9/8.18

Horloge sous interruptions

Une des utilisations les plus spectaculaires d'un ordinateur consiste à utiliser les interruptions pour réaliser un traitement multitâche. Les tâches exécutées simultanément sont cadencées par l'horloge système.

Dans ce paragraphe, nous allons étudier une horloge 24 heures interruptible, manipulable par l'utilisateur au travers d'une RSX. Les fonctions accessibles par l'intermédiaire de la RSX seront les suivantes :

— mise à l'heure de l'horloge ;

— affichage permanent ;

— arrêt de l'affichage.

COMMENT UTILISER LE PROGRAMME

Le programme mettant en œuvre une RSX est forcément écrit en Assembleur. Si vous désirez l'utiliser sous sa forme Assembleur, entrez le listing de la page suivante.

Compilez le programme et installez la RSX en tapant sous Basic :

```
CALL &9024
```

L'horloge peut être mise à l'heure en spécifiant les champs Heures (entre 0 et 23), Minutes (entre 0 et 59) et Secondes (entre 0 et 59) directement après le libellé de la fonction :

```
|HOR,<Heures>, <Minutes>, <Secondes>
```

Par exemple :

```
|HOR,11,15,30
```

initialise l'horloge à 11 heures, 15 minutes, 30 secondes.

Pour afficher l'horloge en permanence sur l'écran, quel que soit le mode d'affichage courant, tapez simplement :

```
|HOR,1
```

