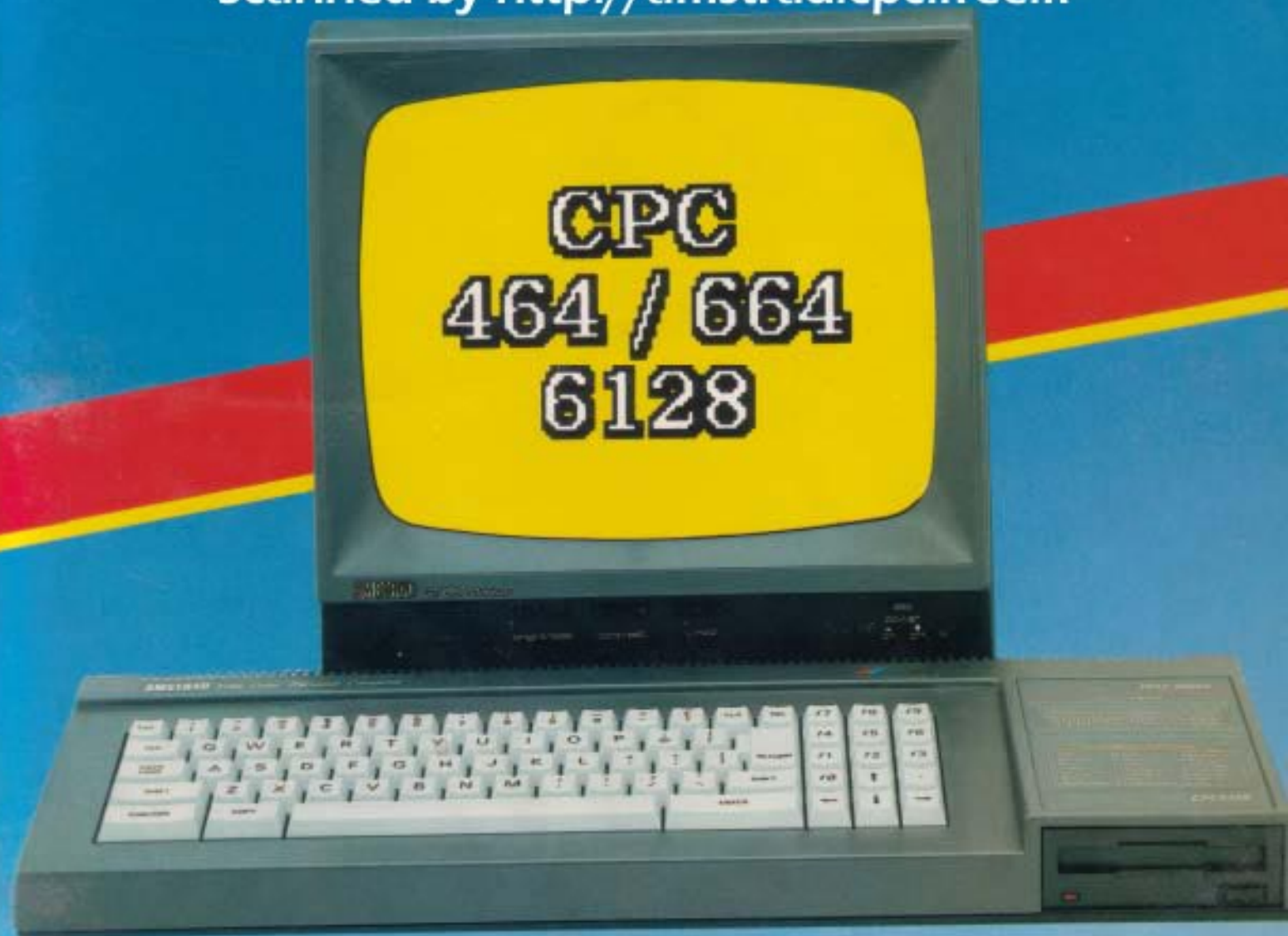


Comment exploiter
toutes les ressources
et augmenter les
performances de votre

AMSTRAD

Scanned by <http://amstrad.cpc.free.fr>



6/0

Table des matières

- 6/1 Définitions
- 6/2 Introduction au circuit sonore AY3-8912
- 6/3 Programmation du circuit sonore en BASIC
- 6/4 Programmation du circuit sonore en ASSEMBLEUR
- 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

