

9/11

Traitement de texte

Nous vous présentons un logiciel de traitement de texte pleine page très perfectionné. Ce logiciel est totalement écrit en assembleur. Vous découvrirez petit à petit comment fonctionne le cœur du traitement de texte (saisie de texte sur un écran en pleine page avec une gestion de scrolling haut, bas, droite et gauche, modes remplacement et insertion, gestion de blocs, insertion/écriture d'une partie du texte, entrée sur disquette, gestion d'un spooler d'imprimante, etc.).

9/11.1

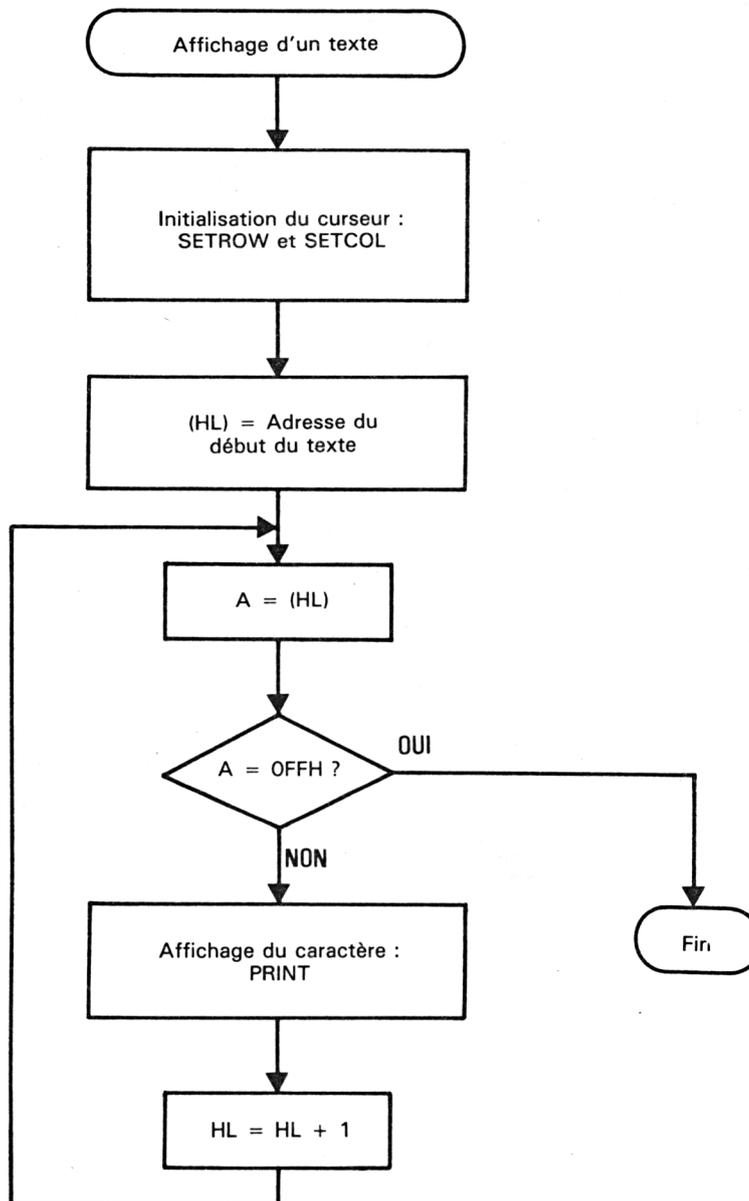
Mise en œuvre d'utilitaires

Avant de traiter le logiciel de traitement de texte à proprement parler, nous allons mettre en œuvre plusieurs utilitaires d'ordre général.

- Le premier utilitaire permet d'afficher un texte alphanumérique à un endroit quelconque sur l'écran.

Cet utilitaire servira à afficher tous les messages d'information (comme par exemple les messages indiquant à l'utilisateur d'entrer un nom de fichier ou une chaîne à rechercher dans le texte), et la ligne d'état en bas de l'écran.