Pour arrêter l'affichage de l'horloge, tapez simplement :

```
|HOR,0
```

```
1                               ORG  9000H
2                               LOAD 9000H
3                   ;
4                   ;-----------------------------------
5                   ; RSX HORLOGE SOUS INTERRUPTIONS
6                   ;-----------------------------------
7                   ;
8  9000 C32490                  JP   DEFRSX              ;Definition RSX
9                   ;
10                  ;-----------------------------------
11                  ; Declaration des constantes et
12                  ; variables du programme
13                  ;-----------------------------------
14                  ;
15 9003 00000000    EVBL:        DB   0,0,0,0
16 9007 00000000                 DB   0,0,0,0
17 900B 00000000                 DB   0,0,0,0,0
17 900F 00
18                  HE:          DS   1                   ;Champ Heures
19                  MI:          DS   1                   ;Champ Minutes
20                  SE:          DS   1                   ;Champ Secondes
21                  SAVA:        DS   1                   ;Sauvegarde de A
22                  SAVCUR:      DS   2                   ;Sauv coord curseur
23                  BUF:         DS   4                   ;Zone RAM pour LOG EXT
24 901A 1F90        PTRTAB:      DW   TABLE               ;Pointeur TABLE
25 901C C33190                   JP   TRAITE              ;Traitement
26 901F 484F        TABLE:       DB   "HO"
27 9021 D2                       DB   "R"+80H
28 9022 00                       DB   0                   ;Fin de table
29                  VALID:       DS   1                   ;Valid/Devalid affich
30
```

```
31                      INITEV:     EQU  0BCEFH          ;INIT EVEN BLOC
32                      ADDEVE:     EQU  0BCE9H          ;ADD EVEN BLOC
33                      DELEVE:     EQU  0BCECH          ;DEL EVEN BLOC
34                      TIMESET:    EQU  0BD10H          ;KL TIME SET
35                      TXTOUT:     EQU  0BB5AH          ;TXT OUTPUT
36                      SETCUR:     EQU  0BB75H          ;TXT SET CURSOR
37                      GETCUR:     EQU  0BB78H          ;TXT GET CURSOR
38                      CUREN:      EQU  0BB7BH          ;TXT CUR ENABLE
39                      CURDIS:     EQU  0BB7EH          ;TXT CUR DISABL
40                      LOGEXT:     EQU  0BCD1H          ;KL LOG EXT
41                      ;
42                      ;-------------------------------
43                      ; Definition de la RSX
44                      ;-------------------------------
45                      ;
46                      DEFRSX:     EQU  $               ;Point d'entree
47 9024 011A90                      LD   BC,PTRTAB       ;Ptr table definition
48 9027 211690                      LD   HL,BUF          ;Buffer pour LOG EXT
49 902A CDD1BC                      CALL LOGEXT          ;Definition de la RSX
50 902D CD5590                      CALL INSTALL         ;Installation de l'IT
51 9030 C9                          RET
52                      ;
53                      ;-------------------------------
54                      ; Traitement de la RSX
55                      ;-------------------------------
56                      ;
57                      TRAITE:     EQU  $
58 9031 FE01                        CP   1
59 9033 CA3B90                      JP   Z,UNPARAM       ;Un parametre
```

```
60 9036 FE03                    CP    3
61 9038 2808                    JR    Z,TROIPAR           ;Trois parametres
62 903A C9                      RET
63                  ;
64                  ;--------------------------------
65                  ; Validation ou devalidation de
66                  ; l'horloge
67                  ;--------------------------------
68                  ;
69                  UNPARAM:    EQU   $
70 903B DD7E00                  LD    A,(IX+0)
71 903E 322390                  LD    (VALID),A           ;Validation affich
72 9041 C9                      RET
73                  ;
74                  ;--------------------------------
75                  ;Initialisation de l'horloge
76                  ;--------------------------------
77                  ;
78                  TROIPAR:    EQU   $
79 9042 DD7E00                  LD    A,(IX+0)
80 9045 321290                  LD    (SE),A              ;Init secondes
81 9048 DD7E02                  LD    A,(IX+2)
82 904B 321190                  LD    (MI),A              ;Init minutes
83 904E DD7E04                  LD    A,(IX+4)
84 9051 321090                  LD    (HE),A              ;Init heures
85 9054 C9                      RET
86                  ;
87                  ;
88                  ;--------------------------------
89                  ; Installation de l'IT
90                  ;------- ------------------------
```

```
 91
 92                 INSTALL:    EQU  $
 93 9055 210990                 LD   HL,EVBL+6
 94 9058 0681                   LD   B,81H
 95 905A 0E00                   LD   C,0
 96 905C 116F90                 LD   DE,TRAIT          ;@ Traitement
 97 905F CDEFBC                 CALL INITEV            ;INIT EVEN BLOC
 98 9062 210390                 LD   HL,EVBL
 99 9065 110100                 LD   DE,1
100 9068 013200                 LD   BC,50
101 906B CDE9BC                 CALL ADDEVE            ;ADD EVEN BLOC
102 906E C9                     RET
103                ;
104                ;--------------------------------
105                ; Routine de traitement
106                ;--------------------------------
107                ;
108                TRAIT:      EQU  $
109 906F F3                     DI
110 9070 F5                     PUSH AF
111 9071 C5                     PUSH BC
112 9072 D5                     PUSH DE
113 9073 E5                     PUSH HL
114 9074 DDE5                   PUSH IX
115 9076 FDE5                   PUSH IY
116
117 9078 3A1290                 LD   A,(SE)
118 907B 3C                     INC  A
119 907C 321290                 LD   (SE),A
```

```
120 907F FE3C                CP   60
121 9081 2802                JR   Z,INCMIN          ;Incremente minute
122 9083 1826                JR   FININC            ;Incrementation terminee
123              INCMIN:     EQU  $
124 9085 AF                  XOR  A
125 9086 321290              LD   (SE),A
126 9089 3A1190              LD   A,(MI)
127 908C 3C                  INC  A
128 908D 321190              LD   (MI),A
129 9090 FE3C                CP   60
130 9092 2802                JR   Z,INCHR           ;Incremente heure
131 9094 1815                JR   FININC            ;Incrementation terminee
132              INCHR:      EQU  $
133 9096 AF                  XOR  A
134 9097 321190              LD   (MI),A
135 909A 3A1090              LD   A,(HE)
136 909D 3C                  INC  A
137 909E 321090              LD   (HE),A
138 90A1 FE18                CP   24
139 90A3 2802                JR   Z,RAZHR           ;RAZ champ heure
140 90A5 1804                JR   FININC            ;Incrementation terminee
141              RAZHR:      EQU  $
142 90A7 AF                  XOR  A
143 90A8 321090              LD   (HE),A
144              ;
145              ;-------------------------
146              ; Affichage de l'heure
147              ;-------------------------
148              ;
149              FININC:     EQU  $
150 90AB 3A2390              LD   A,(VALID)
```

```
151 90AE B7                 OR   A
152 90AF CA3891             JP   Z,FINIT          ;Pas d'affichage
153 90B2 CD78BB             CALL GETCUR           ;Position courante curs
154 90B5 221490             LD   (SAVCUR),HL
155 90B8 CD7EBB             CALL CURDIS           ;TXT CUR DISABLE
156 90BB 2601               LD   H,1              ;Colonne curseur
157 90BD 2E01               LD   L,1              ;Ligne curseur
158 90BF CD75BB             CALL SETCUR           ;Position curseur
159                 ;--------
160                 ; Heures
161                 ;--------
162 90C2 AF                 XOR  A
163 90C3 4F                 LD   C,A              ;Compteur
164 90C4 060A               LD   B,10             ;Dizaines
165 90C6 3A1090             LD   A,(HE)
166                 BOU1:   EQU  $
167 90C9 B8                 CP   B
168 90CA DAD190             JP   C,AFF1           ;Affichage
169 90CD 0C                 INC  C
170 90CE 90                 SUB  B
171 90CF 18F8               JR   BOU1             ;On recommence
172                 AFF1:   EQU  $
173 90D1 321390             LD   (SAVA),A
174 90D4 3E30               LD   A,48
175 90D6 81                 ADD  A,C
176 90D7 CD5ABB             CALL TXTOUT
177 90DA 3A1390             LD   A,(SAVA)
178 90DD 0630               LD   B,48
179 90DF 80                 ADD  A,B
```

```
180 90E0 CD5ABB              CALL  TXTOUT
181 90E3 3E3A                LD    A,58              ;Separateur
182 90E5 CD5ABB              CALL  TXTOUT
183              ;---------
184              ; Minutes
185              ;---------
186 90E8 AF                  XOR   A
187 90E9 4F                  LD    C,A               ;Compteur
188 90EA 060A                LD    B,10              ;Dizaines
189 90EC 3A1190              LD    A,(MI)
190              BOU2:       EQU   $
191 90EF B8                  CP    B
192 90F0 DAF790              JP    C,AFF2            ;Affichage
193 90F3 0C                  INC   C
194 90F4 90                  SUB   B
195 90F5 18F8                JR    BOU2              ;On recommence
196              AFF2:       EQU   $
197 90F7 321390              LD    (SAVA),A
198 90FA 3E30                LD    A,48
199 90FC 81                  ADD   A,C
200 90FD CD5ABB              CALL  TXTOUT
201 9100 3A1390              LD    A,(SAVA)
202 9103 0630                LD    B,48
203 9105 80                  ADD   A,B
204 9106 CD5ABB              CALL  TXTOUT
205 9109 3E3A                LD    A,58              ;Separateur
206 910B CD5ABB              CALL  TXTOUT
207              ;----------
208              ; Secondes
209              ;----------
210 910E AF                  XOR   A
```

