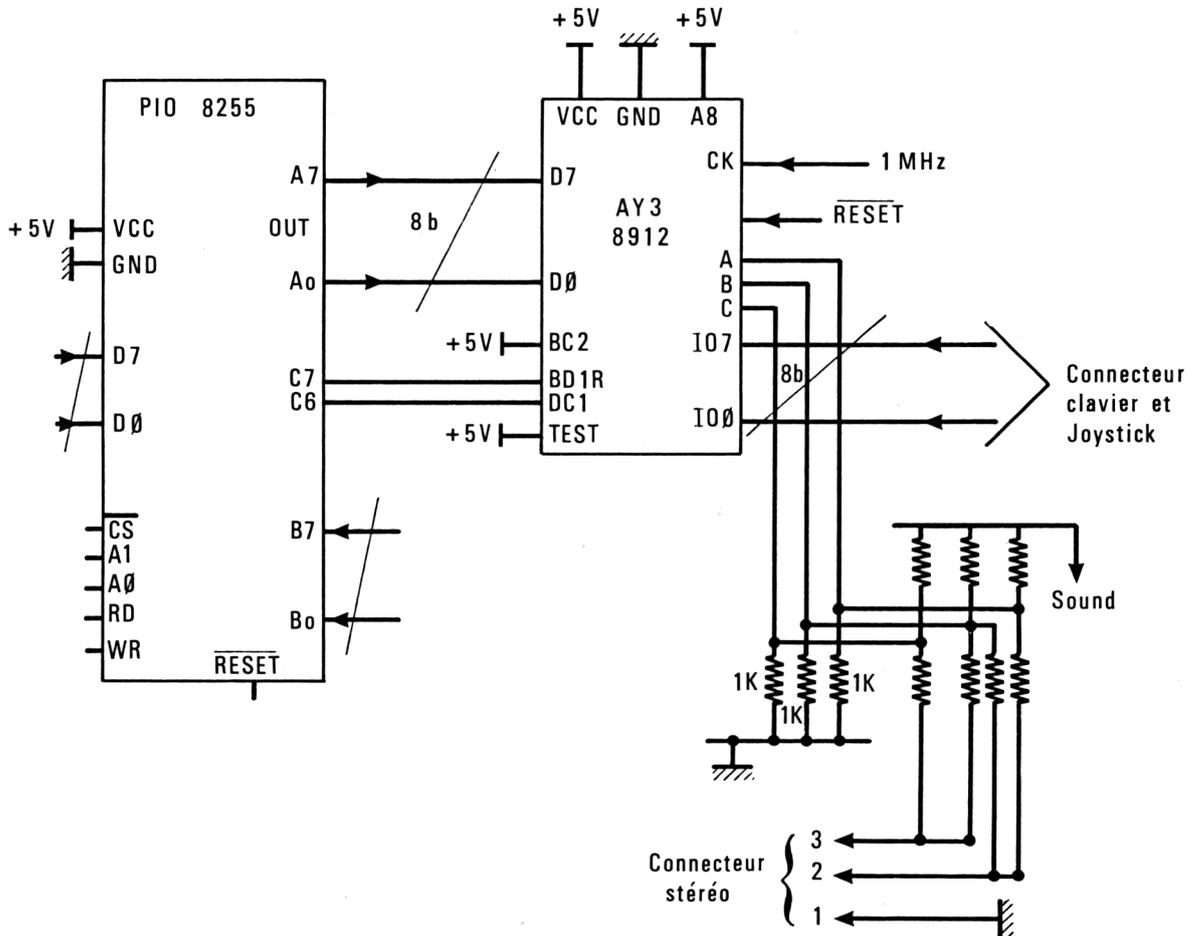
Si BDIR = 1 et BC1 = 0    AY3 est en mode écriture.

Si BDIR = 1 et BC1 = 1    AY3 charge l'adresse d'un de ses registres internes.

## 8°) Buffers d'entrée/sortie

Contiennent les données à écrire ou lues sur AY3.

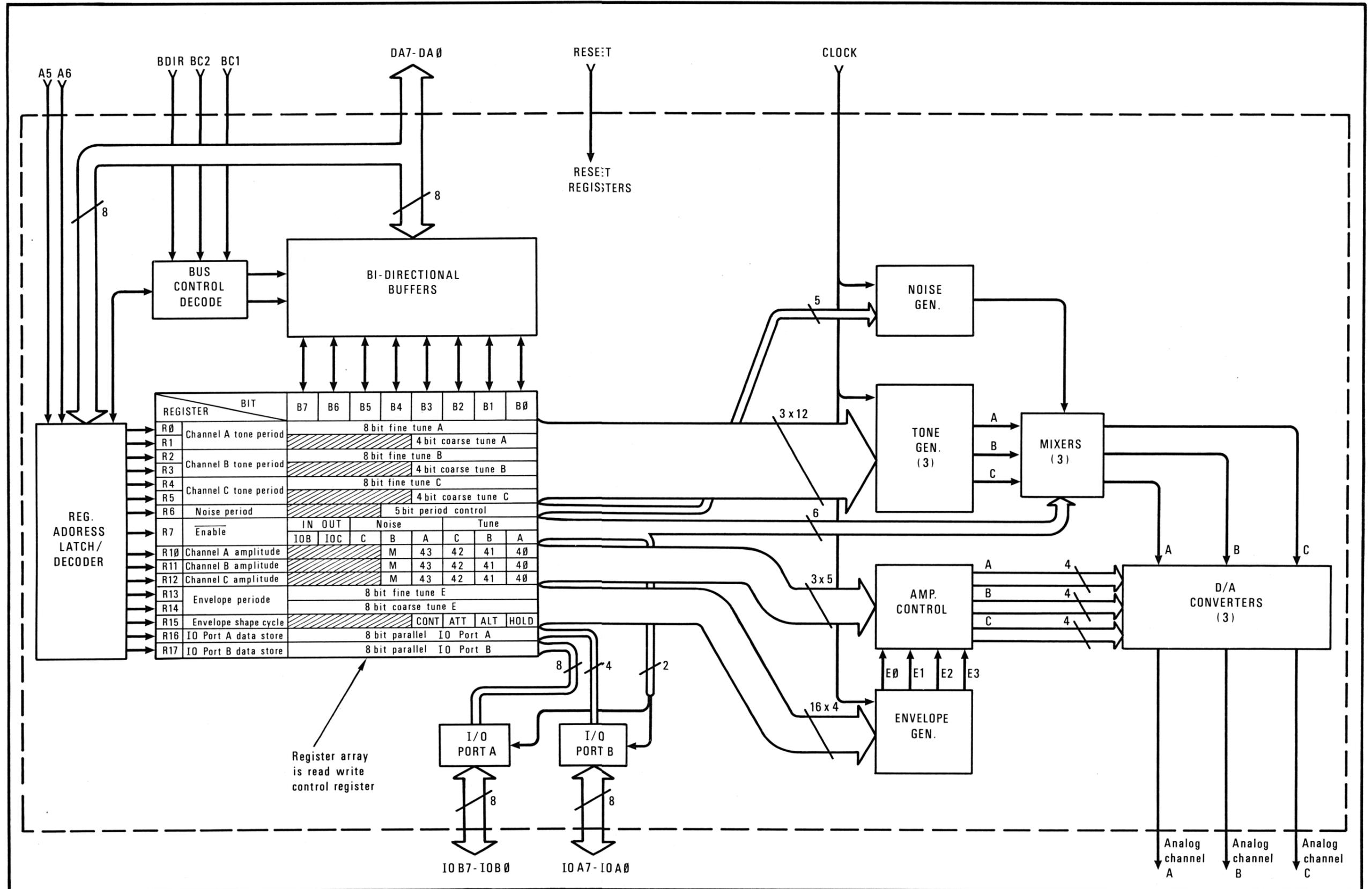
AY3 s'implante sur la carte mère de la façon suivante :



Nous voyons que :

- Les signaux analogiques A, B et C sortent sur le connecteur stéréo par l'intermédiaire d'un pont diviseur.
- Les pattes BDIR et BC1 sont commandées par le PIO 8255.
- De même, le bus de données est directement connecté au port A du PIO.









# 6/3

## Programmation du circuit sonore en BASIC

---

Nous disposons de six ordres en BASIC :

SOUND	Met un son dans la file d'attente.
ENT	Définit une enveloppe de ton.
ENV	Définit une enveloppe de volume.
ON SQ GOSUB	Définit le point d'entrée d'un sous-programme de gestion de file sonore.
RELEASE	Libère un son dans un canal sonore.
SQ	Donne le nombre de places libres dans une file d'attente sonore.

Reportez-vous aux ordres BASIC correspondants (voir Partie 4 chap. 1) pour avoir les détails nécessaires à la programmation du AY3 en BASIC.



## 6/3.1

# Exemples d'utilisation du générateur sonore en Basic

Nous vous proposons trois programmes écrits en Basic qui utilisent les trois voix du générateur sonore. Ces trois programmes sont tous construits sur le modèle sans interruption. Si vous désirez les incorporer dans vos programmes de jeu, il sera très simple de les rendre interrompibles en utilisant ce qui a été dit au sujet des interruptions dans cette partie.

### COMMENT UTILISER LE PROGRAMME

Voici les listings des trois programmes. Pour les exécuter, il suffit de taper RUN.

L'appui sur une touche quelconque du clavier pendant l'exécution d'un morceau le stoppe et redonne le contrôle au Basic.

- Premier listing : « Caraïbo »

```

1000 REM -----
1010 REM Exemple d'utilisation des trois voix
1020 REM du generateur sonore
1030 REM -----
1040 '
1050 ENV 3,2,7,1,1,0,2,5,-1,4 ' Degfinition de l'enveloppe
1060 RESTORE 1300 ' Pointeur de donnees
1070 FOR i=1 TO 128 ' 128 groupes de notes
1080 READ p,p1,p2
1090 a$=INKEY$
1100 IF a$<>" THEN END
1110 IF p<>4 THEN 1160
1120 SOUND 1,p,18,13,3,0,13

```

```

1130 SOUND 2,p1,18,15,3
1140 SOUND 4,p2,18,15,3
1150 GOTO 1240
1160 IF (p OR p1 OR p2)<>1 THEN 1210
1170 SOUND 1,p,18,0
1180 SOUND 2,p1,18,0
1190 SOUND 4,p2,18,0
1200 GOTO 1240
1210 SOUND 1,p,18,15,3
1220 SOUND 2,p1,18,13,3
1230 SOUND 4,p2,18,13,3
1240 NEXT i
1250 '
1260 ' -----
1270 ' Donnees musicales
1280 ' -----
1290 '
1300 DATA 568,150,95,4,150,95,4,150,95,451,150,95,4,169
1310 DATA 95,4,1,1,379,169,106,4,1,1,568,190,142,4,1,1
1320 DATA 4,1,1,451,190,142,4,1,1,4,1,1,379,1,1,4,1,1
1330 DATA 568,150,113,4,150,113,4,150,113,451,150,113,4
1340 DATA 169,126,4,1,1,379,169,106,4,1,1,568,190,142,4
1350 DATA 1,1,4,1,1,451,190,142,4,1,1,4,1,1,379,1,1,4,1,1
1360 DATA 568,150,95,4,150,95,4,150,95,451,150,95,4,169
1370 DATA 95,4,1,1,379,169,106,4,1,1,568,190,142,4,1,1,4
1380 DATA 1,1,451,190,142,4,1,1,4,1,1,379,1,1,4,1,1
1390 DATA 568,150,113,4,150,113,4,150,113,451,150,113,4
1400 DATA 169,126,4,1,1,379,169,106,4,1,1,568,190,142,4
1410 DATA 1,1,4,1,1,451,190,142,4,1,1,4,1,1,379,1,1,4,1,1
1420 DATA 851,169,106,4,169,106,4,1,1,676,159,95,4,1,1,4
1430 DATA 159,95,568,1,1,4,142,84,851,1,1,4,142,84,4,1,1
1440 DATA 676,159,95,4,1,1,4,142,84,568,159,95,4,1,1
1450 DATA 851,169,106,4,1,1,4,169,106,676,1,1,4,190,113
1460 DATA 4,1,1,568,169,106,4,1,1,851,1,1,4,1,1,4,1,1
1470 DATA 678,1,1,4,1,1,4,1,1,568,1,1,4,1,1
1480 DATA 568,225,142,4,225,142,4,1,1,451,213,127,4,1,1,4
1490 DATA 213,127,379,1,1,4,190,113,568,1,1,4,190,113,4,1
1500 DATA 1,451,213,127,4,1,1,4,190,113,379,213,127,4,1,1
1510 DATA 568,225,142,4,1,1,4,225,142,451,1,1,4,253,150,4
1520 DATA 1,1,379,225,142,4,1,1,568,1,1,4,1,1,4,1,1,451,1
1530 DATA 1,4,1,1,4,1,1,379,1,1,4,1,1
1540 GOTO 1060

```

• Second listing : « Percussions »

```

1000 REM -----
1010 REM Exemple d'utilisation des trois voix
1020 REM du generateur sonore
1030 REM -----
1040 '
1050 ENV 2,2,7,1,1,0,2,5,-1,4
1060 FOR i=1 TO 320
1070   READ p
1080   IF (P)=4 THEN SOUND 2,P,10,15,2,0,9:GOTO 1130
1090   IF (P)=1 THEN SOUND 2,p,10,0:GOTO 1130
1100   SOUND 2,p,10,13,2,0,23
1110   a$=INKEY$
1120   IF a$<>" " THEN END
1130 NEXT i
1140 '
1150 '
1160 ' -----
1170 ' Donnees sonores
1180 ' -----
1190 '
1200 DATA 3000,1,3000,1,4,1,3000,1,3000,1,1,1
1210 DATA 4,1,1,3000,3000,1,3000,1,4,1,3000,1
1220 DATA 3000,1,1,1,4,1,1,1,3000,1,3000,1,4
1230 DATA 1,3000,1,3000,1,1,1,4,1,1,3000,3000
1240 DATA 1,3000,1,4,1,3000,1,3000,1,1,1,4,1,1
1250 DATA 1,3000,1,3000,1,4,1,3000,1,3000,1,1
1260 DATA 1,4,1,1,3000
1270 DATA 3000,1,3000,1,4,1,3000,1,3000,1,1,1
1280 DATA 4,1,1,1,3000,1,3000,1,4,1,3000,1,3000
1290 DATA 1,1,1,4,1,1,3000,3000,1,3000,1,4,1
1300 DATA 3000,1,3000,1,1,1,4,1,1,1,3000,1,3000
1310 DATA 1,4,1,3000,1,3000,1,1,1,4,1,1,3000
1320 DATA 3000,1,3000,1,4,1,3000,1,3000,1,1,1
1330 DATA 4,1,4,4
1340 DATA 3000,1,3000,1,4,1,3000,1,3000,1,1,1,4
1350 DATA 1,1,3000,3000,1,3000,1,4,1,3000,1,3000
1360 DATA 1,1,1,4,1,1,1,3000,1,3000,1,4,1,3000,1
1370 DATA 3000,1,1,1,4,1,1,3000,3000,1,3000,1,4
1380 DATA 1,3000,1,3000,1,1,1,4,1,1,1,3000,1,3000
1390 DATA 1,4,1,3000,1,3000,1,1,1,4,1,1,3000
1400 DATA 3000,1,3000,1,4,1,3000,1,3000,1,1,1,4,1
1410 DATA 1,1,3000,1,3000,1,4,1,3000,1,3000,1,1,1
1420 DATA 4,1,1,3000,3000,1,3000,1,4,1,3000,1,3000
1430 DATA 1,1,1,4,1,1,1,3000,1,3000,1,4,1,3000,1
1440 DATA 3000,1,1,1,4,1,1,3000,3000,1,3000,1,4
1450 DATA 1,3000,1,3000,1,1,1,4,1,4,4

```

## • Troisième listing : « Boléro »

```

1000 REM -----
1010 REM Exemple d'utilisation des trois voix
1020 REM du generateur sonore
1030 REM -----
1040 '
1050 FOR I =1 TO 1368
1060   READ P,P1
1070   SOUND 2,P,8,14,3,1
1080   SOUND 1,P1,8,13,4,1
1090   a$=INKEY$
1100   IF a$<>" " THEN END
1110 NEXT I
1120 '
1130 ' -----
1140 ' Donnees musicales
1150 ' -----
1160 '
1170 DATA 638,478,0,0,0,0,0,0,0,0,0,0,638,478,0,0,638,0,0,0,
638,0,0,0,478,379,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,478,426,0,0,0,0,0,0,0,0,638,0,0,0,0,0,0,0,
0,0
1180 DATA 638,478,0,0,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,478,379,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,638,426,0,0,638,0,0,0,638,0,0,0,638,0,0,0,638,0,0,
0,638,0,0,0
1190 DATA 638,119,0,119,0,119,0,119,0,119,0,119,638,119,0,11
9,638,119,0,119,638,119,0,119,478,119,0,119,0,119,0,119,0,11
9,0,119,638,127,0,127,638,127,0,119,638,119,0,119,478,106,0,
106,0,106,0,119,0,119,0,119,638,127,0,127,0,127,0,142,0,142,
0,142
1200 DATA 638,119,0,119,0,119,0,119,0,119,0,0,638,119,0,119,
638,119,0,142,638,142,0,142,478,119,0,119,0,119,0,119,0,119,
0,119,638,119,0,119,638,119,0,119,638,119,0,119,638,119,0,11
9,638,119,0,119,638,119,0,119,638,127,0,127,638,127,0,119,63
8,119,0,119
1210 DATA 638,142,0,142,0,142,0,159,0,159,0,159,638,190,0,19
0,638,190,0,179,638,179,0,179,478,159,0,159,0,159,0,159,0,15
9,0,159,638,159,0,159,638,159,0,159,638,159,0,159,426,159,0,
159,0,159,0,159,0,159,0,159,638,159,0,159,0,159,0,159,0,159,
0,159
1220 DATA 638,159,0,159,0,159,0,179,0,179,0,179,638,190,0,19
0,638,190,0,213,638,213,0,213,478,190,0,190,0,190,0,179,0,17
9,0,179,638,159,0,159,638,159,0,142,638,142,0,142,638,159,0,
159,638,159,0,159,638,159,0,159,638,159,0,159,638,159,0,159,
638,159,0,159
1230 DATA 638,159,0,159,0,159,0,159,0,159,0,159,478,159,0,15
9,638,159,0,159,638,159,0,159,478,159,0,159,0,159,0,142,0,14
2,0,142,638,127,0,127,638,127,0,142,638,142,0,142,478,159,0,
159,0,159,0,179,0,179,0,179,638,190,0,190,0,190,0,213,0,213,
0,213

```

```

1240 DATA 638,190,0,190,0,190,0,213,0,213,0,213,638,239,0,23
9,638,239,0,239,638,239,0,0,478,239,0,239,0,239,0,239,0,239,
0,239,638,239,0,239,638,239,0,213,638,213,0,213,638,190,0,19
0,638,190,0,190,638,190,0,190,638,179,0,179,638,179,0,179,63
8,179,0,179
1250 DATA 638,213,0,213,0,213,0,213,0,213,0,213,478,213,0,21
3,638,213,0,213,638,213,0,213,478,159,0,159,0,159,0,159,0,15
9,0,159,638,159,0,159,638,159,0,159,638,159,0,159,478,159,0,
159,0,159,0,159,0,159,0,159,638,159,0,159,0,159,0,159,0,159,
0,159
1260 DATA 638,159,0,159,0,159,0,159,0,159,0,159,638,159,0,15
9,638,159,0,159,638,159,0,159,478,159,0,159,0,159,0,159,0,15
9,0,159,638,159,0,159,638,159,0,159,638,159,0,159,638,159,0,
159,638,159,0,159,638,159,0,159,638,159,0,159,638,159,0,159,
638,159,0,159
1270 DATA 638,106,0,106,0,106,0,106,0,106,0,106,478,106,0,1
06,638,106,0,106,638,106,0,106,478,106,0,106,0,106,0,106,0,1
06,0,106,638,106,0,106,638,106,0,119,638,119,0,119,478,127,0
,127,0,127,0,142,0,142,0,142,638,127,0,127,0,127,0,119,0,119
,0,119
1280 DATA 638,106,0,106,0,106,0,119,0,119,0,119,638,127,0,12
7,638,127,0,127,638,127,0,127,478,127,0,127,0,127,0,119,0,11
9,0,119,638,127,0,127,638,127,0,142,638,142,0,142,638,119,0,
119,638,119,0,127,638,127,0,127,638,142,0,142,638,142,0,179,
638,179,0,179
1290 DATA 638,478,0,0,0,0,0,0,0,0,0,638,179,0,179,0,638,0,
179,638,179,0,0,478,179,0,179,0,179,0,179,0,179,0,179,638,14
2,0,142,638,142,0,142,638,142,0,142,478,119,0,119,0,119,0,14
2,0,142,0,142,638,127,0,127,0,127,0,159,0,159,0,159
1300 DATA 638,179,0,179,0,179,0,179,0,179,0,0,638,179,0,179,
638,0,0,179,638,179,0,0,478,179,0,179,0,179,0,179,0,179,0,17
9,638,142,0,142,638,142,0,142,638,142,0,142,638,127,0,127,63
8,127,0,159,638,159,0,159,638,142,0,142,638,142,0,179,638,17
9,0,179
1310 DATA 638,213,0,213,0,213,0,213,0,213,0,0,638,213,0,213,
638,213,0,239,638,239,0,239,478,213,0,213,0,213,0,213,0,213,
0,213,638,213,0,213,638,213,0,213,638,213,0,213,478,213,0,21
3,0,213,0,213,0,213,0,0,638,213,0,213,0,0,0,213,0,213,0,0
1320 DATA 638,213,0,213,0,213,0,213,0,213,0,213,638,179,0,17
9,638,179,0,179,638,179,0,179,478,142,0,142,0,142,0,179,0,17
9,0,179,638,159,0,159,638,159,0,190,638,190,0,190,638,213,0,
213,638,213,0,213,638,213,0,0,638,213,0,213,638,213,0,239,63
8,239,0,239
1330 DATA 638,213,0,213,0,213,0,213,0,213,0,213,478,213,0,21
3,638,213,0,213,638,213,0,213,478,213,0,213,0,213,0,213,0,21
3,0,0,638,213,0,213,638,213,0,239,638,239,0,239,478,213,0,21
3,0,213,0,213,0,213,0,213,638,190,0,190,0,190,0,179,0,179,0,
179
1340 DATA 638,159,0,159,0,159,0,159,0,159,0,159,638,159,0,15

