

ACCELERATEUR

— Frédéric SOL



Il s'agit d'un utilitaire permettant d'accélérer la vitesse d'affichage des messages à l'écran.

L'affichage sur Amstrad est en effet un peu lent, ceci étant dû en grande partie à la gestion des fenêtres texte. Aussi, ai-je réécrit une routine d'affichage beaucoup plus rapide.

Le programme a été écrit en assembleur. La routine d'affichage peut être utilisée soit dans des programmes en assembleur, soit en BASIC. Un exemple : pour afficher 20 fois le mot "BONJOUR" en BASIC (donc avec la routine d'affichage de la ROM), il faut 0,3 seconde. Avec ma routine, il ne faut plus que 0,12 seconde. On comprendra donc aisément l'intérêt d'une telle possibilité dans des programmes qui affichent en permanence des données à l'écran.

L'activation de la nouvelle routine est obtenue grâce à deux nouvelles commandes RSX. L'utilisation est quant à elle complètement automatique. Le programme est censé n'être utilisé qu'en mode 2 ; ce mode n'autorise, en effet, que 2 couleurs. Il n'y a donc pas de test de couleur de crayon comme pour le mode 0 ou 1 ; c'est là une des explications de la rapidité d'exécution. D'autre part, il ne faut pas que l'écran ait "roulé" : c'est-à-dire qu'il y ait eu écriture sur la 26^e ligne.

Une fois le programme machine chargé en mémoire à l'adresse 41000, l'activation des nouvelles commandes RSX se fait simplement par un CALL 41000. Deux nouvelles RSX sont alors créées. **RAPID** : qui permet un affichage rapide. Tout ce qui apparaîtra à l'écran sera affiché en mode rapide, sauf les commandes tapées au clavier. Les définitions de fenêtres sont quant à elles ignorées. Par exemple, si on fait PRINT "BONJOUR", le mot "BONJOUR" sera affiché en mode rapide. Le catalogue d'une disquette est lui aussi affiché sous ce mode. Pour utiliser la commande **RAPID**, il faut être sous mode 2.

NORM : permet le retour à un affichage normal.

PRINCIPE GENERAL DE FONCTIONNEMENT DE LA ROUTINE

Lorsque la commande **RAPID** est exécutée, elle détourne le vecteur d'affichage (#BB5A) vers la nouvelle routine. A ce

moment, tout caractère destiné à #BB5A (contenu dans l'accumulateur) sera utilisé par la routine. Le programme distingue :

- les caractères de contrôle (codes ASCII < 32), sauf CR (ASCII 13) et LF (ASCII 10). Si un tel caractère est rencontré, alors il est affiché par la routine de la ROM. Exemple : si l'on fait un PRINT CHR\$(7) et que l'on se trouve en mode rapide, alors on entendra bien un bip et non le caractère n° 7 qui s'affichera. Seuls les caractères 10 et 13 sont traités par la routine car ceux-ci reviennent assez souvent : une chaîne de caractères se termine en effet avec ces deux caractères ; ils sont donc traités par la nouvelle routine afin d'optimiser encore plus la vitesse.

- les autres caractères (code ASCII > 31). Dans ce cas, la routine comporte en entrée le code ASCII du caractère à afficher dans l'accumulateur. On calcule ensuite l'adresse en ROM de la définition du caractère à partir de son code ASCII. Le programme recherche ensuite les coordonnées X et Y du curseur, calcule l'adresse en mémoire vidéo correspondant à cette position et affiche le caractère à cette adresse. La routine augmente ensuite la position du curseur d'une position avec vérification et contrôle d'arrivée en fin de ligne. D'autre part, comme la configuration des adresses écran change lorsque l'on écrit sur la 26^e ligne, un test est effectué pour savoir si l'on écrit sur cette ligne. Si oui, alors la routine émet un bip invitant à taper sur une touche pour pouvoir continuer l'écriture, mais après avoir effacé l'écran. Je terminerai par un simple test d'exemple :

```
FOR I= 1 TO 100:PRINT "REVUE DES CPC";:NEXT
```

En affichage normal, le programme s'exécute en 2.46 secondes. Avec la nouvelle routine rapide, il ne met que 0.6 seconde !

SERVEUR MINITEL

N'oubliez pas 36.15 code MHZ



La B.A.L. permet un dialogue direct avec la rédaction. Les petites annonces ouvrent la porte aux bonnes affaires !