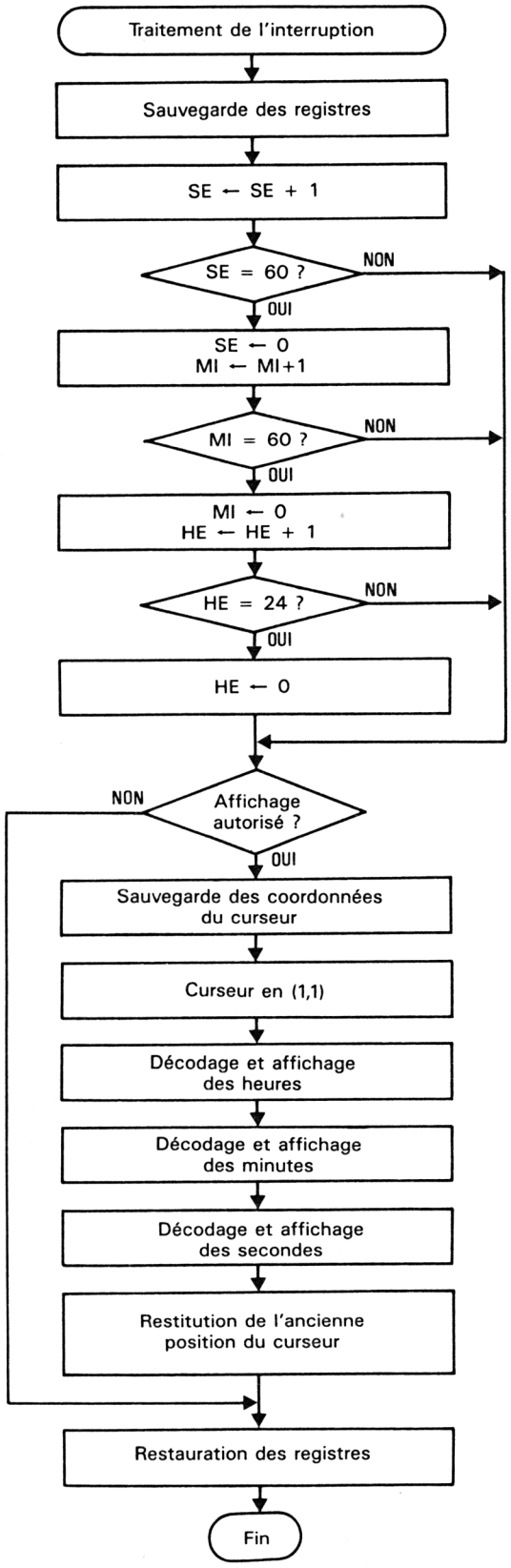
```
211 910F 4F                 LD   C,A              ;Compteur
212 9110 060A               LD   B,10             ;Dizaines
213 9112 3A1290             LD   A,(SE)
214              BOU3:      EQU  $
215 9115 B8                 CP   B
216 9116 DA1D91             JP   C,AFF3           ;Affichage
217 9119 0C                 INC  C
218 911A 90                 SUB  B
219 911B 18F8               JR   BOU3             ;On recommence
220              AFF3:      EQU  $
221 911D 321390             LD   (SAVA),A
222 9120 3E30               LD   A,48
223 9122 81                 ADD  A,C
224 9123 CD5ABB             CALL TXTOUT
225 9126 3A1390             LD   A,(SAVA)
226 9129 0630               LD   B,48
227 912B 80                 ADD  A,B
228 912C CD5ABB             CALL TXTOUT
229
230 912F 2A1490             LD   HL,(SAVCUR)
231 9132 CD75BB             CALL SETCUR           ;Restitution curseur
232 9135 CD7BBB             CALL CURENA           ;TXT CUR ENABLE
233
234              FINIT:     EQU  $
235 9138 FDE1               POP  IY
236 913A DDE1               POP  IX
237 913C E1                 POP  HL
238 913D D1                 POP  DE
239 913E C1                 POP  BC
```

```
240 913F F1                     POP   AF
241 9140 FB                     EI
242 9141 C9                     RET
243                 ;
244                 ;---------------------------------
245                 ; Devalidation de l'IT
246                 ;---------------------------------
247                 ;
248                 DEVALID:    EQU   $
249 9142 210390                 LD    HL,EVBL
250 9145 CDECBC                 CALL  DELEVE              ;DEL EVEN BLOC
251 9148 C9                     RET
252                             END
```

