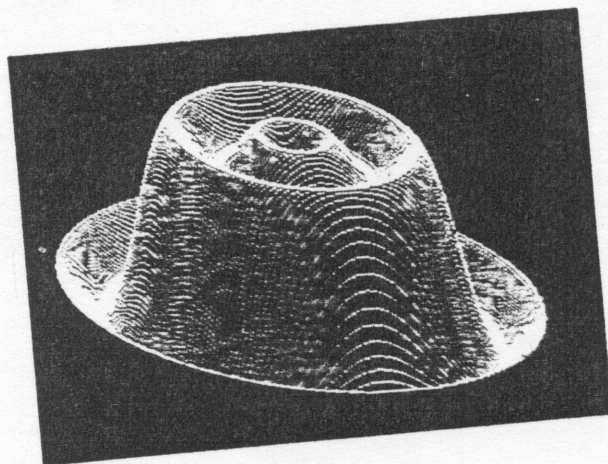


TOOL 64

Vous programmez en BASIC sur Commodore 64. Vous voulez exploiter aisément ses nombreuses capacités... TOOL est donc fait pour vous !

TOOL, c'est 35 nouvelles instructions puissantes et homogènes qui résolvent tous vos problèmes de programmation.



Caractéristiques

gestion d'écran comprenant :

saisie et affichage contrôlés, TLINE, TCOL, DECZ, CARGET, OUT (Print at)...

haute résolution graphique facile avec affichage alphanumérique :

DRAW, PLOT, MOVE, DISPLAY...

sauvegarde des pages écran (comportant le graphisme) sur disque :

SAVE, SLOAD...

instructions BASIC structurées :

IF THEN ELSE...

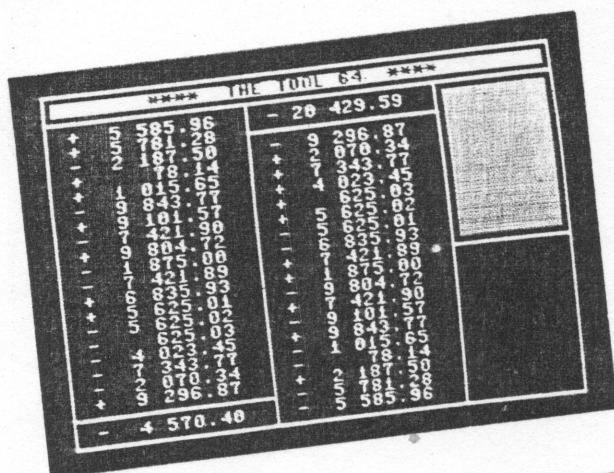
aides à la programmation

AUTO, RENU, DELETE, FIND, TRACE, HCOPY...

DOS support pour une gestion simplifiée du disque...

complément pour l'écriture des jeux en BASIC

JOY, SCROLL, SCREEN, COLOR.



Remarques :

TOOL laisse 30 Ko de RAM pour les utilisateurs et réduit la taille des programmes BASIC grâce aux nouvelles commandes.

Tous les programmes écrits avec SCREEN MASTER sur le VIC 20 sont compatibles avec TOOL sur Commodore 64.

La plupart des instructions existent dans MASTER I pour les CBM 4000, 8000 et Commodore 64, ce qui implique la portabilité de vos programmes.

TOOL est livré sous forme de cartouche, avec un manuel d'utilisation très détaillé en français.

TOOL, c'est l'outil d'une programmation simple et efficace.
TOOL est un produit développé en France.

* THE TOOL64 *

Instructions Complémentaires
au Langage BASIC pour le Commodore C64
Cartouche / Disk / *.PRG

=====

Mon Manuel en 29 double pages
Issues de la photocopie du Manuel réalisée en 1984

scan vers PDF OCR
Matériel :
-scanner EPSON 1640SU
-Logiciel VueScan - x32 (9.4.09) / Edition Professionnelle
2013 HAMRICK Software
Operating System
- Windows 7 Professionnel
Date : L18/01/2021

=====

Notes manuscrites sur la 1ère Page :
- Après chargement
Ne pas prêter attention aux messages
Quand READY est affiché, lancer par SYS64738
RESET par SYS64738

-Mes notes :
Cartouche : SYS64766 désactive TOOL
 SYS64738 réactive
 SYS64765 réactive sans perdre le programme en cours

=====

Paul-Marie BARROO
51 rue d'Hesdin
62 770 BLANGY-SUR-TERNOISE
FRANCE
paul-marie.barroo@orange.fr

Après chargement,
Ne pas perdre attention aux messages
Quand READY est affiché, lancer par SYS 64738.
RESET par SYS 64738.