L I S T I N G

```

10 REM ----- ACCELERATEUR D'AFFICHAGE SOUS MOD >CC
E 2 -----
20 REM >ME
30 REM - Fonctions RSX : RAPID : augmente la >AY
vitesse d'affichage
40 REM - NORM : retour a la >UY
vitesse normale
50 REM >MH
70 REM ----- >TJ
-----
80 MODE 1:MEMORY 40999:l=100:LOCATE 10,10:PRINT"Afficha >RJ
ge dans"
90 c=5:FOR i=41000 TO 41450 STEP 20:LOCATE 24,10:PRINT >WN
INT(c);"s":c=c-0.2
91 s=0:FOR k=i TO i+19:READ a#:a=VAL("%"+a#):s=s+a:POKE >LH
k,a:NEXT
92 READ ctrl#:IF s<>VAL("%"+ctrl#) THEN PRINT"Erreur a >CM
la ligne ";l:END
93 l=l+10:NEXT:CALL 40000 >UD
94 INK 0,0:BORDER 0:MODE 2:IRAPID:PRINT TAB(20)"* ACCEL >BW
ERATEUR D'AFFICHAGE SOUS MODE 2 *"
95 LOCATE 22,4:PRINT"* * * Realise par Frederic Sol * * >LM
*":LOCATE 4,7:PRINT"NOUVELLES COMMANDES RSX DISPONIBLE
S:":LOCATE 10,10:PRINT"IRAPID : autorise l'affichage de
tous les messages en vitesse rapide"
96 LOCATE 10,11:PRINT"INORM : Retour de la vitesse d'a >LD
ffichage en vitesse normale"
97 LOCATE 1,20:PRINT" Attention: cette fonction nouvelle >QY
e n'est valable que sous mode 2, le mode pro. L'ecra
n ne doit pas rouler (Curseur plus bas que 25 lignes).
Si un message":PRINT TAB(15);"d'erreur apparait, mettez
-vous en MODE 2"
98 PRINT TAB(10)"Nouvelle limite haute de la memoire (H >GT
IMEM) : ";HIMEM:LOCATE 1,1:END
100 DATA 21,6D,A0,1,71,A0,CD,D1,BC,21,7C,A0,1,80,A0,CD, >RU
D1,BC,2A,5B,9D7
110 DATA BB,22,D3,A0,3A,9,0,FE,82,20,13,21,CB,B1,22,D2, >QA
A1,21,85,B2,BCD
120 DATA 22,D6,A1,21,C9,B1,22,DB,A1,C9,21,C3,B7,22,D2,A >UP
1,21,26,B7,22,9EB
130 DATA D6,A1,21,C4,B7,22,DB,A1,C9,FC,A6,71,A0,76,A0,C >UC
3,8A,A0,52,41,BC0
140 DATA 50,49,C4,0,6D,A0,50,A0,85,A0,C3,BA,A1,4E,4F,52 >RC
,CD,0,2A,D2,985
150 DATA A1,7E,FE,2,C2,64,A1,2A,DB,A1,7E,B7,C2,70,A1,23 >TG

```

```

,7E,B7,C2,70,B1B
160 DATA A1,3E,C3,32,5A,BB,21,AD,A0,22,5B,BB,C9,E5,D5,C >VF
5,F5,F3,2A,D6,BBF
170 DATA A1,FE,A,20,3,C3,53,A1,FE,D,20,A,2A,D6,A1,3E,0, >MJ
23,77,C3,7F4
180 DATA 5F,A1,FE,20,30,7,F1,C1,D1,E1,CF,0,94,2A,D2,A1, >QU
7E,FE,2,28,A5F
190 DATA 9,CD,BA,A1,CD,64,A1,C3,5F,A1,21,0,0,F1,F5,6F,2 >QB
9,29,29,11,8C8
200 DATA 0,38,19,D9,CB,91,ED,49,D9,11,DA,A1,1,8,0,ED,B0 >PY
,D9,CB,D1,A3C
210 DATA ED,49,D9,2A,D6,A1,23,7E,2B,6E,67,5C,16,0,62,D5 >TH
,54,5D,29,29,7FD
220 DATA 19,29,29,29,29,D1,19,EB,2A,DB,A1,7E,23,66,6F,1 >TM
9,7C,E6,7,67,794
230 DATA 3E,C0,84,67,1,0,8,3E,8,11,DA,A1,F5,1A,77,F1,3D >LZ
,28,4,9,6AD
240 DATA 13,18,F5,2A,D6,A1,23,7E,3C,FE,50,28,3,77,1B,F, >RP
AF,77,2B,7E,784
250 DATA 3C,FE,19,20,5,CD,C6,A1,18,1,77,F1,C1,D1,E1,C9, >RK
21,75,A1,7E,A1E
260 DATA B7,C8,CD,5A,BB,23,18,F7,21,98,A1,18,F2,4C,65,2 >UG
0,73,79,73,74,99B
270 DATA 65,6D,65,20,64,6F,69,74,20,65,74,72,65,20,73,6 >TF
F,75,73,20,6D,74E
280 DATA 6F,64,65,20,32,A,D,0,4C,27,65,63,72,61,6E,20,6 >NE
E,65,20,64,594
290 DATA 6F,69,74,20,70,61,73,20,61,76,6F,69,72,20,72,6 >RB
F,75,6C,65,A,742
300 DATA D,0,2A,D3,A0,3E,CF,32,5A,BB,22,5B,BB,C9,3E,7,C >QZ
D,D2,A0,CD,950
310 DATA 6,BB,CD,6C,BB,C9,CB,B1,0,38,85,B2,C9,B1,0,0,66 >NE
,66,66,3E,950
320 DATA 6,7C,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,82 >RZ