```

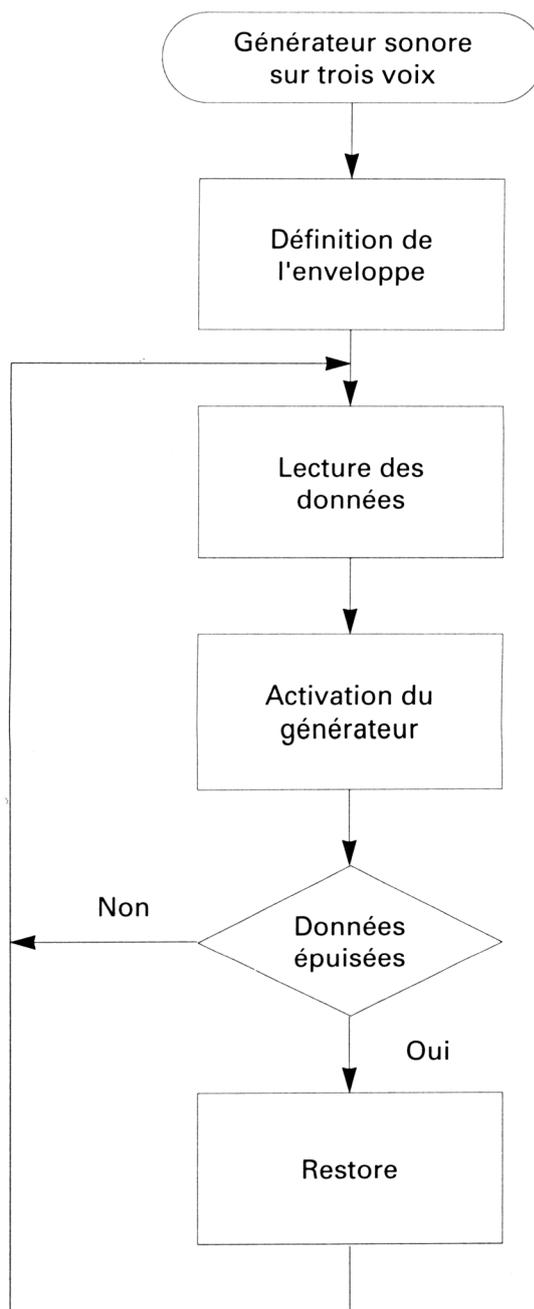
```
9,638,159,0,159,638,159,0,159,478,159,0,159,0,159,0,159,0,15
9,0,159,638,159,0,159,638,159,0,159,638,159,0,159,638,159,0,
159,638,159,0,179,638,179,0,179,638,190,0,190,638,190,0,213,
638,213,0,213
1350 DATA 478,239,0,239,0,239,0,239,0,239,0,239,638,478,0,0,
0,638,0,0,0,638,0,0,478,379,0,0,0,0,0,0,0,0,0,0,638,478,0,0,
0,638,0,0,0,638,0,0,478,426,0,0,0,0,0,0,0,0,0,0,638,0,0,0,0,
0,0,0,0,0,0,0
1360 DATA 638,478,0,0,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,478,379,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,638,426,0,0,638,0,0,0,638,0,0,0,638,0,0,0,638,0,0,
0,638,0,0,0
1370 DATA 638,134,0,134,0,134,0,134,0,134,0,134,638,134,0,13
4,638,134,0,134,638,134,0,134,478,134,0,134,0,134,0,142,0,14
2,0,142,638,159,0,159,638,159,0,179,638,179,0,179,426,134,0,
134,0,134,0,119,0,119,0,119,638,142,0,142,0,142,0,159,0,159,
0,159
1380 DATA 638,134,0,134,0,134,0,134,0,134,0,134,638,142,0,14
2,638,142,0,159,638,159,0,159,478,134,0,134,0,134,0,134,0,13
4,0,134,638,134,0,134,638,134,0,134,638,134,0,134,638,142,0,
142,638,142,0,134,638,134,0,134,638,142,0,142,638,142,0,159,
638,159,0,159
1390 DATA 638,159,0,159,0,159,0,159,0,159,0,159,638,159,0,15
9,638,159,0,159,638,159,0,159,478,159,0,159,0,159,0,179,0,17
9,0,179,638,190,0,190,638,190,0,213,638,213,0,213,478,190,0,
190,0,190,0,190,0,190,0,190,638,190,0,190,0,190,0,190,0,190,
0,190
1400 DATA 638,190,0,190,0,190,0,190,0,190,0,190,638,190,0,19
0,638,190,0,190,638,190,0,190,478,0,0,0,0,0,0,0,0,0,0,638,
134,0,134,638,134,0,134,638,134,0,134,638,119,0,119,638,119,
0,119,638,119,0,119,638,113,0,113,638,113,0,113,638,113,0,0
1410 DATA 638,113,0,113,0,113,0,113,0,113,0,0,638,113,0,113,
638,113,0,113,638,113,0,113,478,113,0,113,0,113,0,113,0,113,
0,0,638,113,0,113,638,113,0,113,638,113,0,0,478,113,0,113,0,
113,0,0,0,113,0,113,638,113,0,0,0,113,0,113,0,113,0,0
1420 DATA 638,113,0,113,0,113,0,113,0,113,0,0,638,113,0,113,
638,0,0,113,638,113,0,0,478,113,0,113,0,113,0,113,0,113,0,11
3,638,119,0,119,638,119,0,134,638,134,0,134,638,113,0,113,63
8,113,0,113,638,113,0,113,638,119,0,119,638,119,0,134,638,13
4,0,134
1430 DATA 638,113,0,113,0,113,0,119,0,119,0,119,638,134,0,13
4,638,134,0,150,638,150,0,150,478,159,0,159,0,159,0,179,0,17
9,0,179,638,190,0,190,638,190,0,190,638,190,0,190,478,190,0,
190,0,190,0,190,0,190,0,190,638,190,0,190,0,190,0,190,0,190,
0,190
1440 DATA 638,190,0,190,0,190,0,190,0,190,0,190,638,190,0,19
0,638,190,0,190,638,190,0,190,478,190,0,190,0,190,0,190,0,19
0,0,190,638,190,0,190,638,190,0,190,638,190,0,190,638,190,0,
190,638,190,0,190,638,190,0,190,638,190,0,190,638,190,0,190,
```

```

638,190,0,190
1450 DATA 638,213,0,213,0,213,0,213,0,213,0,213,638,213,0,21
3,638,213,0,213,638,213,0,213,478,213,0,213,0,213,0,213,0,21
3,0,213,638,190,0,190,638,190,0,190,638,190,0,190,426,213,0,
213,0,213,0,190,0,190,0,190,638,179,0,179,0,179,0,179,0,179,
0,179
1460 DATA 638,179,0,179,0,179,0,179,0,179,0,179,638,179,0,17
9,638,179,0,179,638,179,0,179,478,159,0,159,0,159,0,159,0,15
9,0,159,638,150,0,150,638,150,0,150,638,150,0,150,638,179,0,
179,638,179,0,179,638,159,0,159,638,159,0,159,638,190,0,190,
638,190,0,190
1470 DATA 638,213,0,213,0,213,0,190,0,190,0,190,638,213,0,21
3,638,213,0,239,638,239,0,239,478,239,0,239,0,0,0,0,0,0,0,0,0,0,
638,268,0,268,638,268,0,268,638,268,0,268,426,268,0,268,0,26
8,0,268,0,268,0,268,638,268,0,268,0,268,0,268,0,268,0,268,0,268
1480 DATA 638,268,0,268,0,268,0,239,0,239,0,239,638,268,0,26
8,638,268,0,239,638,239,0,239,478,213,0,213,0,213,0,190,0,19
0,0,190,638,213,0,213,638,213,0,239,638,239,0,239,638,213,0,
213,638,213,0,239,638,239,0,239,638,268,0,268,638,268,0,301,
638,301,0,301
1490 DATA 638,268,0,268,0,268,0,301,0,301,0,301,638,319,0,31
9,638,319,0,319,638,319,0,319,478,319,0,319,0,319,0,0,0,0,0,0,0,
0,638,358,0,358,638,358,0,358,638,358,0,358,478,358,0,358,0,358,0,
358,0,358,0,358,0,358,638,358,0,358,0,358,0,358,0,358,0,358
1500 DATA 638,358,0,358,0,358,0,358,0,358,0,358,0,358,638,358,0,35
8,638,358,0,358,638,358,0,358,478,358,0,358,0,358,0,319,0,31
9,0,319,638,358,0,358,638,358,0,402,638,402,0,402,638,402,0,
402,638,402,0,402,638,402,0,402,638,402,0,402,638,402,0,402,
638,402,0,402
1510 DATA 638,402,0,402,0,402,0,402,0,402,0,402,638,402,0,40
2,638,402,0,402,638,402,0,402,478,358,0,358,0,358,0,402,0,40
2,0,402,638,358,0,358,638,358,0,451,638,451,0,451,478,451,0,
451,0,451,0,451,0,451,0,451,638,451,0,451,0,451,0,451,0,451,
0,451
1520 DATA 638,451,0,451,0,451,0,451,0,451,0,451,638,451,0,45
1,638,451,0,451,638,451,0,451,478,0,0,0,0,0,0,451,0,451,0,45
1,638,402,0,402,638,402,0,451,638,451,0,451,638,358,0,358,63
8,358,0,402,638,402,0,402,638,451,0,451,638,451,0,478,638,47
8,0,478
1530 DATA 638,478,0,0,0,0,0,0,0,0,0,0,638,478,0,0,638,0,0,0,
638,0,0,0,478,379,0,0,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,478,426,0,0,0,0,0,0,0,0,0,0,638,0,0,0,0,0,0,0,0,0,
0,0
1540 DATA 638,478,0,0,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,478,379,0,0,0,0,0,0,0,0,0,0,638,478,0,0,0,638,0,0,
0,638,0,0,638,426,0,0,638,0,0,0,638,0,0,0,638,0,0,0,638,0,0,
0,638,0,0,0
    
```

**LE PROGRAMME EN DÉTAIL**

La logique des trois programmes est assez similaire. Voici par exemple celle du premier programme :



L'enveloppe est définie à l'aide d'une instruction **ENV** :

```
1050 ENV 3,2,7,1,1,0,2,5,-1,4
```

Les données sont réinitialisées ligne 1300 à l'aide d'une instruction **RESTORE** :

```
1060 RESTORE 1300
```

Les instructions suivantes lisent les données musicales qui se trouvent lignes 1300 à 1540 et les passent générateur sonore. Le premier morceau étant composé de 128 notes, la boucle **FOR** de lecture se fait entre 1 et 128 :

```
1070 FOR i=1 TO 128
```

```
...
```

```
1240 NEXT i
```

Les données sont stockées sous la forme de triplets :

```
Voix1, Voix2, Voix3
```

Une instruction **READ** les stocke dans les variables p, p1 et p2 :

```
1080 READ p,p1,p2
```

Le programme teste alors si le clavier a été activé. Si oui, il met fin au programme :

```
1090 a$=inkey$
```

```
1100 IF a$<>" " THEN END
```

Dans le cas contraire, les trois données musicales lues sont envoyées au générateur sonore à l'aide d'instructions **SOUND**. L'émission des sons dépend de la valeur de la première donnée. Cette méthode permet de définir un rythme à moindres frais :

```
1110 IF p<>4 THEN 1160
```

```
...
```

```
1230 SOUND 4,p2,18,13,3
```

Les instructions suivantes définissent les données des trois voix (lignes 1300 à 1530).

La ligne 1540 redonne le contrôle à la ligne 1060, ce qui fait « boucler » le programme.



# 6/4

## Programmation du circuit sonore en assembleur

Comme nous l'avons vu précédemment, AY3 possède 16 registres. Ils sont utilisés comme suit :

- R0 à R5      Générateur de ton
- R6            Générateur de bruit blanc
- R7            Contrôle du mixage du bruit blanc et des 3 canaux
- R8 à R10    Contrôle de l'amplitude des sons
- R11 à R13   Contrôle du générateur d'enveloppe
- R14          Port d'entrée/sortie

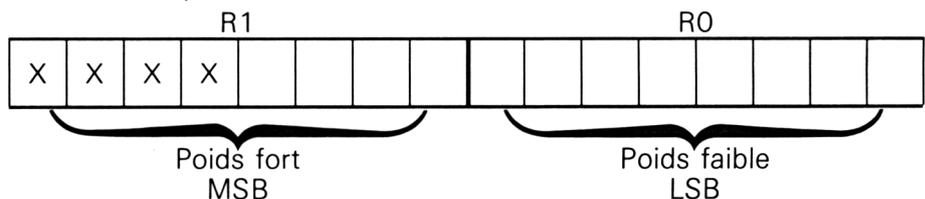
Voyons en détail l'utilisation de chaque registre.

### R0 à R5

Définition de la fréquence du son à émettre.

La fréquence du canal A est décrite dans les registres R0 et R1, la fréquence du canal B dans les registres R2 et R3, et la fréquence du canal C dans les registres R4 et R5.

Les registres R0, R2 et R4 représentent les poids faibles des sons à émettre sur 8 bits, et les registres R1, R3 et R5 les poids forts de ces sons sur les 4 bits de poids faibles :



MSB = Most Significant Byte = Octet de poids fort.  
 LSB = Last Significant Byte = Octet de poids faible.  
 X = non utilisé

En BASIC, nous pouvons calculer LSB et MSB de la façon suivante :  
supposons que nous voulions émettre un son de fréquence 1000 Hz ;  
nous aurons :

$$\text{Per} = 10^6 / (16 \times 1000) = 62.5$$

**MSB = Partie entière (Per/256) = Most significative byte (octet : le plus fort)**

**LSB = Partie entière (Per – MSB × 256) = Last significative byte (octet : le plus faible)**

Les registres « pairs » R0/R1, R2/R3 et R4/R5 sont codés sur 12 bits (8 de LSB et 4 de MSB). Ils peuvent donc prendre des valeurs comprises entre 0 et  $2^{12}$  (soit entre 0 et 4096).

Les fréquences minimum et maximum seront donc :

$$\text{On a } \text{Per} = 10^6 / (16 \times \text{Freq}), \text{ d'où } \text{Freq} = 10^6 / (16 \times \text{Per})$$

$$\text{Freq max} = 10^6 / (16 \times 1) = 62\,500 \text{ Hz}$$

$$\text{Freq min} = 10^6 / (16 \times 4096) = 15.26 \text{ Hz}$$

## R6

Programmation du générateur de bruit blanc.

Seuls les cinq bits de poids faibles sont utilisés.

La période du bruit blanc généré est donnée par la même formule que pour la génération de sons :

$$\text{Freq} = 10^6 / (16 \times \text{Per})$$

La « période » du bruit blanc n'a pas de grande signification, puisqu'un bruit blanc est avant tout une succession de fréquences quelconques. Il définit cependant la période moyenne autour de laquelle sera généré le bruit blanc. Par abus de langage, on parlera de la période du bruit blanc généré.

## R7

Ce registre permet de mixer le bruit blanc et les trois canaux sonores A, B et C.

Il a la structure suivante :

X	PA	NC	NB	NA	SC	SB	SA
---	----	----	----	----	----	----	----

SA, SB et SC valident les canaux A, B ou C. 0 indique que le canal est dévalidé, 1 qu'il est validé.

NA, NB et NC valident l'émission de bruit blanc sur les canaux A, B ou C. 0 indique que le bruit blanc est dévalidé et 1 qu'il est validé.

PA S'il vaut 0, le port A est défini en entrée.

S'il vaut 1, le port A est défini en sortie.

### R8 à R10

Définissent l'amplitude du signal sonore sur les canaux A, B et C. R8 pour le canal A, R9 pour le canal B et R10 pour le canal C. Les bits de ces registres sont codés de la façon suivante :

X	X	X	M	A3	A2	A1	A0
---	---	---	---	----	----	----	----

A0 à A3 donnent l'amplitude codée entre 0 et 15. 0 représente le volume le plus faible, 15 le volume le plus fort.

M donne le mode de fonctionnement du contrôleur d'amplitude. Si M vaut 0, l'amplitude est fixe. Si M vaut 1, l'amplitude est contrôlée par le générateur d'enveloppe.

### R11 et R12

Définissent le contrôle du générateur d'enveloppe.

Comme nous l'avons vu plus haut, l'enveloppe est un signal qui module le volume d'un son issu du générateur de ton.

Si F est la fréquence de l'enveloppe, la période d'enveloppe sera égale à :

$$P = 10^6 / (256 \times F)$$

Elle est codée sur 16 bits. R11 est le registre de poids faible, R12 le registre de poids fort. Comme P peut varier de 0 à 2<sup>16</sup> (de 0 à 65535), les fréquences minimale et maximale de l'enveloppe seront :

$$F_{\min} = 10^6 / (256 \times 65535) = 0.059 \text{ Hz}$$

$$F_{\max} = 10^6 / (256 \times 1) = 3906.25 \text{ Hz}$$

### R13

Ce registre permet de contrôler la forme de la modulation : Seuls les quatre bits de poids faible de R13 sont utilisés de la manière suivante :

X	X	X	X	CO	AT	AL	HO
---	---	---	---	----	----	----	----

CO = COninue

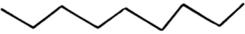
AT = ATtack

AL = ALternate

HO = HOld

Ils ont la signification suivante :

CO	AT	AL	HO	
0	0	X	X	
0	1	X	X	
1	0	0	0	
1	0	0	1	

CO	AT	AL	HO	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Pour que R13 soit pris en compte par le AY3, il faut que le bit MODE (bit 4) des registres de contrôle d'amplitude (R8 à R10) soit à 1.

### R14

C'est par le port A de l'interface parallèle 8255 que va être programmé le AY3. Ce port est accessible par l'intermédiaire du registre R14 du AY3.

La programmation d'un registre du AY3 se déroule comme suit :

- 1°) Adresse du registre à accéder.
- 2°) Bits 6 et 7 du registre C à 11.
- 3°) Valeur à charger/chargée dans le registre du AY3.
- 4°) Bits 6 et 7 du registre C à 10.

Toutes les opérations sont réalisées par la routine du firmware implantée en #BD34.

Pour utiliser cette routine, il faut positionner les registres A et C comme suit :

- le registre A doit contenir le numéro du registre à charger ;
- le registre C doit contenir la donnée à charger.

*Exemple :*

Pour définir le volume du canal B à 13, il faudra faire :

*a) En assembleur :*

```
LD    A,1    ;Volume du canal B dans le registre 1
LD    C,13   ;Volume = 13
CALL  #BD34 ;Programmation
```

*b) En BASIC :*

```
1000 FOR I=0 TO 7
1010  READ A : POKE I+ &9000,A
1020 NEXT I
1030 DATA &3E, 1, &0E, 13, &CD, &34, &BD, &C9
1040 CALL &9000
```

Reportez-vous au chapitre sur les banques ROM (voir Partie 4 chap. 2.7) pour avoir plus de détails.

## 6/4.1

# De nouvelles instructions sonores

---

Nous vous avons déjà proposé quelques effets sonores en Basic pour agrémenter vos programmes, cette fois-ci, ce sont de nouvelles instructions sonores que nous vous proposons de créer pour enrichir le vocabulaire de votre CPC.

### LE PRINCIPE

Ces nouvelles instructions seront créées sous formes de RSX (ReSident eXtension) et utiliseront les vecteurs d'accès au composant sonore, vecteurs décrits en Partie 2 Chapitre 2.7 pages 48 à 53.

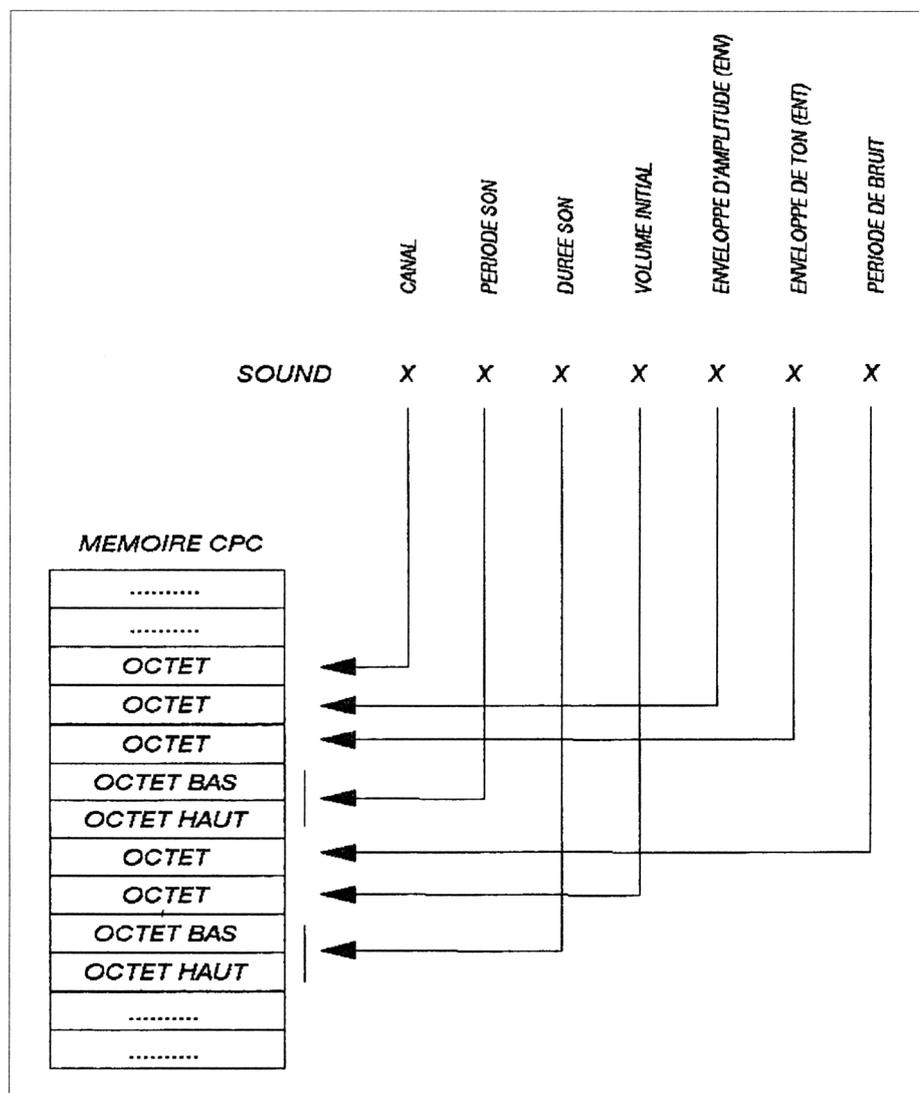
Les trois vecteurs qui nous intéressent plus particulièrement sont situés aux adresses **&BCAA**, **&BCBC** et **&BCBF**, et nous allons les préciser un peu plus, par rapport aux instructions Basic équivalentes.

## SOUND QUEUE

Ce vecteur situé à l'adresse **&BCAA** est l'équivalent de l'instruction **SOUND** du Basic.

L'appel de celui-ci nécessite l'adresse du son à émettre dans le registre **HL**, celui-ci pointant une zone comportant les paramètres semblables à l'instruction **SOUND**.

Pour vous permettre de programmer facilement en assembleur cette instruction, nous vous proposons en figure 1 la correspondance mémoire des différents paramètres de **SOUND**.

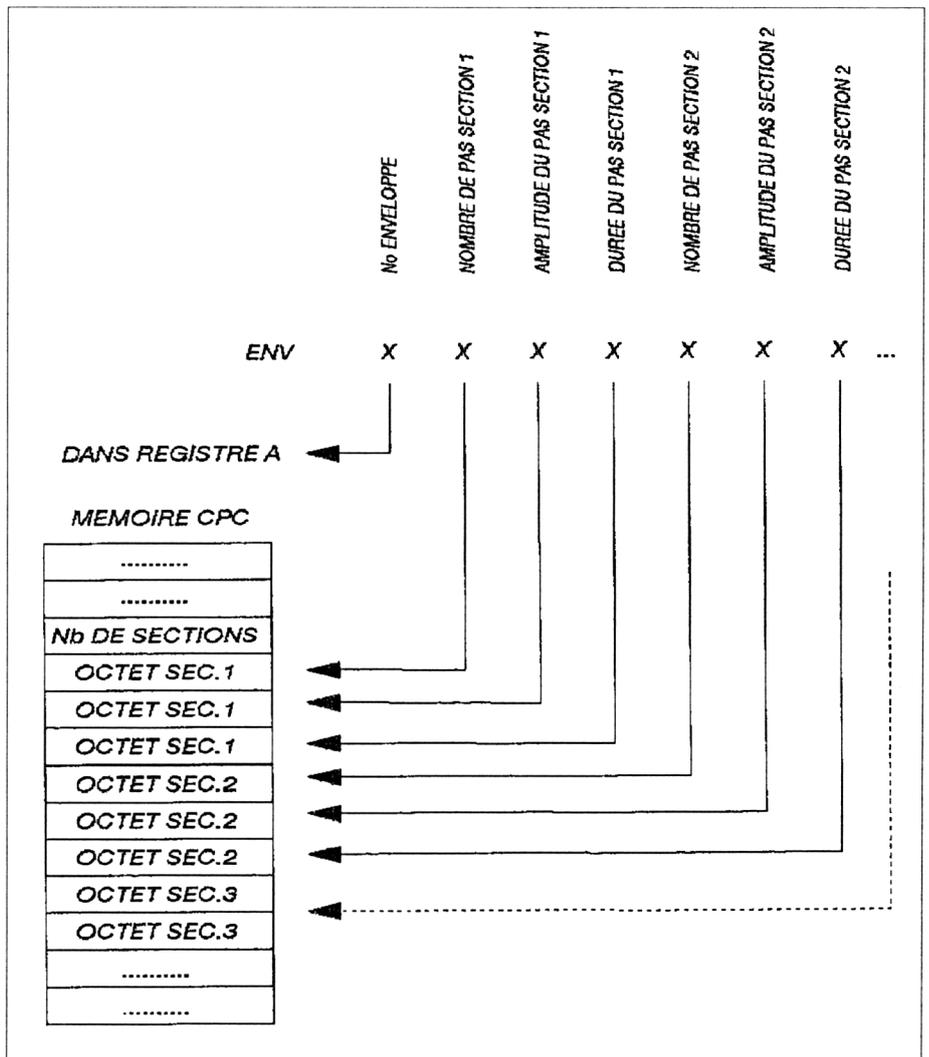


**Fig. 1** : Organisation dans la mémoire des paramètres nécessaires à l'appel du vecteur SOUND-QUEUE pour ajouter un son dans la queue sonore.

**SOUND AMPL VOLUME** Ce vecteur situé à l'adresse &BCBC est l'équivalent de l'instruction ENV du Basic.

L'appel de celui-ci nécessite le numéro de l'enveloppe de volume dans le registre A, et l'adresse de cette enveloppe dans le registre HL.

Vous trouverez en figure 2 la correspondance mémoire de l'enveloppe de définition d'amplitude.



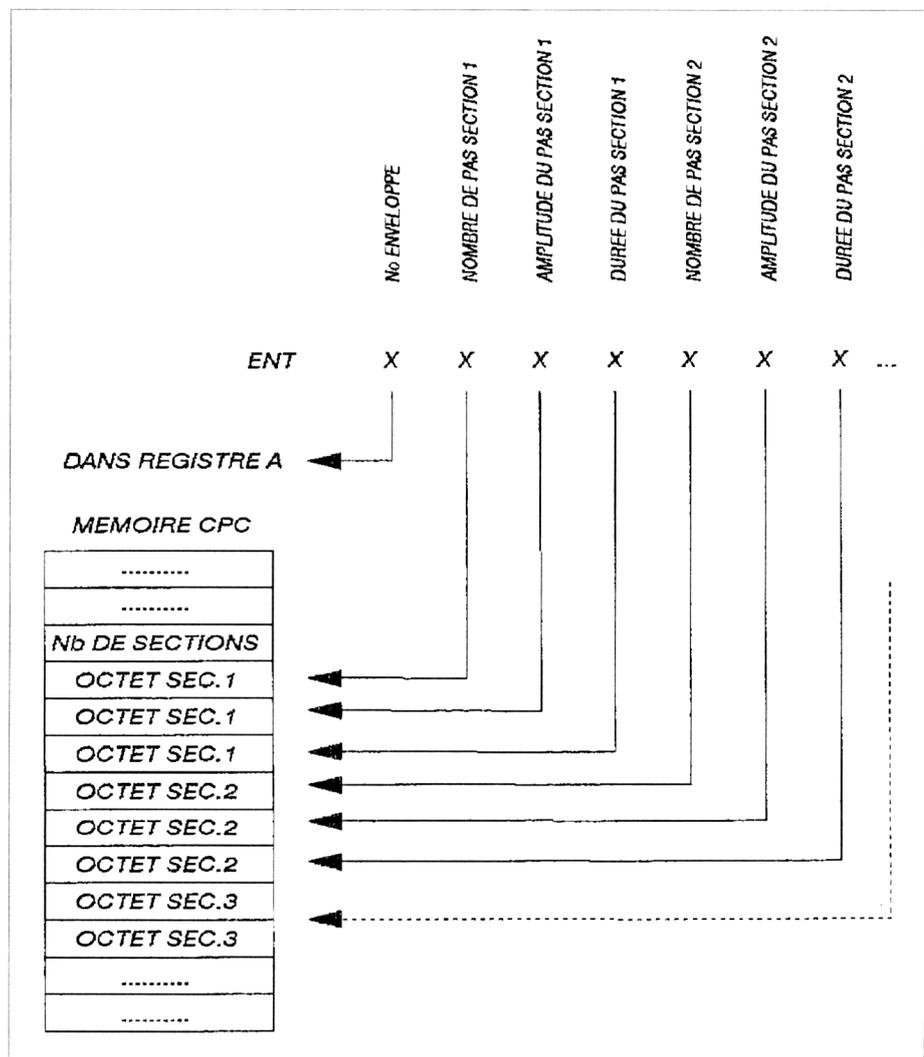
**Fig. 2 :** Organisation dans la mémoire des paramètres nécessaires à l'appel du vecteur SOUND-AMPL-VOLUME pour définir une amplitude de volume.