6/1

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 ou 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^{re}) Générateurs sonores

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

2nd) Générateur de bruit blanc

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

3rd) Mélangeur de canaux

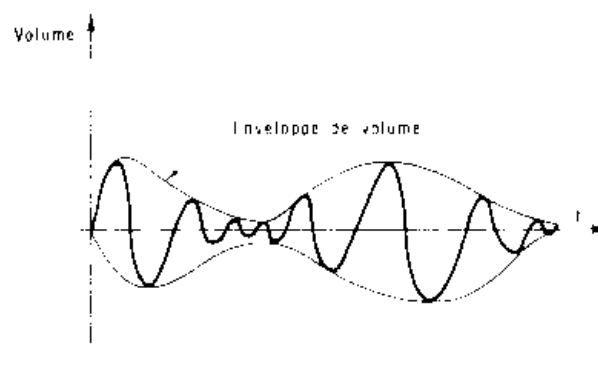
Permet de mixer le bruit blanc et les trois canaux sonores.

4°) Contrôleur d'amplitude

Fournit au convertisseur digital analogique (CDA) le volume sonore de la sortie analogique. Ce volume peut être fixe 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

En 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 BC*.

Si BDIR = 0 et BC* = 0 AY3 est inactif.

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

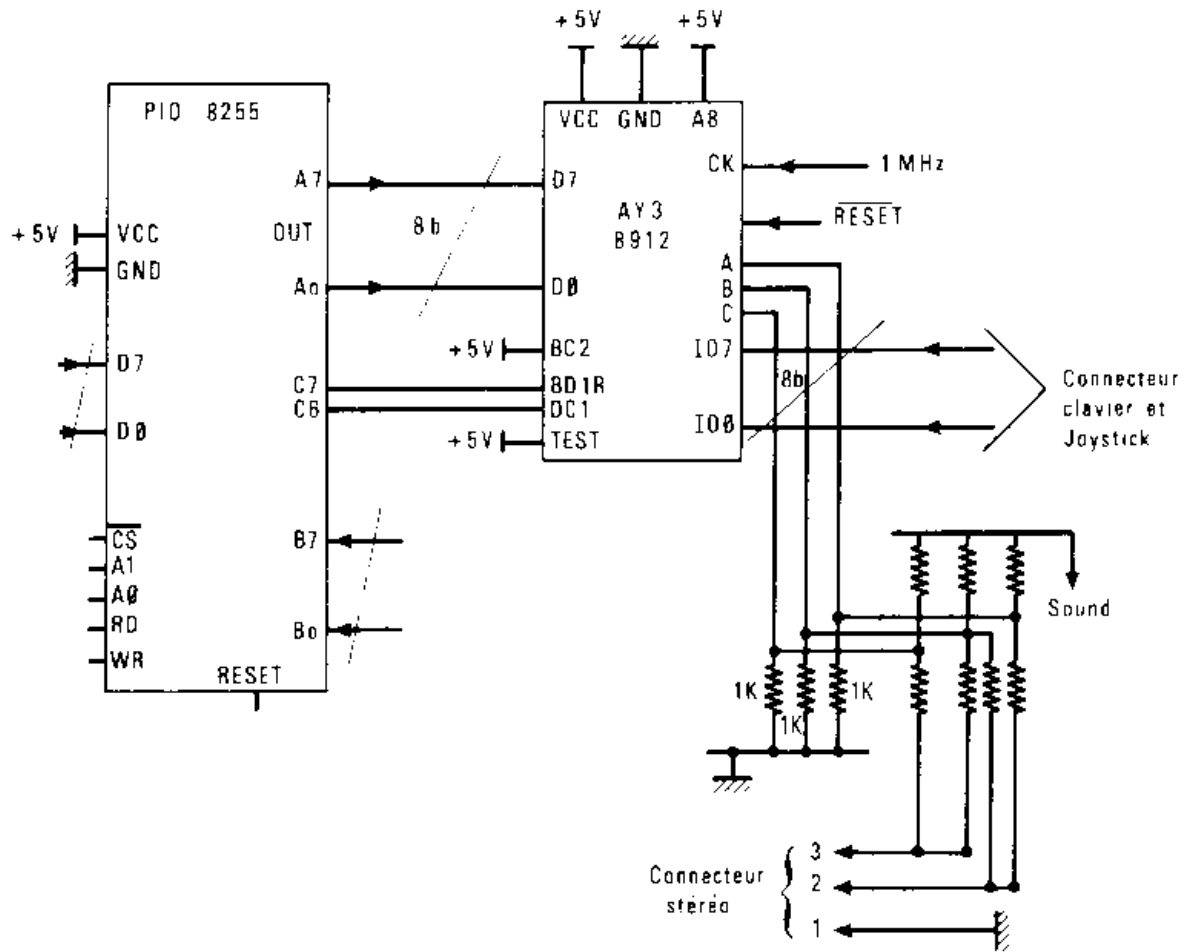
Si BDIR = 1 et BC* = 0 AY3 est en mode écriture.

