

EXTRAT00L

MICRO APPLICATION vous félicite de votre choix. Vous venez d'acquérir le complément indispensable du TOOL 64, qui vous permettra d'accéder très facilement aux possibilités graphiques et sonores de votre COMMODORE 64. Avec EXTRAT00L vous pourrez aussi fusionner des programmes entre eux, imprimer l'écran graphique etc...

En tout, EXTRAT00L et TOOL 64 rajoute 87 fonctions au basic du COMMODORE 64 (dont 42 sont spécifiques à EXTRAT00L).
EXTRAT00L va vous permettre d'aborder facilement :

- le graphique haute résolution (dessin de cercle, d'ellipse, de carré)
- la gestion des sprites (animation)
- impression écran graphique haute résolution
- commandes sonores
- programmation structurée
- fusion de programmes.

Sont également joint au logiciel une démonstration et un tutoriel qui vous aideront dans votre approche d'EXTRAT00L.

Bonne Programmation

INTRODUCTION :

IMPORTANT : AVANT DE CHARGER EXTRATOOL, VOUS DEVEZ AVOIR MIS EN PLACE LA CARTOUCHE TOOL.

- Chargement : Sur cassette, faite :

LOAD '0', 1 et appuyez sur RETURN.

Puis quand READY apparaît à l'écran, tapez RUN et appuyez sur RETURN.

Sur disquette, faite :

LOAD '0', 8 et appuyez sur RETURN.

Après que READY soit apparu à l'écran, tapez RUN et appuyez sur RETURN.

EXTRATOOL va alors vous demander le nombre de sprites que vous voulez définir (pour utiliser la démonstration jointe sur la disquette répondez 4) (on a 8 sprites maximum).

Rentrez ce nombre et appuyez sur RETURN. EXTRATOOL vous demande alors si vous voulez la démonstration jointe au programme. Validez votre choix par l'appui sur la touche RETURN. Pour arrêter la démonstration, appuyez sur la touche RUN/STOP.

NOTE1: Toutes les instructions du TOOL sont opérantes.

NOTE2: Après avoir chargé la démonstration, effacer la mémoire (NEW puis RETURN) pour écrire un programme.

NOTE SUR LA DEMONSTRATION : Cette démonstration est prévue, soit pour tourner seule, soit pour répondre à vos initiatives. Un fond musical est présent lors de son déroulement ; à la fin de certains tableaux le volume sonore baisse, avant de passer à l'écran suivant. Ne vous inquiétez donc pas du léger temps d'attente entre le moment où vous "appuyez sur une touche pour continuer" et la continuation effective de la démonstration.

MIR5

MIR5 définit l'écran haute résolution

SYNTAXE : MIR5 (sans paramètre)

Dans ce mode, la définition est de 200*320 points. Chaque point est adressable indépendamment des autres.

Un point est repéré par ses coordonnées X et Y, l'origine étant en bas à gauche de l'écran

On doit avoir : $0 \leq X \leq 319$; $0 \leq Y \leq 199$.

Il est possible, dans ce mode graphique, d'avoir 2 couleurs différentes dans chaque bloc de 8*8 points, i.e. une couleur pour le fond, plus une couleur pour les points allumés.

MULTI

MULTI définit l'écran multicolore.

SYNTAXE : MULTI (sans paramètre)

Dans ce mode graphique, la définition est de 160*200 points. La résolution horizontale est donc divisée par deux par rapport au mode haute résolution normal. Chaque point est repéré par ses coordonnées X et Y, avec $0 \leq X \leq 159$ et $0 \leq Y \leq 99$.

Mais il est possible, dans ce mode graphique, d'avoir 4 couleurs différentes dans chaque bloc de 8*8 points, i.e. une couleur pour le fond, plus 3 couleurs pour les points allumés.

TEXT

TEXT revient au mode texte, qui est le mode normal du 64.

SYNTAXE : TEXT (sans paramètre)

INK

INK définit la couleur avec laquelle on écrit en mode multicolore.

SYNTAXE : INK COL

COL est le numéro de la couleur avec laquelle on dessine sur l'écran. Ce numéro est compris entre 1 et 3.

