

## 9/8.1.1

### Turbo copie d'écran graphique

Nous vous proposons une autre version du programme développé en Partie 9, chapitre 8.1, dans laquelle la boucle de calcul des éléments graphiques à imprimer est effectuée en Assembleur.

Si vous avez utilisé le programme Basic de copie d'écran graphique, vous avez pu voir à quel point son exécution est lente.

La partie de programme incriminée se trouve entre les lignes 3080 et 3140. Dans ces lignes sont identifiés les pixels dont la couleur est différente de la couleur de fond. Pour une ligne élémentaire, ces pixels sont au nombre de 320. Une ligne d'impression est constituée de 7 pixels les uns au-dessous des autres comme le montre le schéma suivant :

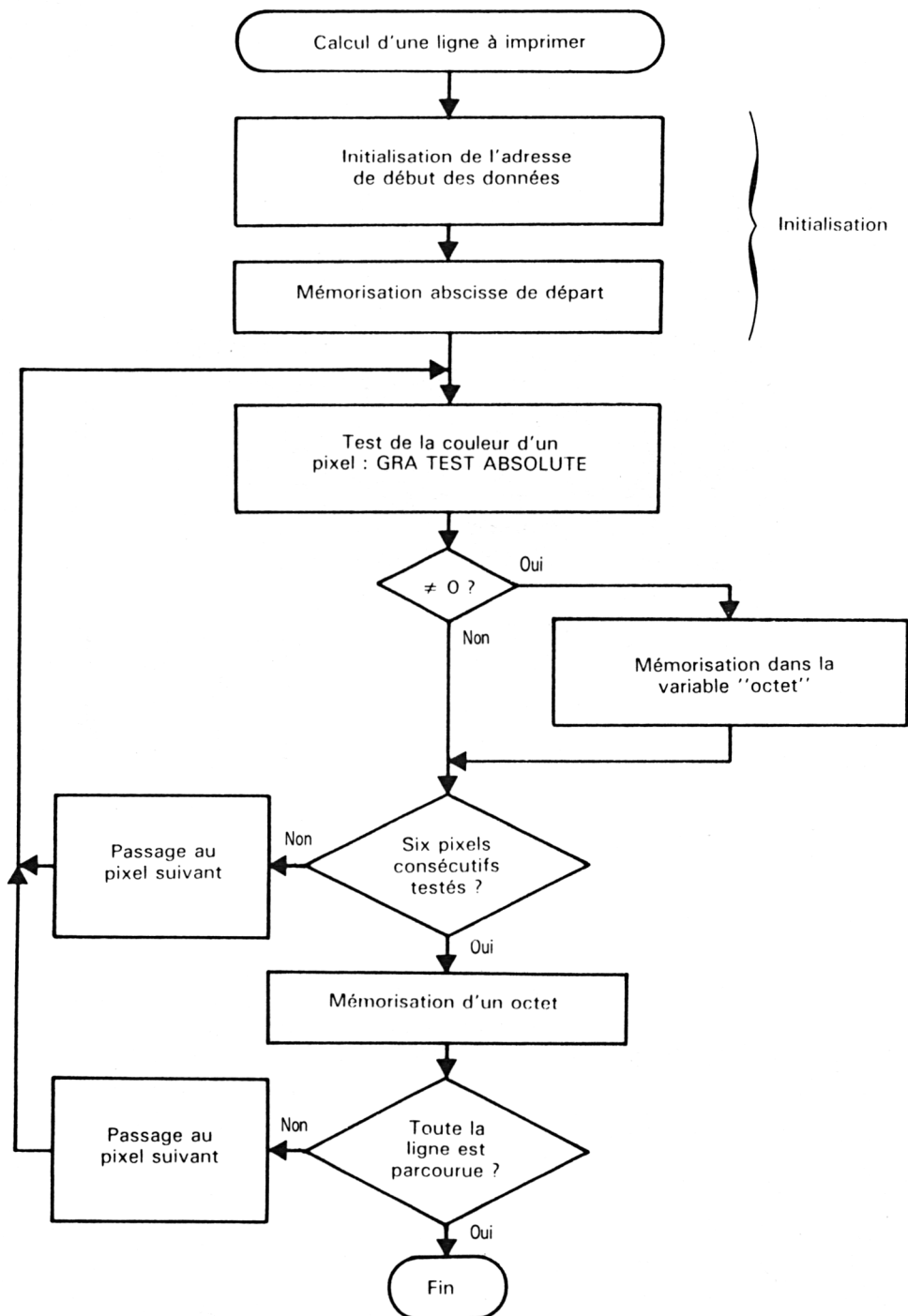
#### Octets

Poids 1 3 5 7 9 11

64						
32						
16						
8						
4						
2						
1						

Pour chaque ligne de 7 pixels, le nombre de points testés est donc de  $320 \times 7 = 2240$ . Cela explique le long temps de calcul (en Basic) entre deux lignes consécutives à imprimer. Grâce au sous-programme Assembleur qui suit, le temps de calcul pour chaque ligne est inférieur à une seconde...

La logique du sous-programme Assembleur est la suivante :



Le sous-programme Assembleur est le suivant :

```

1          ORG 9000H
2          LOAD 9000H
3 9000 216990      LD HL,ADTAB
4 9003 226590      LD (PTRTAB),HL      ;@ init tableau
5 9006 11FEFF      LD DE,OFFFEH      ;Abcisse
6          BO:     EQU $
7 9009 13          INC DE
8 900A 13          INC DE      ;X+2
9 900B 2A5C90      LD HL,(ADVE)      ;@ Variable VE
10 900E AF         XOR A
11 900F 326790     LD (POIDS),A      ;Init Poids a 0
12 9012 326890     LD (OCTET),A      ;Init Octet a 0
13          B1:     EQU $
14 9015 2B         DEC HL
15 9016 2B         DEC HL      ;Y-2
16 9017 D5         PUSH DE
17 9018 E5         PUSH HL      ;Sauvegarde registres
18 9019 CDF0BB     CALL GRATEST      ;Couleur pixel
19 901C E1         POP HL
20 901D D1         POP DE      ;Restitution registres
