

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 aisément réalisable 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).

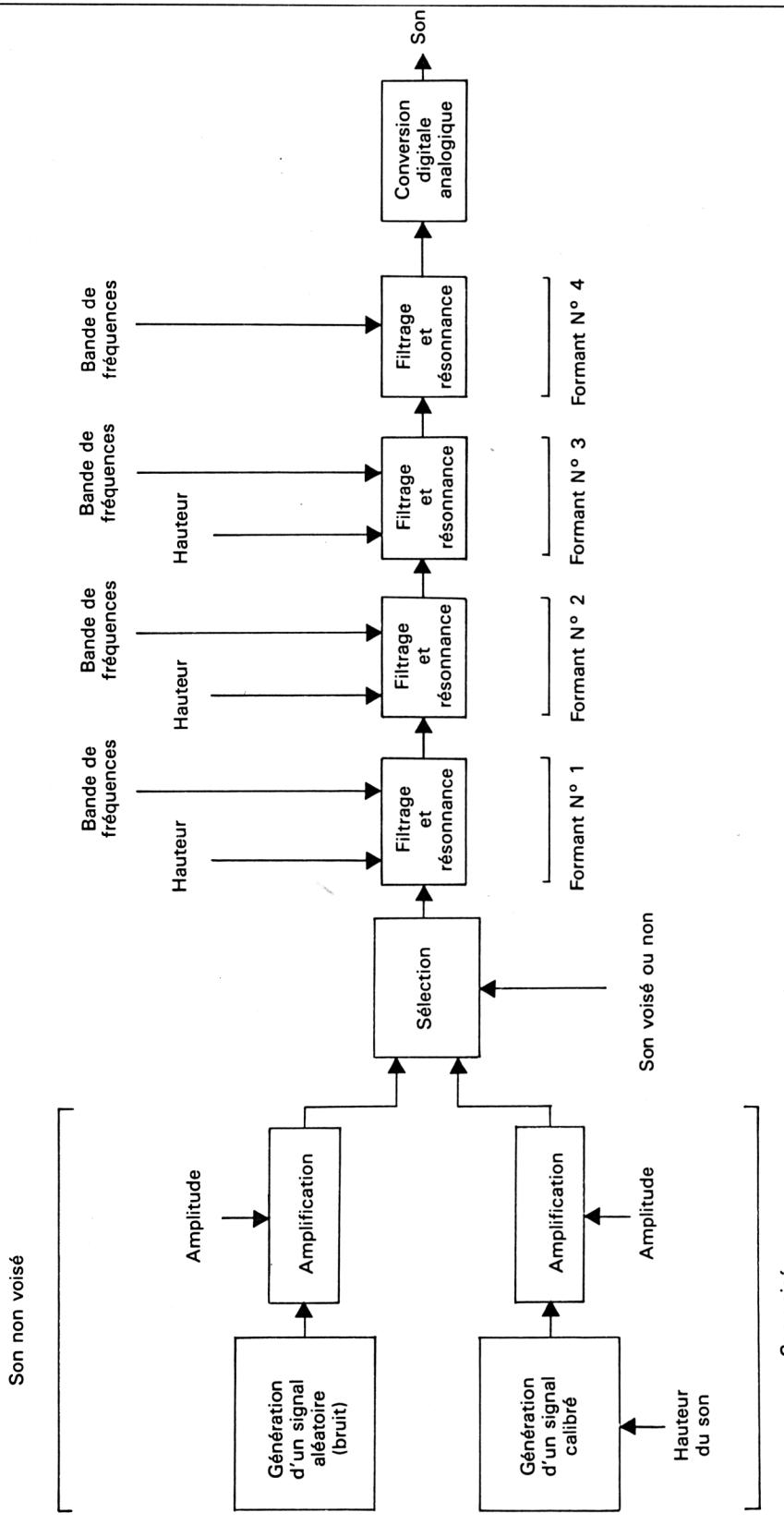


Fig. 1 : Principe de génération d'un son par le composant MEA-8000.