INK 1 permet d'écrire avec la couleur du 1er paramètre HICOLOR

INK 2 permet d'écrire avec la couleur du 2ème paramètre HICOLOR

INK 3 permet d'écrire avec la couleur du 1er paramètre MUCOLOR

Le 2ème paramètre MUCOLOR définit la couleur du fond.

NOTE : En mode Multicolore, on utilise à la fois l'instruction HICOLOR et MUCOLOR (cf NOTE 2 de l'instruction MUCOLOR).

HICOLOR

HICOLOR définit la couleur des points allumés ou éteints en mode graphique haute résolution. (on accède au mode haute résolution par la commande HIRÉS)

SYNTAXE : HICOLOR [COULEUR 1] , [COULEUR 2]

ou HICOLOR [COULEUR 1] , [COULEUR 2] , XO , YO , LX , LY

La première syntaxe met les points allumés en COULEUR 1 et les points éteints en COULEUR 2 sur tout l'écran.

La deuxième syntaxe met les points allumés en COULEUR 1 et les points éteints en COULEUR 2 dans une zone rectangulaire de l'écran. Le rectangle est défini par XO et YO qui sont les coordonnées du coin supérieur gauche du rectangle (entre 0 et 39 pour XO, et entre 0 et 24 pour YO), et par LX et LY qui sont respectivement les largeurs et hauteurs du rectangle. LX et LY, comme XO et YO, doivent être comprises entre 1 et 39 pour LX, et entre 1 et 24 pour LY.

NOTE 1 : Tous les points d'un même carré de 8*8 points ont obligatoirement les mêmes couleurs ; i.e. tous les points allumés de ce carré sont en COULEUR 1 et tous les points éteints en COULEUR 2.

NOTE 2 : L'origine des coordonnées pour les couleurs est en haut à gauche de l'écran.

EXEMPLE : 10 HIRÉS (passe en mode graphique)
20 HICOLOR 1,6 (met tous les points allumés en COULEUR 1 (blanc) et tous les points éteints en COULEUR 6)
30 HICOLOR 7,0 ,10 ,10 ,5,5 (met les points allumés en jaune et les points éteints en noir dans un rect).

MUCOLOR

MUCOLOR définit la couleur des points allumés ou éteints en mode multicolore

SYNTAXE : MUCOLOR [COULEUR 1] , [COULEUR 2]
ou MUCOLOR [COULEUR 1] , [COULEUR 2] , XO , YO , LX , LY
Même syntaxe que HICOLOR.

NOTE 1 : MUCOLOR met de la couleur dans un carré de 8*8 points, donc tous les points d'un carré 8*8 ont les mêmes attributs couleur.

NOTE 2 : En mode multicolore, il y a 4 paramètres couleur différents. MUCOLOR n'en donne que 2. Il est donc nécessaire d'utiliser également l'instruction HICOLOR pour fixer les 4 couleurs de chaque carré 8*8 points en mode multicolore.

NOTE 3 : Les 2 paramètres de MUCOLOR, ainsi que le premier paramètre de l'instruction MUCOLOR peuvent être différents d'un carré de 8*8 points à l'autre, i.e. un carré de 8*8 peut avoir ses 3 premières couleurs différentes des autres carrés 8*8. Par contre, le deuxième paramètre de MUCOLOR est commun à tous les points de l'écran.

EXEMPLE : 10 MULTI (passe en mode multicolore)
20 MUCOLOR 1,2 (les 2 premiers paramètres couleur sont 1 et 2 (blanc et rouge))
30 MUCOLOR 6,7 (les 2 derniers paramètres couleur sont 6 et 7 (bleu et jaune)).

SET dessine un point ou une ligne en mode HIRES ou MULTI.

SYNTAXE : 1) SET X , Y

ou 2) SET X1 , Y1 TO X2 , Y2

ou 3) SET TO X , Y

ou 4) SET X1 , Y1 TO X2 , Y2 TO X3 , Y3 TO X4 , Y4 ...etc

- 1) La première syntaxe dessine un point aux coordonnées X et Y.

EXEMPLE : SET 160,100 en mode HIRES dessine un point au centre de l'écran.

- 2) La deuxième syntaxe dessine une ligne entre les points de coordonnées X1,Y1 et X2,Y2.

EXEMPLE : SET 0,0 TO 319,199 dessine en mode HIRES une ligne diagonale sur l'écran.