## SOUND TONE ENVELOPPE

Ce vecteur situé à l'adresse `&BCBF` est l'équivalent de l'instruction `ENT` du Basic.

L'appel de celui-ci nécessite le chargement des registres `A` et `HL` respectivement par le numéro de l'enveloppe et par l'adresse de cette enveloppe.

La figure 3 présente la correspondance mémoire avec les paramètres de l'instruction `ENT`.



**Fig. 3 :** Organisation dans la mémoire des paramètres nécessaires à l'appel du vecteur `SOUND-TONE-ENVELOPPE` pour définir une enveloppe de tonalité.

### LES INSTRUCTIONS RSX

- I REACTION**                      Cette instruction provoque le bruit d'un avion, ou de tout engin, utilisant la propulsion à réaction.
- I EXPLODE**                      Il n'est pas dit que la vie de votre avion soit infinie, et vous pourrez abréger celle-ci de la manière qui vous plaira, nous vous fournissons le bruitage de la triste fin de celui-ci.
- I PHONE**                         De quoi réveiller l'ami Gaston avec son « téléphone » qui « son » à l'aide de cette instruction provoquant le bruit d'une sonnerie de téléphone au demeurant fort moderne pour notre ami.
- I OCCUPE**                        Pas de chance, Gaston déjà en ligne vous signalera que cette dernière est occupée par cette instruction.
- I GO**                                Vous changez de moyen de transport, l'avion devenant trop dangereux, voici une instruction vous permettant de vous mettre au volant d'une formule 1 attendant le signal de départ.
- I TICTAC**                        Que le temps passe vite sur votre CPC, voici de quoi apprécier sa vitesse avec un tictac discret.
- I DONG**                         Big Ben dans votre CPC, rien d'étonnant, de part l'origine anglo-saxonne de celui-ci, voilà de quoi simuler l'heure du crime et ses douze coups par cette instruction paramétrable.

La syntaxe exacte est celle-ci :

**IDONG,nombredcoupsonnés**

Ainsi **I DONG,5** signalera à notre ami Dutronc qu'il est 5 heures et que Paris s'éveille.
- I PING et I PONG**                Non, ce ne sont pas les noms de deux auteurs japonais écrivant chez WEKA, mais deux instructions simulant les rebonds d'une balle (de pingpong vous avez dit ?).

## LE PROGRAMME ASSEMBLEUR

```

1      ;*****
2      ;*
3      ;* INSTRUCTIONS *
4      ;* PROPOSANT DES *
5      ;* EFFETS SPECIAUX *
6      ;* SOUS FORME DE *
7      ;* RSX BASIC *
8      ;*
9      ;*****
10     ;
11     ;
12     ;*** TABLE DES EQUIVALENCES ***
13     ;
14     ;insertion RSXs
15     ; KL-LOG-TEXT
16     KLOGTX: EQU 0BCD1H
17     ;
18     ;restauration PSG
19     ; SOUND RESET
20     SRESET: EQU 0BCA7H
21     ;
22     ;ajout d'un son
23     ; SOUND QUEUE
24     SQUEUE: EQU 0BCAAH
25     ;
26     ;test du buffer sonore
27     ; SOUND CHECK
28     SCHECK: EQU 0BCADH
29     ;
30     ;arme bloc de donnees
31     ; SOUND ARM EVENT
32     SARMEV: EQU 0BCB0H
33     ;
34     ;execute son en attente
35     ; SOUND RELEASE
36     SRELEA: EQU 0BCB3H
37     ;
38     ;stoppe un son
39     ; SOUND HOLD
40     SHOLD: EQU 0BCB6H
41     ;
42     ;continue un son
43     ; SOUND CONTINUE
44     SCONT: EQU 0BCB9H
45     ;
46     ;amplitude de volume
47     ; SOUND AMPL VOLUME
48     SAMPL: EQU 0BCBCH
49     ;
50     ;tonalite d'enveloppe
51     ; SOUND TONE ENVELOPPE
52     STENV: EQU 0BCBFH
53     ;
54     ;adresse buffer volume
55     ; SOUND A ADRESS
56     SAADDR: EQU 0BCC2H
57     ;
58     ;adresse buffer tonalite
59     ; SOUND T ADRESS
60     STADDR: EQU 0BCC5H

```

```

61          ;
62          ;
63          ;*****
64          ;
65          ;
66          ;*****
67          ;*   EMPLACEMENT DU *
68          ;*   PROGRAMME   *
69          ;*****
70          ;
71          ;           ORG   09800H
72          ;
73          ;           LOAD  09800H
74          ;
75          ;
76          ;*****
77          ;*   INSERTION DES *
78          ;*   RSXs       *
79          ;*****
80          ;
81          DEBUT:      EQU   $
82          9800 210F98  LD   HL,KERNEL      ;A RESERVER POUR RSX
83          9803 011398  LD   BC,VECTEU      ;TABLE INSTRUCTIONS
84          ;
85          9806 3EC9    LD   A,0C9H          ;POUR EVITER NOUVEL
86          9808 320098  LD   (DEBUT),A      ;APPEL
87          ;
88          980B CDD1BC  CALL KLOGTX      ;INSERER RSX
89          ;
90          980E C9     RET                    ;FIN
91          ;
92          ;
93          ;*** RESERVE POUR KERNEL
94          ;
95          KERNEL:    DEFS 4
96          ;
97          ;*** TABLE INSTRUCTIONS ***
98          ;
99          9813 3098   VECTEU:  DEFW TABLE
100         ;
101         9815 C35E98  JP   REAC          ;AVION A REACTION
102         9818 C37198  JP   EXPLOD       ;EXPLOSION
103         981B C39098  JP   PHONE        ;SONNERIE TELEPHONE
104         981E C30699  JP   OCCUPE       ;TELEPHONE OCCUPE
105         9821 C33199  JP   GO
106         9824 C36F99  JP   TICTAC
107         9827 C3CD99  JP   DONG
108         982A C3459A  JP   PING
109         982D C3999A  JP   PONG
110        ;
111        ;
112        TABLE:     EQU   $
113        ;
114         9830 52454143 DEFBS "REACTIO"
114         9834 54494F  DEFBS "N"+080H
115         9837 CE      DEFBS "N"+080H
116        ;
117         9838 4558504C DEFBS "EXPLOD"
117         983C 4F44    DEFBS "E"+080H
118         983E C5      DEFBS "E"+080H
    
```

```

119      ;
120 983F 50484F4E      DEFB "PHON"
121 9843 C5            DEFB "E"+080H
122      ;
123 9844 4F434355      DEFB "OCCUP"
123 9848 50
124 9849 C5            DEFB "E"+080H
125      ;
126 984A 47            DEFB "G"
127 984B CF            DEFB "O"+080H
128      ;
129 984C 54494354      DEFB "TICTA"
129 9850 41
130 9851 C3            DEFB "C"+080H
131      ;
132 9852 444F4E      DEFB "DON"
133 9855 C7            DEFB "G"+080H
134      ;
135 9856 50494E      DEFB "PIN"
136 9859 C7            DEFB "G"+080H
137      ;
138 985A 504F4E      DEFB "PON"
139 985D C7            DEFB "G"+080H
140      ;
141      ;
142      ;*****
143      ;* PROGRAMME EXECUTANT LES *
144      ;*          RSXs          *
145      ;*****
146      ;
147      ;*** AVION A REACTION ***
148      ;
149 REAC:      EQU $
150          CALL SRESET
151          LD HL,REAC1      ;BUFFER SON
152          CALL SQUEUE      ;EMET SON
153          RET              ;FIN
154      ;
155 REAC1:     EQU $          ;BUFFER SON
156          DEFB 1          ;CANAL
157          DEFB 1          ;AMPL ENV
158          DEFB 1          ;TON ENV
159          DEFW 0012H      ;PERIODE TON
160          DEFB 1          ;PERIODE BRUIT
161          DEFB 0FH        ;AMP INIT
162          DEFW 00C8H      ;DUREE ENV
163      ;
164      ;
165      ;*** EXPLOSION ***
166      ;
167 EXPLOD:    EQU $
168          CALL SRESET      ;INIT PSG
169      ;
170          LD B,1FH        ;31 FOIS
171 EXPLO3:    EQU $
172          LD A,B          ;POUR SAUVER
173          LD (EXPLO1+6),A ;MODIFICATION BRUIT
174          PUSH BC        ;SAUVEGARDE
175      ;
176 EXPLO2:    EQU $

```

```

177 987B 218798      LD  HL,EXPLO1      ;BUFFER SON
178 987E CDAABC      CALL SQUEUE       ;EMET SON
179 9881 30F8        JR   NC,EXPLO2    ;SI SON NON EMIS
180
181 9883 C1           ;
182 9884 10F0        POP  BC           ;ENCORE UN SON?
183 9886 C9           DJNZ EXPLO3       ;FIN
184
185                 ;
185                 EXPLO1: EQU $           ;BUFFER SON
186 9887 03          DEFB 03H
187 9888 00          DEFB 00H
188 9889 00          DEFB 00H
189 988A 9001        DEFW 0190H
190 988C 1F          DEFB 01FH
191 988D 00          DEFB 00H           ;BRUIT A MODIFIER
192 988E 3700        DEFW 0037H
193
194                 ;
195                 ;*** SONNERIE TELEPHONE ***
196                 ;
197                 PHONE: EQU $
198 9890 CDA7BC      CALL SRESET       ;INIT PSG
199 9893 0606        LD   B,06H        ;SIX SONNERIES
200
201                 ;
201                 PHONE4: EQU $
202 9895 3E64        LD   A,064H
203 9897 320099      LD   (PHONE1+3),A ;PREMIER BIP
204 989A 210600      LD   HL,0006H
205 989D 220499      LD   (PHONE1+7),HL ;PREMIER BRUIT
206 98A0 CDF298      CALL PHONE2      ;A EMETTRE
207 98A3 3E78        LD   A,078H
208 98A5 320099      LD   (PHONE1+3),A ;DEUXIEME BIP
209 98A8 CDF298      CALL PHONE2      ;A EMETTRE
210 98AB 3E50        LD   A,050H      ;TROISIEME BIP
211 98AD 320099      LD   (PHONE1+3),A ;TROISIEME BIP
212 98B0 CDF298      CALL PHONE2      ;A EMETTRE
213 98B3 210A00      LD   HL,000AH    ;BRUIT POUR
214 98B6 220499      LD   (PHONE1+7),HL ;QUATRIEME BIP
215 98B9 CDF298      CALL PHONE2      ;A EMETTRE
216 98BC 3E78        LD   A,078H
217 98BE 320099      LD   (PHONE1+3),A
218 98C1 CDF298      CALL PHONE2
219 98C4 3E50        LD   A,050H
220 98C6 320099      LD   (PHONE1+3),A
221 98C9 CDF298      CALL PHONE2
222 98CC 3E64        LD   A,064H
223 98CE 320099      LD   (PHONE1+3),A
224 98D1 CDF298      CALL PHONE2
225 98D4 210600      LD   HL,0006H
226 98D7 220499      LD   (PHONE1+7),HL
227 98DA CDF298      CALL PHONE2
228
229 98DD 210300      ;
229 98DD 210300      LD   HL,0003
230                 PHONE6: EQU $
231 98E0 E5          PUSH HL
232 98E1 21FFFF      LD   HL,0FFFFH
233                 PHONE5: EQU $
234 98E4 2B          DEC  HL           ;UN DE MOINS
235 98E5 7D          LD   A,L         ;L DANS A
236 98E6 B4          OR   H           ;POUR TESTER

```

```

237 98E7 20FB          JR  NZ,PHONE5          ;SI HL NUL
238 98E9 E1           POP  HL
239 98EA 2B           DEC  HL
240 98EB 7D           LD   A,L
241 98EC B4           OR   H
242 98ED 20F1         JR   NZ,PHONE6
243                   ;
244 98EF 10A4         DJNZ PHONE4          ;SONNERIE SUIVANTE
245 98F1 C9           RET                   ;FIN
246                   ;
247                   PHONE2: EQU  $
248 98F2 C5           PUSH BC              ;SAUVEGARDE BC
249                   PHONE3: EQU  $
250 98F3 21FD98       LD   HL,PHONE1       ;HL SUR BUFFER SON
251 98F6 CDAABC       CALL SQUEUE         ;EMET SON
252 98F9 30F8        JR   NC,PHONE3      ;SI SON NON EMIS
253 98FB C1           POP  BC              ;RECUPERE BC
254 98FC C9           RET                   ;FIN SOUS-PROGRAMME
255                   ;
256                   PHONE1: EQU  $          ;BUFFER SON
257 98FD 03           DEFB 03H
258 98FE 00           DEFB 00H
259 98FF 00           DEFB 00H
260 9900 0000        DEFW 0000H
261 9902 00           DEFB 00H
262 9903 1F           DEFB 01FH
263 9904 0000        DEFW 0000H
264                   ;
265                   ;
266                   ;*** TELEPHONE OCCUPE ***
267                   ;
268                   OCCUPE: EQU  $
269 9906 CDA7BC       CALL SRESET         ;INIT PSG
270 9909 0606        LD   B,06H         ;SIX SONNERIES OCCUPEE
271                   OCCUP4: EQU  $
272 990B C5           PUSH BC              ;SAUVE BC
273 990C 212899       LD   HL,OCCUP1     ;POINTE BUFFER SON
274 990F CDAABC       CALL SQUEUE         ;A EMETTRE
275                   ;
276 9912 210200       LD   HL,0002H      ;TEMPO
277                   OCCUP2: EQU  $
278 9915 E5           PUSH HL              ;SAUVER UNE FOIS
279 9916 21FFFF       LD   HL,0FFFFH
280                   OCCUP3: EQU  $
281 9919 2B           DEC  HL
282 991A 7D           LD   A,L
283 991B B4           OR   H
284 991C 20FB        JR   NZ,OCCUP3
285 991E E1           POP  HL
286 991F 2B           DEC  HL
287 9920 7D           LD   A,L
288 9921 B4           OR   H
289 9922 20F1        JR   NZ,OCCUP2
290 9924 C1           POP  BC
291 9925 10E4        DJNZ OCCUP4
292                   ;
293 9927 C9           RET
294                   ;
295                   OCCUP1: EQU  $          ;BUFFER SON
296 9928 03           DEFB 03H

```

```

297 9929 00          DEFB 00H
298 992A 00          DEFB 00H
299 992B 6400        DEFW 0064H
300 992D 00          DEFB 00H
301 992E 0F          DEFB 0FH
302 992F 3200        DEFW 0032H
303                  ;
304                  ;
305                  ;*** TOP DEPART ***
306                  ;
307 GO:              EQU $
308 9931 CDA7BC      CALL SRESET          ;INIT PSG
309 9934 0607        LD B,07H          ;SEPT FOIS
310 GO5:             EQU $
311 9936 C5          PUSH BC          ;SAUVE BC
312 GO2:             EQU $
313 9937 215D99      LD HL,GO1          ;BUFFER 1ER SON
314 993A CDAABC      CALL SQUEUE         ;A EMETTRE
315 993D 30F8        JR NC,GO2          ;SI SON NON EMIS
316                  ;
317 993F 210300      LD HL,0003H
318 GO4:             EQU $
319 9942 E5          PUSH HL
320 9943 21FFDF      LD HL,0DFFFH
321 GO3:             EQU $
322 9946 2B          DEC HL
323 9947 7D          LD A,L
324 9948 B4          OR H
325 9949 20FB        JR NZ,GO3
326 994B E1          POP HL
327 994C 2B          DEC HL
328 994D 7D          LD A,L
329 994E B4          OR H
330 994F 20F1        JR NZ,GO4
331                  ;
332 9951 C1          POP BC          ;RECUPERE BC
333 9952 10E2        DJNZ GO5          ;IL EN FAUT DIX
334                  ;
335 GO7:             EQU $
336 9954 216699      LD HL,GO6          ;BUFFER 2EME SON
337 9957 CDAABC      CALL SQUEUE
338 995A 30F8        JR NC,GO7
339                  ;
340 995C C9          RET          ;FIN
341                  ;
342 GO1:             EQU $          ;1ER SON
343 995D 03          DEFB 03H
344 995E 00          DEFB 00H
345 995F 00          DEFB 00H
346 9960 5000        DEFW 0050H
347 9962 00          DEFB 00H
348 9963 09          DEFB 09H
349 9964 2300        DEFW 0023H
350                  ;
351 GO6:             EQU $          ;2EME SON
352 9966 03          DEFB 03H
353 9967 00          DEFB 00H
354 9968 00          DEFB 00H
355 9969 1E00        DEFW 001EH
356 996B 00          DEFB 00H

```

```

357 996C 0C          DEFB 0CH
358 996D 9600       DEFW 0096H
359                 ;
360                 ;
361                 TICTAC: EQU $
362                 ;
363 996F CDA7BC      CALL SRESET          ;INIT PSG
364 9972 0602        LD B,02
365                 TICTA7: EQU $
366 9974 C5          PUSH BC
367 9975 3E02        LD A,02H
368 9977 21A999      LD HL,TICTA1        ;ENVELOPPE SON
369 997A CDBCBC      CALL SAMPL          ;A INITIALISER
370 997D 3E02        LD A,02H
371 997F 21AD99      LD HL,TICTA2        ;ENVELOPPE TON
372 9982 CDBFBC      CALL STENV         ;A INITIALISER
373                 TICTA5: EQU $
374 9985 21B499      LD HL,TICTA3
375 9988 CDAABC      CALL SQUEUE
376 998B 30F8        JR NC,TICTA5
377                 TICTA6: EQU $
378 998D 21BD99      LD HL,TICTA4
379 9990 CDAABC      CALL SQUEUE
380 9993 30F8        JR NC,TICTA6
381 9995 C1          POP BC
382 9996 10DC        DJNZ TICTA7
383 9998 3E02        LD A,02
384 999A 21C599      LD HL,TICTA8
385 999D CDBCBC      CALL SAMPL          ;RESET AMPLITUDE
386 99A0 3E02        LD A,02
387 99A2 21C999      LD HL,TICTA9
388 99A5 CDBFBC      CALL STENV         ;RESET TONALITE
389 99A8 C9          RET
390                 ;
391                 TICTA1: EQU $          ;ENVELOPPE AMPLITUDE
392 99A9 01          DEFB 01H
393 99AA 0F          DEFB 0FH
394 99AB FF          DEFB 0FFH
395 99AC 07          DEFB 07H
396                 ;
397                 TICTA2: EQU $          ;ENVELOPPE TONALITE
398 99AD 82          DEFB 082H        ;BIT 7 A 1
399 99AE 0A          DEFB 0AH
400 99AF FB          DEFB 0FBH
401 99B0 01          DEFB 01H
402 99B1 01          DEFB 01H
403 99B2 14          DEFB 14H
404 99B3 01          DEFB 01H
405                 ;
406                 TICTA3: EQU $          ;SON1 (TIC)
407 99B4 01          DEFB 01H
408 99B5 02          DEFB 02H
409 99B6 00          DEFB 00H
410 99B7 0000        DEFW 0000H
411 99B9 00          DEFB 00H
412 99BA 0F          DEFB 0FH
413 99BB 6900        DEFW 0069H
414                 ;
415                 TICTA4: EQU $          ;SON2 (TAC)
416 99BD 02          DEFB 02H

```

```

417 99BE 02          DEFB 02H
418 99BF 00          DEFB 00H
419 99C0 0000        DEFW 0000H
420 99C2 00          DEFB 00H
421 99C3 0F          DEFB 0FH
422 99C4 69          DEFB 0069H
423                  ;
424                  TICTA8: EQU $
425 99C5 01          DEFB 01H
426 99C6 000000      DEFB 00,00,00
427                  ;
428                  TICTA9: EQU $
429 99C9 81          DEFB 081H
430 99CA 000000      DEFB 00,00,00
431                  ;
432                  ;
433                  ;*** COUPS D'HORLOGE ***
434                  ;
435                  DONG: EQU $
436                  ;
437 99CD B7          OR A
438 99CE C8          RET Z          ;PAS DE PARAMETRES
439                  ;
440 99CF CDA7BC      CALL SRESET          ;INIT PSG
441 99D2 DD7E00      LD A,(IX+00)        ;RECUPERE PARAMETRE
442 99D5 C601        ADD A,1
443 99D7 47          LD B,A
444                  DONG7: EQU $
445 99D8 C5          PUSH BC          ;SAUVE BC
446 99D9 3E01        LD A,01H          ;ENT No 1
447 99DB 21279A      LD HL,DONG1       ;ENVELOPPE D'AMPLITUDE
448 99DE CDBCBC      CALL SAMPL        ;A INITIALISER
449 99E1 3E01        LD A,01H          ;ENT No 1
450 99E3 212B9A      LD HL,DONG2       ;ENVELOPPE DE TON
451 99E6 CDBFBC      CALL STENV        ;A INITIALISER
452                  DONG4: EQU $
453 99E9 21329A      LD HL,DONG3       ;POINTEUR BUFFER SON
454 99EC CDAABC      CALL SQUEUE
455 99EF 30F8        JR NC,DONG4        ;SI SON NON EMIS
456 99F1 C1          POP BC          ;RECUPERE BC
457 99F2 3E01        LD A,01H
458 99F4 A8          XOR B          ;VOIR SI EGAL
459 99F5 2819        JR Z,DONG10
460                  ;
461 99F7 210200      LD HL,0002H
462                  DONG6: EQU $
463 99FA E5          PUSH HL
464 99FB 21FFFF      LD HL,0FFFFH
465                  DONG5: EQU $
466 99FE 2B          DEC HL
467 99FF 7D          LD A,L
468 9A00 B4          OR H
469 9A01 20FB        JR NZ,DONG5
470 9A03 E1          POP HL
471 9A04 2B          DEC HL
472 9A05 7D          LD A,L
473 9A06 B4          OR H
474 9A07 20F1        JR NZ,DONG6
475 9A09 C5          PUSH BC
476 9A0A CDA7BC      CALL SRESET

```

```

477 9A0D C1          POP BC
478 9A0E 10C8       DJNZ DONG7
479                DONG10: EQU $
480 9A10 3E01       LD A,01H
481 9A12 213A9A     LD HL,DONG8
482 9A15 CDBCBC     CALL SAMPL          ;RESET AMPLITUDE
483 9A18 3E01       LD A,01H
484 9A1A 213E9A     LD HL,DONG9
485 9A1D CDBFBC     CALL STENV         ;RESET TONALITE
486 9A20 CDB6BC     CALL SHOLD
487 9A23 CDA7BC     CALL SRESET
488 9A26 C9         RET
489                ;
490                DONG1: EQU $          ;ENVELOPPE D'AMPLITUDE
491 9A27 01         DEFB 01H
492 9A28 0F         DEFB 0FH
493 9A29 FF         DEFB 0FFH
494 9A2A 0A         DEFB 0AH
495                ;
496                DONG2: EQU $          ;ENVELOPPE DE TONALITE
497 9A2B 82         DEFB 082H          ;BIT 7 A 1
498 9A2C 0A         DEFB 0AH
499 9A2D 04         DEFB 04H
500 9A2E 01         DEFB 01H
501 9A2F 01         DEFB 01H
502 9A30 EC         DEFB 0ECH
503 9A31 01         DEFB 01H
504                ;
505                DONG3: EQU $
506 9A32 02         DEFB 02H
507 9A33 01         DEFB 01H
508 9A34 02         DEFB 02H
509 9A35 9001       DEFW 0190H
510 9A37 00         DEFB 00H
511 9A38 0F         DEFB 0FH
512 9A39 00         DEFB 0000H
513                ;
514                DONG8: EQU $
515 9A3A 01         DEFB 01
516 9A3B 000000     DEFB 00,00,00
517                ;
518                DONG9: EQU $
519 9A3E 82         DEFB 82H
520 9A3F 00000000   DEFB 00,00,00,00,00,00
520 9A43 0000
521                ;
522                ;
523                ;*** REBOND PING ***
524                ;
525                PING: EQU $
526                ;
527 9A45 CDA7BC     CALL SRESET        ;INIT PSG
528 9A48 3E01       LD A,01H
529 9A4A 217A9A     LD HL,PING1
530 9A4D CDBCBC     CALL SAMPL         ;INIT ENV
531 9A50 3E01       LD A,01H
532 9A52 217E9A     LD HL,PING2
533 9A55 CDBFBC     CALL STENV         ;INIT ENT
534 9A58 21859A     LD HL,PING3
535 9A5B CDAABC     CALL SQUEUE       ;EMET PING