21 901E B7         OR A
22 901F 2816      JR Z,B2      ;Pixel a 0
23          ; Pixel a 1
24 9021 E5         PUSH HL
25 9022 D5         PUSH DE
26 9023 215E90     LD HL,ADRPOI
27 9026 3A6790     LD A,(POIDS)
28 9029 5F         LD E,A
29 902A 1600      LD D,0
30 902C 19         ADD HL,DE

```

```
31 902D 46          LD  B,(HL)          ;Poids courant
32 902E 3A6890      LD  A,(OCTET)       ;Octet courant
33 9031 B0          OR   B              ;Modification octet coura
34 9032 326890      LD  (OCTET),A
35 9035 D1          POP  DE
36 9036 E1          POP  HL
37                ;
38                B2:   EQU  $
39 9037 3A6790      LD  A,(POIDS)
40 903A FE06        CP   6
41 903C 2806        JR   Z,B3          ;1 caract complet
42 903E 3C          INC  A
43 903F 326790      LD  (POIDS),A
44 9042 18D1        JR   B1
45                ;
46                B3:   EQU  $
47 9044 E5          PUSH HL
48 9045 2A6590      LD  HL,(PTRTAB)
49 9048 3A6890      LD  A,(OCTET)
50 904B 77          LD  (HL),A          ;Memo tableau
51 904C 23          INC  HL
52 904D 226590      LD  (PTRTAB),HL
53 9050 E1          POP  HL
54 9051 7A          LD  A,D
55 9052 FE02        CP   2
56 9054 20B3        JR   NZ,B0
57 9056 79          LD  A,E
58 9057 FE80        CP   80H
59 9059 20AE        JR   NZ,B0          ;Pas fini
60 905B C9          RET
```

```

61          ;-----
62          ;Zone des EQU
63          ;-----
64          GRATEST:   EQU  OBBFOH           ;GRA TEST ABS
65          ;-----
66          ;Zone de communication
67          ;-----
68          ADVE:      DS    2               ;@ var VE
69          ADRPOI:    EQU  $               ;Poids aiguilles
70 905E 40201008      DB    64,32,16,8,4,2,1
70 9062 040201
71          PTRTAB:    DS    2               ;Pointeur ADTAB
72          POIDS:     DS    1               ;Poids courant
73          OCTET:     DS    1               ;Val octet courant
74          ADTAB:     DS    320            ;Tableau ligne
75          END