- 3) La troisième syntaxe dessine une droite entre le dernier point dessiné auparavant et les coordonnées X,Y.

EXEMPLE : SET TO 160,100 dessine en mode HIRES une ligne entre le dernier point précédemment dessiné et le point de coordonnées 160,100.

- 4) La quatrième syntaxe permet de dessiner une ligne brisée continue dont les points anguleux ont pour coordonnées X1,Y1 X2,Y2 etc.

EXEMPLE : SET 0,0 TO 319,0 TO 319,199 TO 0,199 TO 0,0 dessine un rectangle au bord de l'écran.

NOTE 1 :

- Les coordonnées peuvent être données de deux façons différentes :
- en absolu, où l'on donne des valeurs de X et Y comprises entre 0 et 319 pour X et 0 et 199 pour Y.

EXEMPLE : SET 160,100

- en relatif, par rapport à la dernière position précédemment dessinée par l'instruction SET. Les paramètres ne sont plus des coordonnées absolues mais des variations de position. Elles sont exprimées en plus ou en moins.

EXEMPLE : SET +5 , +3 dessinera un nouveau point situé 3 points plus haut et 5 points plus à droite que le point précédemment dessiné.

EXEMPLE : SET -50 , +25 dessine un nouveau point situé 25 points plus haut et 50 points plus à gauche que le point précédemment dessiné.

EXEMPLE : SET +34 , 160 TO 12 , -23 : il est possible, dans la même instruction, de mélanger les deux modes, relatif et absolu.

NOTE 2 : L'instruction SET, ainsi que toutes les instructions dessinant dans l'écran graphique peuvent être utilisées en mode HIRES ou en mode MULTI. Mais en mode HIRES, la valeur de X varie de 0 à 319, alors qu'en mode MULTI, la valeur de X ne varie que de 0 à 159.

EXEMPLE : Programme traçant un cercle sur l'écran

```
10 HIRES (passe en mode graphique)
20 HICOLOR 1,2 (définit les couleurs)
30 MOVE
40 FOR A = 0 TO 6.28 STEP 0.1 (l'angle varie de 0 à 2*PI)
50 X = 160 + COS(A)*50 : Y = 100 + SIN(A)*50 (calcul des coordonnées d'un point du cercle)
60 SET TO X,Y (dessine une partie du cercle)
70 NEXT A
80 GOTO 70
```

RESET

RESET efface une ligne ou un point dans l'écran graphique.

SYNTAXE : 1) RESET X , Y

2) RESET X1,Y1 TO X2,Y2

3) RESET TO X,Y

4) RESET X1,Y1 TO X2,Y2 TO X3,Y3 ...etc

Même syntaxe et remarques que pour l'instruction SET.

FILL

FILL remplit une surface fermée sur l'écran graphique. Il suffit de préciser un point à l'intérieur de cette surface.

SYNTAXE : FILL X,Y

X et Y sont les coordonnées d'un point à l'intérieur de la surface à remplir.

EXEMPLE : A l'exemple précédent, ajouter la ligne :
75 FILL 160,100
et le cercle dessiné va se remplir.

WRITE

WRITE écrit du texte, des variables ou des caractères graphiques dans l'écran graphique.

SYNTAXE : WRITE X,Y CX,CY, "texte ou variables à écrire".

X et Y sont les coordonnées de l'endroit où va être écrit le texte. On doit avoir $0 \leq X \leq 319$ (ou 159 suivant le mode graphique) et $0 \leq Y \leq 199$.

CX et CY sont les dimensions des caractères qui vont être écrits. Une valeur de 1 donne une taille normale. La valeur 2 double la taille, etc....

CX est le facteur d'échelle en X, CY le facteur d'échelle en Y.

Si CX=2 et CY=2, alors les caractères seront deux fois plus hauts et deux fois plus larges que les caractères normaux.

Si CX=1 et CY=4, les caractères seront seulement quatre fois plus haut

SQUARE

SQUARE dessine un rectangle

SYNTAXE : SQUARE X1, Y1, LX, LY

SQUARE dessine le rectangle dont le coin supérieur gauche a pour coordonnées X1, Y1 et dont la largeur et hauteur sont respectivement LX et LY.

