9/8.7.1

Affichage d'un message à une heure prédéfinie

Dans ce paragraphe, nous étudions une RSX qui fonctionne sur le même principe que la RSX horloge présentée à la Partie 9, chapitre 8.18, et dont le but est d'afficher un message en haut et à gauche de l'écran à une heure déterminée.

COMMENT UTILISER LA RSX ?

La RSX MES est bien entendu écrite en Assembleur. Elle est présentée sous sa forme brute et sous la forme d'un chargeur Basic. Voici le listing des deux versions.

RSX Assembleur :

```
 1                          ORG  9000H
 2                          LOAD 9000H
 3                  ;
 4                  ;--------------------------------
 5                  ; RSX Affichage d'un message
 6                  ; a une heure predefinie
 7                  ;--------------------------------
 8                  ;
 9 9000 C32890              JP   DEFRSX            ;Definition RSX
10                  ;
11                  ;--------------------------------
12                  ; Declaration des constantes et
13                  ; variables du programme
14                  ;--------------------------------
15                  ;
16 9003 00000000 EVBL:     DB   0,0,0,0
17 9007 00000000           DB   0,0,0,0
```

```
18 900B 00000000            DB    0,0,0,0,0
18 900F 00
19                MES:      DS    2          ;@ variable a afficher
20                AHE:      DS    1          ;Champ Heures alarme
21                AMI:      DS    1          ;Champ Minutes alarme
22                ASE:      DS    1          ;Champ Secondes alarme
23                HE:       DS    1          ;Champ Heures
24                MI:       DS    1          ;Champ Minutes
25                SE:       DS    1          ;Champ Secondes
26                SAVA:     DS    1          ;Sauvegarde de A
27                SAVCUR:   DS    2          ;Sauv coord curseur
28                BUF:      DS    4          ;Zone RAM pour LOG EXT
29 901F 2490      PTRTAB:   DW    TABLE      ;Pointeur TABLE
30 9021 C33590              JP    TRAITE     ;Traitement
31 9024 4D45      TABLE:    DB    "ME"
32 9026 D3                  DB    "S"+80H
33 9027 00                  DB    0          ;Fin de table
34                ;
35                INITEV:   EQU   0BCEFH     ;INIT EVEN BLOC
36                ADDEVE:   EQU   0BCE9H     ;ADD EVEN BLOC
37                DELEVE:   EQU   0BCECH     ;DEL EVEN BLOC
38                TIMESET:  EQU   0BD10H     ;KL TIME SET
39                TXTOUT:   EQU   0BB5AH     ;TXT OUTPUT
40                SETCUR:   EQU   0BB75H     ;TXT SET CURSOR
41                GETCUR:   EQU   0BB78H     ;TXT GET CURSOR
42                CUREN:    EQU   0BB7BH     ;TXT CUR ENABLE
43                CURDIS:   EQU   0BB7EH     ;TXT CUR DISABL
44                LOGEXT:   EQU   0BCD1H     ;KL LOG EXT
45                LROMEN:   EQU   0B906H     ;KL L ROM EN
46                LROMDI:   EQU   0B909H     ;KL L ROM DIS
47                ;
```

```
48                ;--------------------------------
49                ; Definition de la RSX
50                ;--------------------------------
51                ;
52                DEFRSX:     EQU  $                ;Point d'entree
53 9028 011F90               LD   BC,PTRTAB        ;Ptr table definition
54 902B 211B90               LD   HL,BUF           ;Buffer pour LOG EXT
55 902E CDD1BC               CALL LOGEXT           ;Definition de la RSX
56 9031 CD6E90               CALL INSTALL          ;Installation de l'IT
57 9034 C9                   RET
58                ;
59                ;--------------------------------
60                ; Traitement de la RSX
61                ;--------------------------------
62                ;
63                TRAITE:     EQU  $
64 9035 FE04                 CP   4
65 9037 CA3F90               JP   Z,QUATPAR        ;Quatre parametres
66 903A FE03                 CP   3
67 903C 281D                 JR   Z,TROIPAR        ;Trois parametres
68 903E C9                   RET
69                ;
70                ;--------------------------------
71                ; Enregistrement du message a
72                ; emettre et de l'heure d'emission
73                ;--------------------------------
74                ;
75                QUATPAR:    EQU  $
76 903F DD6601               LD   H,(IX+1)
77 9042 DD6E00               LD   L,(IX+0)
78 9045 221090               LD   (MES),HL         ;Adresse de la chaine
```

```
 79 9048 DD7E02                 LD    A,(IX+2)
 80 904B 321490                 LD    (ASE),A           ;Secondes alarme
 81 904E DD7E04                 LD    A,(IX+4)
 82 9051 321390                 LD    (AMI),A           ;Minutes alarme
 83 9054 DD7E06                 LD    A,(IX+6)
 84 9057 321290                 LD    (AHE),A           ;Heures alarme
 85 905A C9                     RET
 86                  ;
 87                  ;---------------------------------
 88                  ;Initialisation de l'horloge
 89                  ;---------------------------------
 90                  ;
 91                  TROIPAR:    EQU   $
 92 905B DD7E00                 LD    A,(IX+0)
 93 905E 321790                 LD    (SE),A            ;Init secondes
 94 9061 DD7E02                 LD    A,(IX+2)
 95 9064 321690                 LD    (MI),A            ;Init minutes
 96 9067 DD7E04                 LD    A,(IX+4)
 97 906A 321590                 LD    (HE),A            ;Init heures
 98 906D C9                     RET
 99                  ;