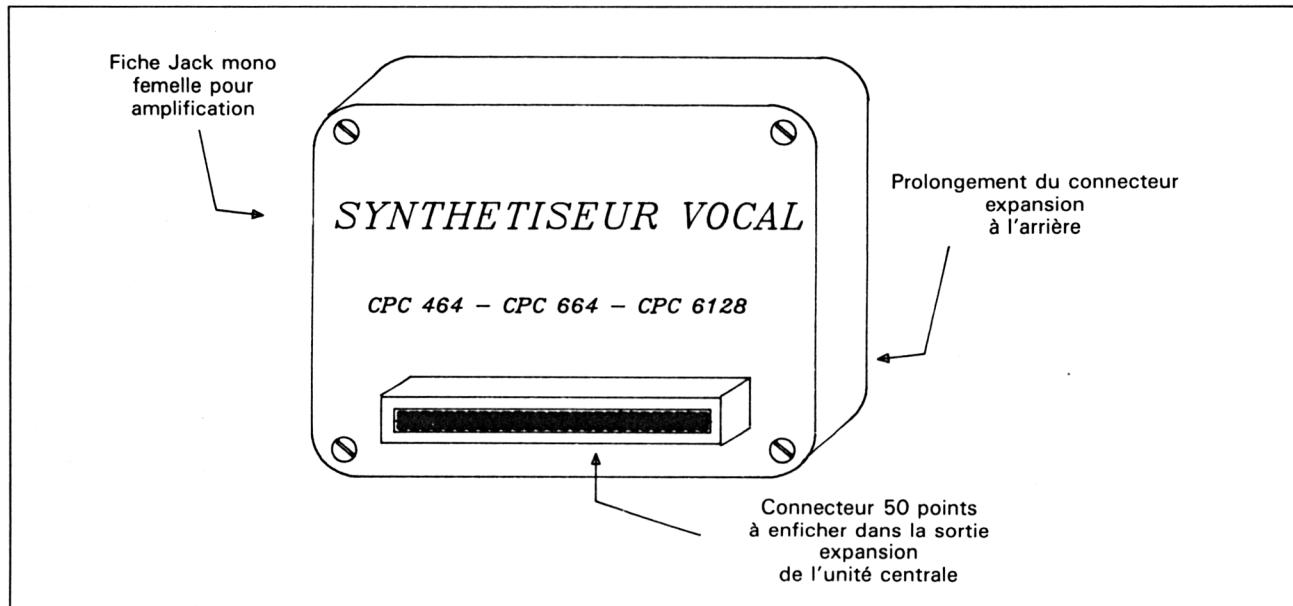


Fig. 2 : Synthétiseur vocal.

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^{er} # seulement
ligne 4310
ligne 5610, le 1^{er} # seulement

```

10 REM ****
20 REM ** SYNTHETISEUR VOCAL V 1.1    **
30 REM ** APRES L'INTRODUCTION D'UNE   **
40 REM ** PHRASE EN FRANCAIS AVEC      **
50 REM ** QUELQUES RESTRICTIONS       **
60 REM ** DE SYNTAXE DES             **
70 REM ** DIFFERENTS PHONEMES        **
80 REM ** RESTITUTION PAR LE H.P.    **
90 REM ****
100 REM
110 DEFINT A-C:DEFINT E-G:DEFINT I-L
120 DEFINT N-R:DEFINT U-Z
130 TEMPS=TIME
140 REM
150 MEMORY &92C2
160 LOAD "PHC"
170 CALL &A655
180 REM
190 MODE 1
200 REM
210 REM ****
220 REM ***** INITIALISATIONS *****
230 REM ****
240 REM
250 DIM tbph$(103)
260 DIM exemple$(102)
270 DIM lecode(256)
280 DIM tbphrase$(256)
290 DIM tbcode(5,256)
300 i=0
310 code = 0
320 WHILE i < 103
330     READ CODE,phoneme$,exemp$
340     tbph$(i)=UPPER$(phoneme$)
350     exemple$(i)=UPPER$(exemp$)
360     i=i+1
370 WEND
380 REM
390 REM ****
400 REM ***** MENU *****
410 REM ****
420 REM
430 CHOIX$="B"
440 MODE 2:INK 0,0:INK 1,26
450 PAPER 0:OPEN 1:BORDER 0
460 ON ERROR GOTO 5370
470 REM
480 WHILE CHOIX$ <> "7"
490     CLS
500     LOCATE 49,1

```

```
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 *** DETECTION DES ENSEMBLES ***
1500 REM ** DE CARACTERES
1510 REM ** EGAUX A EUIL, ILLE, ***
1520 REM ** OUIL, SIO, TION OU ZIO ***
1530 REM ****
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$) <> 1 THEN GOSUB 553
0 ELSE k=0
1980 REM ****
1990 REM ** EXECUTION DE LA PHRASE **
2000 REM ** PRECEDAMMENT INTRODUITE **
2010 REM ****
2020 i=1
2030 WHILE i <= k
2040     QSPEAK,60,lecode(i)
2050     i=i+1
2060 WEND
2070 LOCATE 1,23
2080 PRINT "Entrez le numero de la phrase (1 a 5) : "
2090 no=12
2100 PRINT CHR$(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 tbcode(no,0)=k
2240 i=1
2250 WHILE i <= k
2260     tbcode(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$(lecode(k))