EXEMPLE : 10 WIRIS : HICOLOR 1,2
20 SQUARE 20,150,100,100
30 GOTO 30

SCALE

SCALE dessine un repère d'axe sur l'écran

SYNTAXE : SCALE X,Y

SCALE dessine une droite verticale et une droite horizontale traversant tout l'écran. Les deux droites se coupent au point de coordonnées X,Y.

EXEMPLE : Dessine un repère d'axe centre.

```
10 HRES : HCOLOR 1,2
20 SCALE 160,100
30 GOTO 30
```

CIRCLE

CIRCLE trace un cercle ou une ellipse.

SYNTAXE : CIRCLE X,Y, RX,RY

CIRCLE trace un cercle de centre X,Y de rayon horizontal RX, et de rayon vertical RY.

Pour tracer un cercle, il faut que les rayons RX et RY soient les mêmes.

Pour dessiner une ellipse, il suffit de spécifier des valeurs de RX et RY différentes.

EXEMPLE : trace un cercle et une ellipse.

```
10 HRES : HCOLOR 1,6
20 CIRCLE 160,100,50,50 (trace un cercle)
30 CIRCLE 100,100,50,10 (trace une ellipse)
40 GOTO 40
```

CURVE

CURVE trace un arc de cercle

SYNTAXE : CURVE X,Y, RX,RV, A1,A2

CURVE trace un arc de cercle ou d'ellipse dont le centre a pour coordonnées X,Y.

Les rayons de courbure de l'arc sont RX et RV.

A1 et A2 déterminent quelle portion du cercle va constituer l'arc.

A1 indique l'angle où va commencer l'arc, A2 indique l'angle auquel va s'arrêter l'arc.

Les angles exprimés en degré correspondent aux valeurs suivantes de A1 et A2 :

0 degré	→ A1=0
90 degrés	→ A1=10
180 degrés	→ A1=20
270 degrés	→ A1=30
360 degrés	→ A1=40

EXEMPLE : CURVE 160,100,50,0,10 trace une portion d'un cercle de centre un point de coordonnées (160,100), de rayon 50. L'arc commence à 0 (0 degré) et se termine à 10 (50 degrés). On trace donc un quart de cercle.

CURVE 160,100,10,20,0,20 trace une portion d'ellipse.
L'arc commence à 0 (0 degré) et se termine à 20 (180 degrés).
Il s'agit donc d'une demi ellipse supérieure.

NOTE : Les valeurs de A1 et A2 peuvent varier en continu de 0 à 40.

GESTION DES SPRITES.

A l'initialisation, EXTRATOOL vous a demandé combien de sprites vous voulez définir.

En fonction de ce nombre, EXTRATOOL a réservé de la place pour que le programme BASIC ne déborde pas sur les dessins des sprites.

Il ne vous sera alors pas possible de définir plus de sprites que le nombre que vous avez donné (un maximum de 8 sprites sont définissables). Vous avez la possibilité d'utiliser des sprites en mode TEXI ou MIRE.

SDATA

SDATA sert à dessiner un sprite

SYNTAXE : SDATA 88888888888888888888888888888888

SDATA doit être suivi de 24 caractères binaires. La suite de 21 instructions SDATA constitue la forme du sprite.

EXEMPLE : Dessin d'un ballon à air chaud.

```

1001 SDATA 000000000111111100000000
1002 SDATA 000000011111111110000000
1003 SDATA 000001111111111110000000
1004 SDATA 000001111000111110000000
1005 SDATA 000011101110011111100000
1006 SDATA 000011101111111111100000
1007 SDATA 000011101110011111100000
1008 SDATA 000001111000111110000000
1009 SDATA 000001111111111111100000
1010 SDATA 000001111111111111100000
1011 SDATA 000001011111111110100000
1012 SDATA 000000101111111101000000
1013 SDATA 000000010011111001000000
1014 SDATA 000000010011100100000000
1015 SDATA 000000001001110010000000
1016 SDATA 000000001001001000000000
1017 SDATA 000000000100100100000000
1018 SDATA 000000000111100000000000
1019 SDATA 000000000011110000000000
1020 SDATA 000000000111100000000000
1021 SDATA 000000000011100000000000

```

NOTE 1 : Il est nécessaire de mettre les instructions SDATA au début d'une ligne, et de ne mettre qu'elle sur la ligne.

