Chapitre III.
Le moniteur

La carte mémoire

ADRESSES (hexadécimal)	Taille	Contenu
0000 - 1FFF	8K	Mémoire d'écran
2000 - 20FF	256	Page zéro du moniteur
2100 - 21FF	256	Page zéro du BASIC
2200 - 9FFF	32K	Mémoire utilisateur
A000 - A7BF	1,9K	Mémoire du floppy
A7C0 - A7C3		PIA 6821 système
A7C4 - A7CB		Libre...
A7CC - A7CF		PIA 6821 extension jeux
A7D0 - A7DF		Contrôleur mini-floppy
A7E0 - A7E3		PIA 6821 interface parallèle
A7E4 - A7E7		Gate-Array
A7E8 - A7FF		Libre...
A800 - AFFF	2K	Libre...
B000 - EFFF	16K	Cartouche ROM
C000 - EFFF	12K	BASIC (libre de B000 à BFFF)
F000 - FFFF	4K	Moniteur

La page zéro du moniteur

NOM de registre	Adresse
STATUS	$2019
RANG	$201B
COLN	$201C
TOPRAN	$201E
BOTRAN	$2020
FORME	$2029
COLOUR	$202B
COPCHR	$202F
CHDRAW	$2036
TEMPO	$2039 - $203A
DURÉE	$203B - $203C
TIMBRE	$203D
OCTAVE	$203E - $203F
PR.OPC	$2042
PR.STA	$2043
DK.OPC	$2048
DK.DRV	$2049
DK.TRK	$204A - $204B
DK.SEC	$204C
DK.NUM	$204D
DK.STA	$204E
DK.BUF	$204F - $2050
SWI1PT	$205E - $2060
TIMEPT	$2061 - $2063
IRQPT	$2064 - $2066
FIRQPT	$2067 - $2069
SIMUL	$206A - $206C
CHRPTR	$206D - $206F
USERAF	$2070 - $2072
GENPTR	$2073 - $2075

Fin de la zone de réinitialisation "départ à chaud"	
LATCLV	$2076
CRCODE	$2077
Fin de la zone de réinitialisation "départ à froid"	
DEFDST	$207F
DKFLG	$2080

Fonction

b3 : masque le buzzer.
N° ligne du curseur.
N° colonne du curseur.
N° de la 1ere ligne de la fenêtre.
N° de la dernière ligne de la fenêtre.
Registre couleur pour PLOT et DRAW (-16 à +15).
Registre couleur PBVR PBVR.

Contient le code ASCII du caractère pour PLOT ou DRAW.
Tempo général du générateur musical (1 à 255).
Longueur de la note (1 à 96).
Attaque de la note, du continu 0 au max. 255.
Octave (1 à 16).
Code de commande pour le contrôleur d'imprimante.
État courant du contrôleur d'imprimante.
Code de commande pour le contrôleur de disquette.
Numéro du drive.
Numéro du secteur.
Facteur de saut pour la numérotation des secteurs.
État courant du contrôleur de disquette.
Pointeur du buffer d'E/S de la disquette.
Pointeur d'interruption software SWI.
Pointeur et flag de la temporisation d'interruption utilisateur.
Pointeur et flag du l'interruption $\overline{\text{IRQ}}$.
Pointeur et flag de l'interruption $\overline{\text{FIRQ}}$.
Pointeur et flag de la table des points d'entrée moniteur.
Pointeur et flag de la table de décodage du clavier.
Pointeur et flag du générateur de caractères utilisateur.
Pointeur et flag du générateur de caractères.

Latence du clavier.
Mot de code pour copie graphique de l'écran.

Indicateur de simple ou double densité.
Indicateur de présence du contrôleur de disque.