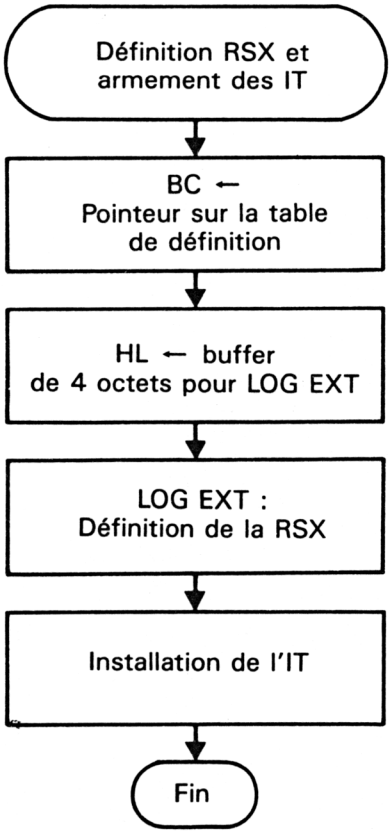
ADDEVE	BCE9	AFF1	90D1	AFF2	90F7	AFF3	911D
BUF	9016	BOU1	90C9	BOU2	90EF	BOU3	9115
CUREN	BB7B	CURDIS	BB7E	DELEVE	BCEC	DEFRSX	9024
DEVALID	9142	EVBL	9003	FININC	90AB	FINIT	9138
GETCUR	BB78	HE	9010	INITEV	BCEF	INSTALL	9055
INCMIN	9085	INCHR	9096	LOGEXT	BCD1	MI	9011
PTRTAB	901A	RAZHR	90A7	SE	9012	SAVA	9013
SAVCUR	9014	SETCUR	BB75	TABLE	901F	TIMESET	BD10
TXTOUT	BB5A	TRAITE	9031	TROIPAR	9042	TRAIT	906F
UNPARAM	903B	VALID	9023				

LE PROGRAMME EN DÉTAIL

La logique du programme obéit à l'ordinogramme suivant :



La partie dédiée à l'installation de la RSX située à l'étiquette DEFRSX obéit à la logique de l'ordinogramme suivant :