Carte 1. SIM 64736 désactive tout
64738 réactive
64760 réactive le programme en cours

THE TOOL 64

1 Présentation Générale.....	1-1
2 Installation.....	2-1
3 Générateur d'Ecran.....	3-1
Présentation	3-1
Instructions d'affichage	3-1
tline	3-3
tcol	3-4
clear	3-5
out	3-6
rev	3-7
scroll	3-8
Instructions d'acquisition de données	3-9
Présentation	3-9
decz	3-10
reqz	3-12
lnz	3-13
outz	3-14
clearz	3-15
cargat	3-16
Instructions de gestion des pages écran	3-17
Présentation	3-17
asave	3-18
aload	3-19
aclear	3-20
Définition des Couleurs	3-21
screen	3-21
4 Instructions de Haute Résolution Graphique.....	4-1
Présentation	4-1
graphic	4-1
move	4-2
draw	4-3
plot	4-4
point	4-5
diaplay	4-6
color	4-7
text	4-8

5 Aide à la programmation.....	5-1
Editeur de programme	5-1
auto	5-1
delete	5-2
renu	5-3
Instructions de Mise au point	5-4
dump	5-4
error	5-5
find	5-6
Execution en Mode Pas à Pas	5-7
trace	5-7
off	5-7
6 Compléments BASIC.....	6-1
hunt	6-2
creatat	6-3
if then else	6-4
hcopy	6-5
joy	6-6
inhibition touche STOP	6-7
5 DOS Support.....	7-1

Appendices

A - Sommaire des Instructions
1 - Générateur d'Ecran
2 - Haute Résolution
Aide à la programmation
Complément BASIC
B - Messages d'erreur

THE TOOL, pour l'ordinateur individuel Commodore 64, est un puissant complément de votre système, pour permettre et faciliter le développement de programmes de qualité et tirer partie de toutes les possibilités de votre COMMODORE 64.

THE TOOL consiste en une gamme de nouvelles instructions BASIC que vous apprendrez facilement pour gérer votre écran, tracer des cadres, réaliser des routines de saisie efficaces, jouer avec les couleurs et l'écran, pour dessiner en haute résolution en superposant du texte et pour sauvegarder ou recharger vos pages écrans (en texte ou haute résolution) sur disquette.

THE TOOL comporte aussi des instructions très pratiques pour concevoir et debugger vos programmes et bien d'autres fonctions telles qu'une variable réservée pour le joystick et des instructions de manipulation de chaîne de caractères.

Les instructions du DOS SUPPORT ont été incluses afin de faciliter la gestion du drive 1541.

Ce document a été conçu pour être à la fois un manuel d'utilisation et un manuel de référence. Une lecture attentive de ce manuel vous permettra d'utiliser THE TOOL de façon efficace et rationnelle.

Afin de connaître rapidement toutes les possibilités de THE TOOL, il est conseillé de lire le manuel et de manipuler en même temps votre COMMODORE 64.

PREFACE

THE TOOL est enregistré sur une cartouche.

Pour utiliser THE TOOL, il vous suffit d'enficher la cartouche dans le connecteur d'extension mémoire situé à l'arrière du COMMODORE 64 et de mettre ce dernier sous tension. Le système s'initialisera automatiquement à la suite de cette manœuvre.

Notes :

L'aspect physique de la cartouche ne permet pas de fausses manœuvres lors de sa connexion sur le COMMODORE 64.