Si BDIR = 1 et BC* = 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.

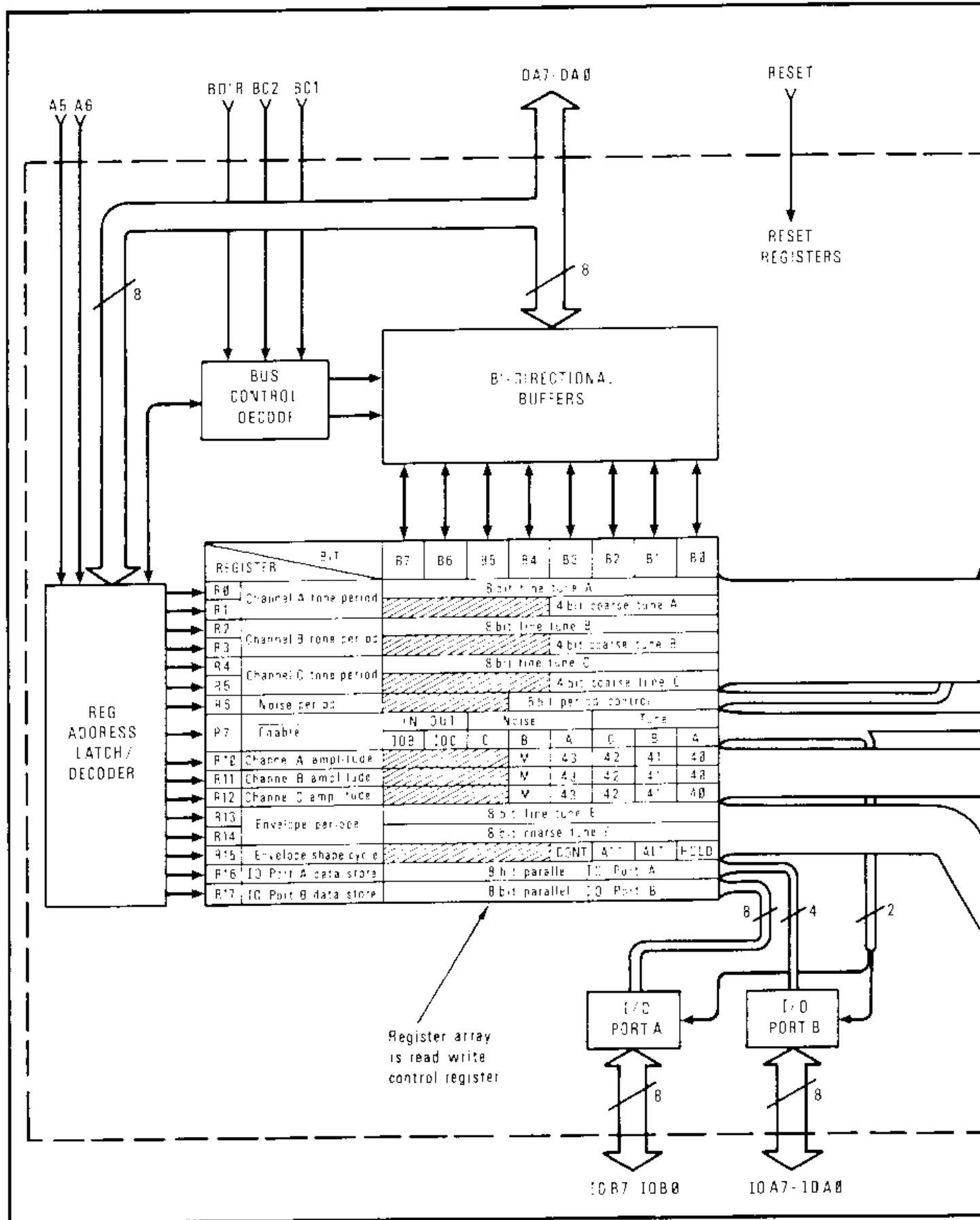
AY3 s'implante sur la carte mère de la façon suivante :