```

```

536                                     ;
537 9A5E 21FFFF                         LD  HL,0FFFFH
538                                     PING4: EQU  $
539 9A61 2B                               DEC  HL
540 9A62 7C                               LD   A,H
541 9A63 B5                               OR   L
542 9A64 20FB                             JR   NZ,PING4
543                                     ;
544 9A66 3E01                             LD   A,01
545 9A68 218E9A                          LD   HL,PING5
546 9A6B CDBCBC                          CALL SAMPL
547 9A6E 3E01                             LD   A,01
548 9A70 21929A                          LD   HL,PING6
549 9A73 CDBFBC                          CALL STENV
550 9A76 CDA7BC                          CALL SRESET
551                                     ;
552 9A79 C9                               RET
553                                     ;
554                                     PING1: EQU  $
555 9A7A 01                               DEFB 01H
556 9A7B 0F                               DEFB 0FH
557 9A7C FF                               DEFB 0FFH
558 9A7D 0A                               DEFB 0AH
559                                     ;
560                                     PING2: EQU  $
561 9A7E 82                               DEFB 082H
562 9A7F 0A                               DEFB 0AH
563 9A80 04                               DEFB 04H
564 9A81 01                               DEFB 01H
565 9A82 01                               DEFB 01H
566 9A83 EC                               DEFB 0ECH
567 9A84 01                               DEFB 01H
568                                     ;
569                                     PING3: EQU  $
570 9A85 02                               DEFB 02H
571 9A86 01                               DEFB 01H
572 9A87 01                               DEFB 01H
573 9A88 2003                             DEFW 0320H
574 9A8A 00                               DEFB 00H
575 9A8B 0F                               DEFB 0FH
576 9A8C 0000                             DEFW 0000H
577                                     ;
578                                     PING5: EQU  $
579 9A8E 01                               DEFB 01H
580 9A8F 000000                          DEFB 00,00,00
581                                     ;
582                                     PING6: EQU  $
583 9A92 02                               DEFB 02H
584 9A93 00000000                       DEFB 00,00,00,00,00,00
584 9A97 0000
585                                     ;
586                                     ;
587 ;*** REBOND PONG ***
588                                     ;
589 PONG: EQU  $
590                                     ;
591 9A99 CDA7BC                          CALL SRESET
592 9A9C 3E01                             LD   A,01H
593 9A9E 21CE9A                          LD   HL,PONG1
594 9AA1 CDBCBC                          CALL SAMPL

```

```

595 9AA4 3E01          LD   A,01H
596 9AA6 21D29A       LD   HL,PONG2
597 9AA9 CDBFBC       CALL STENV
598 9AAC 21859A       LD   HL,PING3
599 9AAF CDAABC       CALL SQUEUE
600                   ;
601 9AB2 21FFFF       LD   HL,0FFFFH
602                   PONG4: EQU $
603 9AB5 2B           DEC  HL
604 9AB6 7C           LD   A,H
605 9AB7 B5           OR   L
606 9AB8 20FB        JR   NZ,PONG4
607                   ;
608 9ABA 3E01          LD   A,01H
609 9ABC 21E29A       LD   HL,PONG5
610 9ABF CDBCBC       CALL SAMPL
611 9AC2 3E01          LD   A,01H
612 9AC4 21E69A       LD   HL,PONG6
613 9AC7 CDBFBC       CALL STENV
614 9ACA CDA7BC       CALL SRESET
615                   ;
616 9ACD C9           RET
617                   ;
618                   PONG1: EQU $
619 9ACE 01           DEFB 01H
620 9ACF 0F           DEFB 0FH
621 9AD0 FF           DEFB 0FFH
622 9AD1 0A           DEFB 0AH
623                   ;
624                   PONG2: EQU $
625 9AD2 82           DEFB 082H
626 9AD3 0A           DEFB 0AH
627 9AD4 06           DEFB 06H
628 9AD5 01           DEFB 01H
629 9AD6 02           DEFB 02H
630 9AD7 EC           DEFB 0ECH
631 9AD8 01           DEFB 01H
632                   ;
633                   PONG3: EQU $
634 9AD9 01           DEFB 01H
635 9ADA 01           DEFB 01H
636 9ADB 02           DEFB 02H
637 9ADC 2003        DEFW 0320H
638 9ADE 00           DEFB 00H
639 9ADF 0A           DEFB 0AH
640 9AE0 4000        DEFW 0040H
641                   ;
642                   PONG5: EQU $
643 9AE2 01           DEFB 01H
644 9AE3 000000      DEFB 00,00,00
645                   ;
646                   PONG6: EQU $
647 9AE6 02           DEFB 02H
648 9AE7 00000000    DEFB 00,00,00,00,00,00
648 9AEB 0000
649                   ;
650                   ;
651                   ;*** FIN DE PROGRAMME ***
652                   END

```

Lignes 11 à 60 : table des équivalences  
Lignes 82 à 139 : implantation des RSXs  
Lignes 149 à 162 : avion à réaction  
Lignes 167 à 192 : explosion  
Lignes 197 à 263 : téléphone sonne  
Lignes 268 à 302 : téléphone occupé  
Lignes 307 à 358 : signal top départ  
Lignes 361 à 430 : tic-tac  
Lignes 435 à 520 : coups d'horloge  
Lignes 525 à 584 : rebond ping  
Lignes 591 à 648 : rebond pong

## LE CHARGEUR BASIC

```

10 REM *****
20 REM **
30 REM ** GENERATION DES CODES **
40 REM ** MACHINES POUR CREER **
50 REM ** DES RSXs D'EFFETS **
60 REM ** SONORES **
70 REM **
80 REM *****
90 REM
100 REM *** RESERVATION MEMOIRE ***
110 REM
120 MEMORY &97FF
130 OPENOUT"bidon."
140 MEMORY HIMEM-1
150 CLOSEOUT
160 REM
170 controle = 0
180 REM
190 REM *** MISE EN MEMOIRE ***
200 REM
210 FOR A= 38912 TO 39659
220 READ B$:POKE A,VAL("&"+B$)
230 controle = controle + VAL("&"+B$)
240 NEXT A
250 REM
260 IF controle = 77713 THEN GOTO 380
270 MODE 2
280 PRINT CHR$(7)
290 PRINT "Vous avez fait une erreur"
300 PRINT "en entrant les DATAs ou le no
mbre de controle 77713"
310 PRINT "Verifiez les svp"
320 REM
330 PRINT:PRINT
340 STOP
350 REM
360 REM *** sauvegarde ***
370 REM
380 MODE 2
390 PRINT "SAUVEGARDE DU PROGRAMME BINAI
RE CREE"
400 PRINT:PRINT
410 SAVE "EFFETSON.BIN",B,&9800,748
420 PRINT:PRINT
430 PRINT "Le programme est sauvegarde s
ous le nom"
440 PRINT
450 PRINT " EFFETSON.BIN"
460 PRINT:PRINT
470 PRINT "Pour utiliser le fichier bina
ire ainsi cree,"
480 PRINT "inserez dans votre programme
les lignes suivantes:"
490 PRINT " MEMORY &97FF"
500 PRINT " OPENOUT " + CHR$(34) + "
BIDON.BID"+CHR$(34)
510 PRINT " MEMORY HIMEM - 1"

```

```
520 PRINT "      CLOSEOUT"
530 PRINT "      LOAD " + CHR$(34) + "EFF
ETSON.BIN" + CHR$(34) + ",&9800"
540 PRINT "      CALL &9800"
550 PRINT "      ... suite du programme
--"
560 PRINT:PRINT
570 END
580 REM
590 REM *** CODES HEXADECIMAUX ***
600 REM
610 REM *** BLOC 1 ***
620 REM
630 DATA 21,0F,98,01,13,98,3E,C9
640 DATA 32,00,98,CD,D1,BC,C9,0F
650 DATA 98,13,98,30,98,C3,5E,98
660 DATA C3,71,98,C3,90,98,C3,06
670 DATA 99,C3,31,99,C3,6F,99,C3
680 DATA CD,99,C3,45,9A,C3,99,9A
690 DATA 52,45,41,43,54,49,4F,CE
700 DATA 45,58,50,4C,4F,44,C5,50
710 REM
720 REM *** BLOC 2 ***
730 REM
740 DATA 48,4F,4E,C5,4F,43,43,55
750 DATA 50,C5,47,CF,54,49,43,54
760 DATA 41,C3,44,4F,4E,C7,50,49
770 DATA 4E,C7,50,4F,4E,C7,CD,A7
780 DATA BC,21,68,98,CD,AA,BC,C9
790 DATA 01,01,01,12,00,01,0F,C8
800 DATA 00,CD,A7,BC,06,1F,78,32
810 DATA 8D,98,C5,21,87,98,CD,AA
820 REM
830 REM *** BLOC 3 ***
840 REM
850 DATA BC,30,F8,C1,10,F0,C9,03
860 DATA 00,00,90,01,1F,00,37,00
870 DATA CD,A7,BC,06,06,3E,64,32
880 DATA 00,99,21,06,00,22,04,99
890 DATA CD,F2,98,3E,78,32,00,99
900 DATA CD,F2,98,3E,50,32,00,99
910 DATA CD,F2,98,21,0A,00,22,04
920 DATA 99,CD,F2,98,3E,78,32,00
930 REM
940 REM *** BLOC 4 ***
950 REM
960 DATA 99,CD,F2,98,3E,50,32,00
970 DATA 99,CD,F2,98,3E,64,32,00
980 DATA 99,CD,F2,98,21,06,00,22
990 DATA 04,99,CD,F2,98,21,03,00
1000 DATA E5,21,FF,FF,2B,7D,B4,20
1010 DATA FB,E1,2B,7D,B4,20,F1,10
1020 DATA A4,C9,C5,21,FD,98,CD,AA
1030 DATA BC,30,F8,C1,C9,03,00,00
1040 REM
1050 REM *** BLOC 5 ***
1060 REM
```

```
1070 DATA 00,00,00,1F,00,00,CD,A7
1080 DATA BC,06,06,C5,21,28,99,CD
1090 DATA AA,BC,21,02,00,E5,21,FF
1100 DATA FF,2B,7D,B4,20,FB,E1,2B
1110 DATA 7D,B4,20,F1,C1,10,E4,C9
1120 DATA 03,00,00,64,00,00,0F,32
1130 DATA 00,CD,A7,BC,06,07,C5,21
1140 DATA 5D,99,CD,AA,BC,30,F8,21
1150 REM
1160 REM *** BLOC 6 ***
1170 REM
1180 DATA 03,00,E5,21,FF,DF,2B,7D
1190 DATA B4,20,FB,E1,2B,7D,B4,20
1200 DATA F1,C1,10,E2,21,66,99,CD
1210 DATA AA,BC,30,F8,C9,03,00,00
1220 DATA 50,00,00,09,23,00,03,00
1230 DATA 00,1E,00,00,0C,96,00,CD
1240 DATA A7,BC,06,02,C5,3E,02,21
1250 DATA A9,99,CD,BC,BC,3E,02,21
1260 REM
1270 REM *** BLOC 7 ***
1280 REM
1290 DATA AD,99,CD,BF,BC,21,B4,99
1300 DATA CD,AA,BC,30,F8,21,BD,99
1310 DATA CD,AA,BC,30,F8,C1,10,DC
1320 DATA 3E,02,21,C5,99,CD,BC,BC
1330 DATA 3E,02,21,C9,99,CD,BF,BC
1340 DATA C9,01,0F,FF,07,82,0A,FB
1350 DATA 01,01,14,01,01,02,00,00
1360 DATA 00,00,0F,69,00,02,02,00
1370 REM
1380 REM *** BLOC 8 ***
1390 REM
1400 DATA 00,00,00,0F,69,01,00,00
1410 DATA 00,81,00,00,00,B7,C8,CD
1420 DATA A7,BC,DD,7E,00,C6,01,47
1430 DATA C5,3E,01,21,27,9A,CD,BC
1440 DATA BC,3E,01,21,2B,9A,CD,BF
1450 DATA BC,21,32,9A,CD,AA,BC,30
1460 DATA F8,C1,3E,01,A8,28,19,21
1470 DATA 02,00,E5,21,FF,FF,2B,7D
1480 REM
1490 REM *** BLOC 9 ***
1500 REM
1510 DATA B4,20,FB,E1,2B,7D,B4,20
1520 DATA F1,C5,CD,A7,BC,C1,10,C8
1530 DATA 3E,01,21,3A,9A,CD,BC,BC
1540 DATA 3E,01,21,3E,9A,CD,BF,BC
1550 DATA CD,B6,BC,CD,A7,BC,C9,01
1560 DATA 0F,FF,0A,82,0A,04,01,01
1570 DATA EC,01,02,01,02,90,01,00
1580 DATA 0F,00,01,00,00,00,82,00
1590 REM
1600 REM *** BLOC 10 ***
1610 REM
1620 DATA 00,00,00,00,00,CD,A7,BC
1630 DATA 3E,01,21,7A,9A,CD,BC,BC
```

```
1640 DATA 3E,01,21,7E,9A,CD,BF,BC
1650 DATA 21,85,9A,CD,AA,BC,21,FF
1660 DATA FF,2B,7C,B5,20,FB,3E,01
1670 DATA 21,8E,9A,CD,BC,BC,3E,01
1680 DATA 21,92,9A,CD,BF,BC,CD,A7
1690 DATA BC,C9,01,0F,FF,0A,82,0A
1700 REM
1710 REM *** BLOC 11 ***
1720 REM
1730 DATA 04,01,01,EC,01,02,01,01
1740 DATA 20,03,00,0F,00,00,01,00
1750 DATA 00,00,02,00,00,00,00,00
1760 DATA 00,CD,A7,BC,3E,01,21,CE
1770 DATA 9A,CD,BC,BC,3E,01,21,D2
1780 DATA 9A,CD,BF,BC,21,85,9A,CD
1790 DATA AA,BC,21,FF,FF,2B,7C,B5
1800 DATA 20,FB,3E,01,21,E2,9A,CD
1810 REM
1820 REM *** BLOC 12 ***
1830 REM
1840 DATA BC,BC,3E,01,21,E6,9A,CD
1850 DATA BF,BC,CD,A7,BC,C9,01,0F
1860 DATA FF,0A,82,0A,06,01,02,EC
1870 DATA 01,01,01,02,20,03,00,0A
1880 DATA 40,00,01,00,00,00,02,00
1890 DATA 00,00,00,00,00,00,00,00
1900 REM
1910 REM *** FIN DES CODES ***
1920 REM
1930 DATA FIN
1940 END
```

Un bon conseil : de par leur nombre élevé, vous aurez soin de vérifier chacune des lignes de DATA que vous entrerez avant de la valider.

Après sauvegarde, vous lancerez le programme Basic et suivrez les instructions affichées.

Pour utiliser le programme, il vous suffira d'entrer les lignes suivantes :

```
10 MEMORY &97FF
20 OPENOUT "TRUC."
30 MEMORY HIMEM - 1
40 CLOSEOUT
50 LOAD "EFFETSON.BIN",&9800
60 CALL &9800
```

### DÉMONSTRATION

Le petit programme suivant vous permettra de vérifier le bon fonctionnement de votre programme de création de RSX sonores.

```

10 REM *****
20 REM **                               **
30 REM **      PROGRAMME DE           **
40 REM **      DEMONSTRATION DES      **
50 REM **      EFFET SONORES         **
60 REM **                               **
70 REM *****
80 REM
90 CALL &BB4E
100 MEMORY &97FF
110 OPENOUT "BIDON.BID"
120 MEMORY HIMEM - 1
130 CLOSEOUT
140 REM
150 LOAD "EFFETSON.BIN",&9800
160 CALL &9800
170 EVERY 50,1 GOSUB 1020
180 REM *****
190 REM *****
200 MODE 2
210 LOCATE 19,9
220 PRINT "APPUYEZ SUR UNE TOUCHE POUR A
RRETER"
230 LOCATE 22,12
240 PRINT "L'HORLOGE ET CONTINUER LA DEM
O"
250 WHILE INKEY$ = ""
260     X% = INT(598 * RND) + 1
270     Y% = INT(398 * RND) + 1
280     PLOT X%,Y%
290 WEND
300 x = REMAIN(1)
310 REM *****
320 REM *****
330 SYMBOL AFTER 250
340 REM
350 MODE 0
360 LOCATE 5,10
370 PRINT "C'EST PARTI
380 |GO : REM C'EST PARTI
390 MODE 2
400 REM *****
410 REM *****
420 SYMBOL 250,0,0,0,0,63,255,0
430 SYMBOL 251,0,2,6,14,254,254,0,0
440 REM
450 FOR I% = 78 TO 1 STEP -1
460     |REACTION
470     LOCATE I%,10
480     FRAME
490     PRINT CHR$(250);CHR$(251);CHR$(32
)
500     FOR J% = 1 TO 10
510     NEXT J%
520 NEXT I%
530 LOCATE 1,10
540 PRINT CHR$(127);CHR$(127)

```

```

550 |EXPLODE
560 REM *****
570 REM *****
580 MODE 1
590 LOCATE 2,2
600 PRINT "ALLO ... ICI GASTON"
610 |PHONE
620 LOCATE 2,2
630 PRINT "ZUT ... Y A PERSON QUI Y REP
OND"
640 |OCCUPE
650 REM *****
660 REM *****
670 MODE 2
680 LOCATE 1,6
690 PRINT CHR$(209)
700 PRINT CHR$(209)
710 LOCATE 70,6
720 PRINT CHR$(211)
730 LOCATE 70,7
740 PRINT CHR$(211)
750 FOR K% = 1 TO 10
760   |PING
770   FOR I% = 2 TO 68
780     LOCATE I%,6
790     PRINT CHR$(32)
800     LOCATE I% + 1,6
810     PRINT CHR$(230)
820   NEXT I%
830   |PONG
840   FOR I% = 68 TO 2 STEP -1
850     LOCATE I% + 1,6
860     PRINT CHR$(32)
870     LOCATE I%,6
880     PRINT CHR$(230)
890   NEXT I%
900 NEXT K%
910 REM *****
920 REM *****
930 MODE 0
940 LOCATE 4,10
950 PRINT "C'EST FINI"
960 X = REMAIN(1)
970 |DONG,12
980 LOCATE 4,10
990 PRINT "BONNE NUIT"
1000 PRINT:PRINT
1010 END
1020 |TICTAC
1030 RETURN

```



# 6/5

## Logiciels

---

### 6/5.1

## Fichiers musicaux sous interruptions

---

Les programmes qui suivent vous permettront de créer simplement des fichiers musicaux exécutables sous interruptions (IT en abrégé), et donc pendant qu'un autre programme se déroule.

Deux programmes sont présentés :

- le premier montre comment activer le processeur sonore sous IT ;
- le second permet de créer des fichiers musicaux exécutables sous IT.

Examinons en détail le système d'IT existant sur le circuit sonore AY3-8912.

#### **ACTIVATION DU PROCESSUS SONORE SOUS IT**

Tout d'abord, signalons aux personnes qui ne sont pas habituées à manipuler les IT que les choses se déroulent en deux temps :

- 1°) Déclaration de l'IT et armement,
- 2°) Définition de la routine qui sera exécutée à chaque IT.

*1°) Déclaration de l'IT et armement :*

Pour déclarer l'IT, nous ferons appel à la macro du firmware « INITEV » (KL INIT EVEN BLOCK #BCEF), dont le but est de définir le type d'IT et son adresse d'exécution (Reportez-vous en 4/2.7 pour avoir plus de détails sur cette macro).

Pour lancer l'IT, nous faisons appel à la macro du firmware « ADDEVE » (KL ADD TICKER #BCE9) dont le but est de définir l'intervalle de temps entre deux IT consécutives. (Pour plus de détails, reportez-vous à la partie 4 chap. 2.7).

*2°) Définition de la routine qui sera exécutée à chaque IT :*

Ces deux routines appelées, il faut définir la routine qui sera exécutée à chaque IT.

La première opération à faire dans une telle routine est de dévalider les interruptions (DI), la deuxième est de sauvegarder les registres qui risquent d'être écrasés (PUSH AF, BC, DE, HL, IX et IY).

Ensuite, la routine d'IT peut être décrite.

Dans notre cas, si l'IT est exécutée pour la première fois, il faudra définir une enveloppe de volume par la macro du firmware « ENV ». Reportez-vous à la partie 4 chap. 2.7 pour avoir plus de détails. Ensuite, que ce soit la première fois ou non, nous allons activer la voie de droite (1), puis les voies de droite et de gauche simultanément (4), puis la voie de gauche (2) par la macro du firmware « SOUND ». Reportez-vous à la partie 4 chap. 1.2.7 pour avoir plus de détails. Si le morceau est fini, le pointeur de notes sera remis au début du morceau ; autrement dit, le morceau s'exécutera sans fin.

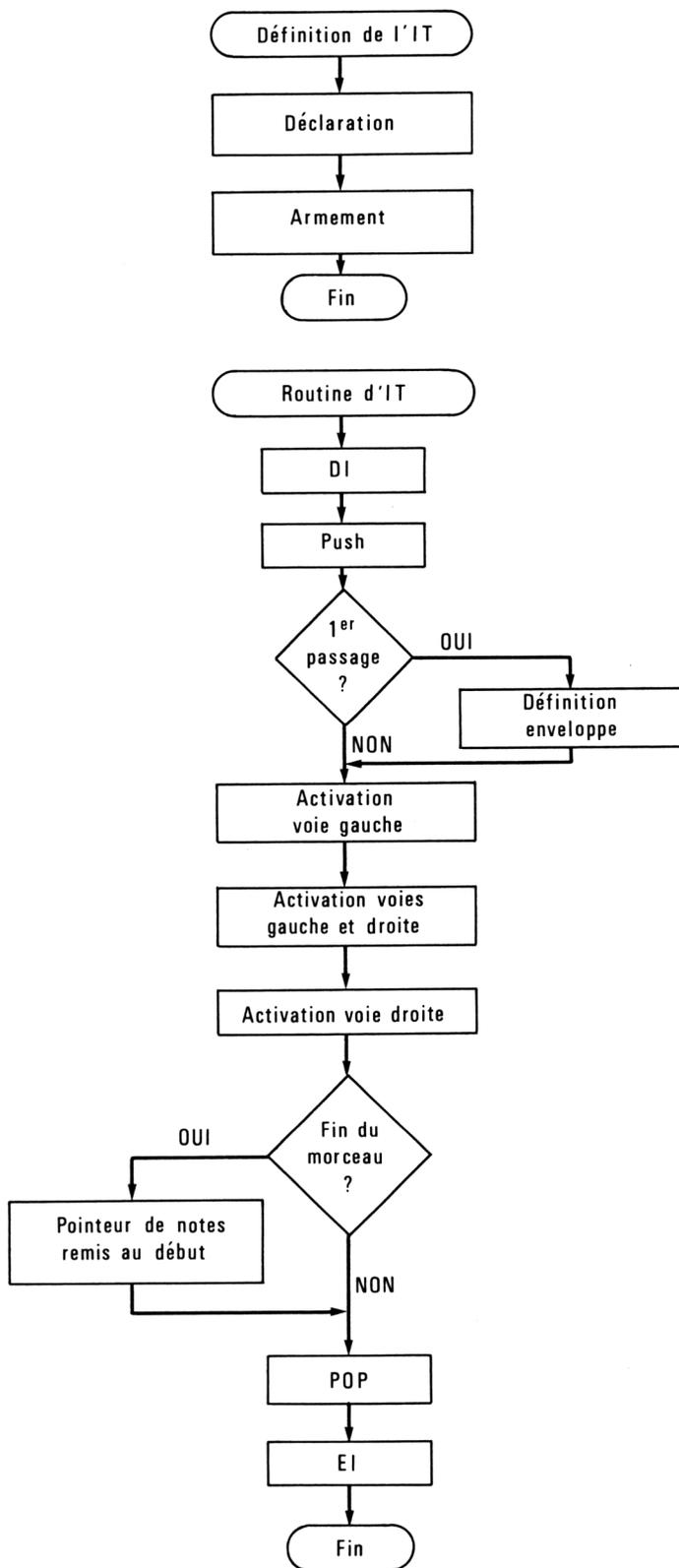
L'IT doit se terminer par la restitution des registres (POP IY, IX, HL, DE, BC, AF) dans l'ordre inverse où ils ont été sauvegardés, et par la revalidation des interruptions (EI), d'où l'organigramme suivant (*Voir page 3*) .