100                  ;
101                  ;---------------------------------
102                  ; Installation de l'IT
103                  ;---------------------------------
104                  ;
105                  INSTALL:    EQU   $
106 906E 210990                LD    HL,EVBL+6
107 9071 0681                  LD    B,81H
108 9073 0E00                  LD    C,0
```

```
109 9075 118890             LD   DE,TRAIT          ;@ Traitement
110 9078 CDEFBC             CALL INITEV            ;INIT EVEN BLOC
111 907B 210390             LD   HL,EVBL
112 907E 110100             LD   DE,1
113 9081 013200             LD   BC,50
114 9084 CDE9BC             CALL ADDEVE            ;ADD EVEN BLOC
115 9087 C9                 RET
116              ;
117              ;--------------------------------
118              ; Routine de traitement
119              ;--------------------------------
120              ;
121              TRAIT:      EQU  $
122 9088 F3                 DI
123 9089 F5                 PUSH AF
124 908A C5                 PUSH BC
125 908B D5                 PUSH DE
126 908C E5                 PUSH HL
127 908D DDE5               PUSH IX
128 908F FDE5               PUSH IY
129              ;
130 9091 3A1790             LD   A,(SE)
131 9094 3C                 INC  A
132 9095 321790             LD   (SE),A
133 9098 FE3C               CP   60
134 909A 2802               JR   Z,INCMIN          ;Incremente minute
135 909C 1826               JR   FININC            ;Incrementation terminee
136              INCMIN:     EQU  $
137 909E AF                 XOR  A
138 909F 321790             LD   (SE),A
139 90A2 3A1690             LD   A,(MI)
```

```
140 90A5 3C                     INC   A
141 90A6 321690                 LD    (MI),A
142 90A9 FE3C                   CP    60
143 90AB 2802                   JR    Z,INCHR           ;Incremente heure
144 90AD 1815                   JR    FININC            ;Incrementation terminee
145                 INCHR:      EQU   $
146 90AF AF                     XOR   A
147 90B0 321690                 LD    (MI),A
148 90B3 3A1590                 LD    A,(HE)
149 90B6 3C                     INC   A
150 90B7 321590                 LD    (HE),A
151 90BA FE18                   CP    24
152 90BC 2802                   JR    Z,RAZHR           ;RAZ champ heure
153 90BE 1804                   JR    FININC            ;Incrementation terminee
154                 RAZHR:      EQU   $
155 90C0 AF                     XOR   A
156 90C1 321590                 LD    (HE),A
157                 ;
158                 ;-------------------------
159                 ; Affichage du message
160                 ;-------------------------
161                 ;
162                 FININC:     EQU   $
163 90C4 3A1590                 LD    A,(HE)
164 90C7 47                     LD    B,A
165 90C8 3A1290                 LD    A,(AHE)
166 90CB B8                     CP    B
167 90CC C21791                 JP    NZ,PASDAF         ;Pas d'affichage
168 90CF 3A1690                 LD    A,(MI)
169 90D2 47                     LD    B,A
170 90D3 3A1390                 LD    A,(AMI)
```

```
171 90D6 B8                     CP    B
172 90D7 C21791                 JP    NZ,PASDAF          ;Pas d'affichage
173 90DA 3A1790                 LD    A,(SE)
174 90DD 47                     LD    B,A
175 90DE 3A1490                 LD    A,(ASE)
176 90E1 B8                     CP    B
177 90E2 C21791                 JP    NZ,PASDAF          ;Pas d'affichage
178                 ;----------
179                 ;Affichage
180                 ;----------
181 90E5 CD78BB               CALL  GETCUR               ;Position courante curs
182 90E8 221990               LD    (SAVCUR),HL
183 90EB CD7EBB               CALL  CURDIS               ;TXT CUR DISABLE
184 90EE 2601                 LD    H,1                  ;Colonne curseur
185 90F0 2E01                 LD    L,1                  ;Ligne curseur
186 90F2 CD75BB               CALL  SETCUR               ;Position curseur
187
188 90F5 CD09B9               CALL  LROMDI               ; Lecture en RAM
189 90F8 2A1090               LD    HL,(MES)
190 90FB 7E                   LD    A,(HL)
191 90FC 47                   LD    B,A                  ;Longueur chaine
192 90FD 23                   INC   HL
193 90FE 7E                   LD    A,(HL)
194 90FF 5F                   LD    E,A
195 9100 23                   INC   HL
196 9101 7E                   LD    A,(HL)
197 9102 57                   LD    D,A
198 9103 EB                   EX    DE,HL                ;@ 1er caractere
199
200               BIS:        EQU   $
201 9104 7E                   LD    A,(HL)
```

```
202 9105 CD5ABB                CALL TXTOUT
203 9108 23                    INC  HL
204 9109 10F9                  DJNZ BIS
205
206 910B CD06B9                CALL LROMEN             ;Fin lect RAM basse
207 910E 2A1990                LD   HL,(SAVCUR)
208 9111 CD75BB                CALL SETCUR             ;Restitution curseur
209 9114 CD7BBB                CALL CUREN              ;TXT CUR ENABLE
210             ;
211             PASDAF:        EQU  $
212 9117 FDE1                  POP  IY
213 9119 DDE1                  POP  IX
214 911B E1                    POP  HL
215 911C D1                    POP  DE
216 911D C1                    POP  BC
217 911E F1                    POP  AF
218 911F FB                    EI
219 9120 C9                    RET
220                            END
```