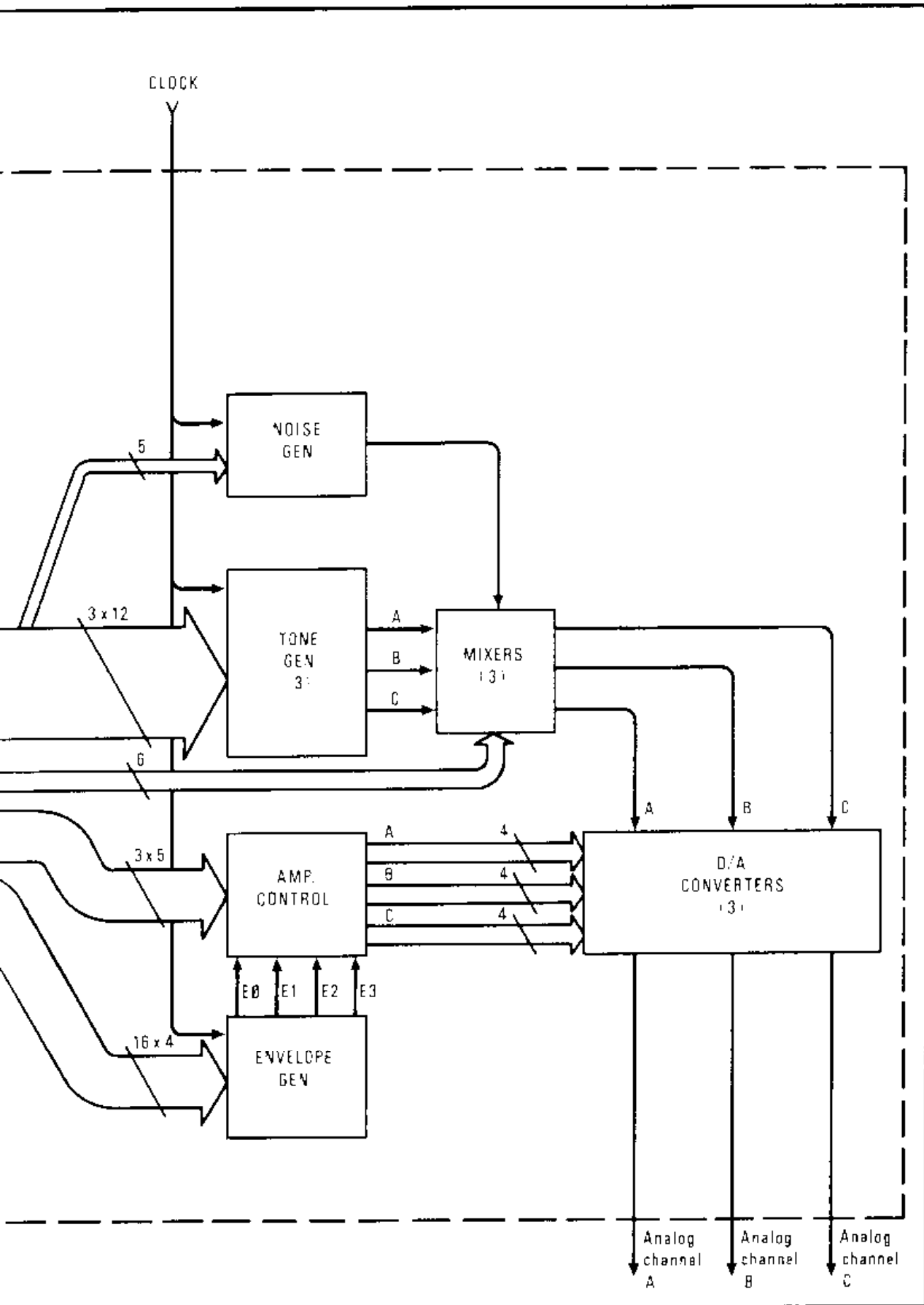


Nous voyons que :

- 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.

6/4

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

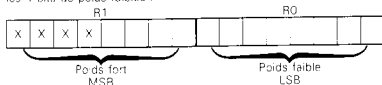
Voyons en détail l'utilisation de chaque registre.

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 2¹² (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 8 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 s'agit 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.

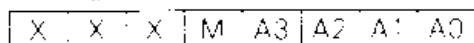
NA, NB et NC valident l'émission de bruit blanc sur les canaux A, B ou C. 0 indique que le bruit blanc est dévalidé et 1 qu'il est validé.

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 :



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 :

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

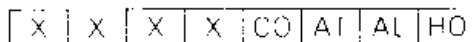
Elle est codée sur 16 bits. R11 est le registre de poids faible, R12 le registre de poids fort. Comme P peut varier de 0 à 2¹⁶ (de 0 à 65535), les fréquences minimale et maximale de l'enveloppe seront :

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

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





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 :



- CO = COntinue
- AT = ATack
- 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	

CC	AT	AL	HC	
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.) pour avoir plus de détails.

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^o Déclaration de l'IT et armement,

2^o 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 #BCFF), 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 » (KI 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 Y).

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.

