

9/9.6

Affichage inversé

Les programmes étudiés dans ce paragraphe ont de nombreuses applications, tant au niveau texte que graphique. Ils permettent d'afficher un texte « la tête en bas ». Cela permettra par exemple de porter des annotations le long d'un axe gradué, ou encore d'afficher un texte qui défile le long d'un cadre.

COMMENT UTILISER LES PROGRAMMES

Avant de pouvoir afficher un texte à l'envers, il faut calculer les données graphiques correspondant à chaque caractère. Pour des raisons de rapidité d'exécution, nous avons confié cette tâche à un programme assembleur. Les caractères redéfinis ont un code ASCII compris entre 33 et 122 :

```
!"#$%&'()*+,-./0123456789:;<=>?  
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_  
abcdefghijklmnopqrstuvwxyz
```

Les caractères redéfinis occupent les codes ASCII 128 à 217. Pour afficher à l'envers le caractère de code ASCII N, il faut utiliser l'instruction Basic :

```
PRINT CHR$(N+95)
```

ou la RSX !PRINT étudiée dans la suite du paragraphe.

Cette RSX a le format suivant :

```
!PRINT, @CH,X,Y,C
```

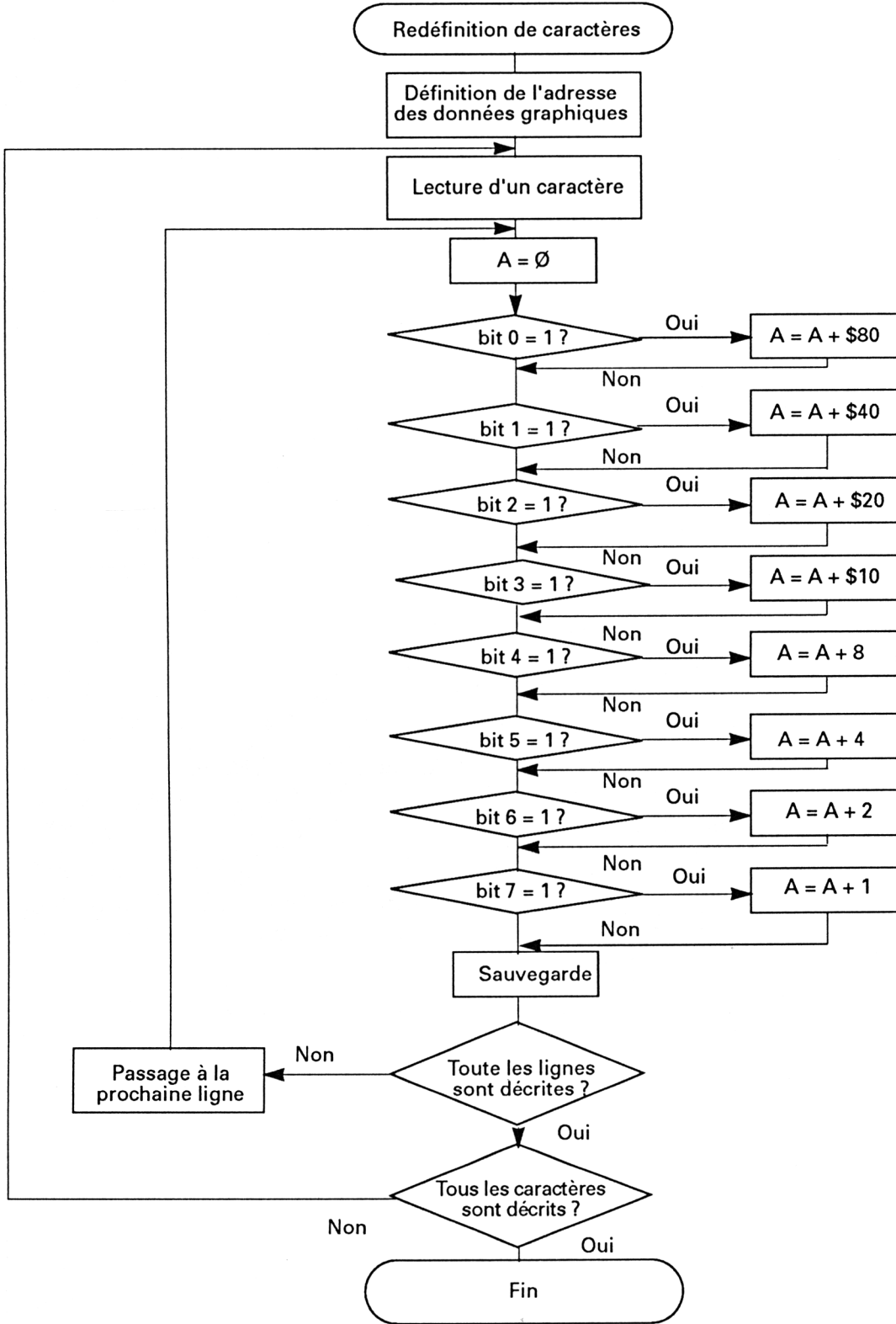
où @CH est l'adresse de la chaîne à afficher, X et Y sont les coordonnées du premier caractère (attention, l'écriture s'effectue de la droite vers la gauche), et C est la couleur d'écriture. Les valeurs possibles pour C dépendent du mode d'affichage :

```
MODE 0 : 0 à 15,  
MODE 1 : 0 à 3,  
MODE 2 : 0 ou 1.
```