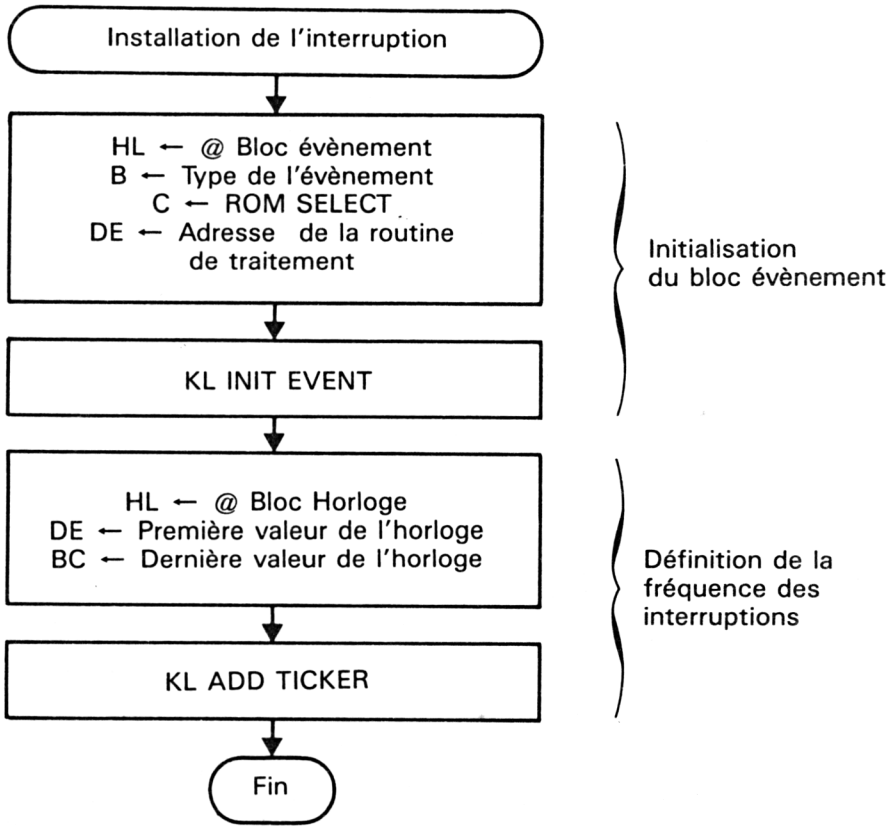
Le registre **BC** est initialisé à l'adresse de la table de définition **PTRTAB** et le registre HL pointe sur un buffer utilisé par **LOGEXT**. Ces deux registres étant initialisés, la macro de définition de la RSX est activée :

```
DEFRSX:  EQU    $
         LD     BC,PTRTAB      ; Ptr table définition
         LD     HL,BUF         ; Buffer pour LOG EXT
         CALL   LOGEXT         ; Définition de la RSX
```

Le programme d'horloge fonctionnant sous interruptions, il est nécessaire de définir le type d'interruption qui le régit avant de rendre le contrôle au programme appelant. C'est le but de la routine située à l'adresse **INSTALL** :

```
         CALL   INSTALL        ; Installation de l'IT
```

La définition du type d'interruption utilisé par le programme se fait en deux phases, comme le montre l'ordinogramme suivant :



La macro **KL INIT EVENT** (Partie 4, Chap. 2.7 page 57) permet d'initialiser le bloc événement. En d'autres termes, elle définit le type de l'événement et l'adresse de la routine de traitement associée :

```
INSTALL:  EQU    $
          LD     HL,EVBL+6     ; Adresse du bloc événement
          LD     B,81H         ; Type de l'événement
          LD     C,0           ; ROM SELECT
          LD     DE,TRAIT      ; Adresse de la routine de trai-
                                 tement
          CALL   INITEV        ; INIT EVENT BLOCK
```

La macro **KL ADD TICKER** (Partie 4, Chap 2.7 page 56) est ensuite activée pour déterminer la fréquence des interruptions (en 1/50 seconde) :

```
          LD     HL,EVBL       ; Adresse du bloc événement
          LD     DE,1          ; Première valeur de l'horloge
          LD     BC,50         ; Dernière valeur de l'horloge
          CALL   ADDEVE        ; ADD EVENT BLOCK
```

La différence entre les dernière et première valeurs d'horloge représente un délai en 1/50 de seconde. Dans notre cas, le délai est de 50 – 1 + 1 = 50/50 de seconde, soit une seconde. La routine de traitement de l'interruption sera donc exécutée toutes les secondes.