Les sous-programmes

1 - Accès à un sous-programme du moniteur

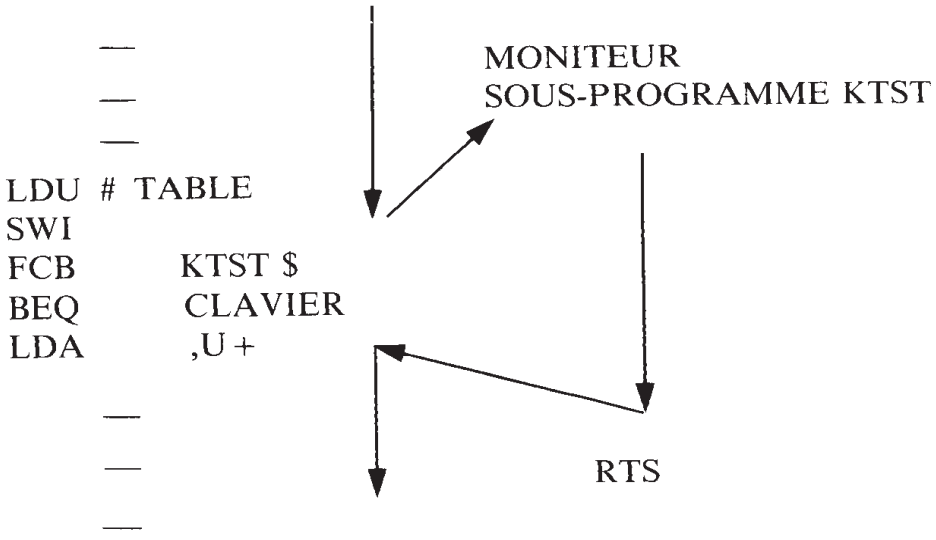
L'accès à un sous-programme du moniteur se fait grâce à une interruption programmée SWI suivie d'un code indiquant le sous-programme sélectionné.

Les codes peuvent appartenir à deux catégories, les CODE $ ou les CODE.

Si un code est du type CODE $, sa valeur numérique est comprise entre $ØØ et $3A. Dans ce cas, l'interruption SWI CODE $ est équivalente à une instruction JSR (Jump to Subroutine) de saut vers le sous-programme de nom CODE $. Ce sous-programme se terminant par une instruction de retour vers le programme principal, RTS, il renverra le 68Ø9 traiter l'instruction qui suit SWI CODE $.

Exemple : Dans le programme principal on souhaite appeler le sous-programme de test rapide du clavier. Ce sous-programme KEY-TEST (en abrégé KTST) a pour code KTST $ = $ØC. On aura donc la structure de fonctionnement suivante :

PROGRAMME PRINCIPAL



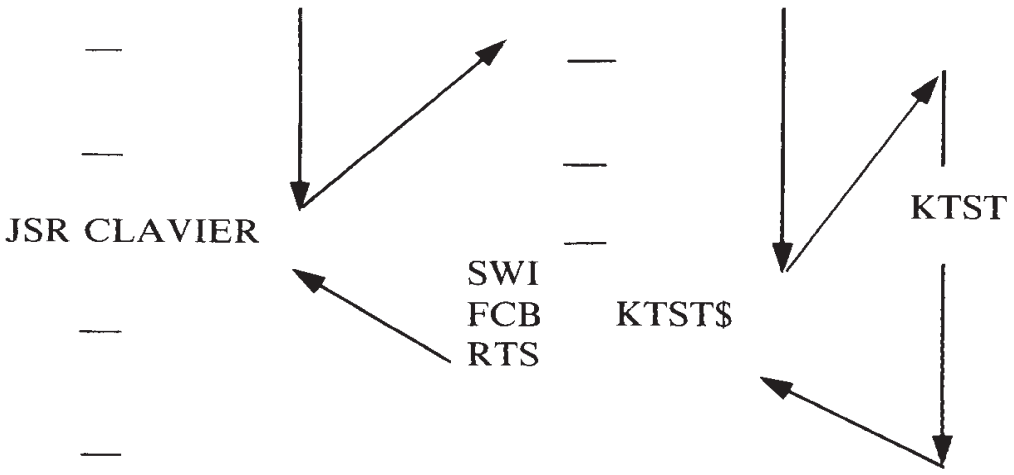
Si un code est du type CODE.,sa valeur numérique est comprise en $8Ø et $BA. La distinction par rapport au CODE $ est faite en portant le bit 7 à 1 dans la valeur numérique du CODE. alors que ce bit 7 est à Ø dans le CODE $. Les autres bits sont inchangés.

Exemple :