Il faut également mettre 21 lignes d'instruction SDATA pour chaque sprite.

NOTE 2 : Les instructions SDATA sont lues par le programme comme des REM. Ces instructions ne servent que lors de l'emploi de l'instruction SCOPY.

SCOPY

SCOPY lit les instructions SOATA et les copie dans un tampon réservé à cet effet.

SYNTAXE : SCOPY NB , NL en mode TEXT
SCOPY #NB , NL en mode HIRES

NB est le numéro du bloc où va être copié le sprite. Si vous avez décidé de définir 4 sprites lors de l'initialisation, alors NB peut varier de 1 à 4.

NL est le numéro de ligne où commence les instructions SOATA.

SPRITE

SPRITE définit les paramètres pour d'un sprite.

SYNTAXE : SPRITE NS, NB, COUL, MOH, XEXP, YEXP, PRI en mode TEXT
SPRITE NS, #NB, COUL, MOH, XEXP, YEXP, PRI en mode HIRES.

NS est le numéro du sprite. Ce nombre peut varier de 1 à 8.

NB est le numéro du bloc où a été copié le sprite par l'instruction SCOPY. Si vous avez décidé de définir 4 sprites lors de l'initialisation, alors NB peut varier de 1 à 4.

COUL est la couleur du sprite.

MOH indique si le sprite est en mode multicolore ou s'il est simplement monochrome. (0=monochrome, 1=multicolore)

XEXP indique si le sprite double sa taille horizontale (0=non, 1=oui)

YEXP indique si le sprite double sa taille verticale (0=non, 1=oui)

PRI indique si le sprite est affiché devant ou derrière le décor (0=devant, 1=derrière)

SOFF

SOFF éteint un sprite de l'écran.

SYNTAXE : SOFF NS

NS est le numéro du sprite (1-8)

NOTE 1 : Le sprite sera rallumé par l'instruction PLAGE

NOTE 2 : Sans paramètre, éteint tous les sprites.

NOTE 3 : Les sprites sont différents en mode TEXT et HIRIS, ils seront éteints lors du passage d'un mode à l'autre.

PLAGE

PLAGE positionne un sprite sur l'écran

SYNTAXE : PLAGE NS, X, Y

NS est le numéro du sprite à déplacer (1-8)

X et Y sont les coordonnées de l'endroit où va être affiché le sprite.

X varie de 0 à 511, Y varie de 0 à 255.

NOTE 1 : Les coordonnées X, Y utilisées pour positionner le sprite ne sont pas les mêmes que les coordonnées de l'écran graphique. Ceci permet de faire disparaître les sprites sur les côtés de l'écran.

NOTE 2 : PLAGE rallume un sprite éteint par l'instruction SOFF.

SCOLL

SCOLL détecte une collision entre sprites. Cette instruction n'est pas une commande mais une fonction (comme peek, sin, etc....)
SCOLL renvoie 0 s'il n'y a pas de collision, et 1 s'il y a une collision.

SYNTAXE : SCOLL (NS)

NS est le numéro du sprite que l'on veut tester.

EXEMPLE : IF SCOLL (1) THEN PRINT "COLLISION" (testé si sprite 1 est entré en collision avec un autre sprite).
IF SCOLL (3) AND SCOLL (4) THEN PRINT "COLLISION" (testé si les sprites 3 et 4 sont entrés en collision).

BCOLL

BCOLL détecte une collision entre un sprite et le fond. Cette instruction n'est pas une commande mais une fonction (comme peek, sin, etc....)
BCOLL renvoie 0 s'il n'y a pas de collision, et 1 s'il y a une collision.

SYNTAXE : BCOLL (NS)

NS est le numéro du sprite que l'on veut tester.

EXEMPLE : IF BCOLL (1) THEN PRINT "COLLISION" (tester si sprite 1 est entré en collision)
IF BCOLL (3) AND BCOLL (4) THEN PRINT "COLLISION" (tester si les sprites 3 et 4 sont entrés en collision avec 1).

SOLOR

SOLOR définit les couleurs des sprites lorsqu'ils sont en mode multicolore

SYNTAXE : SOLOR COL 1, COL 2

COL 1 et COL 2 sont les 2 couleurs des sprites.