2580 i=i+1
2590 RETURN
2600 REM ****
2610 REM *** EXECUTION D'UNE PHRASE ***
2620 REM ****
2630 CLS
2640 PRINT " Execution d'une phrase"
2650 LOCATE 1,5
2660 PRINT "Entrez le numero de la phrase a executer : "
2670 no=9
2680 PRINT CHR$(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$(24)
2810 IF tbcode(no,0) <> 0 THEN 2860
2820     PRINT "Il n'y a rien dans";
2830     PRINT " cette phrase !!!!"
2840     FOR i=1 TO 1000:NEXT i
2850     RETURN
2860 FOR j=1 TO tbcode(no,0)
2870     PRINT tbph$(tbcode(no,j));
2880 NEXT j
2890 PRINT
2900 FOR i=1 TO tbcode(no,0)
2910     QSPEAK,60,tbcode(no,i)
2920 NEXT i
2930 RETURN
2940 REM ****
2950 REM ***** LISTE DES PHRASES *****
2960 REM ****
2970 CLS
2980 PRINT SPACE$(22);
2990 PRINT"LISTE DES PHRASES"
3000 FOR i= 1 TO 5
3010     PRINT"Phrase no : ";i
3020     FOR j=1 TO tbcode(i,0)
3030         PRINT tbph$(tbcode(i,j));
3040     NEXT j
3050     PRINT
3060 NEXT i
3070 CALL &BB06
3080 RETURN
3090 REM ****
3100 REM *** CHARGEMENT DU FICHIER ***
3110 REM ****
3120 CLS
3130 CAT
3140 ON ERROR GOTO 5420
3150 INPUT "Entrez le nom du fichier a charger : ",nomfic$
3160 nomfic$=nomfic$+".phr"
3170 nomfic$=UPPER$(nomfic$)
3180 OPENIN nomfic$
3190 PRINT nomfic$+" en cours de chargement ..."
3200 FOR i=1 TO 5
3210     INPUT#9,tbcode(i,0)
3220     FOR j=1 TO tbcode(i,0)
3230         INPUT #9,tbcode(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 à sauvegarder : ",nomfic$
3350 nomfic$=nomfic$+".phr"
3360 nomfic$=UPPER$(nomfic$)
3370 OPENOUT nomfic$
3380 PRINT nomfic$+" en cours de sauvegarde ..."
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,"¤", "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$(24);
4720 PRINT " CORRECTION D'UNE PHRASE "
4730 GOSUB 3610
4740 LOCATE 1,19
4750 PRINT "Entrez le numero de la phrase a corriger : "
4760 no=9
4770 PRINT CHR$(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$(24)
4880 LOCATE 1,20
4890 IF tbcode(no,0) <> 0 THEN 4940
4900      PRINT "Il n'y a rien dans cette
phrase !!!"