Cet organigramme donne lieu au programme suivant :

```

1          ORG  9000H
2          LOAD 9000H
3 9000 C32F90          JP  SON          ;Debut de programme
4 9003 01000000 SOUND0: DB  1,0,0,0,0,0,0,22,0
4 9007 00000016
4 900B 00
5 900C 01010000 SOUND1: DB  1,1,0,0,0,0,0,22,0
5 9010 00000016
5 9014 00

```



```

6 9015 03020701 ENV1:      DB   3,2,7,1,1,0,2,5,25
6 9019 01000205
6 901D FF04
7 901F 00000000 EVBL:     DB   0,0,0,0,0,0,0,0,0,0,
7 9023 00000000
7 9027 00000000
8 902B 00                  DB   0
9 902C 0000      SAVMUS:   DW   0
10 902E 00      AIG:      DB   0                ;iere IT
11              ENV:      EQU  0BCBCH          ;Init ENV Volume
12              SOUND:    EQU  0BCAAH          ;Ajout d'1 son
13              INITEV:   EQU  0BCEFH          ;INIT EVEN BLOC
14              ADDEVE:   EQU  0BCE9H          ;ADD EVEN BLOC
15              ;-----
16              SON:      EQU  $                ;Activation de l'IT
17 902F 212590          LD   HL,EVBL+6          ;A TICKLIST
18 9032 0681           LD   B,B1H                ;TYPE D'IT
19 9034 0E00           LD   C,0
20 9036 114990          LD   DE,RTSON          ;A touche act. si IT
21 9039 CDEFBC          CALL INITEV            ;INIT EVEN BLOC
22 903C 211F90          LD   HL,EVBL          ;A TICK BLOC
23 903F 110100          LD   DE,1
24 9042 010B00          LD   BC,11
25 9045 CDE9BC          CALL ADDEVE            ;ADD EVEN BLOC
26 9048 C9             RET
27              RTSON:    EQU  $
28 9049 F3             DI
29 904A F5             PUSH AF
30 904B C5             PUSH BC
31 904C D5             PUSH DE
32 904D E5             PUSH HL
33 904E DDE5           PUSH IX

```

```

34 9050 FDE5          PUSH IY
35 9052 3A2E90        LD  A,(AIB)          ;Test si 1ere IT
36 9055 B7            OR  A
37 9056 2015          JR  NZ,SON1         ;Non
38 9058 3E01          LD  A,1
39 905A 322E90        LD  (AIB),A         ;Oui
40 905D 3E01          LD  A,1             ;No enveloppe
41 905F 211590        LD  HL,ENV1         ;À Def enveloppe
42 9062 CDBCBC        CALL ENV            ;Init env vol
43 9065 FD21D390      LD  IY,NOTES
44 9069 FD222C90      LD  (SAVMUS),IY    ;Note en cours
45                      SON1: EQU $
46 906D FD2A2C90      LD  IY,(SAVMUS)    ;à prochaine note
47 9071 0E01          LD  C,1             ;Canal A active
48 9073 CDA090        CALL SONN
49 9076 0E04          LD  C,4             ;2 canaux actives
50 9078 CDA090        CALL SONN
51 907B 0E02          LD  C,2             ;Canal B active
52 907D CDA090        CALL SONN
53 9080 FD7E00        LD  A,(IY+0)
54 9083 FEFF          CP  OFFH
55 9085 200B          JR  NZ,F11
56 9087 FD7E01        LD  A,(IY+1)
57 908A FEFF          CP  OFFH
58 908C 2004          JR  NZ,F11
59 908E FD21D390      LD  IY,NOTES
60                      F11: EQU $
61 9092 FD222C90      LD  (SAVMUS),IY
62 9096 FDE1          POP IY
63 9098 DDE1          POP IX
64 909A E1            POP HL
65 909B D1            POP DE
66 909C C1            POP BC

```

```

67 909D F1          POP  AF
68 909E FB          EI
69 909F C9          RET
70                  SONN: EQU  $
71 90A0 FD7E00      LD   A,(IY+0)      ;Note
72 90A3 B7          OR   A
73 90A4 200F        JR   NZ,NOTE      ;Activation note
74 90A6 FD7E01      LD   A,(IY+1)
75 90A9 B7          OR   A
76 90AA 2009        JR   NZ,NOTE      ;Activation note
77 90AC DD210390    LD   IX,SOUND0    ;Pas de note
78 90B0 DD7100      LD   (IX+0),C     ;Canal
79 90B3 1813        JR   FSON1
80                  NOTE: EQU  $
81 90B5 DD210C90    LD   IX,SOUND1    ;Note
82 90B9 DD7100      LD   (IX+0),C     ;Canal
83 90BC FD7E00      LD   A,(IY+0)
84 90BF DD7703      LD   (IX+3),A     ;LSB Note
85 90C2 FD7E01      LD   A,(IY+1)
86 90C5 DD7704      LD   (IX+4),A     ;MSB Note
87                  FSON1: EQU  $
88 90C8 DDE5        PUSH IX
89 90CA E1          POP  HL          ;Note a jouer
90 90CB CDAABC      CALL SOUND        ;Ajoute 1 son ds queue
91 90CE FD23        INC  IY
92 90D0 FD23        INC  IY
93 90D2 C9          RET
94                  ;-----
95                  NOTES: EQU  $          ;Implantation notes
96                  END

```

Lignes 1 à 14 : Déclarations diverses.

Lignes 17 à 26 : Routine d'activation d'IT.

Lignes 28 à 93 : Routine d'IT.

Ligne 28 : — Entrée dans l'IT.

Ligne 38 : — Premier passage dans l'IT.

Lignes 47 à 52 : — Activation des canaux 1, 4 et 2.

Lignes 53 à 58 : — Test fin de morceau.

Lignes 62 à 69 : — Fin de l'IT.

### **CRÉATION DE FICHIERS MUSICAUX EXÉCUTABLES SOUS INTERRUPTIONS**

Si vous possédez une partition et si vous savez la déchiffrer, il vous sera très simple de créer des fichiers musicaux exécutables sous IT par l'éditeur musical que nous présentons ici. Cet éditeur ne permet de générer que des notes ayant la même durée.

Si vous désirez créer des notes ayant des longueurs différentes, il sera nécessaire de :

1°) modifier le programme d'IT en définissant une enveloppe de volume différente pour chaque longueur de note,

2°) valider l'enveloppe correspondant à la durée de note à jouer avant chaque émission de note.

Dès le début du programme, un menu apparaît à l'écran.

Vous pouvez :

- charger un morceau,
- écrire un morceau,
- l'exécuter,
- le sauvegarder.

L'écriture du morceau fait aussi appel à un menu. Vous pouvez :

- entrer une note,
- corriger une note,
- supprimer une note.

L'option « exécution du morceau » fait entendre les trois voies simultanément, du début à la fin du morceau, en exécutant le programme d'interruptions cité précédemment.

L'option « Sauvegarde » permet de créer un fichier musical directement utilisable par le programme d'IT. Ce fichier pourra être repris par la suite par l'éditeur musical, ou servir de support sonore à l'exécution d'un jeu (par exemple).

L'option « Chargement » permet de mettre en mémoire un fichier musical afin d'y faire des modifications ou d'en continuer l'écriture.

L'éditeur musical correspond au listing suivant :

```

1000 REM EDITEUR MUSICAL
1010 '
1020 '
1030 '
1040 '
1050 '
1060 GOSUB 1110 'Initialisation
1070 GOSUB 1360 'Menu principal
1080 '
1090 END
1100 '-----
1110 REM Initialisation
1120 '
1130 INK 1,10:INK 0,0:BORDER 0:MODE 2
1140 DIM NN(300,3),T1(96),no$(100,3)
1150 '
1160 'Musique sous IT
1170 FOR I=&9000 TO &90D2:READ A:POKE I,A:NEXT
1180 DATA &C3,&2F,&90,&1,&0,&0,&0,&0,&0,&0,&16,&0,&1,&1,&0,&0,&0,&0,&0,&16,&0,&3
,&2,&7,&1,&1,&0,&2,&5,&FF,&4,&0,&0,&0,&0,&0,&0,&0,&0
,&0,&0,&0,&0,&0,&0,&0,&0
1190 DATA &21,&25,&90,&6,&B1,&E,&0,&11,&49,&90,&CD,&EF,&BC,&21,&1F,&90,&11,&1,&0
,&1,&B,&0,&CD,&E9,&BC,&C9,&F3,&F5,&C5,&D5,&E5,&DD,&E
5,&FD,&E5,&3A,&2E,&90,&B7,&20,&15,&3E,&1,&32,&2E,&90,&3E,&1,&21,&15,&90,&CD,&BC,
&BC,&FD,&21,&D3,&90,&FD,&22,&2C,&90
1200 DATA &FD,&2A,&2C,&90,&E,&1,&CD,&A0,&90,&E,&4,&CD,&A0,&90,&E,&2,&CD,&A0,&90,
&FD,&7E,&0,&FE,&FF,&20,&B,&FD,&7E,&1,&FE,&FF,&20,&4,
&FD,&21,&D3,&90,&FD,&22,&2C,&90,&FD,&E1,&DD,&E1,&E1,&D1,&C1,&F1,&FB,&C9,&FD,&7E,
&0,&B7,&20,&F,&FD,&7E,&1,&B7,&20,&9,&DD
1210 DATA &21,&3,&90,&DD,&71,&0,&1B,&13,&DD,&21,&C,&90,&DD,&71,&0,&FD,&7E,&0,&DD
,&77,&3,&FD,&7E,&1,&DD,&77,&4,&DD,&E5,&E1,&CD,&AA,&B
C,&FD,&23,&FD,&23,&C9
1220 '-----

```

```
1230 FOR I=1 TO 96
1240   READ T1(I)
1250 NEXT I
1260 DATA 3822,3608,3405,3214,3034,2863,2703,2551,2408,2273,2145,2025
1270 DATA 1911,1804,1703,1607,1517,1432,1351,1276,1204,1136,1073,1012
1280 DATA 956,902,851,804,758,716,676,638,602,568,536,506
1290 DATA 478,451,426,402,379,358,338,319,301,284,268,253
1300 DATA 239,225,213,201,190,179,169,159,150,142,134,127
1310 DATA 119,113,106,100,95,89,84,80,75,71,67,63
1320 DATA 60,56,53,50,47,45,42,40,38,36,34,32
1330 DATA 30,28,27,25,24,22,21,20,19,18,17,16
1340 '
1350 RETURN
1360 REM Menu principal
1370 '
1380 CLS:PRINT"EDITEUR MUSICAL":PRINT
1390 PRINT"1)Charger un morceau"
1400 PRINT"2)Ecriture du morceau"
1410 PRINT"3)Execution du morceau"
1420 PRINT"4)Sauvegarde magnetique"
1430 PRINT:PRINT"Votre choix"
1440 C#=INKEY#:IF C#="" THEN 1440
1450 C=ASC(C#)-48
1460 IF C>4 OR C<1 THEN SOUND 1,100,30:GOTO 1360 'Choix refuse
1470 POKE &900D,0 'Plus de musique
1480 ON C GOSUB 3270,1590,3000,3480
1490 GOTO 1360
1500 '-----
1590 REM Ecriture du morceau
1600 CLS:PRINT"Ecriture du morceau":PRINT
1610 PRINT"Voulez vous: 1)Entrer des notes"
1620 PRINT"                2)Corriger une note"
```

```
1630 PRINT"                3)Supprimer une note"
1640 A$=INKEY$:IF A$="" THEN 1640
1650 A=ASC(A$)-48
1660 IF ASC(A$)=13 THEN RETURN 'Passage au niveau superieur
1670 IF A<>1 AND A<>2 AND A<>3 THEN SOUND 1,100,30:GOTO 1640
1680 ON A GOSUB 1700,2170,2250
1690 GOTO 1590
1700 '-----
1710 'Entree des notes
1715
1720 CLS
1730 IF N(1)+N(2)+N(3)=0 THEN PRINT"Aucune note en memoire":GOTO 1820
1740 GOSUB 2370 'Affichage des notes en memoire
1750 PRINT:PRINT"Insertion avant cette note (O/N)"
1760 A$=INKEY$:IF A$="" THEN 1760
1770 A$=UPPER$(A$):IF A$<>"O" AND A$<>"N" THEN SOUND 1,100,30:GOTO 1760
1780 IF A$="N" THEN RETURN
1790 AJ=1
1800 LOCATE 1,13:FOR I=1 TO 4:PRINT BAL$;:NEXT
1810 GOTO 1910
1820 GOSUB 2960 'Affichage du "HELP" d'écriture
1830 PRINT:PRINT"Voie (1,2 ou 3) "CHR$(143)
1840 LOCATE 1,15:PRINT:PRINT BAL$
1850 A$=INKEY$:IF A$="" THEN 1850
1860 IF ASC(A$)=13 THEN RETURN
1870 VX=ASC(A$)-48
1880 IF VX>3 OR VX<1 THEN SOUND 1,100,30:GOTO 1850 'Option refusee
1890 LOCATE 16,22:PRINT VX
1900 X=1:Y=16
1910 LOCATE X,Y:SOUND 1,200,10
1920 INPUT A$ 'Lecture Note
1930 IF LEN(A$)=0 AND AJ=0 THEN LOCATE 1,21:GOTO 1830
```

```

1940 IF LEN(A$)=0 THEN RETURN

1950 GOSUB 2050 'Analyse de la note

1960 IF B=1 THEN SOUND 1,300,30:LOCATE X,Y:PRINT SPACE$(80-X):GOTO 1910 'Note re
fusee

1970 IF AJ=1 THEN 2000 'Ajout de notes

1980 NO$(N(VX),VX)=A$:N(VX)=N(VX)+1 Memorisation

1990 NT=NT+1:IF X<>73 THEN LOCATE X,Y:PRINT" ":LOCATE X,Y+1:PRINT" ":X=X+6:GOTO
1910 ELSE LOCATE 1,16:PRINT BAL$:X=1:Y=16:GOTO 1910

2000 IF NT>=N(A) THEN VX=A:GOTO 1980

2010 FOR I=N(A) TO NT STEP -1:NO$(I+1,A)=NO$(I,A):NEXT

2020 NO$(NT,A)=A$:N(A)=N(A)+1 'Memorisation

2030 GOTO 1990

2040 '-----

2050 'Analyse des notes entrees

2060 '

2070 A$=UPPER$(A$) 'Majuscules

2080 L=LEN(A$):IF L>4 THEN B=1:GOTO 2150

2090 L$=LEFT$(A$,2) 'Note

2100 IF L$<>"DO" AND L$<>"RE" AND L$<>"MI" AND L$<>"FA" AND L$<>"SO" AND L$<>"LA
" AND L$<>"SI" THEN B=1:GOTO 2150 'Mauvaise note

2110 S$=MID$(A$,3,1):IF L=4 AND (S$<>"#" AND S$<>"B") THEN B=1:GOTO 2150 'Muvais
signe

2120 R=ASC(RIGHT$(A$,1))-48 'Octave

2130 IF R<1 OR R>8 THEN B=1:GOTO 2150 'Mauvaise octave

2140 B=0 'Note correcte

2150 RETURN

2160 '-----

2170 'Correction d'une note

2180 '

2190 GOSUB 2370 'Gestion du curseur

2200 LOCATE 6,14:INPUT A$

2210 IF A$="" THEN 2230

2220 GOSUB 2050:IF B=1 THEN SOUND 1,300,30:GOTO 2200 ELSE NO$(NT,A)=A$

2230 RETURN

```

```
2240 '-----
2250 'Suppression d'une note
2260 '
2270 GOSUB 2370 'Gestion du curseur
2280 PRINT:PRINT"Confirmation (O/N)"
2290 A$=INKEY$:IF A$="" THEN 2290
2300 A$=UPPER$(A$):IF A$<>"O" AND A$<>"N" THEN SOUND 1,100,30:GOTO 2290
2310 IF A$="N" THEN 2350 'Pas de suppression
2320 'Suppression
2330 FOR I=NT TO N(A):NO$(I,A)=NO$(I+1,A):NEXT
2340 NO$(I,A)="":N(A)=N(A)-1
2350 RETURN
2360 '-----
2370 'Affichage des notes en memoire
2380 '
2390 PAGE=0:PX=2 'No page et Position de la fleche au depart
2400 IF N(1)>N(2) THEN SUP=N(1) ELSE SUP=N(2)
2410 IF SUP<N(3) THEN SUP=N(3)
2420 FOR I=0 TO SUP STEP 16
2430   CLS:PRINT" Touche 'SPACE' pour changer de page, Touches fleches pour"
2440   PRINT"pour deplacer le curseur et touche 'ENTER' pour selectionner une no
te"
2450   FOR K=1 TO 3
2460     LOCATE 1,2*K+1
2470     FOR J=0 TO 15
2480       A$=NO$(I+J,K)+" ":PRINT MID$(A$,1,4) " ";
2490     NEXT J
2500     LOCATE 1,2*K+2
2510     PRINT:PRINT
2520   NEXT K
2530   LOCATE PX,12:PRINT"^"
2540   A$=INKEY$:IF A$="" THEN 2540 'Attente action clavier
2550   A=ASC(A$)
```

```
2560 IF A<>242 AND A<>243 AND A<>32 AND A<>13 THEN SOUND 1,100,30:GOTO 2540 'R
efus
2570 IF A<>243 THEN 2590
2580 IF PX<>77 THEN LOCATE PX,12:PRINT " ":PX=PX+5:LOCATE PX,12:PRINT"^":GOTO 2
540 ELSE 2540
2590 IF A<>242 THEN 2610
2600 IF PX<>2 THEN LOCATE PX,12:PRINT " ":PX=PX-5:LOCATE PX,12:PRINT"^":GOTO 25
40 ELSE 2540
2610 IF A<>32 THEN 2630
2620 IF I+16<SUP THEN PAGE=PAGE+1:GOTO 2650 ELSE I=SUP:LOCATE 1,20:PRINT"Plus
de page":LOCATE 1,13 :GOTO 2540
2630
2640 IF A=13 THEN I=SUP:NT=INT((PX-3)/5+1+PAGE*16) 'Note trouvee
2650 NEXT I
2660 PRINT"Voie (1 a 3)"
2670 A#=INKEY$:IF A#="" THEN 2670
2680 A=ASC(A#)-48
2690 IF A>3 OR A<0 THEN SOUND 1,100,30:GOTO 2670
2700 PRINT"Note : ";NO$(NT,A)
2710 RETURN
2720 '-----
2960 'Affichage du HELP d'écriture
2970 '
2980 LOCATE 1,18:PRINT:PRINT"Notes : de D01 a S18":PRINT
2990 RETURN
2995 '-----
3000 'Execution du morceau
3010
3020 AD=&90D3 'Adresse 1ere note a jouer
3030 CLS:PRINT"Compilation en cours... Patientez"
3040 A=0
3050 FOR I=1 TO 3
3060 IF N(I)>A THEN A=N(I)
3070 NEXT I
```

```
3080 FOR I=0 TO A-1
3090   FOR J=1 TO 3
3100     IF NO$(I,J)="" THEN MSB=0:LSB=0:GOTO 3190   Rien a jouer
3110     L$=LEFT$(NO$(I,J),2)
3120     L=- (L$="DO") - (L$="RE")*3 - (L$="MI")*5 - (L$="FA")*6 - (L$="SO")*8 - (L$="LA")*
10 - (L$="SI")*12
3130     IF MID$(L$,3,1)="#" THEN L=L+1
3140     IF MID$(L$,3,1)="B" THEN L=L-1
3150     R$=RIGHT$(NO$(I,J),1):R=ASC(R$)-48
3160     L=(R-1)*12+L
3170     NT=T1(L) 'Note a jouer
3180     MSB=INT(NT/256):LSB=NT-MSB*256
3190     POKE AD,LSB:AD=AD+1:POKE AD,MSB:AD=AD+1
3200   NEXT J
3210 NEXT I
3220 POKE AD,&FF:AD=AD+1:POKE AD,&FF
3230 POKE &900D,1 'Musique validee
3240 CALL &902F
3250 RETURN
3260 -----
3270 'Chargement d'un morceau
3280
3290 CLS:PRINT"Chargement d'un morceau"
3300 PRINT:INPUT"Nom du morceau ?";N$
3310 OPENIN N$
3320 INPUT #9,NB 'Nombre de notes
3330 FOR I=0 TO NB-1
3340   FOR J=1 TO 3
3350     INPUT #9,NO$(I,J)
3360   NEXT J
3370 NEXT I
3380 CLOSEIN
3390 FOR I=1 TO 3 Recherche du nombre de notes sur chaque voie
```

```
3400 X=NB
3410 WHILE NO$(X,I)=""
3420     X=X-1
3430 WEND
3440 N(I)=X+1
3450 NEXT I
3460 RETURN
3470 '-----
3480 'Sauvegarde d'un morceau
3490 '
3500 CLS:PRINT"Sauvegarde d'un morceau"
3510 PRINT:INPUT"Nom du morceau :";N$
3520 A=0
3530 FOR I=1 TO 3
3540     IF N(I)>A THEN A=N(I) 'Nombre max de notes
3550 NEXT I
3560 OPENOUT N$
3570 PRINT #9,A 'Nombre de notes
3580 FOR I=0 TO A
3590     FOR J=1 TO 3
3600         PRINT #9,NO$(I,J)
3610     NEXT J
3620 NEXT I
3630 CLOSEOUT
3640 RETURN
```

Lignes 1160 à 1210 : Chargement du programme musical sous IT.

Lignes 1220 à 1330 : Chargement des fréquences des notes.

Lignes 1360 à 1490 : Menu.

Lignes 1590 à 1690 : Ecriture d'un morceau.

Lignes 2070 à 2150 : Analyse des notes entrées.

Lignes 2190 à 2230 : Correction d'une note.

Lignes 2270 à 2350 : Suppression d'une note.

Lignes 2390 à 2710 : Affichage des notes déjà en mémoire.

Lignes 3020 à 3250 : Exécution du morceau.

Lignes 3290 à 3460 : Chargement d'un morceau.

Lignes 3500 à 3640 : Sauvegarde d'un morceau.

Le sous-programme assembleur utilisé est le programme d'exécution de musique sous IT précédemment analysé.

*Remarques :*

— Pour arrêter la musique, faites **POKE &900D,0**

— Pour redémarrer la musique à l'endroit où elle avait été interrompue, faites **POKE &900D,1**.

— Si, lors de l'exécution d'un morceau, les trois voies vous semblent être légèrement décalées (souvent après la compilation d'un morceau déjà compilé), sauvez le morceau actuel et réexécutez le programme.

## 6/5.2

# Emission d'un Beep sonore en utilisant la macro du firmware MC SOUND REGISTER

---

La macro **MC SOUND REGISTER** (voir Partie 4 chap. 2.7 p. 63) permet d'accéder aux divers registres du circuit sonore et de placer une donnée dans l'un d'eux. Pour émettre un son grâce à cette macro, il faut :

- choisir la fréquence du beep (registres 0 et 1) ;
- choisir l'amplitude du beep (registre 8) ;
- valider un canal d'émission (registre 7) ;

Pour émettre un Beep, il faut :

- émettre un son par les manipulations que nous venons de décrire ;
- attendre un certain temps (la durée du beep) ;
- mettre l'amplitude du son émis à zéro.

Le programme Assembleur qui réalise ces diverses actions est le suivant.

```
1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ; Emission d'un BEEP en utilisant
5          ; la macro MC SOUND REGISTER
6          ;-----
7          ;
8          ;Declaration de constante
9          ;
```

```
10          SOUND:      EQU  OBD34H
11          ;
12          ;
13 9000 3E00          LD   A,0           ;Registre freq canal 0
14 9002 0E00          LD   C,0           ;mis a zero
15 9004 CD34BD        CALL SOUND
16 9007 3E01          LD   A,1           ;Registre freq canal 0
17 9009 0E01          LD   C,1           ;mis a un
18 900B CD34BD        CALL SOUND
19 900E 3E08          LD   A,8           ;Registre amplitude
20 9010 0E06          LD   C,6           ;Valeur de l'amplitude
21 9012 CD34BD        CALL SOUND
22 9015 3E07          LD   A,7           ;Registre validation
23 9017 0E08          LD   C,8           ;canal A
24 9019 CD34BD        CALL SOUND
25          ;
26          ;Boucle d'attente
27          ;
28 901C 21FFFF        LD   HL,0FFFFH
29          BOU:      EQU  $
30 901F 2B            DEC  HL
31 9020 7C            LD   A,H
32 9021 B5            OR   L
33 9022 20FB          JR   NZ,BOU
34          ;
35          ;Arret du son emis
36          ;
37 9024 3E08          LD   A,8           ;Registre amplitude
```

```

38 9026 0E00          LD   C,0           ;mis a zero
39 902B CD34BD        CALL SOUND
40 902B C9           RET
41                   END

```

Pour activer un Beep, il vous faudra entrer **CALL 9000**.

Voici la version « Chargeur Basic » du programme précédent :

```

1000 REM BEEP avec la macro MC SOUND REGISTER
1010 REM =====
1020 REM Memorisation des codes op
1030 REM -----
1040 FOR I=&9000 TO &902B
1050   READ A$
1060   A$="&"+A$
1070   POKE I,VAL(A$)
1080 NEXT I
1090 REM -----
1100 REM Activation du BEEP
1110 REM -----
1120 CALL &9000
1130 '
1140 REM -----
1150 REM Donnees correspondant aux codes op
1160 REM -----
1170 DATA 3E,0,E,0,CD,34,BD,3E,1,E,1,CD,34,BD,3E,8
1180 DATA E,6,CD,34,BD,3E,7,E,8,CD,34,BD,21,FF,FF,2B
1190 DATA 7C,B5,20,FB,3E,8,E,0,CD,34,BD,C9

```



## 6/5.2.1

# Emission d'un beep sonore paramétrable

Ce chapitre complète le précédent. Il propose d'utiliser la macro du firmware **MC SOUND REGISTER** de façon paramétrique. En clair, la note à émettre et son volume peuvent être directement choisis à partir du Basic.