```

Pour éviter d'entrer les codes opératoires, voici un chargeur Basic qui fabrique le fichier binaire « **hcbn** » à partir de données hexadécimales définies en DATA:

```

1000 '-----
1010 ' Constitution du fichier de donnees
1020 '-----
1030 FOR I=&9000 TO &906F
1040   READ A$ 'Lecture d'une donnee
1050   A=VAL("&"+A$) 'Conversion numerique de la donnee
1060   POKE I,A 'Mise en memoire de la donnee
1070 NEXT I
1080 SAVE "hcbn",B,&9000,&70 'Sauvegarde du fichier binaire
1090 '-----
1100 ' Zone de donnees
1110 '-----
1120 DATA 21,69,90,22,65,90,11,FE,FF,13,13,2A,5C,90,AF,32
1130 DATA 67,90,32,68,90,2B,2B,D5,E5,CD,FO,BB,E1,D1,B7,28
1140 DATA 16,E5,D5,21,5E,90,3A,67,90,5F,16,0,19,44,3A,68
1150 DATA 90,B0,32,68,90,D1,E1,3A,67,90,FE,6,28,6,3C,32
1160 DATA 67,90,18,D1,E5,2A,65,90,3A,68,90,77,23,22,65,90
1170 DATA E1,7A,FE,2,20,B3,7B,FE,80,20,AE,C9,0,0,40,20
1180 DATA 10,8,4,2,1,0,0,0,0,0,0,0,0,0,0,0

```

Utilisez le programme de checksum donné en Partie 9 chapitre 8.4 pour vérifier que les lignes de DATA entrées sont bien correctes (une seule erreur peut « planter » l'ordinateur, voire endommager la disquette qui se trouve dans le lecteur...).

Les données affichées par le programme de checksum doivent être les suivantes :

**62 43 8B F3 CD 25 1F**

Le programme de copie d'écran est le suivant :

```

1000 '-----
1010 ' Hard-Copy graphique rapide DMP 2000
1020 '-----
1030 '-----
1040 ' Initialisation
1050 '-----
1060 WIDTH 255 'Pas de retour charriot automatique
1070 MEMORY &9000 : LOAD "hcbn" 'Chargement du S/P Assembleur
1080 PRINT#8,CHR$(27);"@":PRINT#8,CHR$(27);"1"
1090 ORIGIN 0,0 : ve%=401
1100 '-----
1110 ' Programme principal
1120 '-----
1130 a=@ve% : amsb=PEEK(a+1) : alsb=PEEK(a) 'Interface avec le S/P Assembleur
1140 POKE &905C,alsb : POKE &905D,amsb
1150 CALL &9000 'Calcul d'une ligne graphique
1160 GOSUB 1230 'Trace
1170 ve%=ve%-14
1180 IF ve%>0 THEN 1130 'Passage a la prochaine colonne
1190 END
1200 '-----
1210 ' Sous-programme d'impression
1220 '-----
1230 a=0:af=0:a(0)=1:a(1)=100:a(2)=100:a(3)=120
1240 FOR j=1 TO 3
1250   a=a+a(j-1):af=af+a(j)
1260   PRINT#8,CHR$(27);"K";CHR$(a(j));CHR$(0);
1270   FOR k=a TO af
1280     PRINT#8,CHR$(PEEK(&9069+k-1));
1290   NEXT k
1300 NEXT j
1310 PRINT#8
1320 RETURN

```

*Remarques :*

- ligne 1060, l'utilisation de la commande **WIDTH** permet de spécifier que les passages à la ligne sur l'imprimante ne se font pas de manière automatique.
- ligne 1070, chargement du fichier binaire « hcbn ».
- ligne 1130, calcul de la valeur de la variable **ve%**.
- ligne 1140, stockage de la valeur de **ve%** en **&905C** et **&905D** pour le sous-programme Assembleur.