Ces deux macros (**INIT EVENT BLOCK** et **ADD EVENT BLOCK**) exécutées, la routine de traitement située à l'adresse **TRAIT** est activée toutes les secondes.

L'utilisateur du programme peut :

— mettre l'horloge à l'heure en tapant :

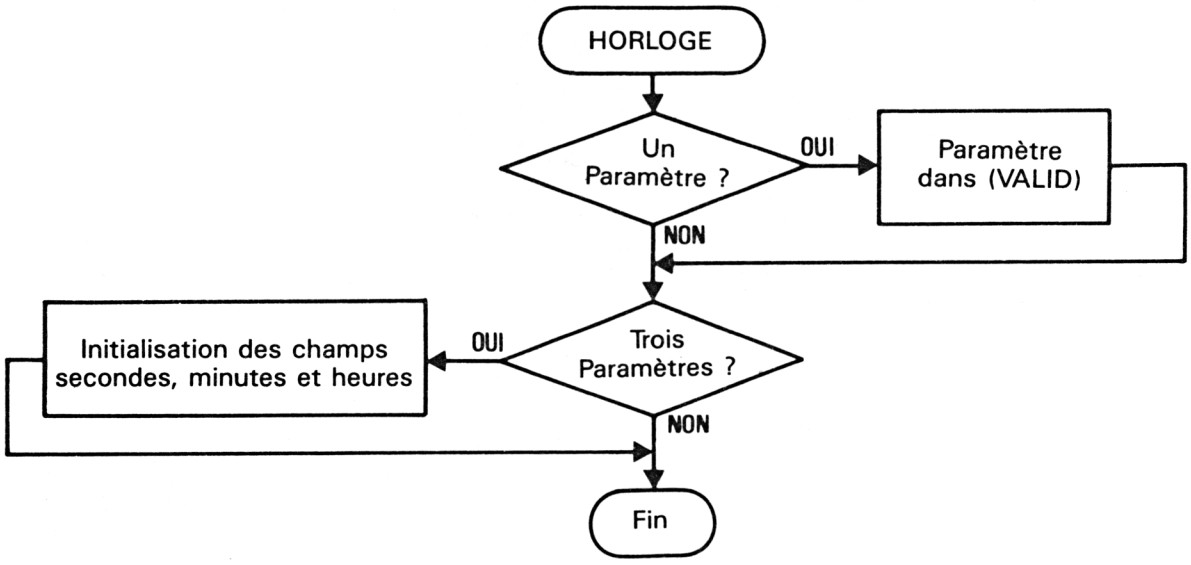
HOR,<Heures>, <Minutes>, <Secondes>

— **autoriser l'affichage de l'horloge en haut et à gauche de l'écran en tapant :**

|HOR,1

— interdire l'affichage de l'horloge en tapant : **|HOR,0**

Ces diverses commandes sont interprétées par le court programme situé à l'étiquette **TRAITE**. Ce programme est activé par l'interpréteur de commandes Basic à chaque fois qu'il rencontre l'instruction **|HOR**. La logique de ce programme apparaît dans l'ordinogramme suivant :



Le nombre de paramètres passés à la RSX se trouve dans le registre A en entrée de la RSX.

Si A vaut 1, le programme situé à l'étiquette **UNPARAM** est exécuté. Si A vaut 3, le programme situé à l'étiquette **TROIPAR** est exécuté :

```
TRAITE:   EQU   $
          CP    1
```

```
        JP      Z,UNPARAM    ; Un paramètre
        CP      3
        JR      Z,TROIPAR    ; Trois paramètres
```

Les programmes situés en **UNPARAM** et en **TROIPAR** sont très simples. Lorsqu'un paramètre est identifié, **UNPARAM** se contente de stocker la valeur de ce paramètre dans la variable **VALID** :

```
UNPARAM:EQU     $
        LD      A,(IX+0)
        LD      (VALID),A    ; Validation de l'affichage
```

Lorsque trois paramètres sont identifiés, l'utilisateur désire modifier l'heure de l'horloge. Dans ce cas :

— le premier paramètre contient le champ des heures ;

— le second paramètre contient le champ des minutes ;

— le troisième paramètre contient le champ des secondes.