### COMMENT UTILISER LE PROGRAMME

Si vous désirez utiliser la version Assembleur du programme, entrez le listing suivant :

```
ORG 9000H
LOAD 9000H

;-----
; Emission d'un BEEP parametuable
; en utilisant la macro
; MC SOUND REGISTER
;-----
;
;Declaration de constante
;
SOUND: EQU 0BD34H
FREQH: EQU 9F00H ;MSB frequence
FREQL: EQU 9F01H ;LSB frequence
VOL: EQU 9F02H ;Volume du son emis
;
;
```

```
9000 3A009F      LD  A,(FREQH)      ;MSB Frequence
9003 4F          LD  C,A            ;dans le registre C
9004 3E00        LD  A,0            ;Registre 0
9006 CD34BD      CALL SOUND
9009 3A019F      LD  A,(FREQL)      ;LSB Frequence
900C 4F          LD  C,A            ;dans C
900D 3E01        LD  A,1            ;Registre 1
900F CD34BD      CALL SOUND
9012 3A029F      LD  A,(VOL)        ;Volume sonore
9015 4F          LD  C,A            ;Dans C
9016 3E08        LD  A,8            ;Registre 8
9018 CD34BD      CALL SOUND
901B 3E07        LD  A,7            ;Registre validation
901D 0E08        LD  C,8            ;canal A
901F CD34BD      CALL SOUND

;
;Boucle d'attente
;
9022 21FFFF      LD  HL,OFFFH
BOU:             EQU $
9025 2B          DEC  HL
9026 7C          LD  A,H
9027 B5          OR   L
9028 20FB        JR   NZ,BOU

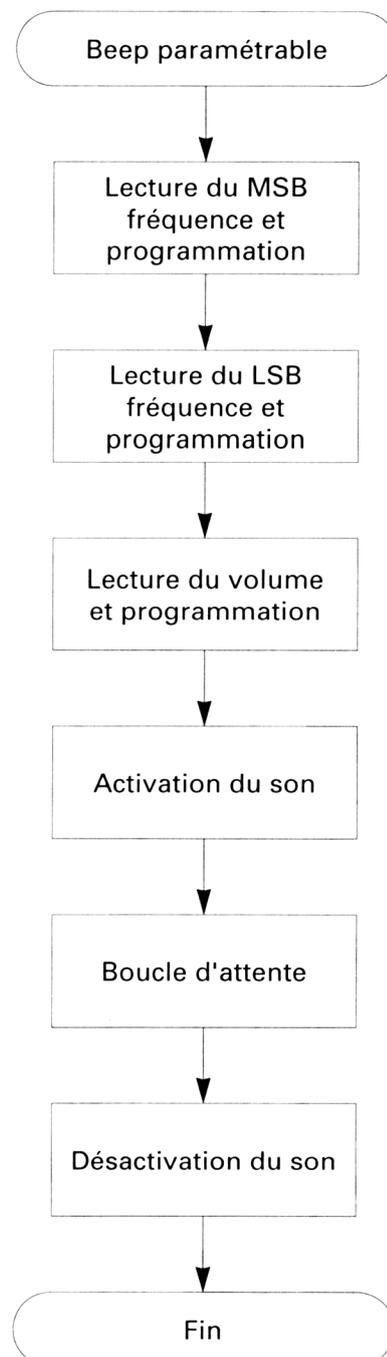
;
;Arret du son emis
;
902A 3E08        LD  A,8            ;Registre amplitude
902C 0E00        LD  C,0            ;mis a zero
902E CD34BD      CALL SOUND
9031 C9          RET
END
```



Voici les données de checksum correspondantes :  
5E 4C E2 87

### LE PROGRAMME EN DÉTAIL

La logique du programme Assembleur apparaît dans l'ordinogramme suivant :



Les données passées au programme dans les mémoires 9F00 à 9F02 sont lues et passées à la macro SOUND :

```
LD    A,(FREQH)  —► Fréquence haute
LD    C,A
LD    A,0
CALL  SOUND
```

```
...
LD    A,(VOL)    —► Volume
LD    C,A
LD    A,8
CALL  SOUND
```

Le canal A est ensuite validé, ce qui provoque l'émission du son précédemment programmé :

```
LD    A,7
LD    C,8
CALL  SOUND
```

Une boucle d'attente basée sur la décrémentation de la paire de registres HL maintient l'émission du son pendant un bref instant :

```
LD    HL,0FFFFH
BOU:  EQU    $
      DEC   HL
      LD    A,H
      OR   L
      JR   NZ,BOU
```

Le son est ensuite coupé en dévalidant le canal A :

```
LD    A,8
LD    C,0
CALL  SOUND
```



# 6/6

## La synthèse vocale

---

Il est parfois dur de supporter la solitude engendrée par la passion de la micro-informatique. Il est en effet vrai que les programmeurs s'enferment généralement dans leur petit monde dont l'unique population est composée d'octets, parfois ingrats à l'annonce d'une indésirable erreur **Type Mismatch**, après des heures d'effort de programmation. Seuls musiques et graphismes, au demeurant fort réussis sur AMSTRAD CPC, égayent la vie des fanatiques du clavier et du joystick.

Il ne manquerait à notre AMSTRAD que la parole pour en faire un partenaire informatique à part entière.

La société TECHNI-MUSIQUE a probablement entendu cet appel, et a créé un synthétiseur vocal, simple d'utilisation, et possédant de surcroît une bibliothèque de sons pré-programmés et en français.

### En savoir plus sur la synthèse vocale

La synthèse vocale fut, et reste encore, un secteur de l'électronique spécialisée.

Une des voies suivies pour synthétiser vocalement les sons, commença par l'étude des mécanismes de la parole humaine. Cette étude fut suivie par une modélisation de ces mécanismes, puis l'application électronique de ces modèles.

#### LE MÉCANISME HUMAIN

Le mécanisme humain de formation des sons semble en général très simple, mais il suffit d'observer l'évolution de la parole chez le très jeune enfant qui, malgré des efforts considérables pour communiquer avec nous, ne commence à prononcer correctement les mots qu'après plusieurs années.

Aucun son ne serait produit sans un dosage adéquat de l'air que nous respirons. Cet air, inspiré dans les poumons, va servir d'énergie, qui sera véhiculée dans la gorge, et permettra, à l'aide des muscles, de faire vibrer les cordes vocales, et résonnera ensuite dans les cavités du pharynx, de la bouche et des fosses nasales (voir Fig. 1).

Par contre, tous les sons ne feront pas forcément agir toutes les parties du système vocal. Certains sons, par exemple, ne font pas vibrer les cordes vocales (CH, S, ...). Les sons peuvent ainsi être classés en deux catégories : les sons dits *voisés* et les sons *non voisés*.

Les sons voisés correspondent à ceux qui font vibrer les cordes vocales (A, B, D, E, G, I, J, L, ...). Les sons non voisés sont ceux qui ne font pas vibrer les cordes vocales (F, K, P, CH, T, ...). Ces sons, bien sûr, doivent être pris dans leur contexte, car les lettres prononcées isolées font intervenir plusieurs sons (par exemple, la lettre P fait intervenir les deux sons P et é).

Les diverses études ont abouti à une liste de sons élémentaires, permettant de créer toutes les combinaisons possibles pour prononcer tous les mots d'une même langue. Ces sons élémentaires sont appelés phonèmes. La liste des phonèmes est généralement donnée dans les premières pages des dictionnaires, et l'on retrouve la prononciation en phonétique des mots entre parenthèses.

Des études encore plus approfondies ont permis de déterminer les fréquences de résonance utilisées pour former les sons, chacune de ces fréquences est appelée formant.

En résumé : une phrase est composée d'une suite de formants, constituant eux-mêmes les différents phonèmes qui, associés, génèrent le message final.

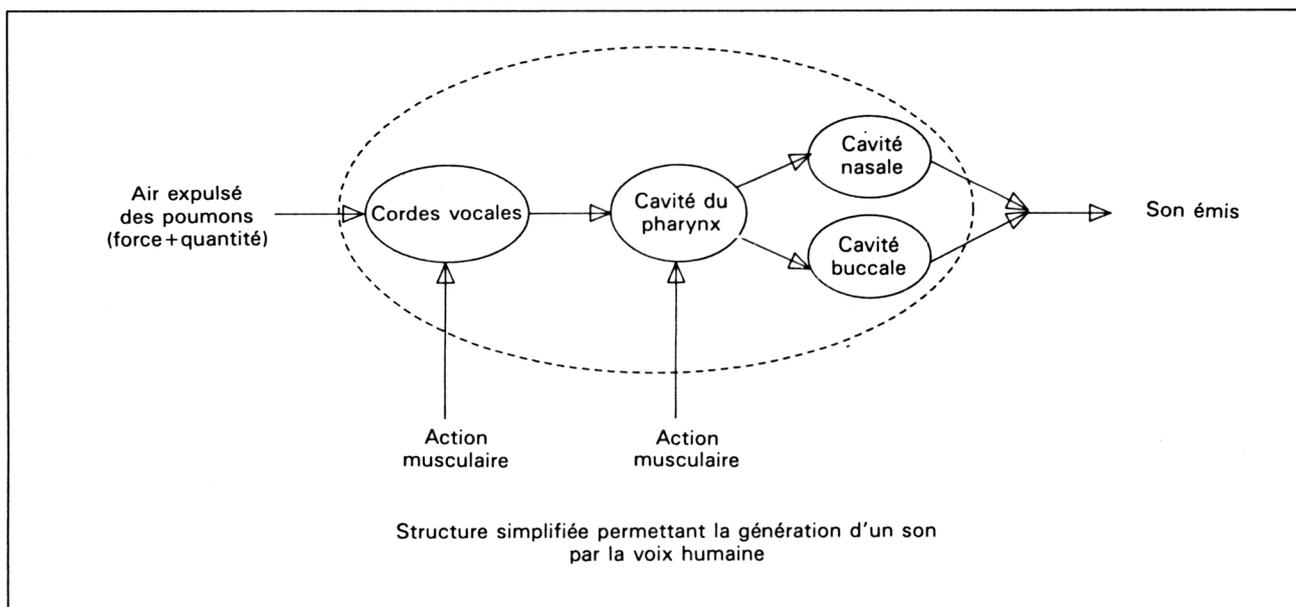


Fig. 1 : Structure simplifiée permettant la génération d'un son par la voix humaine.

**APPLICATION A L'ÉLECTRONIQUE**

La recherche en fabrication des composants de synthèse vocale s'est appuyée sur les modèles humains pour créer les différentes fonctions internes incorporées dans le boîtier.

Deux composants sont largement répandus dans le domaine de la micro-informatique pour synthétiser les sons : les MEA-8000 et PCF-8200 fabriqués par RTC. Leurs principales caractéristiques sont :

- \* MEA-8000 : — tension d'alimentation 5 volt
  - trames de parole : 8, 16, 32 ou 64 ms
  - nombre de formants : 4
  - fréquence de l'horloge : 4 Mhz
- \* PCF-8200 : — tension d'alimentation : 5 volt
  - trames de paroles : programmables de 8.8 à 88 ms
  - nombre de formants : 5 en voix masculine  
4 en voix féminine



## 6/6.1

# Le synthétiseur vocal TECHNI-MUSIQUE

---

Le synthétiseur vocal TECHNI-MUSIQUE utilise le MEA-8000, qui permet d'obtenir une synthèse de qualité acceptable pour un prix abordable, un interfaçage aisé au bus du Z80 de l'AMSTRAD (la fréquence d'horloge de 4 MHz fut probablement décisive), et une programmation relativement facile.

Nous nous intéresserons donc plus particulièrement à ses caractéristiques, dont le schéma fonctionnel est donné en figure 1.

On distingue les différentes fonctions permettant de générer les sons voisés ou non, qui seront ensuite filtrés et mis en résonance dans des bandes de fréquences définies, permettant de générer les formants caractéristiques du son.

La programmation, si l'on veut obtenir une synthèse correcte, doit essentiellement s'effectuer en Assembleur, et nous l'aborderons dans un paragraphe suivant.

### **I. Le synthétiseur vocal TECHNI-MUSIQUE et le non-initié**

#### **PRÉSENTATION DU SYNTHÉTISEUR VOCAL**

Le synthétiseur vocal TECHNI-MUSIQUE se présente en boîtier plastique ne nécessitant aucune alimentation externe, accompagné d'une cassette ou d'une disquette (selon le type de CPC que vous possédez), contenant un fichier de sons pré-établis et un logiciel de gestion des sons par une instruction supplémentaire ajoutée au Basic (RSX), plus un logiciel de démonstration (voir Fig. 2).

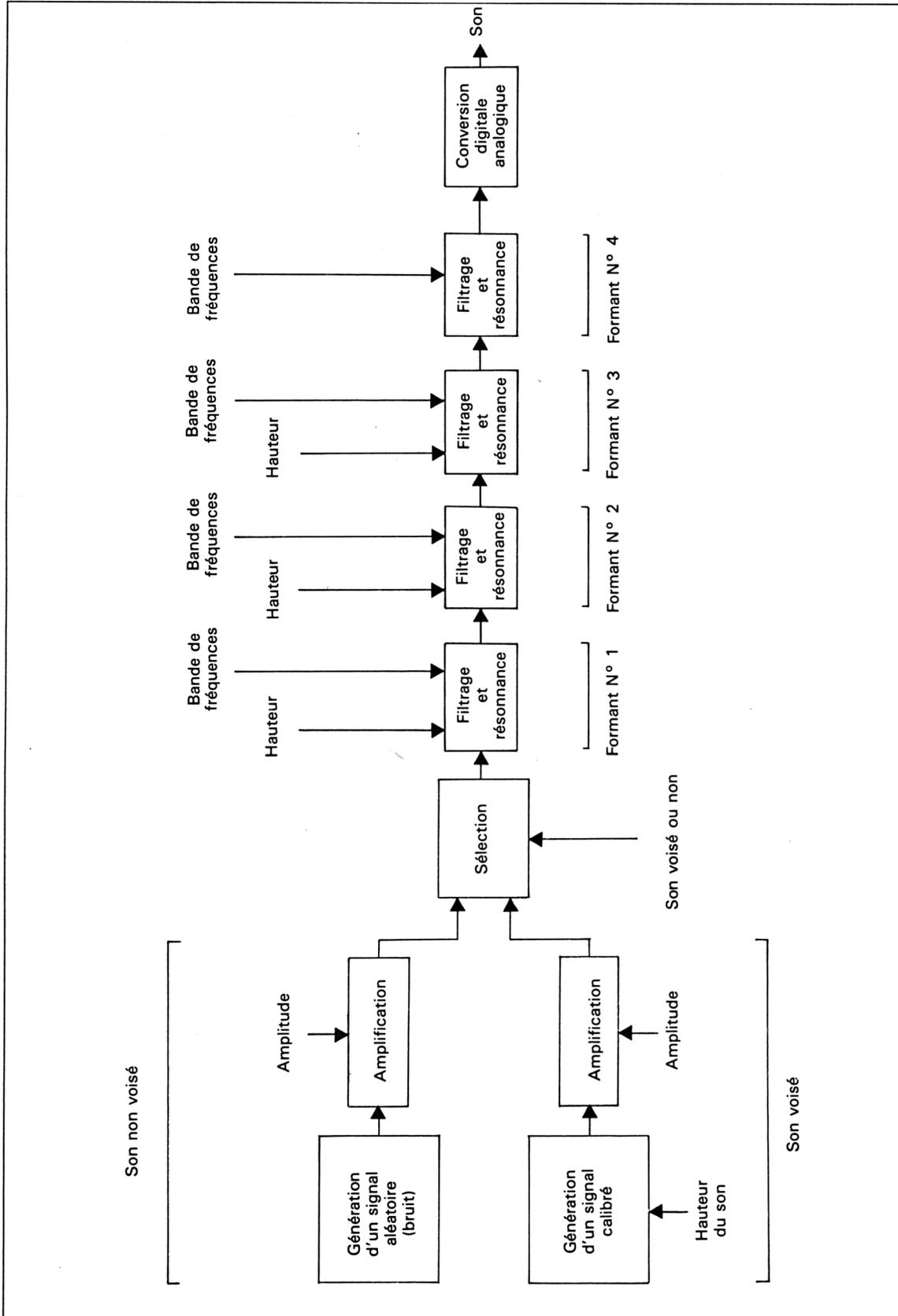


Fig. 1 : Principe de génération d'un son par le composant MEA-8000.

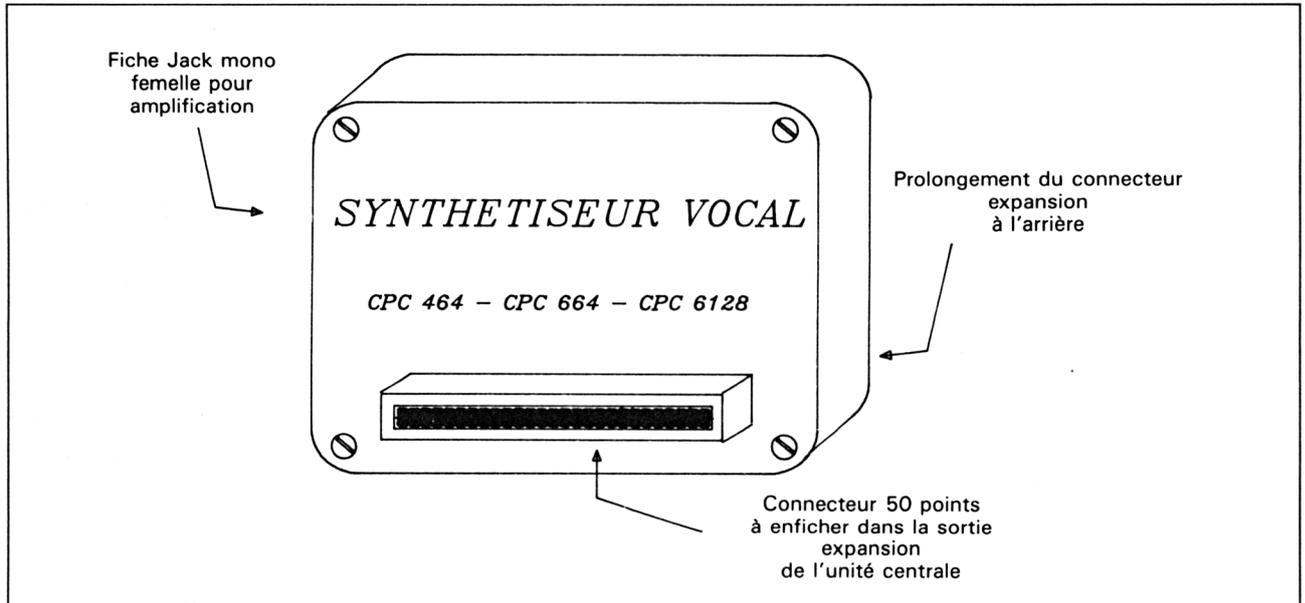


Fig. 2 : Synthétiseur vocal.

### CONNEXION ET DEMONSTRATION

Pour avoir un aperçu des possibilités du synthétiseur vocal, nous allons mettre en œuvre le logiciel de démonstration.

La connexion du boîtier est relativement simple, et devra absolument s'effectuer HORS TENSION. Il suffit d'enficher le boîtier à l'arrière de votre CPC, sur le connecteur noté « EXPANSION », l'alimentation étant fournie par la ligne + 5 volt du connecteur. Certains modèles de CPC demanderont un adaptateur, car étant livrés avec des connecteurs femelles de type CENTRONIC.

Pour la démonstration, il vous faudra lancer le programme appelé « DEMO » (dans notre version), qui propose trois choix :

- le choix numéro 1 permet de mettre à l'heure l'horloge parlante qui s'affiche au menu, et est réactualisée toutes les dix secondes avec une signalisation orale ;
- le choix numéro 2 vous propose une démonstration de chants, plus particulièrement une berceuse, mais qui, étant donné la qualité sonore nasillarde du haut parleur AMSTRAD, ne risque pas de vous endormir ;
- le choix numéro 3 permet de créer soi-même de petites phrases et de les parfaire, grâce à un tableau accompagné de codes à aligner pour composer des mots.

## II. Utilisation sous Basic

Deux possibilités s'offrent à vous pour utiliser le logiciel sous Basic.

### LE FICHIER PHC.BIN

La première possibilité charge les principaux sons de la langue française ainsi que quelques sons complexes (AIL, EUILLE, ZIO, ...).

Le chargement s'effectuera de la façon suivante (pour la version du logiciel que nous possédons) :

```
MEMORY &92C2
LOAD « PHC.BIN »
CALL &A655
```

Vous disposez alors d'une fonction supplémentaire dénommée ùSPEAK, et dont la syntaxe est la suivante :

```
ùSPEAK,hauteurdeson,son1,son2,son3,...
```

Les numéros des sons correspondent à la liste donnée dans la notice, et affichée dans le choix 3 de la démonstration, ce qui vous permet de créer des phrases (0 = A, 1 = AIL, ...)

et de modifier les intonations (PITCH). La notice étant suffisante, nous ne nous attarderons pas ici.

### LE FICHIER HOP.BIN

En examinant de près le programme de démonstration et le fichier HOP.BIN appelé dans ce logiciel (dans notre version du synthétiseur), la notice étant discrète sur le sujet, nos conclusions sont les suivantes : ce fichier diffère du précédent par l'adjonction de 28 mots supplémentaires permettant de prononcer facilement les nombres de zéro à cinquante-neuf, et l'heure.

La séquence d'installation est dans ce cas :

```
MEMORY & 8A44
LOAD « HOP.BIN »
CALL &A655
```

La commande supplémentaire reste la même, et possède la même syntaxe, par contre les codes ajoutés sont :

0 : ZERO	15 : QUINZE
1 : UN	16 : SEIZE
2 : DEUX	17 : DIX-SEPT
3 : TROIS	18 : DIX-HUIT
4 : QUATRE	19 : DIX-NEUF
5 : CINQ	20 : VIN
6 : SISSE	21 : TREN
7 : SEPT	22 : QUARAN
8 : HUIT	23 : CINQUAN

9 : NEUF	24 : TET
10 : DISSE	25 : TE
11 : ONZE	26 : HEURE
12 : DOUZE	27 : MINUTE
13 : TREIZE	28 : SECONDE
14 : QUATORZE	

Les codes 24 et 25 permettent d'effectuer des liaisons ou non des chiffres des dizaines avec les unités, par exemple :

41 = QUARAN + TET + UN  
45 = QUARAN + TE + CINQ

On retrouve ensuite les mêmes codes que pour les sons du fichier PHC.BIN, mais auxquels il faut ajouter 29 (29 = A, 30 = AIL, ... etc.)

### III. Utilitaire de conversion

Après quelques instants d'utilisation, on s'aperçoit très rapidement qu'il est fastidieux de composer des phrases relativement longues à l'aide des codes de la notice.

Nous vous proposons ci-après un programme permettant de créer plus facilement vos phrases en convertissant celles-ci en codes, que vous pourrez sauvegarder sur disquette. Il sera facile ensuite, de récupérer ces codes par la lecture du fichier dans un programme désirant les utiliser.

#### PRESENTATION DU PROGRAMME

Nous avons choisi de proposer un menu, géré par une structure de type arborescence simple, proposant sept choix :

1 - Conception d'une phrase : permet d'entrer au clavier une phrase qui sera traduite, puis vocalisée par le synthétiseur. Il vous sera ensuite demandé un numéro (de 1 à 5) permettant de la sauvegarder. La phrase devra comporter au maximum 240 caractères. Un rappel du tableau des sons audibles vous est aussi affiché. La traduction vocale s'effectue automatiquement suite à l'appui sur la touche <RETURN>.

2 - Exécution d'une phrase : permet de réécouter l'une des cinq phrases mémorisées, pour en vérifier les sonorités.

3 - Liste des phrases : affiche le contenu des cinq phrases en mémoire.

4 - Chargement : permet de charger un fichier de codes de cinq phrases préalablement sauvegardées.

5 - Sauvegarde : permet de sauvegarder dans un fichier les cinq phrases créées.

6 - Modification d'une phrase : permet de corriger une phrase si les sonorités ne vous ont pas convaincues. Vous devez pour cela récupérer tous les caractères corrects à l'aide de la touche <COPY> et effectuer les corrections aux endroits désirés.

7 - Fin : retourne au Basic sans perdre le contenu de la mémoire, ni l'instruction ùSPEAK.

*Remarque :*

— Lors de la création ou de la correction d'une phrase, vous devez tenir compte de l'affichage des caractères ou groupes de caractères non reconnus, et modifier les sonorités en fonction de la liste des sons affichés. Par contre, vous n'êtes pas obligés de vous préoccuper des majuscules ou minuscules, le logiciel s'en chargeant.

— Lors des sauvegardes ou lectures de fichiers, ne pas mettre d'extension, le logiciel place automatiquement l'extension « PHC ».

### ORGANISATION DU PROGRAMME

Nous ne nous étendrons pas ici sur l'algorithme du programme, mais signalons tout de même les principaux tableaux de variables utilisés :

- `tbph$` contient tous les sons disponibles ;
- `exemple$` contient tous les exemples des sons ;
- `lecode` est un tableau temporaire des codes utilisés ;
- `tbcode` est la table des codes pour les cinq phrases possibles, c'est un tableau à deux dimensions. Nous en reparlerons dans le paragraphe indiquant la méthode pour récupérer les codes créés ;
- `tbphrase$` est un tableau temporaire pour la décomposition des phrases en codes.

### LE PROGRAMME

Nous avons, lors de la frappe, respecté l'indentation (décalage) requise par l'algorithmique, et nous vous conseillons de faire de même ; cela afin que vous puissiez mieux cerner les différentes boucles de traitement et les étudier, de limiter les erreurs de frappes, et de faciliter les corrections éventuelles (le listing respecte le MODE 1 de l'affichage sur écran).

Les versions de CPC ne possédant pas les instructions élaborées de gestion des erreurs devront éliminer les lignes 460, 3140, 3330 et les lignes 5370 à 5490, mais une erreur d'utilisation du logiciel (sur disquette) redonnera la main au Basic, et un nouveau <RUN> efface toutes les variables en mémoire.