La logique mise en œuvre pour réaliser cet utilitaire est la suivante :



Les routines du Firmware utilisées par cet utilitaire sont les suivantes :

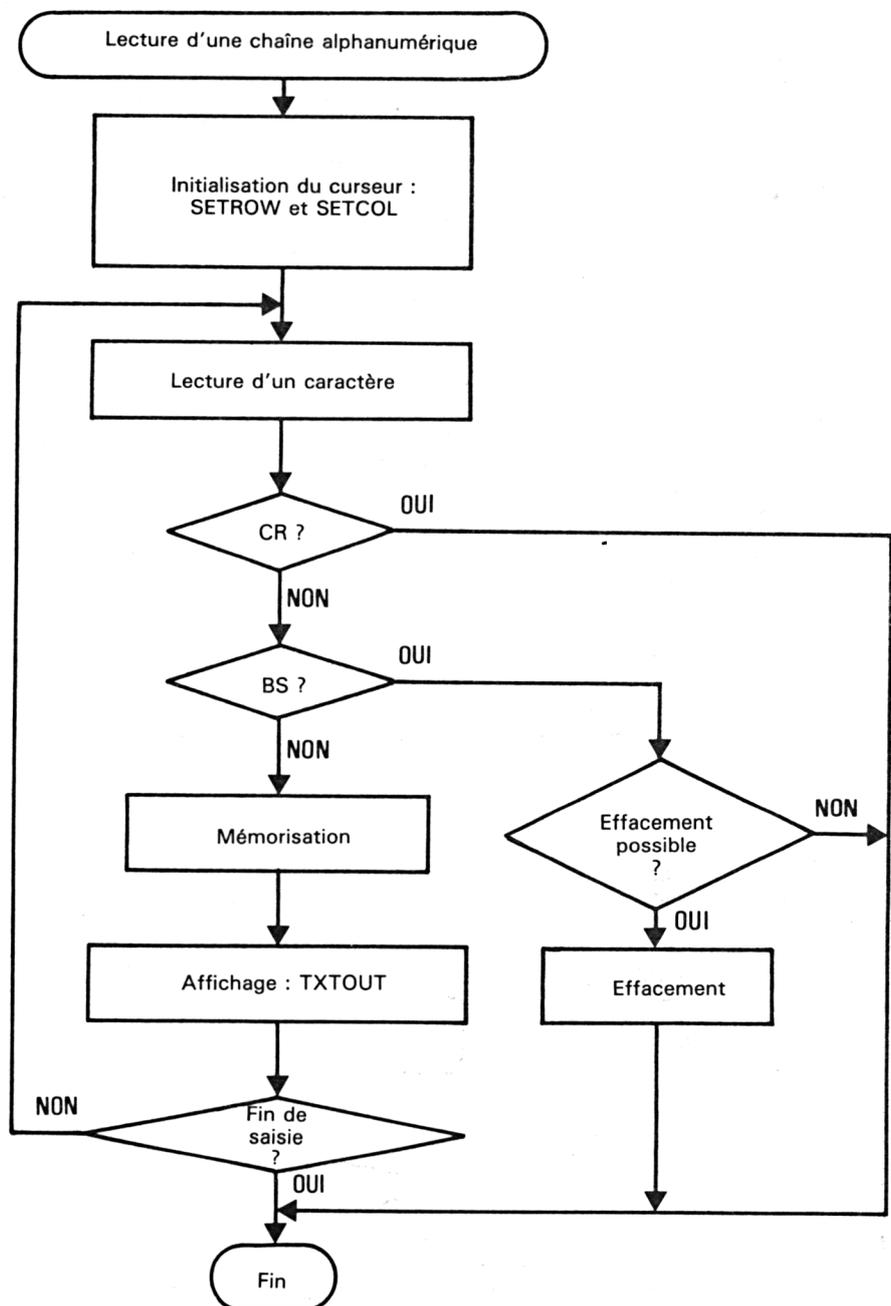
- 0BB72H: TXT SET ROW
(Positionnement de la ligne du curseur)
- 0BB6FH: TXT SET COLUMN
(Positionnement de la colonne du curseur)

Reportez-vous à la Partie 4, chapitre 2.7 page 78, pour avoir des détails à ce sujet.