Ces informations sont respectivement recopiées dans les variables **HE**, **MI** et **SE** :

```
TROIPAR: EQU    $
        LD      A,(IX+0)
        LD      (SE),A       ; Initialisation des secondes
        LD      A,(IX+2)
        LD      (MI),A       ; Initialisation des minutes
        LD      A,(IX+4)
        LD      (HE),A       ; Initialisation des heures
```

Peut-être vous demandez-vous comment l'horloge peut fonctionner toute seule ? Dans ce cas, vous oubliez que le programme situé à l'étiquette **TRAIT** est exécuté sous interruptions toutes les secondes. Etudions son fonctionnement.

La première action effectuée par le programme consiste à dévalider les interruptions et à sauvegarder le contexte lors de son appel :

```
TRAIT:  EQU     $
        DI
        PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    IX
        PUSH    IY
```

Le contexte étant sauvegardé, le programme lit la valeur qui se trouve dans la variable **SE** (secondes), l'incrémente, la sauvegarde et la compare à la constante 60 :

```
        LD      A,(SE)
        INC     A
        LD      (SE),A
        CP      60
```

Si (SE) est égal à 60, le nombre de secondes doit être mis à zéro, et le nombre de minutes incrémenté d'un :

```
INCMIN : EQU    $
         XOR    A
         LD     (SE),A      ; Champ <Secondes> mis à zéro
         LD     A,(MI)
         INC    A
         LD     (MI),A      ; Incrémentation du champ <Minutes>
```

Dans le cas contraire, le programme se débranche vers l'étiquette **FININC** dans le but éventuel (si **(VALID)** vaut 1) d'afficher l'heure sur l'écran.

Si, après incrémentation, le champ **<Minutes>** est égal à 60, il doit être mis à zéro, et le champ **<Heures>** doit être incrémenté :

```
         CP     60
         JR     Z,INCHR
         ...
INCHR:   EQU    $
         XOR    A
         LD     (MI),A      ; Champ <Minutes> mis à 0
         LD     A,(HE)
         INC    A
         LD     (HE),A      ; Incrémentation du champ
                            <Heures>
```

Dans le cas contraire, le programme se débranche vers l'étiquette **FININC** dans le but éventuel (si **(VALID)** vaut 1) d'afficher l'heure sur l'écran.

Si après incrémentation, le champ **<Heures>** est égal à 24, il doit être mis à zéro :

```
         CP     24
         JR     Z,RAZHR
         ...
RAZHR:   EQU    $
         XOR    A
         LD     (HE),A      ; Champ <Heures> mis à 0
```

L'affichage de l'heure sur l'écran est subordonné à la valeur contenue dans la variable **VALID**. Si cette variable est nulle, l'affichage n'a pas lieu :

```
FININC:  EQU   $
         LD    A,(VALID)
         OR    A
         JP    Z,FINIT          ; Pas d'affichage
```

Dans le cas où la variable **VALID** n'est pas nulle, l'affichage a lieu. L'affichage des champs **<Heures>**, **<Minutes>** et **<Secondes>** sur l'écran met en œuvre les macros suivantes :

— **GETCUR** : Lecture de la position du curseur sur l'écran,

— **CURDIS** : Effacement du curseur,

— **SETCUR** : Positionnement du curseur sur l'écran,

— **TXTOUT** : Affichage d'un caractère sur l'écran.

La première action à effectuer, avant tout affichage, consiste à mémoriser la position courante du curseur sur l'écran dans le but de la restituer après l'affichage :

```
         CALL  GETCUR       ; Position courante curseur
         LD    (SAVCUR),HL  ; Sauvegarde
```

Le curseur est ensuite effacé de l'écran pour éviter sa visualisation durant l'affichage de l'heure :

```
CALL     CURDIS             ; TXT CUR DISABLE
```

L'affichage des divers champs horaires sur l'écran met en œuvre une routine de conversion décimal/ASCII intéressante. Les caractères affichables sur l'écran doivent en effet être passés à la macro d'affichage sous la forme de codes ASCII.

Supposons que le champ **<Minutes>** ait pour valeur 45. Les valeurs 52 (code ASCII de 4) et 53 (code ASCII de 5) devront successivement être passées à la routine d'affichage.