NOM du SOUS-PROGRAMME	CODE $	CODE.
KTST	KTST$ = $ØC	KTST. = $8C

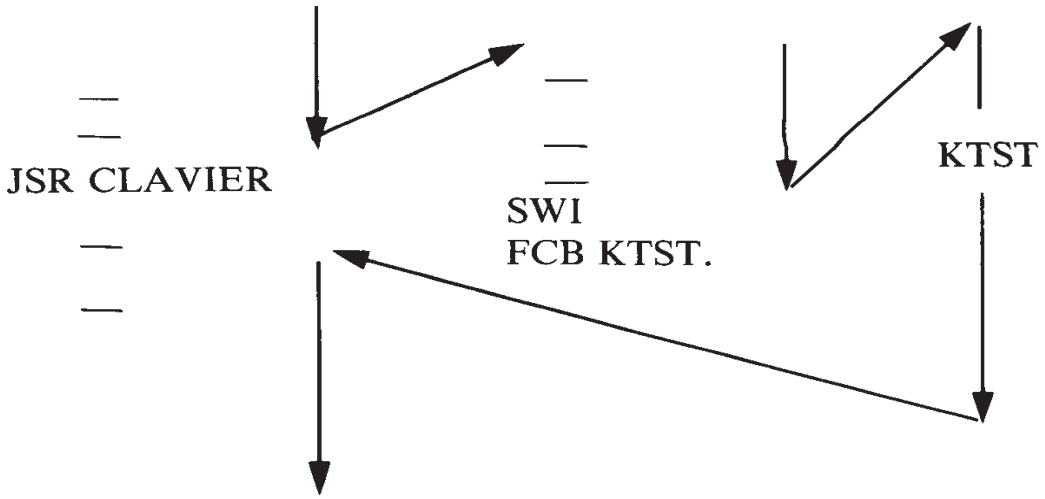
Dans un code de ce type, l'interruption SWI CODE. est équivalente à un JMP (Jump), saut sans attente d'instruction de retour de sous-programme (RTS). Le sous-programme du moniteur se terminant comme précédemment par une instruction de retour RTS renverra donc le 68Ø9 à l'instruction suivant le dernier appel de sous-programme.
Cette procédure simplifiera, dans certains cas l'écriture des programmes. C'est le cas de l'exemple 2 ci-après qui réalise la même fonction que l'exemple 1 ci-joint :

Exemple 1 :



Exemple 2 :

PROGRAMME PRINCIPAL SOUS-PROGRAMME MONITEUR



Que l'on utilise la procédure CODE $ ou CODE., l'instruction d'appel est dans les deux cas une interruption logicielle SWI. Donc, dans tous les cas tous les registres sont sauvegardés sur la pile. D'autre part, le registre de page directe DP est mis à la valeur $2Ø (page zéro du moniteur) et le registre d'index U est chargé avec l'adresse du PIA système, $A7CØ.
Au retour du sous-programme moniteur, tous les registres, sont "restorés" par dépilage à l'exception du registre code condition CCR et de ceux des registres contenant le résultat de l'opération réalisée par le sous-programme moniteur, c'est-à-dire B, X ou Y.

Il est vivement conseillé d'utiliser l'une des deux procédures ci-dessus pour appeler une routine du moniteur. En effet, celle-ci supposera dans son fonctionnement que tous les registres sont sur la pile, et que DP et U sont correctement initialisés.

Les valeurs numériques des CODE $ sont données dans le tableau ci-contre. On obtient celles de CODE. en leur ajoutant $8Ø.

NOM du SOUS-PROGRAMME	VALEUR NUMÉRIQUE DU CODE $	FONCTION
MENU	Ø	Retour au menu MO 5.
PUTC	2	Gestion de l'écran.
FRMØ	4	Mémoire couleur.
FRM1	6	Mémoire caractère.
BIIP	8	Gestion du buzzer.
GETC	$ØA	Lecture du clavier.
KTST	$ØC	Lecture rapide du clavier.
DRAW	$ØE	Tracé d'une ligne
PLOT	$1Ø	Gestion du point graphique ou de la case caractère.
CHPL	$12	Affichage d'un caractère.
GETP	$14	Lecture de la couleur d'un point de l'écran.
LPIN	$16	Test de l'interrupteur du crayon optique.
GETL	$18	Lecture des coordonnées visées par le crayon.
GETS	$1A	Lecture de l'écran.
JOYS	$1C	Gestion des manettes.
NOTE	$1E	Gestion des notes.
K7CO	$2Ø	Gestion des cassettes.
K7MO	$22	Gestion du moteur du LEP
PRCO	$24	Gestion du l'imprimante.
DKCO	$26	Gestion du lecteur de disquettes.

2 - Redéfinition des pointeurs système

Les pointeurs utilisés par le moniteur sont des registres à 3 octets situés dans la page zéro du moniteur de $2ØØØ à $2ØFF.

Ces registres doivent être manipulés avec précaution. Les deux premiers octets contiennent l'adresse soit d'une table, soit d'un sous-programme de gestion d'une interruption et le troisième octet contient un indicateur (FLAG).