*Note de l'éditeur :* Dans certaines lignes du listing suivant, le signe # est à remplacer par £ :

ligne 1610  
ligne 4120, le 1<sup>er</sup> # seulement  
ligne 4310  
ligne 5610, le 1<sup>er</sup> # seulement

```

10 REM ***** 510
20 REM ** SYNTHETISEUR VOCAL V 1.1 ** 520
30 REM ** APRES L'INTRODUCTION D'UNE ** 530
40 REM ** PHRASE EN FRANCAIS AVEC ** 540
50 REM ** QUELQUES RESTRICTIONS ** 550
60 REM ** DE SYNTAXE DES ** 560
70 REM ** DIFFERENTS PHONEMES ** 570
80 REM ** RESTITUTION PAR LE H.P. ** 580
90 REM ***** 590
100 REM 600
110 DEFINT A-C:DEFINT E-G:DEFINT I-L 610
120 DEFINT N-R:DEFINT U-Z 620
130 TEMPS=TIME 630
140 REM 640
150 MEMORY &92C2 650
160 LOAD"PHC" 660
170 CALL &A655 670
180 REM 680
190 MODE 1 690
200 REM 700
210 REM ***** 710
220 REM ***** INITIALISATIONS ***** 720
230 REM ***** 730
240 REM 740
250 DIM tbph$(103) 750
260 DIM exemple$(102) 760
270 DIM lecode(256) 770
280 DIM tbphrase$(256) 780
290 DIM tbcodes(5,256) 790
300 i=0 800
310 code = 0 810
320 WHILE i < 103 820
330 READ CODE,phoneme$,exem$ 830
340 tbph$(i)=UPPER$(phoneme$) 840
350 exemple$(i)=UPPER$(exem$) 850
360 i=i+1 860
370 WEND 870
380 REM 880
390 REM ***** 890
400 REM ***** MENU ***** 900
410 REM ***** 910
420 REM 920
430 CHOIX$="8" 930
440 MODE 2:INK 0,0:INK 1,26 940
450 PAPER 0:PEN 1:BORDER 0 950
460 ON ERROR GOTO 5370 960
470 REM 970
480 WHILE CHOIX$ <> "7" 980
490 CLS 990
500 LOCATE 49,1 1000

```

```
510     PRINT CHR$(24);
520     PRINT SPACE$(31)
530     LOCATE 49,2
540     PRINT SPACE$(31)
550     LOCATE 49,3
560     PRINT SPACE$(31)
570     LOCATE 49,4
580     PRINT " -1- Conception d'une";
590     PRINT " phrase      "
600     LOCATE 49,5
610     PRINT SPACE$(31)
620     LOCATE 49,6
630     PRINT " -2- Execution d'une";
640     PRINT " phrase      "
650     LOCATE 49,7
660     PRINT SPACE$(31)
670     LOCATE 49,8
680     PRINT " -3- Liste des";
690     PRINT " phrases      "
700     LOCATE 49,9
710     PRINT SPACE$(31)
720     LOCATE 49,10
730     PRINT " -4- Chargement d'une";
740     PRINT " phrase      "
750     LOCATE 49,11
760     PRINT SPACE$(31)
770     LOCATE 49,12
780     PRINT " -5- Sauvegarde";
790     PRINT " d'une phrase  "
800     LOCATE 49,13
810     PRINT SPACE$(31)
820     LOCATE 49,14
830     PRINT " -6- Correction";
840     PRINT " d'une phrase  "
850     LOCATE 49,15
860     PRINT SPACE$(31)
870     LOCATE 49,16
880     PRINT " -7- Fin";
890     PRINT SPACE$(23)
900     LOCATE 49,17
910     PRINT SPACE$(31)
920     LOCATE 49,18
930     PRINT "   Entrez votre";
940     PRINT " choix :      "
950     LOCATE 49,19
960     PRINT SPACE$(31)
970     PRINT CHR$(24)
980     LOCATE 1,1
990     PRINT "Temps passe sur le";
1000    PRINT " logiciel : "
```

```

1010     CHOIX$="  "
1020     LOCATE 73,18
1030     PRINT CHOIX$
1040     WHILE CHOIX$ <> "1" AND
        CHOIX$ <> "2" AND CHOIX$ <>
        "3" AND CHOIX$ <> "4" AND
        CHOIX$ <> "5" AND CHOIX$ <>
        "6" AND CHOIX$ <> "7"
1050         LOCATE 74,18
1060         PRINT CHOIX$
1070         CHOIX$=INKEY$
1080         GOSUB 5170
1090     WEND
1100     LOCATE 74,18
1110     PRINT CHOIX$
1120     CHOIX=VAL(CHOIX$)
1130     ON CHOIX GOSUB 1260,2630,2950
        ,3100,3290,4680
1140 WEND
1150 END
1160 REM
1170 REM *****
1180 REM * FIN DU PROGRAMME PRINCIPAL *
1190 REM *****
1200 REM
1210 REM
1220 REM *****
1230 REM ** CONCEPTION D'UNE PHRASE **
1240 REM *****
1250 REM
1260 GOSUB 3610
1270 boul$="n"
1280 WHILE boul$="n"
1290     LOCATE 1,19
1300     PRINT "Entrez votre phrase";
1310     PRINT ", pas plus de 240";
1320     PRINT " caracteres :";
1330     PRINT CHR$(24)
1340     LOCATE 1,20
1350     FOR i=1 TO 240
1360     PRINT "_";
1370     NEXT i
1380     LOCATE 1,23
1390     PRINT CHR$(24)+SPACE$(28)
1400     PRINT CHR$(24)
1410     LOCATE 1,20
1420     INPUT "",phrase$
1430     phrase$=UPPER$(phrase$)
1440     PRINT CHR$(24)

```

```

1450         IF LEN(phrase$) < 241 THEN
              boul$ = "o" ELSE LOCATE 34,19
              :PRINT CHR$(24):PRINT "240":
              PRINT CHR$(24):LOCATE 34,19:
              PRINT "240" :PRINT CHR$(24):
              PRINT"240":PRINT CHR$(24)
1460 WEND
1470 PHRASE$=PHRASE$+"      "
1480 REM
1490 REM *****
1500 REM ** DETECTION DES ENSEMBLES **
1510 REM **      DE CARACTERES
1520 REM **      EGAUX A EUIL, ILLE,   **
1530 REM ** OUIL, SIO, TION OU ZIO **
1540 REM *****
1550 FOR i=1 TO LEN(phrase$)
1560     IF MID$(phrase$,i,4) <> "EUIL"
              THEN GOTO 1600
1570     MID$(PHRASE$,I,1)="!"
1580     DECAL=3
1590     GOSUB 4140
1600     IF MID$(phrase$,i,4) <> "ILLE"
              THEN GOTO 1640
1610     MID$(PHRASE$,I,1)="#"
1620     DECAL=3
1630     GOSUB 4140
1640     IF MID$(phrase$,i,4) <> "OUIL"
              THEN GOTO 1680
1650     MID$(PHRASE$,I,1)="#"
1660     DECAL=3
1670     GOSUB 4140
1680     IF MID$(phrase$,i,3) <> "SIO"
              THEN GOTO 1720
1690     MID$(PHRASE$,I,1)="#"
1700     DECAL=3
1710     GOSUB 4140
1720     IF MID$(phrase$,i,4) <> "TION"
              THEN GOTO 1760
1730     MID$(PHRASE$,I,1)="à"
1740     DECAL=3
1750     GOSUB 4140
1760     IF MID$(phrase$,i,3) <> "ZIO"
              THEN GOTO 1800
1770     MID$(PHRASE$,I,1)=""
1780     DECAL=3
1790     GOSUB 4140
1800 NEXT i
1810 REM *****
1820 REM *** DETECTION DES AUTRES ***
1830 REM *** SYLLABES OU CARACTERES ***
1840 REM *****

```

```

1850 LOCATE 1,24
1860 PRINT "Caracteres non reconnus ";
1870 PRINT "....."
1880 LOCATE 1,25
1890 PRINT "Caracteres ou groupes de";
1900 PRINT " caracteres reconnus ...."
1910 m=1
1920 phrase$=LEFT$(phrase$, (LEN(phrase$)
-3))
1930 WHILE m < LEN(phrase$)
1940     tbphrase$(m)=MID$(phrase$,m,1)
1950     m=m+1
1960 WEND
1970 IF LEN(phrase$) <> 1 THEN GOSUB 553
0 ELSE k=0
1980 REM *****
1990 REM ** EXECUTION DE LA PHRASE **
2000 REM ** PRECEDAMMENT INTRODUITE **
2010 REM *****
2020 i=1
2030 WHILE i <= k
2040     @SPEAK,60,lecode(i)
2050     i=i+1
2060 WEND
2070 LOCATE 1,23
2080 PRINT "Entrez le numero de la phras
e (1 a 5) : "
2090 no=12
2100 PRINT CHR$(24)
2110 LOCATE 40,23
2120 PRINT " "
2130 WHILE no > 5 OR no < 1
2140     LOCATE 41,23
2150     a$=INKEY$
2160     no=VAL(a$)
2170     LOCATE 41,23
2180     PRINT a$
2190     LOCATE 41,23
2200     PRINT a$
2210 WEND
2220 PRINT CHR$(24)
2230 tbcodes(no,0)=k
2240 i=1
2250 WHILE i <= k
2260     tbcodes(no,i)=lecode(i)
2270     i=i+1
2280 WEND
2290 RETURN

```

```

2300 REM *****
2310 REM *** CARACTERE OU GROUPE DE ***
2320 REM *** CARACTERE NON RECONUS ***
2330 REM *** PAR LE PROGRAMME ***
2340 REM *****
2350 IF LEN(morceau$) <> 1 THEN 2450
2360 LOCATE 50,24
2370 PRINT " "
2380 LOCATE 50,24
2390 PRINT tbphrase$(i)
2400 lecode(k)=101
2410 k=k+1
2420 i=i+1
2430 morceau$=""
2440 GOTO 2470
2450 k=k+1
2460 morceau$=""
2470 RETURN
2480 REM *****
2490 REM ** CARACTERE OU GROUPE DE **
2500 REM ** CARACTERES RECONNUS PAR **
2510 REM ** LE PROGRAMME **
2520 REM *****
2530 lecode(k)=j
2540 LOCATE 50,25
2550 PRINT " "
2560 LOCATE 50,25
2570 PRINT tbph$(lecode(k))
2580 i=i+1
2590 RETURN
2600 REM *****
2610 REM *** EXECUTION D'UNE PHRASE ***
2620 REM *****
2630 CLS
2640 PRINT " Execution d'une phr
ase"
2650 LOCATE 1,5
2660 PRINT "Entrez le numero de la phras
e a executer : "
2670 no=9
2680 PRINT CHR$(24)
2690 LOCATE 43,5
2700 PRINT " "
2710 WHILE no > 5 OR no < 1
2720 LOCATE 44,5
2730 a$=INKEY$
2740 no=VAL(a$)
2750 LOCATE 44,5
2760 PRINT a$
2770 LOCATE 44,5
2780 PRINT a$

```

```

2790 WEND
2800 PRINT CHR$(24)
2810 IF tbcodé(no,0) <> 0 THEN 2860
2820     PRINT "Il n'y a rien dans";
2830     PRINT " cette phrase !!!"
2840     FOR i=1 TO 1000:NEXT i
2850     RETURN
2860 FOR j=1 TO tbcodé(no,0)
2870     PRINT tbph$(tbcodé(no,j));
2880 NEXT j
2890 PRINT
2900 FOR i=1 TO tbcodé(no,0)
2910     @SPEAK,60,tbcodé(no,i)
2920 NEXT i
2930 RETURN
2940 REM *****
2950 REM ***** LISTE DES PHRASES *****
2960 REM *****
2970 CLS
2980 PRINT SPACE$(22);
2990 PRINT"LISTE DES PHRASES"
3000 FOR i= 1 TO 5
3010     PRINT"Phrase no : ";i
3020     FOR j=1 TO tbcodé(i,0)
3030         PRINT tbph$(tbcodé(i,j));
3040     NEXT j
3050     PRINT
3060 NEXT i
3070 CALL &BBO6
3080 RETURN
3090 REM *****
3100 REM *** CHARGEMENT DU FICHER ***
3110 REM *****
3120 CLS
3130 CAT
3140 ON ERROR GOTO 5420
3150 INPUT "Entrez le nom du fichier a c
harger : ",nomfic$
3160 nomfic$=nomfic$+".phr"
3170 nomfic$=UPPER$(nomfic$)
3180 OPENIN nomfic$
3190 PRINT nomfic$+" en cours de chargem
ent ..."
3200 FOR i=1 TO 5
3210     INPUT#9,tbcodé(i,0)
3220     FOR j=1 TO tbcodé(i,0)
3230         INPUT #9,tbcodé(i,j)
3240     NEXT
3250 NEXT
3260 CLOSEIN
3270 RETURN

```

```

3280 REM *****
3290 REM *** SAUVEGARDE DU FICHIER ***
3300 REM *****
3310 CLS
3320 CAT
3330 ON ERROR GOTO 5460
3340 INPUT "Entrez le nom du fichier a s
auvegarder : ",nomfic$
3350 nomfic$=nomfic$+".phr"
3360 nomfic$=UPPER$(nomfic$)
3370 OPENOUT nomfic$
3380 PRINT nomfic$+" en cours de sauvega
rde ..."
3390 FOR i=1 TO 5
3400     PRINT#9,tbcode(i,0)
3410     FOR j=1 TO tbcode(i,0)
3420         PRINT #9,(tbcode(i,j))
3430     NEXT j
3440 NEXT i
3450 CLOSEOUT
3460 RETURN
3470 REM *****
3480 REM * LISTE EN DATA DES PHONEMES *
3490 REM *****
3500 DATA 0,"A","BAS",1,"AIL","PAILLE",2
,"AN","MAMAN",3,"B","BOA",4,"BE","TOURBE
",5,"BL","BLEU",6,"BLE","CABLE",7,"BR","
ABRIT",8,"BRE","ARBRE",9,"C","CORPS",10,
"CE","PLAQUE",11,"CH","CHAT",12,"CL","CL
OU"
3510 DATA 13,"CLE","RACLE",14,"CR","CRAB
E",15,"CT","ACTE",16,"D","DOUX",17,"DE",
"RIDE",18,"DR","DRAIN",19,"DRE","HYDRE",
20,"E","PEUX"
3520 DATA 21,"EU","COULEUR",22,"EI","ETE
",23,"AI","PARAIT",24,"EIL","TREILLE",25
,"EUIL","TREUIL",26,"F","FORCE",27,"FL",
"FLAQUE",28,"FLE","BAFLE",29,"FR","FRANC
E",30,"FRE","AFFRE",31,"G","GOUT",32,"GE
","BAGUE"
3530 DATA 33,"GL","GLISSE",34,"GLE","ANG
LE",35,"GN","PIGNON",36,"GNE","GAGNE",37
,"GR","GRAND",38,"GRE","OGRE",39,"I","ID
EE"
3540 DATA 40,"IA","VIA",41,"IE","DIDIER"
,42,"IEN","CHIEN",43,"IEU","VIEUX",44,"I
LLE","BILLE",45,"IN","PAIN",46,"IO","CHI
OT",47,"ION","PION",48,"J","JOIN",49,"JE
","SAGE",50,"L","LOIN",51,"LE","VOCAL",5
2,"M","MOINS"

```

```

3550 DATA 53,"ME","RAME",54,"N","NOIX",5
5,"NE","REINE",56,"AU","BATEAU",57,"O","
BOSSE",58,"OI","BOIS",59,"OIN","POING"
3560 DATA 60,"ON","BON",61,"OU","ROUE",6
2,"OUIL","ROUILLE",63,"P","POIDS",64,"PE
","GROUPE",65,"PL","PLAGE",66,"PLE","SOU
PLE",67,"PR","PROSE",68,"PRE","APRE",69,
"PS","EPSYLON",70,"PT","OPTIQUE",71,"R",
"RUDE"
3570 DATA 72,"RE","RARE",73,"S","SOIN",7
4,"SE","GLACE",75,"SIO","RATIO",76,"SP",
"ASPECT",77,"ST","STATIQUE",78,"STR","ST
RATE"
3580 DATA 79,"STRE","ASTRE",80,"T","TORE
",81,"TE","ROUTE",82,"TION","RATION",83,
"TR","TRACE",84,"TRE","AUTRE",85,"U","UN
E",86,"UI","TUILE",87,"V","VOIX",88,"VE"
,"BRAVE",89,"VR","VRILLE",90,"VRE","LIVR
E",91,"X","AXIOME"
3590 DATA 92,"XE","AXE",93,"X","XAVIER",
94,"Z","ZOE",95,"ZE","ROSE",96,"ZIO","ZO
ZIO",97,"/","PITCH -2",98,"ç","PITCH +2"
,99,"°","24 mS",100,"§","40 mS",101,":",
"72 mS",102," " ," " ,"136 mS"
3600 REM *****
3610 REM ** LISTE DES PHONEMES ECRAN **
3620 REM *****
3630 CLS
3640 J=0
3650 FOR i=1 TO 17
3660     LOCATE 1,i
3670     PRINT tbph$(j)
3680     LOCATE 5,i
3690     PRINT exemple$(j)
3700     J=J+1
3710 NEXT i
3720 FOR i=1 TO 17
3730     LOCATE 13,i
3740     PRINT tbph$(j)
3750     LOCATE 18,i
3760     PRINT exemple$(j)
3770     j=j+1
3780 NEXT i
3790 FOR i=1 TO 17
3800     LOCATE 27,i
3810     PRINT tbph$(j)
3820     LOCATE 32,i
3830     PRINT exemple$(j)
3840     J=J+1

```

```

3850 NEXT i
3860 FOR i=1 TO 17
3870     LOCATE 40,i
3880     PRINT tbph$(j)
3890     LOCATE 45,i
3900     PRINT exemple$(j)
3910     j=j+1
3920 NEXT i
3930 FOR i=1 TO 17
3940     LOCATE 54,i
3950     PRINT tbph$(j)
3960     LOCATE 59,i
3970     PRINT exemple$(j)
3980     j=j+1
3990 NEXT i
4000 FOR i=1 TO 18
4010     LOCATE 68,i
4020     PRINT tbph$(j)
4030     LOCATE 73,i
4040     PRINT exemple$(j)
4050     j=j+1
4060 NEXT i
4070 RETURN
4080 REM *****
4090 REM *   INSERTION DU CARACTERE   *
4100 REM * CARACTERISANT UNE SYLLABE **
4110 REM *           EUIL ILLE ... :   *
4120 REM *           ! OU # OU # % à   *
4130 REM *****
4140 DROITE$=RIGHT$(PHRASE$(LEN(PHRASE$)-
(I+DECAL)))
4150 GAUCHE$=LEFT$(PHRASE$,I)
4160 PHRASE$=GAUCHE$+DROITE$
4170 RETURN
4180 REM *****
4190 REM *   CODAGE DANS LECODE()   *
4200 REM *   DES CARACTERES       *
4210 REM * CARACTERISANT UNE SYLLABES *
4220 REM *           EUIL ILLE ...   *
4230 REM *****
4240 IF morceau$ <> "!" THEN 4310
4250     lecode(k)=25
4260     LOCATE 50,25
4270     PRINT "   "
4280     LOCATE 50,25
4290     PRINT "ILLE"
4300     i=i+1
4310 IF morceau$ <> "#" THEN 4380
4320     lecode(k)=44
4330     LOCATE 50,25
4340     PRINT "   "

```

```
4350     LOCATE 50,25
4360     PRINT "ILLE"
4370     i=i+1
4380 IF morceau$ <> "#" THEN 4450
4390     lecode(k)=62
4400     LOCATE 50,25
4410     PRINT "   "
4420     LOCATE 50,25
4430     PRINT "OUIL"
4440     i=i+1
4450 IF morceau$ <> "%" THEN 4520
4460     lecode(k)=75
4470     LOCATE 50,25
4480     PRINT "   "
4490     LOCATE 50,25
4500     PRINT "SIO"
4510     i=i+1
4520 IF morceau$ <> "à" THEN 4590
4530     lecode(k)=82
4540     LOCATE 50,25
4550     PRINT "   "
4560     LOCATE 50,25
4570     PRINT "TION"
4580     i=i+1
4590 IF morceau$ <> "'" THEN 4660
4600     lecode(k)=96
4610     LOCATE 50,25
4620     PRINT "   "
4630     LOCATE 50,25
4640     PRINT "ZIO"
4650     i=i+1
4660 RETURN
4670 REM *****
4680 REM ** CORRECTION D'UNE PHRASE **
4690 REM *****
4700 CLS
4710 PRINT SPACE$(24);
4720 PRINT " CORRECTION D'UNE PHRASE "
4730 GOSUB 3610
4740 LOCATE 1,19
4750 PRINT "Entrez le numero de la phras
e a corriger : "
4760 no=9
4770 PRINT CHR$(24)
4780 LOCATE 43,19
4790 PRINT "   "
4800 WHILE no > 5 OR no < 1
4810     LOCATE 44,19
4820     a$=INKEY$
4830     no=VAL(a$)
```

```

4840         LOCATE 44,19
4850         PRINT a$
4860 WEND
4870 PRINT CHR$(24)
4880 LOCATE 1,20
4890 IF tbcodé(no,0) <> 0 THEN 4940
4900         PRINT "Il n'y a rien dans cette
phrase !!!"
4910         FOR i= 1 TO 1000
4920             NEXT
4930             GOTO 4950
4940 GOSUB 5030
4950 LOCATE 1,20
4960 PRINT SPACE$(240)
4970 LOCATE 1,20
4980 FOR j=1 TO tbcodé(no,0)
4990         PRINT tbph$(tbcodé(no,j));
5000 NEXT j
5010 FOR i=1 TO 1000:NEXT i
5020 RETURN
5030 LOCATE 1,20
5040 FOR I=1 TO 240
5050         PRINT "_";
5060 NEXT i
5070 LOCATE 1,20
5080 FOR j=1 TO tbcodé(no,0)
5090         PRINT tbph$(tbcodé(no,j));
5100 NEXT j
5110 LOCATE 1,20
5120 INPUT "",phrase$
5130 phrase$=UPPER$(phrase$)
5140 phrase$=phrase$+" "
5150 IF phrase$=" " THEN GOSUB 2070 E
LSE IF LEN(phrase$)<245 THEN GOSUB 1550
ELSE LOCATE 1,23:PRINT "PHRASE TROP LONGU
E !!!"
5160 RETURN
5170 '***** CALCUL DU TEMPS
ECOULE *****
5180 TEMPSPASSE=TIME-TEMPS
5190 TEMPSPASSE=TEMPSPASSE/300
5200 H1=TEMPSPASSE/3600
5210 H2=FIX(H1)' heures
5220 M1=(H1-H2)*60
5230 M2=FIX(M1)' Minutes
5240 S1=(M1-M2)*60
5250 S2=FIX(S1)' Secondes
5260 D1=(S1-S2)*100
5270 D2=FIX(D1)' Centiemes
5280 PRINT CHR$(24)

```

```

5290 LOCATE 31,1
5300 PRINT USING"## H ## M ## S ##";H2;M
2;S2;D2
5310 PRINT CHR$(24)
5320 RETURN
5330 REM *****
5340 REM ***** GESTION DES ERREURS *****
5350 REM *****
5360 '
5370 REM ***** ERREURS PROGRAMME *****
5380 PRINT
5390 PRINT "ERREUR No ";ERR
5400 PRINT "EN LIGNE ";ERL
5410 CALL &BB06:RESUME 390
5420 REM *** CHARGEMENT IMPOSSIBLE ***
5430 PRINT "Le fichier ";nomfic$;" n'exi
ste pas"
5440 CALL &BB06
5450 RESUME 430
5460 REM *** SAUVEGARDE IMPOSSIBLE ***
5470 PRINT "Le fichier ";nomfic$;" ne pe
ut pas etre sauvegarde "
5480 CALL &BB06
5490 RESUME 430
5500 REM *****
5510 REM *** TRAITEMENT *****
5520 REM *****
5530 i=1:k=1
5540 morceau$=""
5550 WHILE LEN(phrase$) > i
5560     morceau$=morceau$+tbphrase$(i
)
5570     j=0
5580     WHILE tbph$(j) <> morceau$
AND j < 103
5590         j=j+1
5600     WEND
5610     IF morceau$="!" OR morceau$=
"# " OR morceau$="à " OR
morceau$="%" OR morceau$="'"
OR MORCEAU$="#" THEN GOSUB
4240 ELSE IF j > 102 THEN
GOSUB 2350 ELSE GOSUB 2490
5620 WEND
5630 RETURN

```

### RECUPERATION DES CODES

Il est possible de récupérer les codes créés et sauvegardés dans un programme de votre réalisation utilisant le synthétiseur vocal.

Nous allons auparavant expliciter le tableau tbcodes (5,256) qui sert à la sauvegarde des codes des phrases, dont la structure est donnée en figure 3.

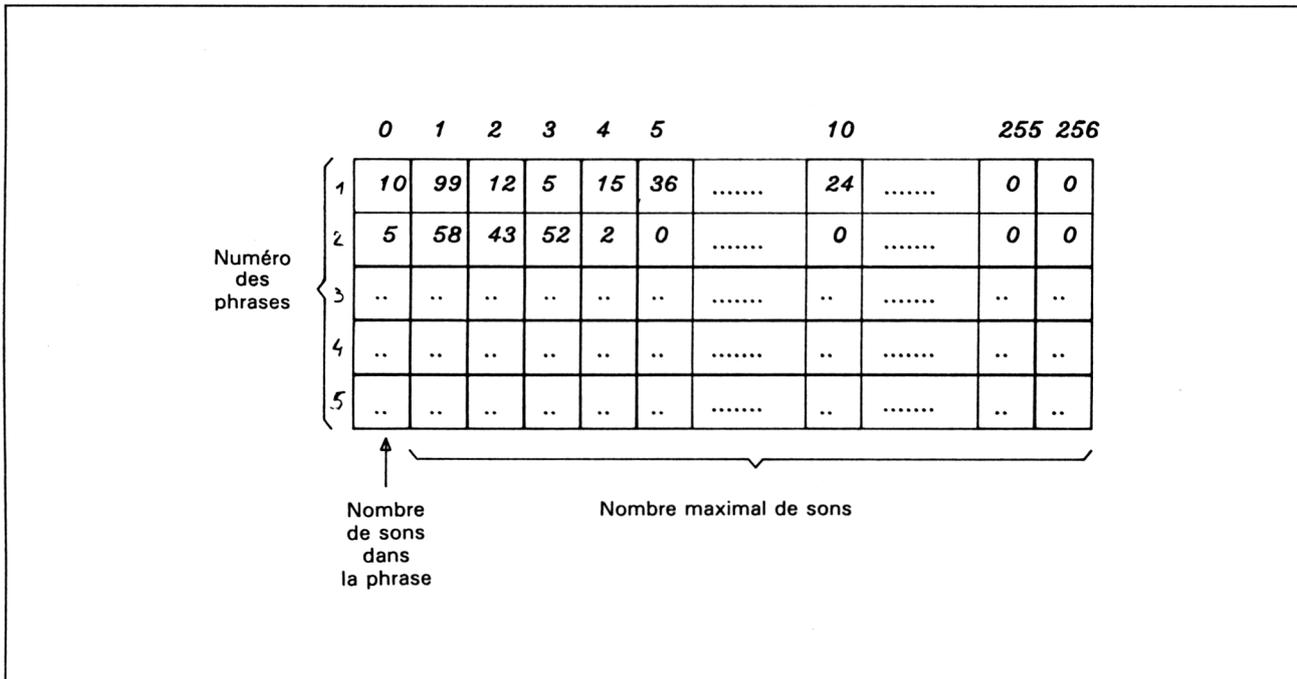


Fig. 3 : Organisation de tbcodes (i,j).