4910      FOR i= 1 TO 1000
4920      NEXT
4930      GOTO 4950
4940 GOSUB 5030
4950 LOCATE 1,20
4960 PRINT SPACE$(240)
4970 LOCATE 1,20
4980 FOR j=1 TO tbcode(no,0)
4990      PRINT tbph$(tbcode(no,j));
5000 NEXT j
5010 FOR i=1 TO 1000:NEXT i
5020 RETURN
5030 LOCATE 1,20
5040 FOR I=1 TO 240
5050      PRINT "_";
5060 NEXT i
5070 LOCATE 1,20
5080 FOR j=1 TO tbcode(no,0)
5090      PRINT tbph$(tbcode(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$)<245 THEN GOSUB 1550
ELSE LOCATE 1,23:PRINT "PHRASE TROP LONGUE !!!"
5160 RETURN
5170 ***** CALCUL DU TEMPS
      ECOULE *****
5180 TEMPSPASSE=TIME-TEMPS
5190 TEMPSPASSE=TEMPSPASSE/300
5200 H1=TEMPSPASSE/3600
5210 H2=FIX(H1) ' heures
5220 M1=(H1-H2)*60
5230 M2=FIX(M1) ' Minutes
5240 S1=(M1-M2)*60
5250 S2=FIX(S1) ' Secondes
5260 D1=(S1-S2)*100
5270 D2=FIX(D1) ' Centièmes
5280 PRINT CHR$(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'existe pas"
5440 CALL &BB06
5450 RESUME 430
5460 REM *** SAUVEGARDE IMPOSSIBLE ***
5470 PRINT "Le fichier ";nomfic$;" ne peut pas etre sauvegarde"
5480 CALL &BB06
5490 RESUME 430
5500 REM ****
5510 REM ***      TRAITEMENT      ***
5520 REM ****
5530 i=1:k=1
5540 morceau$=""
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 tbcode (5,256) qui sert à la sauvegarde des codes des phrases, dont la structure est donnée en figure 3.

Fig. 3 : Organisation de tbcode (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.

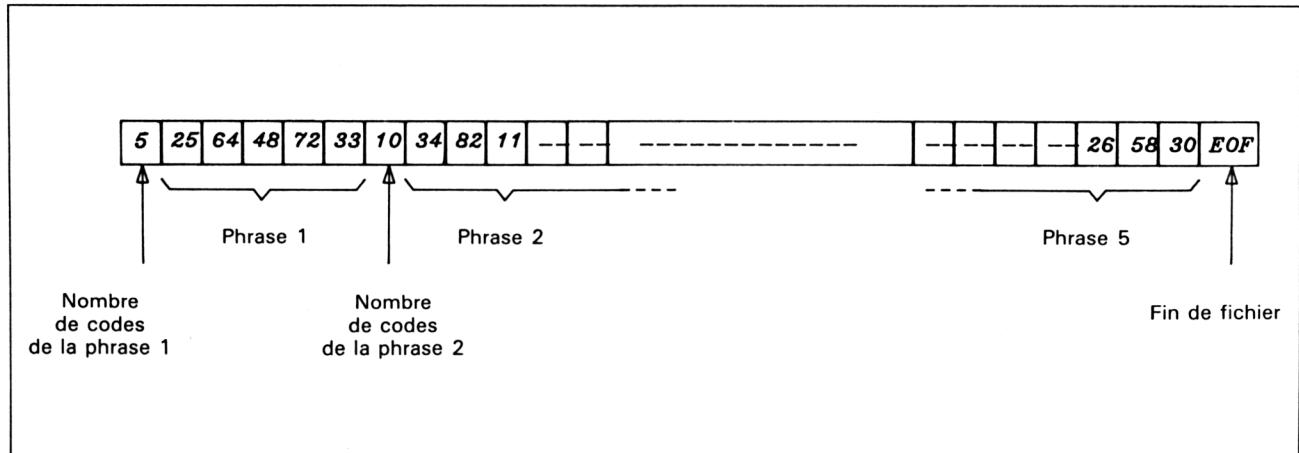


Fig. 4 : Gestion de fichier à longueur variable.