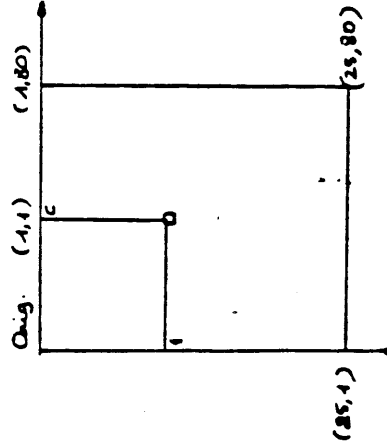
INTRODUCTION

Le générateur d'écran intégré dans THE TOOL a été étudié pour faciliter l'acquisition des données sur l'écran. La saisie en est donc plus simple qu'avec des routines BASIC et le temps d'exécution, primordial en saisie de données, est beaucoup plus rapide puisque ce générateur est écrit en langage machine.

THE TOOL contrôle les fonctions suivantes :

- commandes d'affichage pour dessiner des tableaux et effectuer des défilements (scroll) dans l'écran.
- commandes de saisie pour définir des zones de saisies, associer à ces zones des contrôles, transférer le contenu de ces zones dans des variables BASIC, etc.,...
- commandes de manipulation d'écrans comprenant des sauvegardes et des chargements de pages à partir du disque 1541.

Dans les pages qui suivent, les noms de variables choisis sont arbitraires et sont utilisés dans le texte afin d'en faciliter la compréhension. N'importe quel nom de variable peut être utilisé à l'exception de "zo" et "ok" qui sont des variables réservées.



L'écran est défini comme une table à dessin dont l'origine est en haut et à gauche.

Les coordonnées de l'origine sont 1,1. N'importe quel endroit de l'écran est donc défini par :

- l = numéro de ligne
- c = numéro de colonne

INSTRUCTION (suite)

En règle générale :

- l : numéro de ligne $(1 \leq l \leq 25)$
- c : numéro de colonne $(1 \leq c \leq 40)$
- ln : longueur (ou nombre de colonnes) - ligne horizontale
- lg : hauteur (ou nombre de lignes) - ligne verticale

Pour une meilleure compréhension de ce chapitre, il est vivement conseillé de manipuler ces instructions sur votre système, soit en mode direct soit en mode programmé.

Une Page écran est le contenu d'un écran à un moment donné. Il est donc possible de créer différentes pages écran, de les sauvegarder sur disquette, de les charger directement en mémoire et de les remplir avec des données.

MODE GRAPHIC / MODE TEXT

Nous verrons plus loin qu'il est possible d'afficher du texte en mode graphique. Les instructions graphiques seront présentées dans le chapitre suivant. Mais, pour l'instant, toutes les instructions qui suivent ne seront utilisées qu'en mode TEXT.

Nous avons donc deux modes de travail : un mode TEXT et un mode GRAPHIC. Le chapitre 3 présente les instructions du mode TEXT et le chapitre 4 présente les instructions du mode GRAPHIC. Le mode par défaut est le mode TEXT.

tline

Fonction : tline dessine sur l'écran une ligne horizontale définie par son origine et sa longueur.

Syntaxe : tline ln, l, c

avec ln : longueur de la ligne,
l : numéro de ligne de l'origine,
c : numéro de colonne de l'origine.

Utilisation : La commande tline est utilisée pour dessiner des tableaux sur l'écran ou pour souligner des titres.

Exemple : tline 40,1,1
tline 40,12,1

Ces deux instructions dessinent deux lignes parallèles dans la partie haute de l'écran.

tcol

Fonction : tcol dessine sur l'écran une ligne verticale définie par son origine et sa longueur.

Syntaxe : tcol lg, l, c

avec lg : hauteur de la ligne verticale,
l : numéro de ligne de l'origine,
c : numéro de colonne de l'origine.

Utilisation : La commande tcol est utilisée pour dessiner des tableaux avec la commande tline.

Exemple : tcol 12,1,1
tcol 12,1,40

Ces deux instructions dessinent deux colonnes parallèles dans les 12 premières lignes de l'écran.

Notes:

10 tline 40,1,1 : tline 40,12,1
20 tcol 12,1,1 : tcol 12,1,40

Ces deux lignes Basic dessine un cadre dans l'écran dont les intersections sont correctement dessinées.

clear

Fonction : clear efface une partie de l'écran.

Syntaxe : clear l, c, lg, ln

avec l : numéro de ligne de l'origine.
c : numéro de colonne de l'origine,
lg : hauteur (nombre de lignes)
ln : longueur (nombre de colonnes)

Utilisation : clear est utilisée pour effacer une fenêtre sur l'écran.