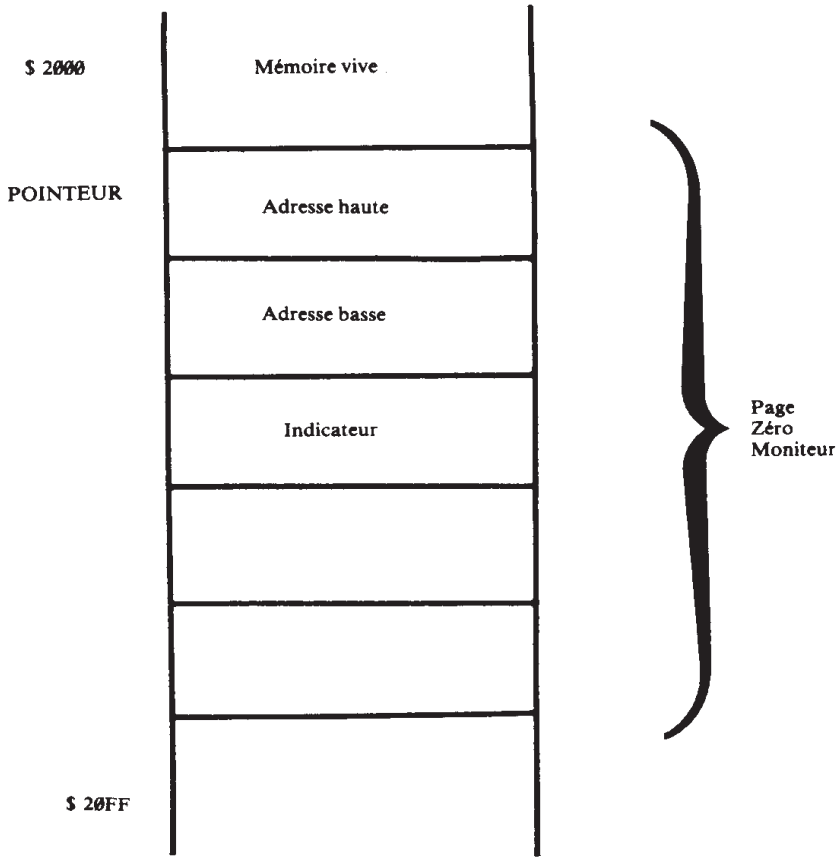


Figure 45
Occupation mémoire d'un pointeur

Cet indicateur doit contenir :
- La valeur Ø si le pointeur contient l'adresse standard.

- La valeur 1 si le pointeur contient une adresse définie par l'utilisateur.

C'est l'utilisateur qui doit mettre l'indicateur à 1 s'il change l'adresse du pointeur, et qui doit le remettre à Ø s'il restaure l'adresse standard.

Les pointeurs sont les suivants :
SWI1PT, TIMEPT, IRQPT, FIRQPT, SIMUL, CHRPTR, USERAF et GENPTR (voir leur adresse au chapitre 4.2).

3 - Gestion de tables

La mémoire du MO 5 est divisé en segments de 2K octets depuis l'origine $0000. Ces segments sont donc :

$0000 à $07FF Segment n° 1
$0800 à $0FFF Segment n° 2
$1000 à $17FF Segment n° 3
$1800 à $1FFF Segment n° 4
...etc.

Les tables utilisées dans certaines applications doivent toujours être contenues dans un seul et même segment de 2 Koctets.

Leur taille maximum est donc de 2 Koctets, et elles ne doivent en aucun cas être ''à cheval'' sur des segments différents.

Exemple : une table située de $154A à $182E n'est pas trop longue mais chevauche les segments n° 3 et n° 4, ce qui est interdit.

4 - Le fonctionnement des principaux sous-programmes

* PLOT	Entrées : X, Y (6809) FORME, CHDRAW, COLOUR (page 0)

En mode graphique, allume le point X, Y de couleur FORME si CHDRAW = $00.

En mode caractère, écrit en X, Y le caractère dont le code ASCII est dans CHDRAW avec les couleurs définies par COLOUR.

* DRAW	Entrées : X, Y (6809) PLOTX, PLOTY, CHDRAW, FORME, COLOUR (page 0).

En mode graphique, tire un trait de (PLOTX, PLOTY) à (X, Y) dans la couleur définie par FORME avec CHDRAW à 0.

En mode caractère, tire une ligne de caractères.