Pour séparer le poids fort et le poids faible d'un nombre code en décimal, nous procédons comme suit :

La valeur décimale à convertir est placée dans le registre **A**. Le registre **B** est initialisé à la valeur constante 10 et le registre **C** (compteur) est mis à zéro. Le programme soustrait de **A** le registre **B** autant de fois que nécessaire pour que sa valeur soit inférieure à 10. Le registre **C** est incrémenté à chaque soustraction. Après n soustractions, le registre **C** contient le poids fort (les dizaines) et le registre **A** le poids faible (les unités). Par exemple, pour extraire les dizaines et unités du champ **<Heures>** :

```
         XOR   A
         LD    C,A          ; Dizaines
         LD    B,10         ; Valeur de soustraction
```

```
        LD      A,(HE)          ; Valeur à convertir
BOU1:   EQU     $
        CP      B
        JP      C,AFF1
        INC     C
        SUB     B
        JR      BOU1            ; Boucle jusqu'à ce que A<10
```

L'affichage d'un champ horaire se fait en trois parties :

— affichage du poids fort,

— affichage du poids faible,

— affichage du séparateur (pour les champs <Heures> et <Minutes>).

Par exemple, pour le champ <Heures> :

```
AFF1:   EQU     $
        LD      (SAVA),A
        LD      A,48
        ADD     A,C
        CALL    TXTOUT          ; Affichage des dizaines
        LD      A,(SAVA)
        LD      B,48
        ADD     A,B
        CALL    TXTOUT          ; Affichage des unités
        LD      A,58
        CALL    TXTOUT          ; Affichage du séparateur
```

La routine de traitement de l'interruption se termine par la restitution de la position du curseur :

```
        LD      HL,(SAVCUR)
        CALL    SETCUR
```

l'autorisation de l'affichage du curseur sur l'écran :

```
        CALL    CUREN
```

la validation des interruptions et la restitution du contexte avant l'activation de la routine de traitement :

```
        POP     IY
        POP     IX
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        EI
```

Comme toujours, voici le chargeur Basic correspondant au programme :

```
1000 '----------------------------
1010 ' Chargeur de la RSX HORLOGE
1020 '----------------------------
1030 '
1040 FOR I=&9000 TO &9148
1050   READ A$
1060   A=VAL("&"+A$)
1070   POKE I,A
1080 NEXT I
1090 '
1100 '----------------------------
1110 ' Codes op de la RSX HORLOGE
1120 '----------------------------
1130 '
1140 DATA C3,24,90,0,0,0,0,0,0,0,0,0,0,0,0,0
1150 DATA 7F,11,90,15,90,1A,90,C3,29,90,1F,90,C3,31,90,48
1160 DATA 4F,D2,0,11,1,1A,90,21,16,90,CD,D1,BC,CD,55,90
1170 DATA C9,FE,1,CA,3B,90,FE,3,28,8,C9,DD,7E,0,32,23
1180 DATA 90,C9,DD,7E,0,32,12,90,DD,7E,2,32,11,90,DD,7E
1190 DATA 4,32,10,90,C9,21,9,90,6,81,E,0,11,6F,90,CD
1200 DATA EF,BC,21,3,90,11,1,0,1,32,0,CD,E9,BC,C9,F3
1210 DATA F5,C5,D5,E5,DD,E5,FD,E5,3A,12,90,3C,32,12,90,FE
1220 DATA 3C,28,2,18,26,AF,32,12,90,3A,11,90,3C,32,11,90
1230 DATA FE,3C,28,2,18,15,AF,32,11,90,3A,10,90,3C,32,10
1240 DATA 90,FE,18,28,2,18,4,AF,32,10,90,3A,23,90,B7,CA
1250 DATA 38,91,CD,78,BB,22,14,90,CD,7E,BB,26,1,2E,1,CD
1260 DATA 75,BB,AF,4F,6,A,3A,10,90,B8,DA,D1,90,C,90,18
1270 DATA F8,32,13,90,3E,30,81,CD,5A,BB,3A,13,90,6,30,80
1280 DATA CD,5A,BB,3E,3A,CD,5A,BB,AF,4F,6,A,3A,11,90,B8
1290 DATA DA,F7,90,C,90,18,F8,32,13,90,3E,30,81,CD,5A,BB
1300 DATA 3A,13,90,6,30,80,CD,5A,BB,3E,3A,CD,5A,BB,AF,4F
1310 DATA 6,A,3A,12,90,B8,DA,1D,91,C,90,18,F8,32,13,90
1320 DATA 3E,30,81,CD,5A,BB,3A,13,90,6,30,80,CD,5A,BB,2A
1330 DATA 14,90,CD,75,BB,CD,7B,BB,FD,E1,DD,E1,E1,D1,C1,F1
1340 DATA FB,C9,21,3,90,CD,EC,BC,C9,0,0,0,0,0,0,0
```

et les données de checksum afférentes :

```
78 6C B6 E 1A CF D8 C 15 6F E0 BE C5 37 E3 BA D3 B2 76 AF BI
```