- Le second utilitaire permet de lire une chaîne de caractères tapée au clavier. La longueur de la chaîne est paramétrable, et la touche (effacement du caractère à gauche du curseur) est prise en compte.

Cet utilitaire servira à lire toutes les données entrées par l'utilisateur suite aux messages d'information affichés par le traitement de texte (comme par exemple les messages indiquant à l'utilisateur d'entrer le nom d'une chaîne à rechercher dans le texte).

La logique mise en œuvre pour réaliser cet utilitaire est la suivante :



Les routines du Firmware utilisées par cet utilitaire sont les suivantes :

- OBB72H: TXT SET ROW
(Positionnement de la ligne du curseur)
- OBB6FH: TXT SET COLUMN
(Positionnement de la colonne du curseur)
- OBB06H: KM WAIT CHAR
(Lecture d'un caractère au clavier)
- OBB5AH: TXT OUTPUT
(Affichage d'un caractère, y compris le caractère de contrôle)
- OBB5DH: TXT WR CHAR
(Affichage d'un caractère, sauf caractère de contrôle)

Pour montrer comment fonctionnent ces deux utilitaires, nous avons réalisé une courte démonstration (appelée « PROGRAMME PRINCIPAL » dans le programme ci-dessous).

Elle consiste à afficher un texte sur l'écran en ligne 1, colonne 1, et à saisir un texte de longueur comprise entre 1 et 6 caractères. Le texte entré se trouve dans le buffer « BUF » à partir de l'adresse &H908C.

Le listing Assembleur de ces deux utilitaires est le suivant :

```

1          ORG 9000H
2          LOAD 9000H
3 9000 C3AB90      JP EXEC ;Execution
4          FIN:    EQU $
5 9003 00         NOP
6          ;
7          ;-----
8          ; Affichage d un texte à l'ecran
9          ;-----
10         ;Entree: HL=pointeur memoire
11         ;      DE=Ligne,Colonne
12         ;-----
13         .AFFICH: EQU $
14 9004 E5         PUSH HL
15 9005 F5         PUSH AF

```

```

16 9006 7A          LD  A,D
17 9007 CD72BB      CALL SETROW          ;Init ligne
18 900A 7B          LD  A,E
19 900B CD6FBB      CALL SETCOL          ;Init colonne
20 900E F1          POP  AF
21 900F E1          POP  HL
22 9010 E5          PUSH HL
23 9011 F5          PUSH AF              ;Init et sauveg registre
24                  AT1: EQU  $              ;Boucle d'affichage
25 9012 7E          LD  A,(HL)
26 9013 FEFF        CP   OFFH          ;Delimiteur?
27 9015 2806        JR   Z,AT2          ;Oui=>Fin affichage
28 9017 CD5ABB      CALL PRINT           ;Aff caractere
29 901A 23          INC  HL
30 901B 18F5        JR   AT1
31                  AT2: EQU  $
32 901D F1          POP  AF
33 901E E1          POP  HL
34 901F C9          RET
35                  ;
36                  ;
37                  ;-----
38                  ; Lecture de n caracteres
39                  ;-----
40                  ;Entree: B=Nbre de caract a lire
41                  ;          HL=Pointeur buffer
42                  ;          DE=Ligne,Colonne
43                  ;Sortie: Buf=buffer de lecture
44                  ;-----
45                  LECTN: EQU  $
46 9020 3A8990      LD  A,(PX)