LES PROGRAMMES EN DÉTAIL

Redéfinition de caractères

La logique du programme de redéfinition des caractères apparaît dans l'ordinogramme ci-après (voir page suivante).



Ce programme utilise deux macros du FIRMWARE :

- **TXT SET M TABLE** permet de définir l'adresse de la table dans laquelle se trouvent les données de redéfinition des caractères. Cette macro demande deux paramètres en entrée :

- le registre **DE** doit contenir le code ASCII du premier caractère redéfini ;

- le registre **HL** doit contenir l'adresse des données de redéfinition.

- **TXT GET MATRIX** permet de connaître l'adresse mémoire où se trouvent les données correspondant à un caractère dont le code ASCII est précisé dans le registre **A**. L'adresse des données correspondantes est renvoyée dans le registre **HL**.

La méthode utilisée pour redéfinir les caractères est la suivante :

la macro **TXT GET MATRIX** donne l'adresse des données correspondant à un caractère. Ces données sont inversées (de haut en bas et de droite à gauche) et stockées dans un buffer en mémoire. Lorsque tous les caractères sont ainsi redéfinis, la macro **TXT SET M TABLE** permet leur utilisation à travers la fonction Basic **CHR\$** ou la **RSX!IPRINT**.

Le programme de redéfinition des caractères est situé à l'étiquette **REDEF**. Il débute par l'activation des macros situées en **0B906H** et **0B900H** lignes 25 et 26 : validation des ROM supérieure (**HI:KL U ROM ENABLE**) et inférieure (**HI:KL L ROM ENABLE**).

Il définit ensuite le code du premier caractère redéfini, et l'implantation des données de redéfinition à l'aide de la macro **SETMTABL** (lignes 28 à 30).

L'adresse des données graphiques du caractère de code ASCII 32 est lue à l'aide de la macro **GETMAT** (lignes 32 et 33). Chacune des 8 données correspondant à un caractère est ensuite inversée entre les lignes 36 et 75 :

Les bits 0 à 7 sont testés. Lorsque le bit **N** est à 1, le registre **A** est augmenté d'une valeur qui correspond à $2^{(7-N)}$.

En d'autres termes, la donnée analysée est inversée :

- bit 0 placé dans le bit 7 ;

- bit 1 placé dans le bit 6 ;

...

- bit 7 placé dans le bit 0.

Cela produit une inversion horizontale du caractère.