Chargeur Basic :

```
1000 REM ------------------------------------------------
1010 REM Affichage d'un message a une heure determinee
1020 REM ------------------------------------------------
1030 REM
1040 FOR i=&9000 TO &9120
1050   READ a$
1060   POKE i,VAL("&"+a$)
1070 NEXT i
1080 REM
1090 REM --------
1100 REM Donnees
1110 REM --------
1120 REM
1130 DATA C3,28,90,0,0,0,0,0,0,0,0,0,0,0,0,0
1140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24
1150 DATA 90,C3,35,90,4D,45,D3,0,1,1F,90,21,1B,90,CD,D1
1160 DATA BC,CD,6E,90,C9,FE,4,CA,3F,90,FE,3,28,1D,C9,DD
1170 DATA 66,1,DD,6E,0,22,10,90,DD,7E,2,32,14,90,DD,7E
1180 DATA 4,32,13,90,DD,7E,6,32,12,90,C9,DD,7E,0,32,17
1190 DATA 90,DD,7E,2,32,16,90,DD,7E,4,32,15,90,C9,21,9
1200 DATA 90,6,81,E,0,11,88,90,CD,EF,BC,21,3,90,11,1
1210 DATA 0,1,32,0,CD,E9,BC,C9,F3,F5,C5,D5,E5,DD,E5,FD
1220 DATA E5,3A,17,90,3C,32,17,90,FE,3C,28,2,18,26,AF,32
1230 DATA 17,90,3A,16,90,3C,32,16,90,FE,3C,28,2,18,15,AF
1240 DATA 32,16,90,3A,15,90,3C,32,15,90,FE,18,28,2,18,4
1250 DATA AF,32,15,90,3A,15,90,47,3A,12,90,B8,C2,17,91,3A
1260 DATA 16,90,47,3A,13,90,B8,C2,17,91,3A,17,90,47,3A,14
1270 DATA 90,B8,C2,17,91,CD,78,BB,22,19,90,CD,7E,BB,26,1
1280 DATA 2E,1,CD,75,BB,CD,9,B9,2A,10,90,7E,47,23,7E,5F
1290 DATA 23,7E,57,EB,7E,CD,5A,BB,23,10,F9,CD,6,B9,2A,19
1300 DATA 90,CD,75,BB,CD,7B,BB,FD,E1,DD,E1,E1,D1,C1,F1,FB
1310 DATA C9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Données de Checksum correspondantes :