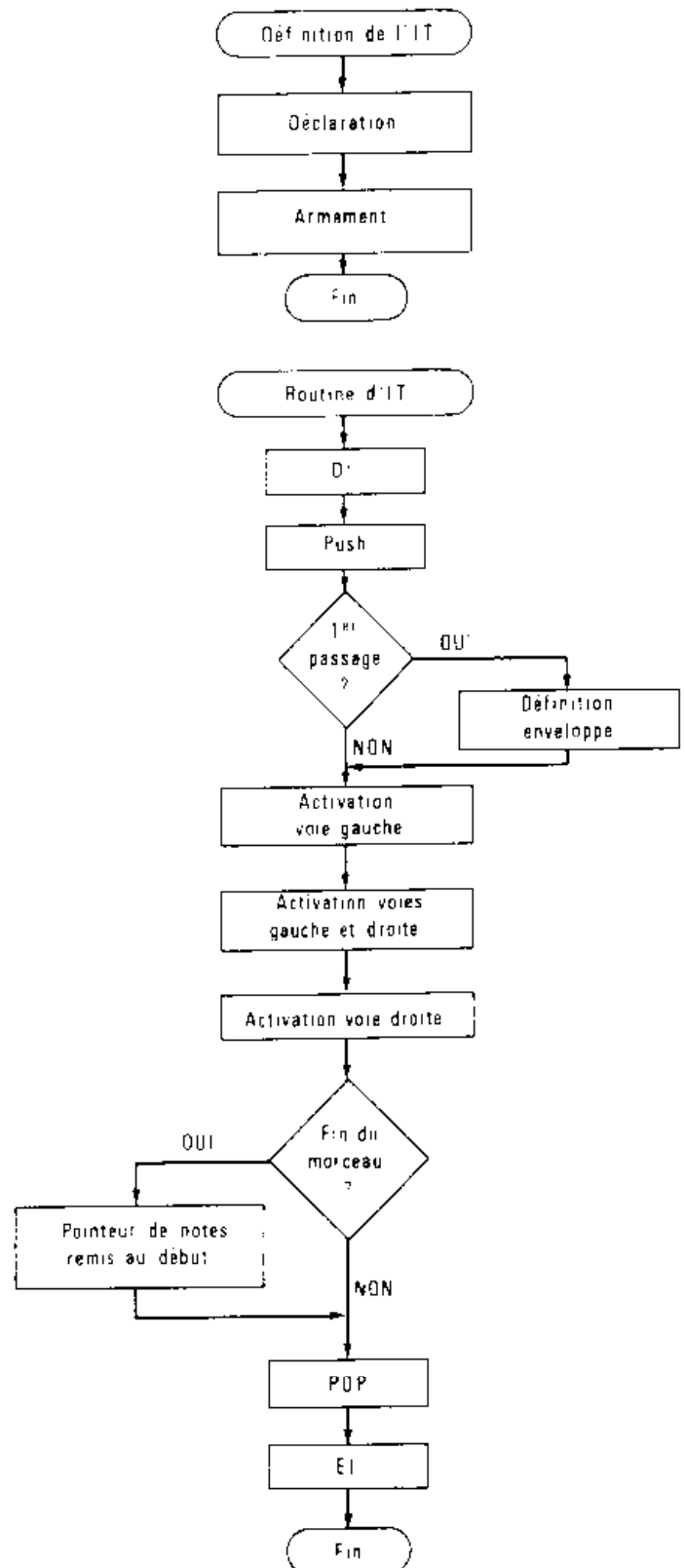
L'IT doit se terminer par la restitution des registres (POP IY, IX, HL, DE, BC, AF) dans l'ordre inverse où ils ont été sauvegardés, et par la revalidation des interruptions (EI), d'où l'organigramme suivant (Voir page 3)