NOTE : Les couleurs sont les mêmes pour tous les sprites.

COMMANDES SONORES.

VOL.

VOL définit le volume sonore.

SYNTAXE : VOL V

V est le volume. V peut varier de 0 à 15.
(0=faible, 15=fort)

VOICE

VOICE définit les paramètres d'une voix

SYNTAXE : VOICE NV, A, D, S, R, MOD, PULS

NV est le numéro de la voix que l'on veut définir (1-3)
A, D, S, R sont les paramètres constituant l'enveloppe du son.

A : attaque
D : 777
S : soutient
R : relâchement

Ces quatre paramètres indiquent le temps que dure chacune des phases du son. Ils varient de 0 (temps nul) à 15 (à 16 sec suivent la phase)

MOD indique la modulation choisie.
MOD peut être :

- soit une lettre indiquant la modulation :

N : bruit (Noise)
P : signaux carrés (Pulse)
T : signaux triangulaires
S : signaux en dent de scie

- soit un nombre binaire à 8 bits. Chaque bit mis à 1 indique un état :

bit7 : indique que la modulation est en bruit
bit6 : indique que la modulation est en signaux carrés
bit5 : indique que la modulation est en signaux triangulaires
bit4 : indique que la modulation est en signaux dent de scie
bit3 : 77

bit2

bit1

bit0

PULS représente la longueur d'onde des signaux carrés.
PULS doit varier de 0 à 4096.

PULS ne doit être spécifié que lorsque la modulation est en signaux carrés.

EXEMPLE : VOICE 1,0,0,9,0,1

est équivalent à : VOICE 1,0,0,9,0,00100000

VOICE 1,0,0,9,0,P,256

est équivalent à : VOICE 1,0,0,9,0,01000000,256

SOUND

SOUND joue une note.

SYNTAXE : SOUND NV, NO

NV est le numéro de voix (1-3)

NO représente la note. NO peut avoir deux formes :

- un nombre entre 0 et 32767 précédé de ' '
- le numéro d'octave (1-8) et le numéro de la note (1-12) séparés par une virgule

Les numéros de notes sont codés de la manière suivante :

DO : 1
DO# : 2
RE : 3
RE# : 4
MI : 5
FA : 6
FA# : 7
SOL : 8
SOL# ou LAB : 9
LA : 10
LA# ou SIB : 11
SI : 12

EXEMPLES : SOUND 1,1,1 joue le DO de la 1ère octave sur la 1ère voix
SOUND 3,4,5 joue le MI de la 4ème octave sur la 3ème voix
SOUND 2, 0 joue un soupir sur la 2ème octave
SOUND 2, 2346 joue un son sur la 2ème octave.

NOTE : Vous devez, en préambule à l'utilisation de l'instruction SOUND avoir utilisé les instructions VOL et VOICE.

MOATA

MOATA stocke les valeurs de notes qui sont destinées à être jouées plus tard.

SYNTAXE : MOATA NO1, DU1, NO2, DU2, NO3, DU3, etc....

NO1 représente la valeur de la note qui est destinée à être jouée. Il y a 2 représentations possibles :

- un nombre précédé de '1'.
- le numéro d'octave et le numéro de note séparés par une virgule.

DU1 représente la durée de la note en soixantièmes de seconde.

NOTE 1 : Pour pouvoir jouer de la musique sur plusieurs voix, il suffit de séparer les partitions en parties distinctes, chaque partie étant jouée par une voix. Pour que EXTRA-100L distingue les parties, il faut indiquer la fin d'une partie par '2'.

NOTE 2 : Chaque instruction MOATA doit commencer au début d'une ligne. Aucune instruction ne doit être intercalée entre deux lignes de MOATA d'un même morceau. Sinon, EXTRA-100L considèrerait le morceau comme terminé.

NOTE 3 : Les instructions MOATA sont considérées par le programme comme des instructions REM. Elles ne sont lues que par l'instruction MOOPY.

MOOPY

MOOPY copie une partition en MOATA dans un tampon en fin de mémoire.

SYNTAXE : MOOPY NL

NL est le numéro de ligne où se trouve la 1ère instruction MOATA.

PLAY

PLAY joue la partition qui a été copiée dans un tampon par l'instruction COPY.