Pour opérer une inversion verticale, il suffit de stocker :

– la donnée correspondant à la première ligne élémentaire dans la donnée correspondant à la 7^e ligne élémentaire ;

...

– la donnée correspondant à la 7^e ligne élémentaire dans la donnée correspondant à la 1^{ère} ligne élémentaire.

La ligne 61 utilise une instruction **LD** pour stocker une donnée graphique dans le buffer mémoire pointé de **DE** :

LD (DE), A

Le registre **C** initialisé à 8 en début de boucle est décrémenté. Le programme redonne le contrôle à l'étiquette **BIS2** tant que **C** différent de 0, c'est-à-dire tant qu'un caractère n'a pas été entièrement inversé (lignes 64 et 65).

Les instructions qui suivent réalisent une boucle sur 91 caractères. Le registre **C** est initialisé à 91 en début de boucle. Il est décrémenté de 1 chaque fois qu'un caractère a été inversé. Le programme redonne le contrôle à l'étiquette **BIS1** tant que les 91 caractères n'ont pas été redéfinis.

Le listing du programme est le suivant :

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ; Redefinition des caracteres de
5          ; code ASCII compris entre 33 et
6          ; 122 en operant une rotation.
7          ; Les nouveaux codes ASCII sont
8          ; stockes a partir du code 128
9          ;-----
10         ;
11         ;
12         ;-----
13         ; Declaration des constantes
14         ; du programme
15         ;-----
16         ;
17         GETMAT: EQU 0BB5H          ;TXT GET MATRIX
18         SETMTABL: EQU 0BB6H       ;TXT SET M TABLE
19         ;
20         ;-----
21         ; Point d'entree
22         ;-----
23         ;
24         REDEF: EQU $
25 9000 CD06B9          CALL 0B906H
26 9003 CD00B9          CALL 0B900H
27         ;
28 9006 118000          LD DE,128      ;1er caract redefini
29 9009 210080          LD HL,8000H    ;@ caract redefinis
30 900C CDABBB          CALL SETMTABL ;Redefinition
31         ;

```

```

32 900F 3E20      LD  A,32          ;1er caract a redefinir
33 9011 CDASBB    CALL GETMAT       ;HL=@ source
34 9014 110700    LD  DE,0007H     ;DE=@ destination
35 9017 0E5B      LD  C,91         ;Nb de caract a redefinir
36              BIS1: EQU  $          ;Redef des 91 caract
37 9019 C5        PUSH BC          ;Sauvegarde
38 901A 0E08      LD  C,8          ;
39              BIS2: EQU  $          ;Redef d'1 caractere
40 901C 46        LD  B,(HL)
41 901D AF        XOR  A
42              ;
43 901E CB40      BIT  0,B
44 9020 202F      JR  NZ,A00
45 9022 CB40      R1:  BIT  1,B
46 9024 202F      JR  NZ,A40
47 9026 CB50      R2:  BIT  2,B
48 9028 202F      JR  NZ,A20
49 902A CB50      R3:  BIT  3,B
50 902C 202F      JR  NZ,A10
51 902E CB60      R4:  BIT  4,B
52 9030 202F      JR  NZ,A08
53 9032 CB68      R5:  BIT  5,B
54 9034 202F      JR  NZ,A04
55 9036 CB70      R6:  BIT  6,B
56 9038 202F      JR  NZ,A02
57 903A CB78      R7:  BIT  7,B
58 903C 202F      JR  NZ,A01
59              R8:  EQU  $
60              ;

```