```

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

```

ORG 41000
ENT #
LD h1,debut1
LD bc,tabsx1
CALL #bcd1
LD h1,debut2
LD bc,tabsx2
CALL #bcd1
LD h1,(#bb5b)
LD (adrom),h1
LD a,(9)
CP #82
JR nz,c664

```




```

LD h1,#b1c8
LD (mode),h1
LD h1,#b285
LD (poscur),h1
LD h1,#b1c9
LD (roll),h1
RET
c664 LD h1,#b7c3
LD (mode),h1
LD h1,#b726
LD (poscur),h1
LD h1,#b7c4
LD (roll),h1
RET
debut1 DEFS 4
tabrsx1 DEFW nom1
JP comm1
nom1 DEFM RAPI
DEFB "D"+#80
DEFB 0
debut2 DEFS 4
tabrsx2 DEFW nom2
JP comm2
nom2 DEFM NOR
DEFB "M"+#80
DEFB 0
comm1 LD h1,(mode)
LD a,(h1)
la CP 2
JP nz,errmod
LD h1,(roll)
LD a,(h1)
OR a
JP nz,erroll
INC h1
LD a,(h1)
OR a
JP nz,erroll
LD a,195
LD (#bb5a),a
LD h1,ici
LD (#bb5b),h1
RET
ici PUSH h1
PUSH de
PUSH bc
PUSH af
DI
LD h1,(poscur)
CP 10
JR nz,pas10
JP augmlig
pas10 CP 13
JR nz,pas13
LD h1,(poscur)
LD a,0
INC h1
LD (h1),a ; colonne=0
JP teri
pas13 CP 32
JR nc,ok
POP af
POP bc
POP de
POP h1
affmes RST 08
adrom DEFW 0
ok LD h1,(mode)
LD a,(h1)
CP 2
JR z,cebon
CALL comm2 ; restaure #bb5a
CALL errmod ; affiche message
JP teri
cebon LD h1,0
POP af
PUSH af
LD l,a
ADD h1,h1
ADD h1,h1
ADD h1,h1
LD de,#3800
ADD h1,de
EXX
RES 2,c
OUT (c),c
EXX
LD de,bloc
LD bc,8
LDIR ; cherche definition du
EXX ; caractere en ROM et
SET 2,c ; le place en RAM a l'adresse
OUT (c),c ; bloc
EXX
LD h1,(poscur)
INC h1
LD a,(h1) ; colonne
DEC h1

```

```

LD l,(h1) ; ligne
LD h,a
LD e,h
LD d,0
LD h,d
PUSH de ; calcule l'adresse en memoire
LD d,h ; video de la position courante
LD e,l ; du curseur
ADD h1,h1
ADD h1,h1
ADD h1,de
ADD h1,h1
ADD h1,h1
ADD h1,h1
ADD h1,h1
POP de
ADD h1,de
EX de,h1
LD h1,(roll)
LD a,(h1)
INC h1
LD h,(h1)
LD l,a
ADD h1,de
LD a,h
AND 7
LD h,a
LD a,#c0
ADD a,h
LD h,a ; h1=adr ecran
LD bc,#800
LD a,8
LD de,bloc
lire PUSH af
LD a,(de)
LD (h1),a ; affiche le caractere
POP af
DEC a
JR z,apres
ADD h1,bc
INC de
JR lire
apres LD h1,(poscur)
INC h1
LD a,(h1)
INC a
CP 80 ; atteint fin de ligne ?
JR z,ligne ; si oui, increm No ligne
LD (h1),a
JR teri
ligne XOR a
LD (h1),a
DEC h1
augmlig LD a,(h1)
INC a
CP 25 ; 26eme ligne ?
JR nz,pas26
CALL efface ; bip et cls
pas26 LD (h1),a
teri POP af
POP bc
POP de
POP h1
RET
errmod LD h1,mess1
gloup LD a,(h1)
OR a
RET z
CALL #bb5a
INC h1
JR gloup
erroll LD h1,mess2
JR gloup
mess1 DEFM Le systeme doit etr
DEFM e sous mode 2
DEFB 10,13,0
mess2 DEFM L'ecran ne doit pa
DEFM s avoir roule
DEFB 10,13,0
comm2 LD h1,(adrom)
LD a,#cf
LD (#bb5a),a
LD (#bb5b),h1
RET
efface LD a,7
CALL affmes
CALL #bb06
CALL #bb6c
RET
mode DEFS 2
debut DEFW #3800
poscur DEFS 2
roll DEFS 2
bloc DEFS 8

```

Text:26869
Hmem:36153

End:28769

1900 Bytes