SYNTAXE : PLAY (sans paramètre)

NOTE : Lorsque EXTRA-TOOL joue une partition, il rend la main à l'utilisateur et celui-ci peut faire autre chose (par exemple exécuter un programme BASIC) pendant que la musique est jouée.

MWAIT

MWAIT attend que la musique générée par l'instruction PLAY soit terminée avant de continuer l'exécution du programme.

SYNTAXE : MWAIT (Sans paramètre)

COMMANDES PERMETTANT UNE PROGRAMMATION STRUCTUREE

REPEAT et UNTIL

REPEAT et UNTIL permettent d'exécuter une boucle jusqu'à ce qu'une condition soit vérifiée.

REPEAT indique le début de la boucle

UNTIL indique la fin de la boucle. Il sert aussi à déterminer la condition qui, suivent sa véracité, déterminera si le programme recommence la boucle ou la termine.

SYNTAXE : REPEAT : [instructions...] : UNTIL condition

Lorsque le programme lit l'instruction UNTIL, la condition sera testée. Si elle est fausse, alors le programme reprendra l'exécution au début de la boucle (symbolisé par REPEAT). Si la condition est vraie, alors le programme lit l'instruction suivent UNTIL.

EXEMPLE : 10 REPEAT
20 A=A+1
30 PRINT A
40 UNTIL A=5

La boucle située entre REPEAT et UNTIL sera effectuée autant de fois qu'il le faut jusqu'à ce que la condition (A=5) soit vraie. On aura donc sur l'écran :

1
2
3
4
5

NOTE 1 : On peut imbriquer plusieurs niveaux de REPEAT-UNTIL

EXEMPLE : 10 DIM A(4,4)
20 REPEAT
30 I=0
40 REPEAT
50 A(I,J)=1
60 I=I+1
70 UNTIL I=5
80 J=J+1
90 UNTIL J=5
100 END

NOTE 2 : Quelle que soit la condition exprimée après UNTIL, la boucle sera exécutée au moins une fois. Si l'on veut que la boucle ne soit exécutée qu'une seule fois, il faut utiliser WHILE-WEND.

WHILE et WEND

WHILE et WEND font exécuter une boucle tant qu'une condition est vraie.

WHILE indique le début de la boucle et sert à définir la condition qui, suivent sa véracité, déterminera si le programme exécute ou non les instructions constituant la boucle.

WEND indique la fin de la boucle.

SYNTAXE : WHILE condition : [instructions] : WEND

Tant que la condition est vraie, les instructions entre WHILE et WEND sont exécutées. Sinon, le programme continue son exécution après WEND.

EXEMPLE : 10 WHILE A < 5
20 PRINT A
30 A=A+1
40 WEND
50 PRINT "TERMINE"

Tant que A < 5, les lignes 20 et 30 seront exécutées. Lorsque A = 5, la ligne 50 sera exécutée.

On aura à l'exécution :

0
1
2
3
4
TERMINE

NOTE 1 : Il est possible d'imbriquer plusieurs boucles WHILE-WEND.

PROC

PROC sert à donner un nom à un sous programme.

SYNTAXE : PROC, sprog

Sprog est le nom donné au sous programme (8 caractères alphabétiques maximum)

NOTE : Après une instruction PROC, il est interdit de rajouter des instructions sur la même ligne.

ENDPROC

ENDPROC sert à indiquer la fin d'un sous programme déclaré par PROC et qui sera appelé par CALL. (à la manière d'un RETURN)

SYNTAXE : ENDPROC (sans paramètres)

CALL

Appelle un sous programme repéré par PROC.

SYNTAXE : CALL sprog

Sprog est le nom donné au sous programme que l'on veut appeler.

JMP

Branchement à un sous programme repéré par PROC.

SYNTAXE : JMP sprog

Sprog est le nom donné au sous programme.

NOTE : CALL appelle une routine à la manière de GOSUB, JMP appelle le sous programme à la manière de GOTO.

MERGE (NE FONCTIONNE PAS SUR CASSETTE)

MERGE fusionne un fichier programme existant sur disque ou cassette avec un programme déjà en mémoire.

Les lignes du programme sur disque sont insérées entre les lignes déjà existantes en mémoire. Si une ligne du programme sur disque ou cassette et une ligne du programme en mémoire ont le même numéro de ligne, la ligne en mémoire n'est pas modifiée.