```
61 903E 12          LD   (DE),A
62 903F 23          INC  HL
63 9040 1B          DEC  DE
64 9041 0D          DEC  C
65 9042 20D8        JR   NZ,BIS2
66                ;
67 9044 E5          PUSH HL
68 9045 211000      LD   HL,16
69 9048 19          ADD  HL,DE
70 9049 54          LD   D,H
71 904A 5D          LD   E,L
72 904B E1          POP  HL
73 904C C1          POP  BC
74 904D 0D          DEC  C
75 904E 20C9        JR   NZ,BIS1
76                ;
77 9050 C9          RET
78                ;
79 9051 C680        A80:  ADD  A,80H
80 9053 18CD        JR   R1
81 9055 C640        A40:  ADD  A,40H
82 9057 18CD        JR   R2
83 9059 C620        A20:  ADD  A,20H
84 905B 18CD        JR   R3
85 905D C610        A10:  ADD  A,10H
86 905F 18CD        JR   R4
87 9061 C608        A08:  ADD  A,08H
88 9063 18CD        JR   R5
89 9065 C604        A04:  ADD  A,04H
90 9067 18CD        JR   R6
91 9069 C602        A02:  ADD  A,02H
```



```

92 906B 18CD          JR  R7
93 906D 3C          A01:  INC  A
94 906E 18CE          JR  R8
95                   ;
96                   END

A80      9051 A40      9055 A20      9059 A10      905D
A08      9061 A04      9065 A02      9069 A01      906D
BIS1     9019 BIS2     901C GETMAT     BBAS REDEF     9000
R1       9022 R2       9026 R3       902A R4       902E
R5       9032 R6       9036 R7       903A R8       903E
SETMTABL BBAB

```

Comme toujours, voici le chargeur Basic correspondant :