```

```

47 9023 328B90      LD   (PDX),A          ;Pos depart en X
48 9026 E5          PUSH HL
49 9027 7A          LD   A,D
50 9028 CD72BB      CALL SETROW          ;Init ligne
51 902B 7B          LD   A,E
52 902C CD6FBB      CALL SETCOL          ;Init colonne
53 902F E1          POP  HL
54 9030 218C90      LD   HL,BUF          ;Adresse du buffer
55                LO:  EQU  ±          ;Boucle principale
56 9033 CD06BB      CALL WAIT            ;Lect 1 caractere
57 9036 FE0D        CP   CRLF
58 903B 2B26        JR   Z,L1
59 903A FE7F        CP   BS
60 903C 2B23        JR   Z,L2
61                ; Caractere normal
62 903E E5          PUSH HL              ;Sauveg HL
63 903F 32AA90      LD   (CAR),A        ;Memo du caractere
64 9042 3AB990      LD   A,(PX)
65 9045 5F          LD   E,A
66 9046 1600        LD   D,0            ;DE=Depl dans buffer
67 904B 1B          DEC  DE
68 9049 19          ADD  HL,DE
69 904A 3AAA90      LD   A,(CAR)
70 904D 77          LD   (HL),A        ;Stockage caract
71 904E C5          PUSH BC
72 904F CD5DBB      CALL TXTOUT         ;Affichage caractere
73 9052 C1          POP  BC
74 9053 E1          POP  HL            ;Restit registres
75 9054 3AB990      LD   A,(PX)
76 9057 3C          INC  A

```

```

77 905B 328990      LD   (PX),A
78 905B 05          DEC  B           ;Nb caract a saisir-1
79 905C 78          LD   A,B
80 905D B7          OR   A
81 905E 20D3        JR   NZ,LO      ;On continue
82                  L1:   EQU  $           ;Sortie du S/P
83 9060 C9          RET
84                  L2:   EQU  $           ;
85 9061 3AB990      LD   A,(PDX)
86 9064 C601        ADD  A,1
87 9066 4F          LD   C,A
88 9067 3A8990      LD   A,(PX)
89 906A B9          CP   C
90 906B 38C6        JR   C,LO      ;BS impossible
91 906D 3A8990      LD   A,(PX)
92 9070 3D          DEC  A
93 9071 328990      LD   (PX),A
94 9074 C5          PUSH BC
95 9075 E5          PUSH HL      ;BC,HL detruits par SETCC
96 9076 CD6FBB      CALL SEICOL
97 9079 3E2D        LD   A,2DH      ;Caract -
98 907B CD5DBB      CALL IXTOUT     ;Affich caract -
99 907E 3A8990      LD   A,(PX)
100 9081 CD6FBB     CALL SEICOL     ;Repos colonne
101 9084 E1         POP  HL
102 9085 C1         POP  BC      ;Restit registres
103 9086 04         INC  B
104 9087 18AA       JR   LO      ;Suite boucle
105                  ;-----
106                  ;Zone des EQU
107                  ;-----

```

```

108      CRLF:      EQU  13          ;Code CR
109      BS:       EQU  127         ;Code BS
110      ;
111      WAIT:     EQU  0B06H       ;KM WAIT CHAR
112      PRINI:    EQU  0BB5AH      ;TXT OUTPUT
113      TXTOUT:   EQU  0BB5DH      ;TXT WR CHAR
114      SETCOL:   EQU  0BB6FH      ;TXT SET COLUMN
115      SETROW:   EQU  0BB72H      ;TXT SET ROW
116      ;
117      ;-----
118      ;Zone des DS
119      ;-----
120      PX:       DS    1          ;Position en X
121      PY:       DS    1          ;Position en Y
122      PDX:      DS    1          ;Pod depart en X
123      BUF:      DS    30         ;Buffer de lecture
124      CAR:      DS    1          ;Stock tempo 1 caract
125      ;
126      ;-----
127      ; PROGRAMME PRINCIPAL
128      ;-----
129      EXEC:     EQU  *
130 90AB 110101      LD    DE,101H    ;Ligne,Colonne
131 90AE 21C990      LD    HL,TEXIE   ;Pointeur texte a aff
132 90B1 CD0490      CALL AFFICH
133      ;
134 90B4 3E16        LD    A,22
135 90B6 32B990      LD    (PX),A    ;Abcisse saisie
136 90B9 3E05        LD    A,5
137 90BB 32BA90      LD    (PY),A    ;Colonne saisie
138 90BE 0606        LD    B,6      ;Lgr texte saisi