```
7C 24 9D DF 8 80 F3 91 9E 63 DF 2A E9 67 B1 50 45 97 C9
```

Si vous avez choisi d'utiliser le programme sous sa forme Assembleur, compilez-le et installez la RSX en exécutant la routine située en **&H9028**.

Si vous avez choisi d'utiliser le programme sous sa forme Basic, exécutez-le, et installez la RSX en tapant sous Basic :

CALL &H9000

Le programme utilise une horloge interne gérée sous interruptions. Avant toute chose, vous devez initialiser cette horloge 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 RSX :

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

Par exemple :

```
|MES,11,15,30
```

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

A partir de cet instant, et jusqu'à la mise hors tension de l'ordinateur, cette horloge sera maintenue de manière transparente.

Pour obtenir l'affichage d'un message, il vous suffit de préciser l'heure d'émission du message et le libellé du message en tapant :

```
|MES,<Heures>,<Minutes>,<Secondes>,<Var message>
```

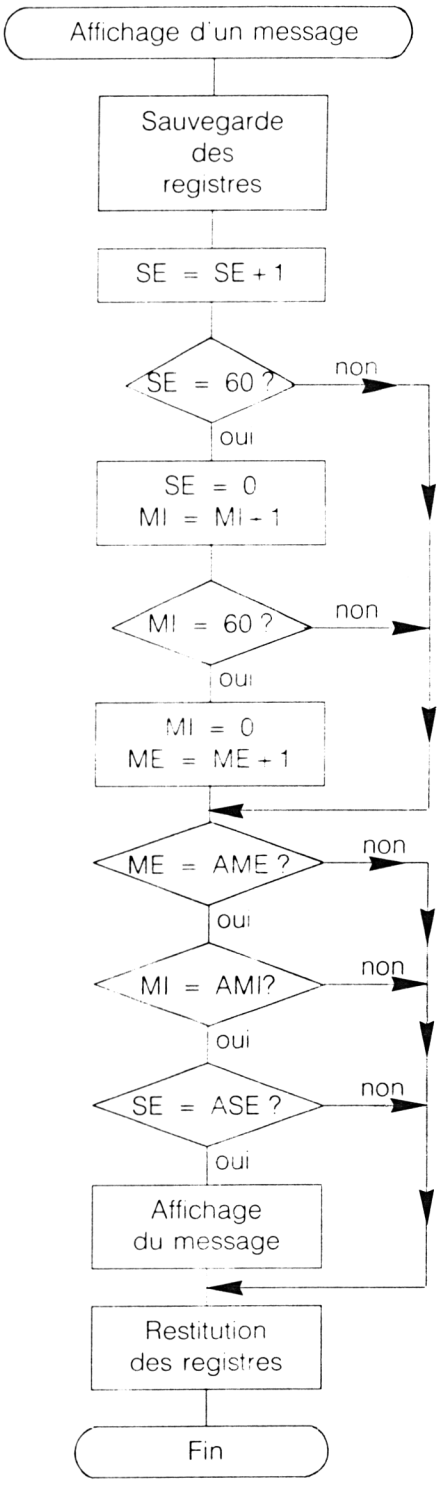
Par exemple :

```
a$="A table"
|MES,12,00,00,a$
```

affiche le message **"A table"** en haut et à gauche de l'écran à douze heures précises.

LA RSX EN DÉTAIL

La logique du traitement de la RSX apparaît dans l'ordinogramme suivant :



La première instruction du programme donne le contrôle à la routine d'installation en **&H9028** :

```
;Définition RSX
JP DEFRSX
```

Le programme se poursuit par diverses déclarations de constantes et de variables. Remarquez entre autres :

— le bloc d'événement **EVBL**,

— la variable **PTRTAB** qui pointe sur la table de définition de la RSX et sur l'adresse de traitement,

— diverses routines Firmware.

La routine d'installation de la RSX est des plus classiques. Elle utilise la routine **LOGEXT** (voir Partie 4, chapitre 2.7, page 24) du Firmware pour définir la syntaxe de la nouvelle RSX :

```
DEFRSX:   EQU     $
          LD      BC,PTRTAB
          LD      HL,BUF
          CALL    LOGEXT
```

et les routines **INITEVENT** et **EVENT** (voir Partie 4,chapitre 2.7, page 57) du Firmware pour initialiser le bloc événement (type d'interruption et fréquence) :