Cet organigramme donne lieu au programme suivant :

```

1          ORG 9000H
2          LOAD 9000H
3 9000 032F90      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     ;▲ TICKLIST
18 9032 0681           LD   B,B1H         ;TYPE D'IT
19 9034 0E00           LD   C,0
20 9036 114990          LD   DE,RTSON     ;▲ touche act. si IT
21 9039 CDEFBC         CALL INITEV       ;INIT EVEN BLOC
22 903C 211F90          LD   HL,EVBL     ;▲ TICK BLOC
23 903F 110100          LD   DE,1
24 9042 010B00          LD   BC,11
25 9045 CDEFBC         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 lere IF
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         ;A Def enveloppe
42 9062 CDBCBC        CALL ENV            ;Init env val
43 9065 FD21D390      LD  IY,NOTES
44 9069 FD222C90      LD  (SAVMUS),IY    ;Note en cours
45                      SON1: EQU  $
46 906D FD2A2C90      LD  IY,(SAVMUS)    ;A 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,FI1
56 9087 FD7E01        LD  A,(IY+1)
57 908A FEFF          CP  OFFH
58 908C 2004          JR  NZ,FI1
59 908E FD21D390      LD  IY,NOTES
60                      FI1: 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 1B13        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 DD7704      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  $           ;Implementation 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 exécutable 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^o) modifier le programme d'IT en définissant une enveloppe de volume différente pour chaque longueur de note,

2^o) 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,%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,%F3,%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 ND$(N(VX),VX)=A$: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$:X=1:Y=16:GOTO 1910
2000 IF NT=N(A) THEN VX=A:GOTO 1980
2010 FOR I=N(A) TO NT STEP -1:ND$(I+1,A)=ND$(I,A):NEXT
2020 ND$(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$>"DD" AND L$>"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 'Mauvaise
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$="" THEN 2230
2220 GOSUB 2050:IF B=1 THEN SOUND 1,300,30:GOTO 2200 ELSE ND$(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
refus

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 '-----

2760 'Affichage du HELP d'écriture

2770 '

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 n-1
3090   FOR J=1 TO 3
3100     IF ND*(I,J)="" THEN MID=0:LSB=0:GOTO 3190   Rien à jouer
3110     L#=LEFT$(ND*(I,J),2)