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:OPEN 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 QSPEAK,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 QSPEAK,65,72,22,88,39,94,60,102,2
340 QSPEAK,65,74,2,6,102,50,0,50,50,99
350 QSPEAK,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 QSPEAK,65,0,63,63,86,102,73,85,72
420 QSPEAK,65,102,50,22,80,61,11,102,16
430 QSPEAK,65,20,102,50,0,50,50,99,26
440 QSPEAK,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     QSPEAK,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 @SPEAK,65,74,20,102,54,22,102,63,0
1350 @SPEAK,65,0,102,85,55,102,50,23,80
1360 @SPEAK,65,83,20,102,17,20,102,50,0
1370 @SPEAK,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).

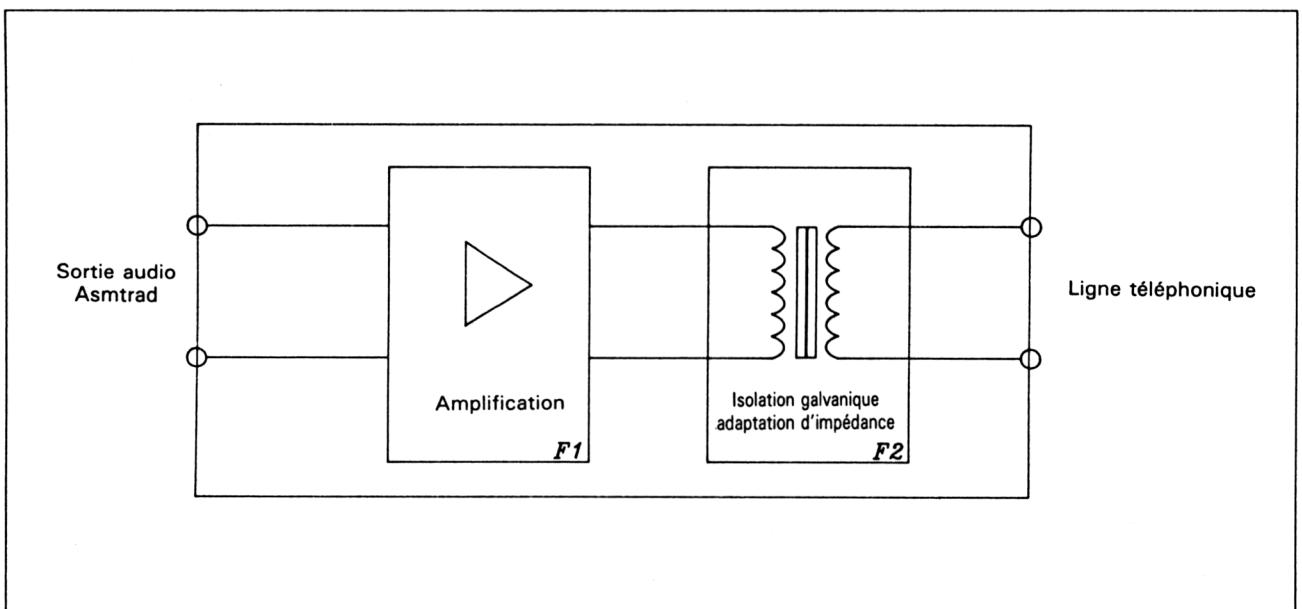


Fig. 5 : Schéma fonctionnel du répondeur téléphonique.