Exemples : Pour effacer la première ligne de l'écran :

clear 1,1,1,40

Pour effacer le cadre déjà dessiné ainsi que ce qu'il y a à l'intérieur du cadre :

30 clear 1,1,12,40

(efface une "fenêtre" de 12 lignes et 40 colonnes).

out

Fonction : out affiche sur l'écran un texte

Syntaxe : out a\$,l,c

avec a\$: une chaîne de caractères,
l : numéro de ligne de l'origine,
c : numéro de colonnes de l'origine.

Utilisation : out affiche un texte, contenu dans la variable a\$, dans l'écran sans avoir à utiliser les touches curseurs dans une chaîne de caractères.

Exemple

40 a\$ = "MICRO APPLICATION"

50 out a\$,5,10

Ces deux lignes afficheront MICRO APPLICATION à la cinquième ligne et à la dixième colonne.

rev

Fonction : rev change la couleur d'une fenêtre dans l'écran

Syntaxe : rev l,c,lg,ln [,co]

With : l : numéro de ligne de l'origine
c : numéro de colonne de l'origine
lg : hauteur (ou nombre de colonnes)
ln : longueur (ou nombre de lignes)
co : nouvelle couleur : facultatif
(0 à 15 - cf page 3-21)

Utilisation : Rev sans paramètre couleur permet, dans une fenêtre de l'écran, de mettre les caractères dans la couleur de l'écran et l'écran dans la couleur des caractères. Avec un paramètre couleur, la couleur du caractère est inchangé et le fonds est de la couleur spécifié.

Exemple : Pour changer les couleurs du nom de l'exemple précédent, composez la ligne suivante :

140 rev 5,10,1,5,3 : rev 5,16,1,11,4

scroll

Fonction : défilement en haut, en bas, à gauche, à droite de tout ou partie de l'écran, défini par les paramètres.

Syntaxe :

scroll l,c,lg,ln,ty
avec

l : numéro de ligne de l'origine
c : numéro de colonne de l'origine
lg : hauteur de la fenêtre (ou nombre de colonnes)
ln : longueur de la fenêtre (ou nombre de lignes)
ty : type de défilement
- u pour haut
- d pour bas
- l pour gauche
- r pour right

Utilisation : Scroll permet la consultation de tableaux de données à l'écran ; tableaux trop grands pour être affichés en entier. Par une programmation simple, il est possible de faire défiler des données sur l'écran.

Exemple :

```
10 print " "
30 a$="ERIC":b$="SERGE":c$="JULIE"
40 out a$,11,4:out b$,11,10:out c$,11,17
50 for i=0 to 9
60 scroll 10-i, 4,2,4,u
70 clear 11-1, 4,1,4
80 scroll 11+1,17,2,5,d
90 clear 11+1,17,1,5
100 for j=1 to 200:next j
110 next i
120 end
```

Tapez ce programme et regardez ce qu'il se passe. Vous pouvez exécuter ce programme sans les lignes 70 et 90 et voir la différence. Vous pouvez, de même faire des défilements latéraux.

Notes: - Un défilement n'efface pas la ligne ou colonne de départ.

GENÉRATEUR D'ECRAN Instructions d'acquisition de données

DECZ

Fonction : decz est utilisée pour définir tous les paramètres de contrôle dans la zone.

Syntaxe : decz n,l,c,ln [,ty] [,f\$]
avec
n : numéro d'identification (0 = n = 127)
l : numéro de ligne de l'origine
c : numéro de colonne de l'origine
ln : longueur totale de la z (1 = ln = 255)
ty : type de contrôle dans la zone.
avec les types :

- . ty = n type numérique,
- . ty = m type majuscules,
- . ty = r sortie autre que RETURN
- . pour numérique : artie sur toutes les touches sauf numériques,
- . pour alphanumérique : toutes les touches non alphanumériques.

. Le code ASCII de la touche enfoncée pour sortir de la zone, est stocké dans la variable zo comme par exemple :

```
f1 to f8      : zo = 133 à 140
CASR / DOWN   : zo = 17
CASR / UP      : zo = 145
SHIFT/RETURN  : zo = 141
CTRL/ A à Z    : zo = 1 à 26
```

. ty = p zone définie en printing + f\$

f\$: chaîne contenant le format du printing.