```

1000 REM -----
1010 REM Chargeur BASIC de la redefinition des caracteres
1020 REM -----
1030 REM
1040 FOR i=&9000 TO &906F
1050   READ a$
1060   a$="&"+a$
1070   POKE i,VAL(a$)
1080 NEXT i
1090 CALL &9000 'Transfert
1100 END
1110 REM - - - - -
1120 REM Donnees du programme de redefinition
1130 REM - - - - -
1140 REM
1150 DATA CD,6,B9,CD,0,B9,11,80,0,21,0,80,CD,AB,BB,3E
1160 DATA 20,CD,A5,BB,11,7,80,E,5B,C5,E,8,46,AF,CB,40
1170 DATA 20,2F,CB,48,20,2F,CB,50,20,2F,CB,58,20,2F,CB,60
1180 DATA 20,2F,CB,68,20,2F,CB,70,20,2F,CB,78,20,2F,12,23
1190 DATA 1B,D,20,D8,E5,21,10,0,19,54,5D,E1,C1,D,20,C9
1200 DATA C9,C6,80,18,CD,C6,40,18,CD,C6,20,18,CD,C6,10,18
1210 DATA CD,C6,8,18,CD,C6,4,18,CD,C6,2,18,CD,3C,18,CE

```

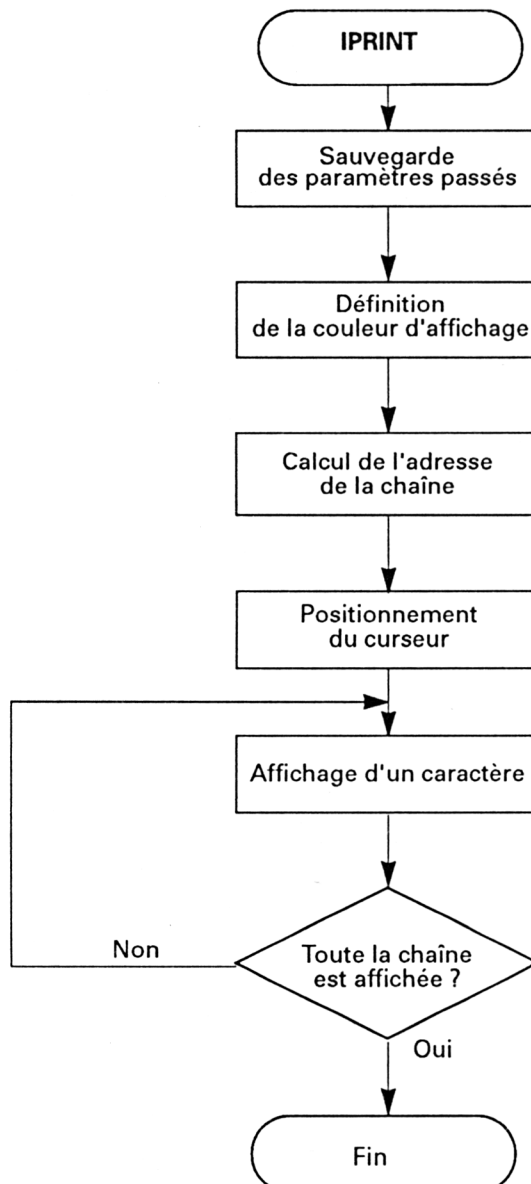
et les données de checksum qui permettent de vérifier l'exactitude des données entrées :

BB 2F BD 27 9D 9F 5

RSX d'affichage

Lorsque les caractères ont été redéfinis par le programme assembleur ou par le chargeur Basic, vous pouvez utiliser l'instruction **CHR\$** du Basic pour afficher un caractère inversé. Si vous désirez afficher une chaîne, vous devez effectuer une boucle ou bien utiliser la RSX **!IPRINT** présentée dans la suite du paragraphe.

La logique de la RSX **!IPRINT** apparaît dans l'ordinogramme suivant :



L'adresse de traitement de la RSX : **IPRINT** est... **IPRINT**. La RSX débute par la sauvegarde des données qui lui sont passées (lignes 55 à 63).

Elle initialise ensuite la couleur d'affichage à l'aide de la macro **TXT SET PEN** (lignes 70 et 71).

L'adresse passée à la RSX (soit **AD** cette adresse) pointe sur une zone mémoire dans laquelle se trouvent les données suivantes :

(**AD**) = Longueur de la chaîne en octets ;
(**AD+1**) = Partie basse de l'adresse de la chaîne ;
(**AD+2**) = Partie haute de l'adresse de la chaîne.

La longueur de la chaîne est stockée dans la variable **LEN** lignes 78 à 80.

L'adresse du début de la chaîne est ensuite stockée dans le registre **DE** lignes 81 à 84.

L'adresse du dernier caractère de la chaîne est obtenue par l'addition :

- de l'adresse du premier caractère de la chaîne ;
- de la longueur de la chaîne (lignes 85 à 88).

Le curseur est positionné en X, Y à l'aide de la macro **TXT SET CUR** (lignes 91 à 99).

Le programme se termine par une boucle (ligne 106 à 111) qui affiche les caractères de la chaîne, de droite à gauche.

Le listing du programme est le suivant :

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ; RSX IPRINT
5          ; Format : !IPRINT,@Ch,X,Y,C
6          ; Entree : Ch = chaine a afficher
7          ;          Y=Ordonnee affichage
8          ;          X=Abscisse affichage
9          ;          C=Couleur d'affichage
10         ; Sortie : Affichage du texte
11         ;-----
12         ;
13         ;
14         ;-----
15         ; Declaration des constantes
16         ; et des variables du programme
17         ;-----
18         ;
19         SETCUR:    EQU 0BB75H          ;TXT SET CURSOR
20         TXTOUT:   EQU 0BB5AH          ;TXT OUTPUT
21         LOGEXT:   EQU 0BCD1H          ;KL LOG EXT
22         SETPEN:   EQU 0BB90H          ;TXT SET FEN
23         BUF:      DS 4                ;ZONE RAM POUR LOG EXT
24 9004 0990 PTRTAB: DW TABLE          ;Pointeur TABLE
25 9006 C32090      JP IPRINT           ;Affichage inverse
26 9009 49505249 TABLE: DB "IPRIN"
26 900D 4E
27 900E D4          DB "T"+80H
28 900F 00          DB 0                ;Fin de table
29         X:       DS 1                ;Abscisse coin sup gauche
30         Y:       DS 1                ;Ordonnee coin sup gauche