```

```

139 90C0 111601          LD   DE,116H          ;Ligne,Colonne
140 90C3 CD2090          CALL LECTN          ;Lecture
141 90C6 C30390          JP    FIN
142                      ;
143                      ;
144 90C9 456E7472 TEXTE: DB   "Entrez 6 caracter
144 90CD 657A2036
144 90D1 20636172
144 90D5 61637465
144 90D9 7265733A
145 90DD 202D2D2D          DB   "  ----"
145 90E1 2D2D2D
146 90E4 FF              DB   OFFH
147                      END
    
```

Partie 9 : Programmes

Le chargeur Basic de ce même programme est le suivant :

```

1000 REM -----
1010 REM Boucle de lecture des donnees et memorisation
1020 REM -----
1030 FOR I=&9000 TO &90E4
1040   READ A$
1050   A=VAL("&H"+A$)
1060   POKE I,A
1070 NEXT I
1080 REM -----
1090 REM Appel des S/P Assembleur
1100 REM -----
1110 MODE 1
1120 CALL &9000
1130 END
1140 REM -----
1150 REM Donnees
1160 REM -----
1170 DATA C3,AB,90,C9,E5,F5,7A,CD,72,BB,7B,CD,6F,BB,F1,E1
1180 DATA E5,F5,7E,FE,FF,28,6,CD,5A,BB,23,18,F5,F1,E1
1190 DATA C9,3A,89,90,32,8B,90,E5,7A,CD,72,BB,7B,CD,6F,BB
1200 DATA E1,21,8C,90,CD,6,BB,FE,D,28,26,FE,7F,28,23,E5
1210 DATA 32,AA,90,3A,89,90,5F,16,0,1B,19,3A,AA,90,77,C5
1220 DATA CD,5D, BB,C1,E1,3A,89,90,3C,32,89,90,5,7B,B7,20
1230 DATA D3,C9,3A,8B,90,C6,1,4F,3A,89,90,B9,38, C6,3A,89
1240 DATA 90,3D,32,89,90,C5,E5,CD,6F,BB,3E,2D,CD,5D,BB,3A
1250 DATA 89,90,CD,6F,BB,E1,C1,4,1B,AA,0,0,0,0,0,0
1260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1270 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,11,1,1,21
1280 DATA C9,90,CD,4,90,3E,16,32,89,90,3E,5,32,8A,90,6
1290 DATA 6,11,16,1,CD,20,90,C3,3,90,45, 6E,74,72,65,7A
1300 DATA 20,36,20,63,61,72,61,63,74,65,72,65,73,3A,20,2D
1310 DATA 2D,2D,2D,2D,2D,FF,0,0,0,0,0,0,0,0,0,0

```

Utilisez ce chargeur plutôt que le code assembleur présenté pour passer moins de temps à saisir le programme. Les performances des deux versions du programme sont totalement identiques puisque le chargeur Basic se contente de placer le code binaire du programme Assembleur en mémoire et d'exécuter le code binaire.

Vous avez sans doute remarqué que chaque ligne de DATA contient 16 codes hexadécimaux. Cela n'est pas un hasard. De cette manière, vous pourrez vérifier rapidement que les codes que vous avez tapés sont bien conformes à ceux du livre. Pour cela, utilisez le programme de checksum (voir Partie 9, chap. 8.4) et comparez le résultat de son exécution avec les données suivantes :

```
64 3A 55 0A B9 C2 98 44 F3 00 FD 30 98 2C B4
```