Chaque phrase occupant une ligne du tableau, le premier code étant le nombre de codes de sons dans la phrase, suivis de ces sons.

Le tableau est ensuite sauvegardé dans un fichier à longueur variable, sur disquette, dont l'organisation est représentée en figure 4.

Lorsque l'on veut récupérer les différents codes, il suffit d'ouvrir le fichier en lecture, de lire la première valeur qui est le nombre de codes de la phrase 1, puis, à l'aide d'une boucle, de lire les codes, soit dans un tableau, soit en les exécutant de suite (en n'oubliant pas d'ajouter la tonalité de départ), et ainsi de suite pour les autres phrases : on pourra prendre exemple sur les lignes 3090 à 3260 du programme.

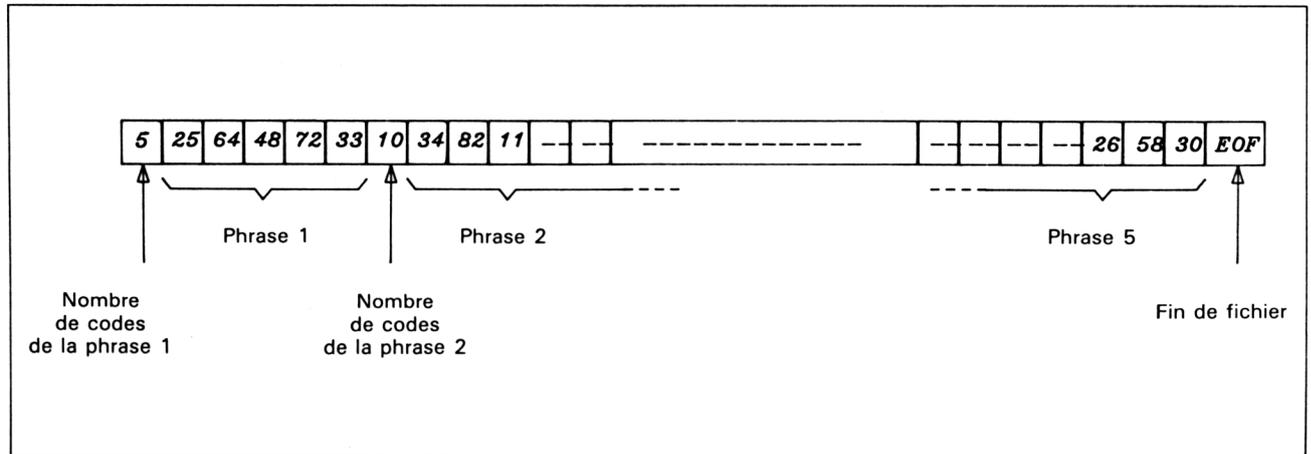


Fig. 4 : Gestion de fichier à longueur variable.

#### IV. Le synthétiseur pédagogique

L'utilisation du synthétiseur vocal vous semble peut-être un peu futile, mais elle rend de grands services pédagogiques et aide à la programmation pour les non-voyants.

Nous vous proposons un programme simple que vous pourrez expérimenter avec des enfants pour l'apprentissage de l'alphabet, et qui vous apportera, nous l'espérons, des idées nouvelles permettant d'éveiller le désir d'apprendre chez les tout-petits.

Ce programme, après un « **BONJOUR** » vocal et coloré invite à frapper sur les touches alphabétiques du clavier, et les épelle à voix haute lors de l'appui sur celles-ci.

```

10 REM *****
20 REM *** REVISION DE L'ALPHABET ***
30 REM *****
40 REM
50 REM *** CHARGEMENT PHONEMES ***
60 REM *ET INITIALISATION INSTRUCTION *
70 REM *****
80 MEMORY &92C2
90 LOAD "PHC"
100 CALL &A655
110 REM *****
120 REM *** PRESENTATION *****
130 REM *****
140 INK 0,0:INK 1,18
150 BORDER 0:PAPER 0:PEN 1
160 MODE 0
170 LOCATE 7,10

```

```
180 PRINT"BONJOUR"
190 FOR I = 24 TO 1 STEP -1
200 FOR J = 1 TO 100:NEXT J
210 INK 1,I
220 NEXT I
230 ÛSPEAK,65,97,3,60,48,61,72: REM BONJ
OUR
240 FOR I = 1 TO 24
250 FOR J = 1 TO 100:NEXT J
260 INK 1,I
270 NEXT I
280 FOR i = 1 TO 2000:NEXT i
290 MODE 1
300 INK 1,25
310 LOCATE 2,5
320 PRINT "Revisons ensemble l'alphabet"
330 ÛSPEAK,65,72,22,88,39,94,60,102,2
340 ÛSPEAK,65,74,2,6,102,50,0,50,50,99
350 ÛSPEAK,65,26,0,99,3,22
360 PRINT:PRINT
370 FOR i = 1 TO 2000:NEXT i
380 INK 0,27
390 INK 1,6
400 PRINT "Appuies sur les touches de l'
alphabet..."
410 ÛSPEAK,65,0,63,63,86,102,73,85,72
420 ÛSPEAK,65,102,50,22,80,61,11,102,16
430 ÛSPEAK,65,20,102,50,0,50,50,99,26
440 ÛSPEAK,65,0,99,3,22
450 FOR i = 1 TO 2000: NEXT i
460 INK 0,0:INK 1,21
470 REM *****
480 REM *** PROGRAMME PRINCIPAL ***
490 REM *****
500 lettre$ = INKEY$
510 IF lettre$ = "" THEN 500
520 MODE 2
530 LETTRE$=UPPER$(LETTRE$)
540 IF lettre$ < "A" OR lettre$ > "Z" TH
EN GOTO 1250
550 PRINT lettre$
560 BORDER 0
570 REM *****
580 REM *** DESSINNE LE CARACTERE ***
590 REM *****
600 FOR I = 1 TO 50
610 X = INT(RND*79)+1
620 Y = INT(RND*23)+1
630 LOCATE X,Y
640 PRINT LETTRE$
```

```

650 NEXT I
660 LOCATE 38,11
670 FOR a = &C000 TO &F800 STEP &800
680 b = PEEK(a)
690 a$ = BIN$(b,8)
700 FOR i = 1 TO 8
710 IF MID$(a$,i,1) = "1" THEN PRINT CHR
$(127); ELSE PRINT CHR$(32);
720 NEXT i
730 FOR i=1 TO 8
740 PRINT CHR$(8);
750 NEXT i
760 PRINT CHR$(10);
770 NEXT a
780 REM *****
790 REM *** RECHERCHE CARACTERE ***
800 REM *****
810 RESTORE 960
820 READ A$
830 IF LETTRE$=A$ THEN 860
840 GOTO 820
850 STOP
860 I=0
870 WHILE I <> 255
880 READ I
890 @SPEAK,65,I: REM EPELE LA
LETTRE
900 WEND
910 GOTO 500
920 STOP
930 REM *****
940 REM ** PHONEMES DES CARACTERES **
950 REM *****
960 DATA A,0,102,255
970 DATA B,4,22,255
980 DATA C,73,22,255
990 DATA D,16,22,255
1000 DATA E,20,102,255
1010 DATA F,22,26,255
1020 DATA G,48,22,255
1030 DATA H,0,11,255
1040 DATA I,39,102,255
1050 DATA J,48,39,255
1060 DATA K,9,0,255
1070 DATA L,22,50,255
1080 DATA M,22,53,255
1090 DATA N,22,55,255
1100 DATA O,56,102,255
1110 DATA P,63,22,255
1120 DATA Q,09,85,255

```

```

1130 DATA R,23,72,255
1140 DATA S,22,73,255
1150 DATA T,80,22,255
1160 DATA U,85,85,255
1170 DATA V,87,22,255
1180 DATA W,16,61,06,20,87,22,255
1190 DATA X,39,91,255
1200 DATA Y,39,37,22,9,255
1210 DATA Z,94,23,16,255
1220 REM *****
1230 REM ***** ERREUR *****
1240 REM *****
1250 FOR i = 1 TO 10:PRINT CHR$(7):NEXT
i
1260 BORDER 26
1270 MODE 1
1280 PRINT "Ce n'est pas une lettre de l
'alphabet"
1290 PRINT:PRINT:PRINT:PRINT
1300 FOR k=ASC("A") TO ASC("Z")
1310 PRINT CHR$(k);" ";CHR$(k+&20);
1320 PRINT" ";
1330 NEXT k
1340 ùSPEAK,65,74,20,102,54,22,102,63,0
1350 ùSPEAK,65,0,102,85,55,102,50,23,80
1360 ùSPEAK,65,83,20,102,17,20,102,50,0
1370 ùSPEAK,65,50,50,99,26,0,99,3,22
1380 BORDER 3
1390 FOR i = 1 TO 2000: NEXT i
1400 PRINT:PRINT:PRINT:PRINT
1410 GOTO 400

```

Nous vous invitons à améliorer ce programme, à le modifier selon le caractère et l'âge de vos enfants, et à créer par exemple le programme inverse qui demanderait l'appui sur une touche particulière de l'alphabet. Vous pouvez même créer vos programmes d'apprentissage des chiffres, des opérations, de dictées ..., etc.

## V. Un répondeur téléphonique

Pour ceux qui ont quelques connaissances en électronique, nous proposons le schéma fonctionnel permettant de créer un répondeur téléphonique à partir du synthétiseur vocal (voir Fig. 5).

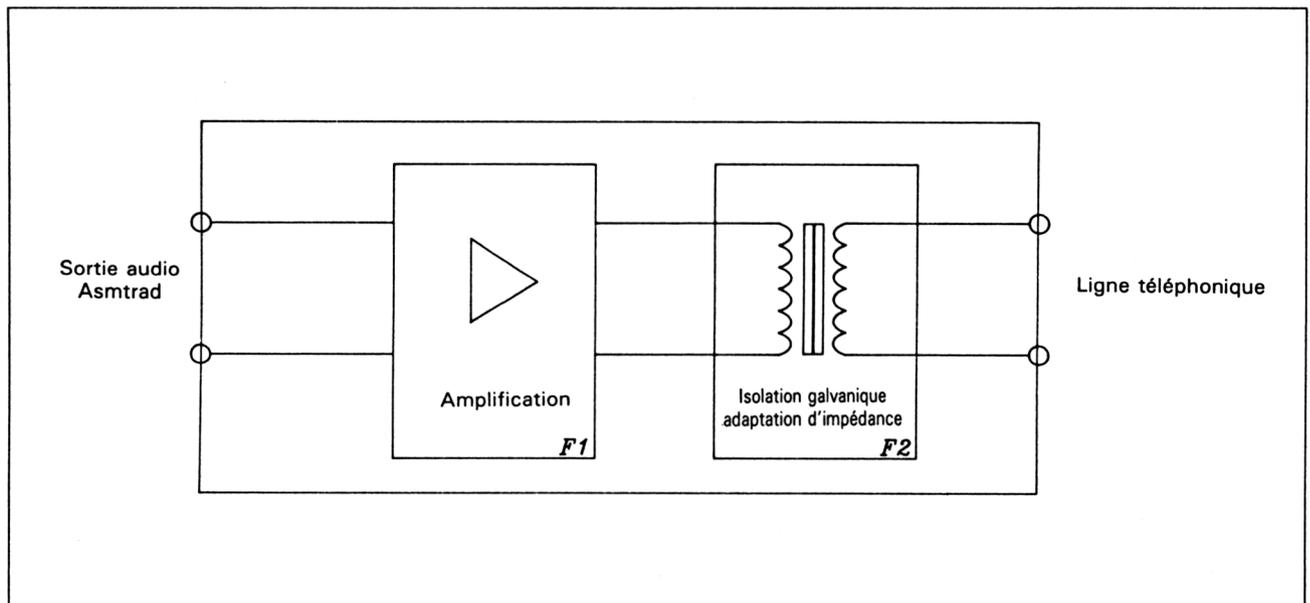


Fig. 5 : Schéma fonctionnel du répondeur téléphonique.

## VI. Le synthétiseur vocal et langage machine

Le programmeur en langage machine sera certainement intéressé par la démarche permettant de créer ses propres sons réutilisables en assembleur.

Mais avant tout, procurez-vous la documentation du composant MEA-8000 auprès de la société LA RADIOTECHNIQUE COMPELEC (RTC), ou de revendeurs électroniques, ainsi que des notes d'application.

### LE PRINCIPE DE PROGRAMMATION DU MEA-8000

Le MEA-8000 possède deux registres internes permettant de mettre en œuvre ses fonctions de synthèse vocale : un registre de contrôle et un registre de données tous deux de huit bits.

Le registre de contrôle fait travailler le MEA-8000 dans différents modes en y écrivant un octet définissant le mode. Il permet aussi de connaître l'état du composant, par une lecture de ce même registre : son émis, composant libre pour un nouveau son, ou son en cours d'émission, composant occupé.

Le registre de données s'utilise en écriture et pour programmer le type de son à synthétiser.

### LE REGISTRE CONTRÔLE

En lecture, ce registre permettra de connaître la phase d'émission des sons par le composant, grâce au bit 7 de l'octet qui y est lu.

Ce bit, lorsqu'il est à zéro indique qu'il est possible d'émettre un son et, lorsqu'il est à un, signale que le composant n'est pas disponible pour une nouvelle donnée.

En écriture, il sert à initialiser le composant, et l'utilisation la plus classique est de charger ce registre avec la valeur &1A, correspondant à un caractère appelé STOP (on prendra soin de vérifier l'occupation du composant auparavant).

La meilleure utilisation de ce registre est de réinitialiser le composant à l'aide d'un STOP avant et après l'émission d'un son.

### LE REGISTRE DE DONNÉES

Le registre de données sert à émettre les sons et requiert un certain nombre de paramètres organisés d'une façon particulière.

Le premier paramètre est un octet appelé PITCH et définissant la hauteur du son.

Les données suivantes assureront la synthèse des sons, et seront émises par blocs (appelés trames) composés obligatoirement de quatre octets. Il est possible d'émettre autant de blocs de quatre octets que l'on veut, tant que le composant n'est pas réinitialisé, sinon il faudra à nouveau entrer le PITCH.

Les trames sont organisées de la façon suivante :

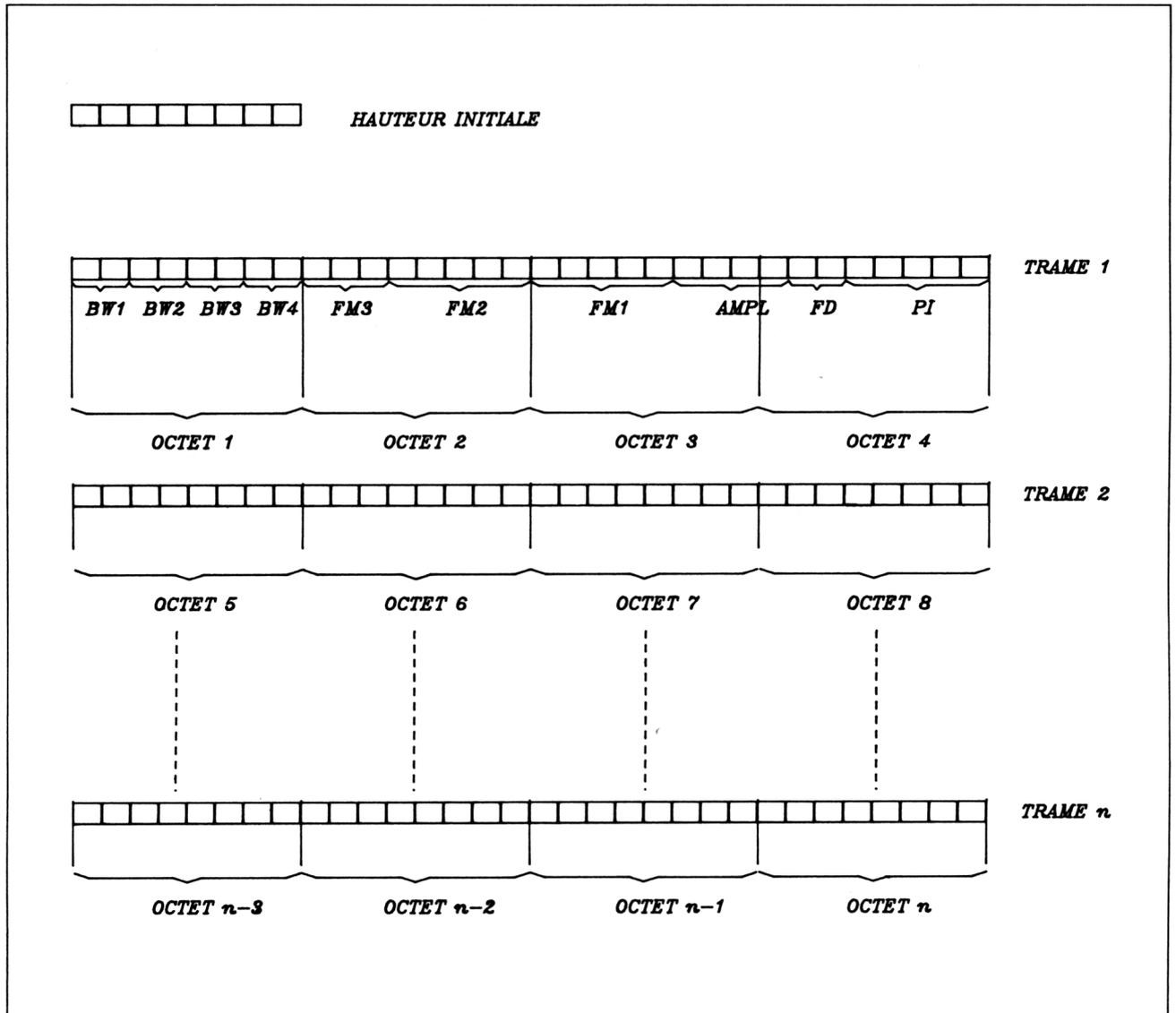


Fig. 6 : Architecture des blocs (trames).

- BW1 : Largeur de bande du premier formant (2 bits)
- BW2 : Largeur de bande du deuxième formant (2 bits)
- BW3 : Largeur de bande du troisième formant (2 bits)
- BW4 : Largeur de bande du quatrième formant (2 bits)
- FM3 : Fréquence du troisième formant (3 bits)
- FM2 : Fréquence du deuxième formant (5 bits)
- FM1 : Fréquence du premier formant (5 bits)

AMPL : Amplitude (4 bits)

FD : Durée de la trame de parole (2 bits)

PI : Incrément du pitch, ou sélection du bruit (5 bits)

Pour ceux qui sont intéressés par la programmation de ces différents paramètres, nous vous conseillons de vous référer à des notes d'applications et à des ouvrages spécialisés sur la synthèse vocale, ou encore de décoriquer le logiciel fourni avec le synthétiseur.

### LE LOGICIEL SUR CPC

Le logiciel fourni avec le synthétiseur a retenu les principes expliqués précédemment et utilise une table comportant la suite des sons émis.

Pour le fichier PHC.BIN, vous retrouverez la table des sons entre les adresses &92C3 et & A654. Pour chacun des sons, vous trouverez d'abord le nombre d'octets pour le son, codé sur 16 bits (2 octets), suivis d'un octet nul séparateur et des octets du son. A partir de cette table sont effectués des calculs relativement complexes pour retrouver l'adresse des différents sons.

Si vous désirez réécrire un programme plus modulaire, et permettant d'ajouter des sons ultérieurement, ou d'effectuer des corrections, nous vous conseillons d'organiser la table selon la figure 7.

A titre d'expérimentation, nous vous fournissons ci-dessous les deux séries de codes hexadécimaux manquant pour compléter le programme HOP.BIN afin de lui faire dire les nombres de zéro à 999 :

```
SOIXAN : 1A F7 F8 10 1B F7 FC 10
          1A F7 FC 30 1A F7 FC F0
          92 B0 7B C0 DA 8E 6B 20
          67 B1 8F 48 BB 92 9F A4
          BA 93 9F A0 A0 B4 86 5D
          2A F7 9D D0 1A F7 9C F0
          15 D6 8D 50 B6 B3 7E 22
          B2 B3 97 3E 36 D1 9F 5D
          31 CF 8E 7C 32 AE 94 FF
          20 8E 53 C2 21 8D 48 63
CENT :   1A F7 C8 10 1A F7 CB 90
          1A F7 CB B0 1A F7 AD 70
          1A F4 8D F0 B5 D2 86 A0
          66 D1 9F AE 22 CF 8F 40
          12 CD 8E DC 52 AA 84 DD
          52 AD 83 3E 10 AE 78 41
```

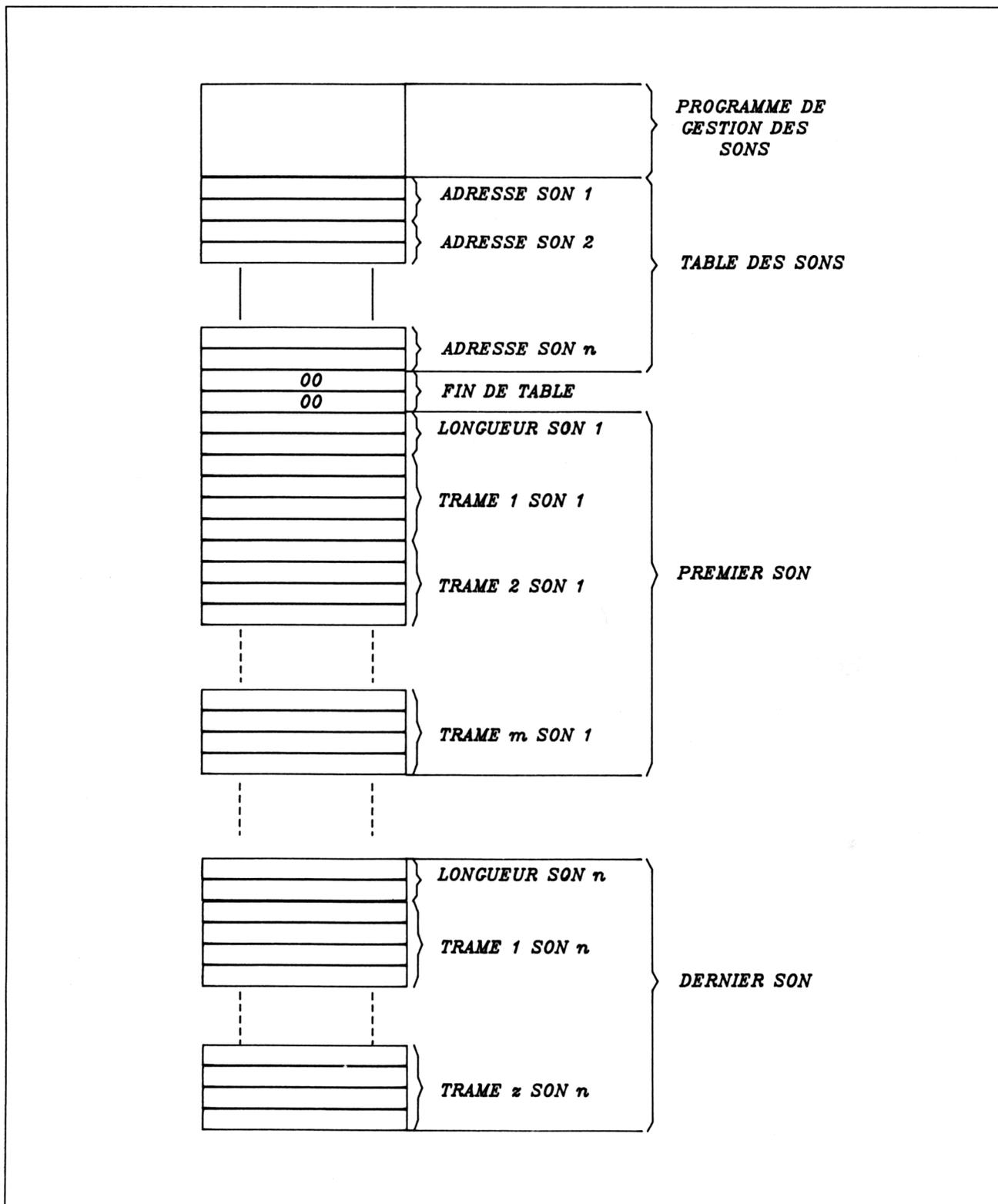


Fig. 7 : Organisation de la table des sons.