```
          LD      HL,EVBL+6
          LD      B,81H
          LD      C,0
          LD      DE,TRAIT
          CALL    INITEV
          LD      HL,EVBL
          LD      DE,1
          LD      BC,50
          CALL    ADDEVE
```

Lorsque la RSX est activée par une instruction |MES, le programme exécuté se trouve à l'étiquette **TRAITE**. Il teste le nombre de paramètres passés et exécute la routine **TROIPAR** ou **QUATPAR** :

```
TRAITE:   EQU     $
          CP      4
          JP      Z,QUATPAR ;Quatre paramètres
          CP      3
          JP      Z,TROIPAR ;Trois paramètres
```

Lorsque la RSX a détecté quatre paramètres, la routine **QUATPAR** sauvegarde l'adresse de la chaîne à afficher et l'heure d'affichage :

```
QUATPAR:  EQU     $
          LD      HL,(IX+1)
          LD      L(IX+0)
          LD      (MES),HL    ;Adresse de la chaîne
          LD      A,(IX+2)
```

```
        LD      (ASE),A     ;Secondes alarme
        LD      A,(IX+4)
        LD      (AMI),A     ;Minutes alarme
        LD      A,(IX+6)
        LD      (AHE),A     ;Heures alarme
```

Lorsque la RSX a détecté trois paramètres, la routine TROIPAR initialise l'horloge temps réel en fonction des données passées :

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

La routine d'interruption TRAIT est activée toutes les secondes. Elle maintient l'heure en incrémentant les variables HE, MI et SE.

La première action effectuée dans cette routine 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 d'afficher le message demandé (si l'heure système est égale à l'heure d'alarme).

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 d'afficher le message demandé (si l'heure système est égale à l'heure d'alarme).

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
```

La routine située en FININC compare l'heure système et l'heure d'alarme. Si un des champs horaires des deux heures (heures, minutes ou secondes) n'est pas égal, aucun affichage ne doit être effectué :

```
FININC: EQU   $
        LD    A,(HE)
        LD    B,A
        LD    A,(AHE)
        CP    B
        JP    NZ,PASDAF
        LD    A,(MI)
        LD    B,A
        LD    A,(AMI)
        CP    B
        JP    NZ,PASDAF
        LD    A,(SE)
        LD    B,A
        LD    A,(ASE)
        CP    B
        JP    NZ,PASDAF
```

Lorsque les trois champs horaires (système et alarme) sont égaux, le message situé en (ADR) est affiché sur l'écran.

La position courante du curseur est sauvegardée dans la variable **SAVCUR** :

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

L'affichage du curseur est dévalidé à l'aide de la macro **CURDIS** du Firmware :

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

Le curseur est positionné en ligne 1 et colonne 1 (en haut et à gauche de l'écran) :

```
        LD      H,1 ; Colonne curseur
        LD      L,1 ;Ligne curseur
        CALL    SETCUR ;Position curseur
```

Pour pouvoir accéder à la variable chaîne à afficher, il faut au préalable valider la RAM inférieure à l'aide de la macro **LROMDIS** du Firmware :

```
        CALL    LROMDI ;Lecture en RAM
```

L'adresse de la chaîne est stockée dans le registre **HL**, et sa longueur dans le registre **B** :

```
        LD      HL,(MES)
        LD      A,(HL)
        LD      B,A ;Longueur chaîne
        INC     HL
        LD      A,(HL)
        LD      E,A
        INC     HL
        LD      A,(HL)
        LD      D,A
        EX      DE,HL ;@ 1er caractère
```

Une boucle utilisant la puissante instruction **DJNZ** affiche enfin la chaîne en haut de l'écran à l'aide de la macro **TXTOUT** du Firmware :

```
BIS:    EQU     $
        LD      A,(HL)
        CALL    TXTOUT
        INC     HL
        DJNZ    BIS
```

La ROM basse est revalidée :

```
        CALL    LROMEN ;Fin lect RAM basse
```

Le curseur est repositionné en **SAVCUR**, et son affichage est validé :

```
        LD      HL,(SAVCUR)
        CALL    SETCUR ;Restitution curseur ;Restitution curseur
        CALL    CUREN ;TXT CUR ENABLE
```

La routine se termine par la restitution du contexte sauvegardé dans les registres du microprocesseur, et la revalidation des interruptions :

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