3120     L#=(L#="DO")+(L#="RE")*3+(L#="MI")*5+(L#="FA")*6+(L#="SO")*8+(L#
#="LA")*10+(L#="SI")*12
3130     IF MID$(ND*(I,J),3,1)="#" THEN L=L+1
3140     IF MID$(ND*(I,J),3,1)="B" THEN L=L-1
3150     R#=RIGHT$(ND*(I,J),1):R=ASC(R#)-48
3160     L#=(R-1)*12+L
3170     NT=TI(L)   Note à 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 &P00D,1   Musique validée
3240 CALL &90FF
3250 RETURN
3260 -----
3270   Chargement d'un morceau
3280
3290 CLS:PRINT"Chargement d'un morceau"
3300 PRINT:INPUT"Nom du morceau ?";N$
3310 OPEN IN N$
3320 INPUT #9,NR   Nombre de notes
3330 FOR I=0 TO NR-1
3340   FOR J=1 TO 3
3350     INPUT #9,ND*(I,J)
3360   NEXT J
3370 NEXT I
3380 CLOSE IN
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          IRC = 9000H
2          IAD = 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    OBD1AH
11          ;
12          ;
13 9000 3E00          LD     A,0          ;Registre trou canal 0
14 9002 0E00          LD     A,0          ;mis a zero
15 9004 CD3480       CALL SOUND
16 9007 3E01          LD     A,1          ;Registre freq canal 0
17 9009 0E01          LD     A,1          ;mis a 01
18 900B CD74ED       CALL SOUND
19 900E 7E00          LD     A,0          ;Registre amplitude
20 9010 0E00          LD     A,0          ;Mueur de l'amplitude
21 9012 CD74ED       CALL SOUND
22 9015 3E07          LD     A,7          ;Registre validation
23 9017 0E08          LD     A,8          ;canal 0
24 9019 CD3480       CALL SOUND
25          ;
26          ;Boucle d'attente
27          ;
28 901C 21FFFF       LD     HL,0FFFFH
29          SOUND:      EQU    *
30 901E 2B          DCR     HL
31 9020 7C          LD     A,H
32 9021 35          OR     A
33 9022 20FFD       JS     NZ,SOUND
34          ;
35          ;Arrêt du son emis
36          ;
37 9024 7E08          LD     A,8          ;Registre amplitude

```

```

38 902E BEEF          LD      R10,          1015, a, 0000
39 902D CD34BD       CALL   SOUND
40 902E D9          RET
41                  END

```

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

Voici la version « Chargeur Bas ic » du programme précédent :

```

1000 REM BEEP avec le macro MD SOUND REGISTER
1010 REM =====
1020 REM Memorisation des codes op
1030 REM -----
1040 FOR I=39000 TO 3902B
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 Données correspondant aux codes op
1160 REM -----
1170 DATA 7E,0,E,0,00,74,BD,3E,1,E,1,00,34,BD,3E,9
1180 DATA 5,6,0E,34,BD,0F,7,E,8,00,74,1D,21,FF,FF,2E
1190 DATA 2C,0F,00,FB,3E,31,F,0,00,74,BD,09

```