SYNTAXE : MERGE np , "nprog"

Np est le numéro du périphérique sur lequel est enregistré le programme à fusionner (1 pour les cassettes, 8 pour les disques)

NPROG est le nom du programme à fusionner.

NOTE1: Pour fusionner deux programmes à des endroits précis, on peut utiliser l'instruction RENU pour renumérotter le programme à fusionner et ensuite utiliser MERGE.

NOTE2: Lors de l'emploi de l'instruction RENU du TOOL 64 les lignes où figurent des instructions de branchement telles que GOTO ou GOSUB sont modifiées en conséquence. Par contre, les lignes où figurent des instructions, spécifiques à EXTRA TOOL, avec branchement, ne sont pas modifiées.

GCOPY

GCOPY fait une recopie de l'écran haute résolution sur l'imprimante VC 1525 ou MPS 801 de numéro de périphérique 4.

SYNTAXE : GCOPY (sans paramètre)

NOTE : On peut interrompre l'impression par STOP.

STOPOFF

STOPOFF interdit l'emploi de la touche STOP ainsi que des touches STOP/RESTORE.

Attention, car il ne vous sera plus possible d'arrêter votre programme !

SYNTAXE : STOPOFF (sans paramètre)

STOPON

STOPON rétablit l'usage normal des touches STOP et RESTORE.

SYNTAXE : STOPON (sans paramètre)

TABLE DES COULEURS

TABLE DES NOTES

Couleur	CODE	NOTE	CODE
Noir	0	DO	1
Blanc	1	DO #	2
Rouge	2	RE	3
Jaune	3	RE #	4
Vert	4	MI	5
Bleu	5	FA	6
Orange	6	FA #	7
Marron	7	SOL	8
Rouge clair	8	SOL # ou LAB	9
Gris 1	9	LA	10
Gris 2	10	LA # ou SIB	11
Vert clair	11	SI	12
Bleu clair	12		
Gris 3	13		
	14		
	15		

LISTE DES INSTRUCTIONS

- Commandes Graphiques

- Hires (sans paramètre)
- Multi (sans paramètre)
- Ink Col
- Text (sans paramètre)
- Micolor [coul 1] , [coul 2] , X0 , Y0 , LX , LY
- Micolor [coul 1] , [coul 2] , X0 , Y0 , LX , LY
- Set X,Y
- Set X1 , Y1 TO X2 , Y2
- Set X1 , Y1 TO X2 , Y2 TO X3 , Y3 TO X4 , Y4 ...
- RESET X,Y
- Repeat X1 , Y1 TO X2 , Y2
- Repeat X1 , Y1 TO X2 , Y2 TO X3 , Y3 TO X4 , Y4 ...
- Fill X,Y
- Write X,Y , CX,CY , "Texte ou variable à écrire"
- Square X1,Y1 , LX,LY
- Scale X,Y
- Circle X,Y , RX,RV
- Curve X,Y , RX,RV , A1,A2

- Commandes Gestion de sprites

- SOATA BB8 BB
- SCOPY NB , NL
- SCOPY #NB , NL
- SPRITE NS , NB , COUL , MOH , XEXP , YEXP , PRI
- SPRITE NS , #NB , COUL , MOH , XEXP , YEXP , PRI
- SOFF NS

- PLACE NS , X , Y
- SCOLL (NS)
- BCOLL (NS)
- SCOLL [coul 1] , [coul 2]
- Commandes Sonores
- VOL V
- VOICE NV , A , D , S , R , MOD , PULS
- SOUND NV , NO
- MOATA NO1 , DU1 , NO2 , DU2 , NO3 , DU3 , ...
- MOOPY NL
- PLAY (sans paramètre)
- MUAIT (sans paramètre)
- Instructions Basic supplémentaires
- Repeat : [Instructions] : Until condition
- While condition : [Instructions] : Wend
- PROC SPROG
- ENDPROC (sans paramètre)
- CALL SPROG
- JUMP SPROG
- Key NT , "Texte"
- Show (sans paramètre)
- Merge np , "nprog"
- Copy (sans paramètre)
- Stopoff (sans paramètre)
- Stopon (sans paramètre)