*** GETP** Entrées : X, A (68Ø9)
Sortie : B
Met dans B le code ASCII du caractère situé en (X, A).

*** PUTCH** Entrée : B
Affiche dans la position courante le caractère dont le code ASCII est dans B, puis avance d'une case.
Pour les GR$, il faut initialiser le pointeur USERAF.

Cette routine permet également l'initialisation de l'écran.

*** KTST** Sorties : CCR, B (68Ø9).
Si une touche du clavier est enfoncée, met le bit 7 du CCR à Ø.
B revient alors avec le code ASCII du caractère.

*** GETC** Sorties : CCR, B (68Ø9).
Cette fois B peut contenir le code BASIC en utilisation de cette fonction.

*** JOYST** Entrée : A (68Ø9).
Sortie : B, CCR (68Ø9).

Met dans B la position numérique de la manette dont le numéro est dans A.

Si le bouton de cette manette est enfoncé, le bit C du CCR est mis à Ø.

*** PRCO** Entrée : B (68Ø9) PROPC, GRCODE (page Ø).
Sortie : CCR (68Ø9) PR. STA (page Ø).

PR. OPC = 1 → Écriture d'un octet
= 2 → Copie graphique de l'écran
= 4 → Ouverture mode parallèle
= 16 → Fermeture

Envoie vers l'imprimante l'octet dont le code ASCII est dans B.

* NOTE Entrée : B (68Ø9) OCTAVE, DURÉE, TEMPO, TIMBRE (page Ø).

Joue la note dont la valeur est dans B en respectant les valeurs des registres OCTAVE,...
Les notes sont :

PAUSE = $3Ø
DO = $31
DO# = $32
⋮
SI = $3C

* LPIN Sortie : CCR (68Ø9).
C est mis à 1 si le bouton du crayon optique est enfoncé.

* GETL Sortie : X, Y (68Ø9).
Passe dans X, Y les coordonnées pointées par le crayon optique.

5 - Quelques applications

Pour terminer voici deux programmes en assembleur qui vous permettront de créer des BRUITS dans le convertisseur de l'extension jeux. Dans le deuxième programme, on module la fréquence en ajoutant au registre DURÉE à chaque passage.