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 :

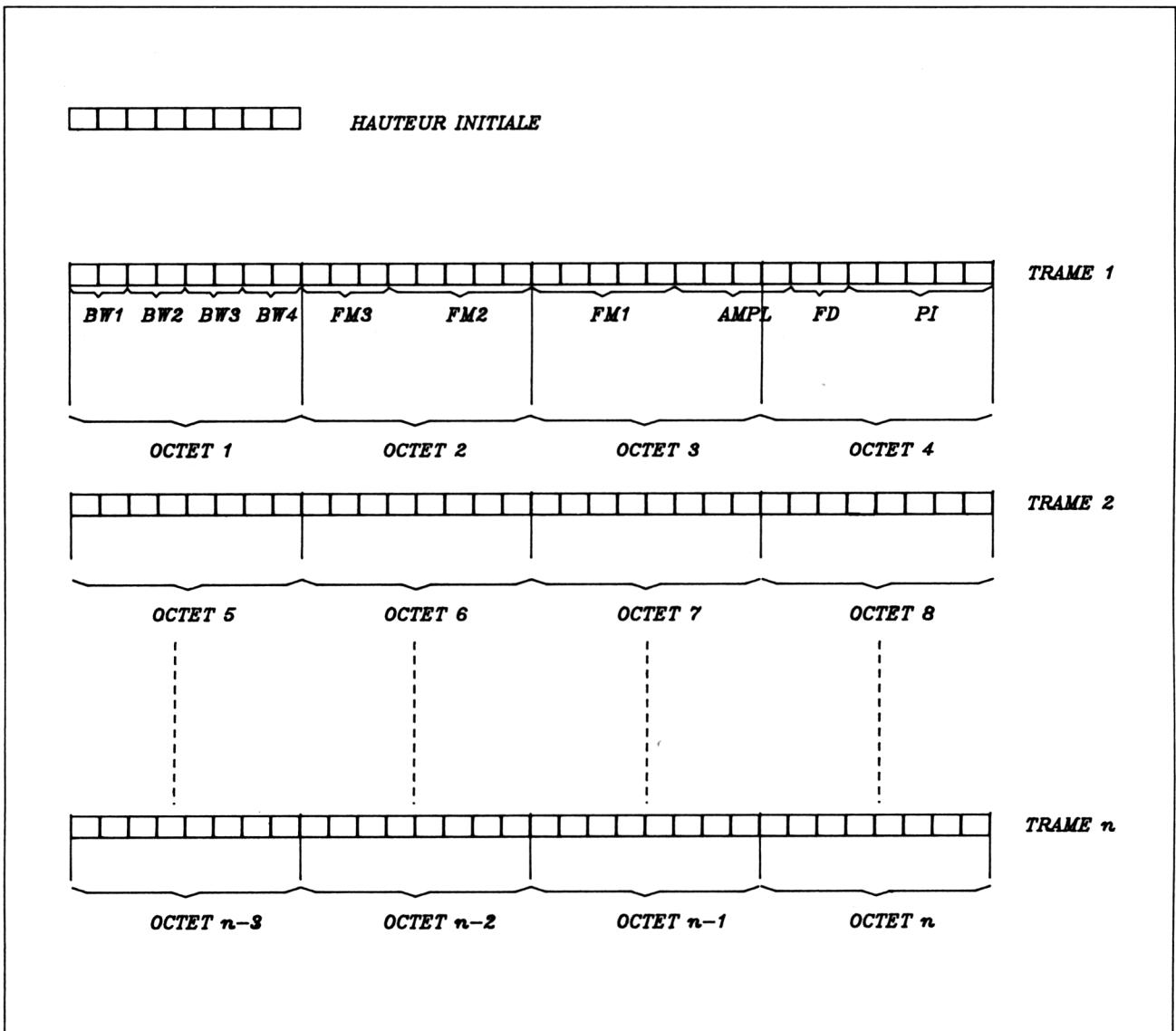


Fig. 6 : Architecture des blocs (trames).

- 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 décorer le logiciel fourni avec le synthétiseur.