```

```

31          CH:          DS    2          ;Adresse de la chaine
32          COUL:        DS    1          ;Couleur d'affichage
33          LEN:         DS    1          ;Longueur de la chaine
34          ;
35          ;-----
36          ; Definition de la RSX
37          ;-----
38          ;
39          DEFRSX:      EQU    $          ;Point d'entree
40 9016 010490          LD    BC, PTRTAB   ;Ptr table definition
41 9019 210090          LD    HL, BUF     ;Buffer pour LOG EXT
42 901C CDD1BC          CALL LOGEXT      ;Definition de la RSX
43 901F C9              RET
44          ;
45          ;-----
46          ; Traitement de IPRINT
47          ;-----
48          ;
49          IPRINT:      EQU    $          ;Point d'entree
50          ;
51          ;-----
52          ; Lecture des donnees passees
53          ;-----
54          ;
55 9020 DD7E00          LD    A, (IX+0)
56 9023 321490          LD    (COUL), A   ;Couleur de trace
57 9026 DD7E02          LD    A, (IX+2)
58 9029 321190          LD    (Y), A     ;Ordonne affichage
59 902C DD7E04          LD    A, (IX+4)

```

```

60 902F 321090          LD   (X),A           ;Abscisse affichage
61 9032 DD6607          LD   H,(IX+7)
62 9035 DD6E06          LD   L,(IX+6)
63 9038 221290          LD   (CH),HL         ;Adresse chaine
64                      ;
65                      ;-----
66                      ; Initialisation de la couleur
67                      ; d'affichage de la chaine
68                      ;-----
69                      ;
70 903B 3A1490          LD   A,(COUL)
71 903E CD90BB          CALL SETPEN          ;Couleur de trace
72                      ;
73                      ;-----
74                      ; Calcul de l'adresse reelle
75                      ; reelle de la chaine
76                      ;-----
77                      ;
78 9041 2A1290          LD   HL,(CH)
79 9044 7E              LD   A,(HL)
80 9045 321590          LD   (LEN),A         ;Longueur chaine
81 9048 23              INC  HL
82 9049 5E              LD   E,(HL)
83 904A 23              INC  HL
84 904B 56              LD   D,(HL)          ;@ reelle chaine
85 904C 2600           LD   H,0
86 904E 6F              LD   L,A
87 904F 19              ADD  HL,DE
88 9050 2B              DEC  HL              ;@ dernier caractere
89 9051 E5              PUSH HL             ;Sauvegarde
90                      ;

```

```

91 9052 3A1590      LD  A,(LEN)
92 9055 47          LD  B,A
93 9056 3A1090      LD  A,(X)
94 9059 90          SUB  B
95 905A 3C          INC  A
96 905B 67          LD  H,A
97 905C 3A1190      LD  A,(Y)
98 905F 6F          LD  L,A
99 9060 CD75BB      CALL SETCUR      ;Positionne curseur
100                ;
101 9063 E1         POP  HL      ;@ dernier caract
102 9064 3A1590      LD  A,(LEN)
103 9067 47         LD  B,A      ;Nb car a ecrire
104 9068 0E60       LD  C,96     ;Decalage caractere
105                ;
106                BIS:      EQU  $
107 906A 7E         LD  A,(HL)
108 906B 81         ADD  A,C
109 906C CD5ABB      CALL TXTOUT     ;Ecrit 1 caractere
110 906F 2B         DEC  HL
111 9070 10F8       DJNZ BIS
112                ;
113 9072 C9         RET
114                END

BUF      9000 BIS      906A CH      9012 COUL      9014
DEFRSX   9016 IPRINT  9020 LOGEXT   BCD1 LEN      9015
PTRTAB   9004 SETCUR  BB75 SETPEN   BB90 TXTOUT   BBSA
TABLE    9009 X      9010 Y      9011

```