En changeant le contenu de la table, on peut changer le bruit.

```
                ****************************************
                *                                      *
                *   PROGRAMME DE CREATION DE BRUITS    *
                *                                      *
                *        DANS LE CONVERTISSEUR         *
                *                                      *
                *                                      *
                *   ENTREES :                          *
                *                                      *
                *    DUREE = Duree de la temporisation *
                *    PAS   = Pas d'echantillonage      *
                *    TABLE = Debut de la table         *
                *    NBR   = Nombre d'echantillons     *
                *                                      *
                *                                      *
                *   SORTIES :                          *
                *                                      *
                *    FINTA = Fin de la table           *
                *                                      *
                *                                      *
                *   PORTB2 en $A7CD                    *
                *                                      *
                *                                      *
                *                                      *
                ****************************************


          A7CD PORTB2 EQU    $A7CD
          A7CD DDRB2  EQU    $A7CD
          A7CF CRB2   EQU    $A7CF

7000                 ORG    $7000

                *
                *       Initialisation du PIA
                *

7000 5F
7001 F7   A7CF       CLR    CRB2     Mise a 0 du CRB2
7004 CC   3F04       LDD    #$3F04
7007 B7   A7CD       STA    DDRB2    B0 a B5 en sorties
700A F7   A7CF       STB    CRB2     Acces au PORTB
```

```
                    *
                    *       Creation du bruit
                    *

700D 34   7E           PSHS   U.Y,X,DP,B,A   On sauve
700F 1A   10           ORCC   #$10     Ne pas interrompre

7011 8E   7033 BOUC0   LDX    #TABLE   On pointe
7014 F6   7031         LDB    PAS      Pas de lecture


7017 A6   84   BOUC1   LDA    ,X       Lire une valeur
7019 30   85           LEAX   B,X      Pointer la suivant
701B B7   A7CD         STA    PORTB2   Envoyer en sortie
701E 10BE 702F         LDY    DUREE    Debut de tempo.

7022 31   3F   BOUC2   LEAY   -1,Y
7024 26   FC           BNE    BOUC2    Fin de tempo
7026 8C   704B         CMPX   #FINTA+1  Fin de Table?
7029 2D   EC           BLT    BOUC1    Non, alors boucler
702B 20   E4           BRA    BOUC0    Non, alors boucler

702D 35   FE           PULS   A,B,DP,X,Y,U,PC

                    *
                    *       Initialisation de la table
                    *

702F       0070 DUREE   FDB    $0070
7031       01   PAS     FCB    1
7032       18   NBR     FCB    24

7033       00   TABLE   FCB    0
7034       05           FCB    5
7035       0A           FCB    10
7036       0F           FCB    15
7037       15           FCB    21
7038       1A           FCB    26
7039       1F           FCB    31
703A       24           FCB    36
703B       2A           FCB    42
703C       2F           FCB    47
703D       34           FCB    52
703E       39           FCB    57
703F       3F           FCB    63
```

```
7040        39              FCB     57
7041        34              FCB     52
7042        2F              FCB     47
7043        2A              FCB     42
7044        24              FCB     36
7045        1F              FCB     31
7046        1A              FCB     26
7047        15              FCB     21
7048        0F              FCB     15
7049        0A              FCB     10
704A        05      FINTA   FCB     5

            0000            END

00000 Total Errors


BOUC0   7011
BOUC1   7017
BOUC2   7022
CRB2    A7CF
DDRB2   A7CD
DUREE   702F
FINTA   704A
NBR     7032
PAS     7031
PORTB2  A7CD
TABLE   7033
```

```
*****************************************
*                                       *
*   PROGRAMME DE CREATION DE BRUITS     *
*                                       *
*        DANS LE CONVERTISSEUR          *
*                                       *
*                                       *
*   ENTREES :                           *
*                                       *
*    DUREE = Duree de la temporisation  *
*    PAS   = Pas d'echantillonage       *
*    TABLE = Debut de la table          *
*    NBR   = Nombre d'echantillons      *
*                                       *
*                                       *
*   SORTIES :                           *
*                                       *
*    FINTA = Fin de la table            *
*                                       *
*                                       *
*   PORTB2 en $A7CD                     *
*                                       *
*                                       *
*****************************************

          A7CD PORTB2 EQU    $A7CD
          A7CD DDRB2  EQU    $A7CD
          A7CF CRB2   EQU    $A7CF

7000                  ORG    $7000

                      *
                      *      Initialisation du PIA
                      *

7000 5F
7001 F7   A7CF        CLR    CRB2     Mise a 0 du CRB2
7004 CC   3F04        LDD    #$3F04
7007 B7   A7CD        STA    DDRB2    B0 a B5 en sorties
700A F7   A7CF        STB    CRB2     Acces au PORTB

                      *
                      *      Creation du bruit
                      *
```

```
700D 34   7E          PSHS   U,Y,X,DP,B,A   On sauve
700F 1A   10          ORCC   #$10     Ne pas interrompre

7011 8E   7040 BOUC0  LDX    #TABLE   On pointe
7014 F6   703E        LDB    PAS      Pas de lecture
7017 A6   84   BOUC1  LDA    ,X       Lire une valeur
7019 30   85          LEAX   B,X      Pointer la suivant
701B B7   A7CD        STA    PORTB2   Envoyer en sortie
701E 10BE 703C        LDY    DUREE    Debut de tempo.

7022 31   3F   BOUC2  LEAY   -1,Y
7024 26   FC         BNE    BOUC2    Fin de tempo
7026 8C   7058       CMPX   #FINTA+1  Fin de Table?
7029 2D   EC         BLT    BOUC1    Non, alors boucler
702B 7A   703D       DEC    DUREE+1
702E 26   E1         BNE    BOUC0    Boucler
7030 108E 0100       LDY    #$0100
7034 10BF 703C       STY    DUREE
7038 20   D7         BRA    BOUC0

703A 35   FE         PULS   A,B,DP,X,Y,U,PC

                *
                *    Initialisation de la table
                *

703C      0100 DUREE  FDB    $0100
703E      01   PAS    FCB    1
703F      18   NBR    FCB    24

7040      00   TABLE  FCB    0
7041      05         FCB    5
7042      0A         FCB    10
7043      0F         FCB    15
7044      15         FCB    21
7045      1A         FCB    26
7046      1F         FCB    31
7047      24         FCB    36
7048      2A         FCB    42
7049      2F         FCB    47
704A      34         FCB    52
704B      39         FCB    57
704C      3F         FCB    63
704D      39         FCB    57
704E      34         FCB    52
704F      2F         FCB    47
7050      2A         FCB    42
```

```
7051        24              FCB     36
7052        1F              FCB     31
7053        1A              FCB     26
7054        15              FCB     21
7055        0F              FCB     15
7056        0A              FCB     10
7057        05      FINTA   FCB     5

            0000            END

00000 Total Errors
```