LE LOGICIEL SUR CPC

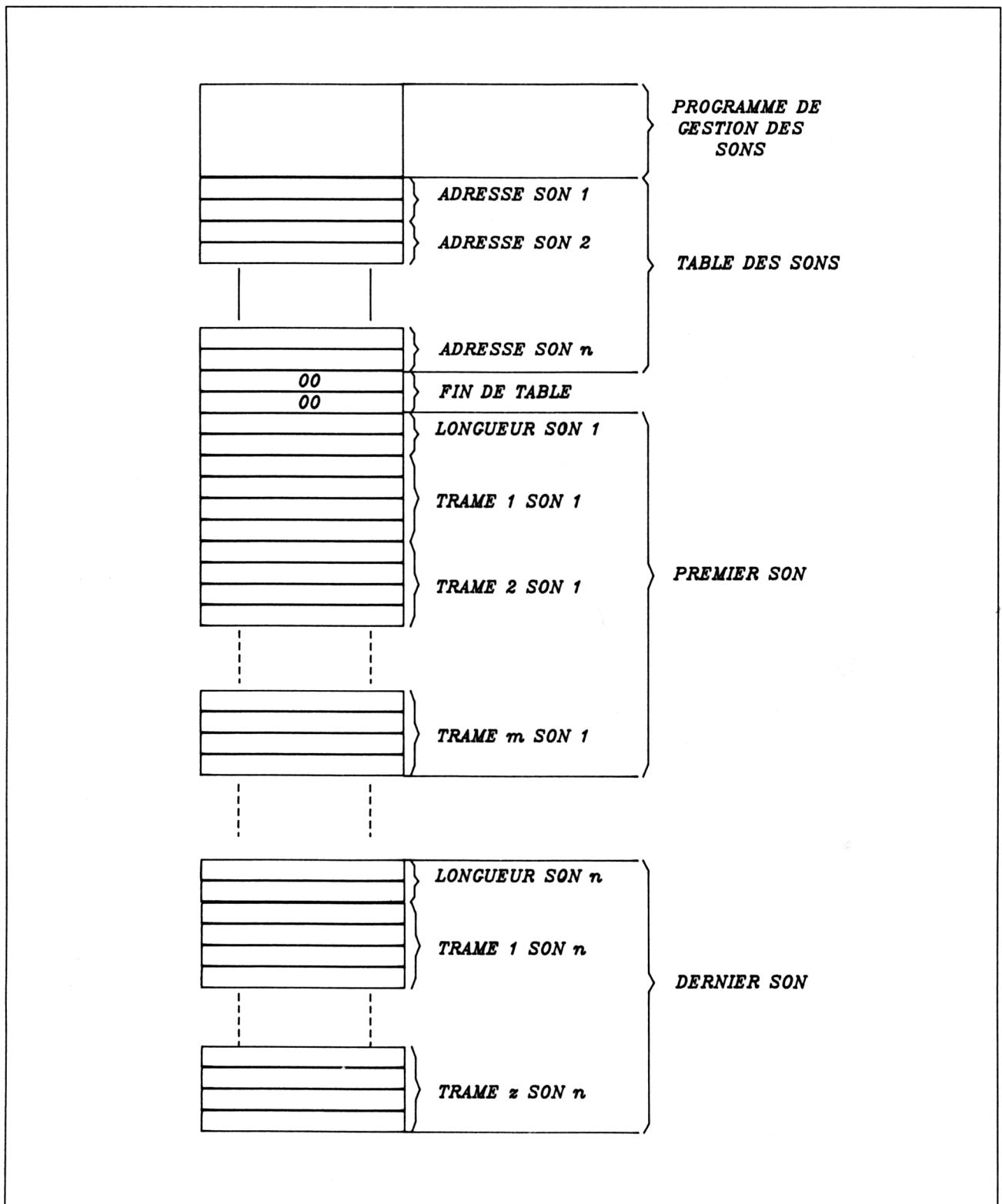
Le logiciel fourni avec le synthétiseur a retenu les principes expliqués précédemment et utilise une table comportant la suite des sons émis.

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.