Un format est utilisé pour définir l'arrangement des données numériques dans une zone donnée. Le format est stocké dans une chaîne de caractères (f\$). Les caractères de contrôle sont :

- 9 : numérique
- 8 : numérique et zéro si nul
- 1 : position signe (seulement -)
- 2 : position signe (avec + ou -)

Nimporte quel caractère alpha peut être mis dans un format.

Ex : f\$ = "\$ 19,999,998.88"

Utilisation : decz permet de définir aussi précisément que possible une zone de saisie. Dans cette zone, les fonctions Editeur du 64 seront conservées. Decz est associé aux instructions de type2 par simple référence aux numéros de zones.

GENÉRATEUR D'ECRAN Instructions d'acquisition de données

INTRODUCTION :

Les instructions d'acquisition de données ont été étudiées pour remplacer INPUT et GET afin de rendre le saisie plus homogène, plus aisée et confortable.

Cette section développe ces instructions. Le saisie se fait donc par l'intermédiaire d'une zone à laquelle est associée un numéro d'identification ainsi qu'un certain nombre de contrôles.

Une fois la zone définie, une instruction permet d'effectuer une saisie dans cette zone. C'est à dire que le curseur peut se déplacer dans la zone et la saisie est enfin validée par l'action d'une touche de fonction. Ensuite le contenu de la zone est transféré dans une zone mémoire définie par un nom de variable. Un maximum de 128 zones peuvent être définies pour une page écran.

```
input a$ -      decz 1 : définition de la zone
reqz 1 : saisie dans la zone
inz 1,a$: transfert dans a$
```

Nous vous donnons plus de détails dans les pages suivantes.

Instructions d'acquisition de données

Instructions d'acquisition de données

decz

exemple : decz 2,1,2,10,n,p,"199 999.88"

Cet exemple définit une zone numéro 2 en ligne 1, colonne 2 et de longueur 10. Seuls des nombres seront acceptés (n). Le format est utilisé avec le forçage du align '...' et les décimales forcées à zéro si elles sont nulles. Ce nombre aura donc 5 chiffres avant la virgule et 2 chiffres après la virgule.

Notes :
 . Plus d'un type peuvent être utilisés dans la même zone.
 Toutes ces possibilités sont expliquées ci-dessous.

n	:	Nombres	-RETURN pour quitter la zone
n,r	:	Nombres	-Sortie sur touche de fonction
n,p	:	Nombres avec format	-RETURN pour quitter la zone
n,r,p	:	Nombres avec format	-Sortie sur touche de fonction
m	:	Majuscules	-RETURN pour quitter la zone
m,r	:	Majuscules	-Sortie sur touche de fonction
r	:	Tous les caractères	-Sortie sur touche de fonction
p	:	(équivalent à n,p)	
r,p	:	(équivalent à n,r,p)	
rien	:	Tous les caractères.	-RETURN pour quitter la zone

Si les données dépassent la zone, elle est mise en inversion vidéo, le contenu de la zone est effacé et le curseur revient au début de la zone : ceci afin de ne pas quitter la zone si le format n'est pas correct.

Si vous utilisez decz sans utiliser un reqz ensuite, il ne se passera rien mais la zone sera définie dans le système.

reqz

Fonction : reqz exécute la saisie d'une donnée dans une zone définie par une instruction decz.

Syntaxe : reqz n

avec n : numéro d'identification de la zone.

Utilisation : reqz est utilisé en association avec un decz.
 Si la zone n'existe pas, un message d'erreur sera généré par le système.

Avec cette commande, THE TOOL saisit une information sous les contrôles associés. L'exécution de cette commande positionne le curseur au début de la zone.

Les touches CLR, HOME, CURSOR BACK et FORWARD, INST/DEL. peuvent être utilisées dans cette zone.

Exemple : Si le précédent exemple a été composé, (decz 2,...), tapez simplement :

reqz 2

pour voir ce qui se passe. Seuls des nombres seront acceptés dans cette zone.

inz

Fonction : inz transfère le contenu d'une zone dans une variable chaîne de caractères.

Syntaxe : inz n,a\$

avec n : numéro d'identification de la zone
a\$: une chaîne de caractères.

Utilisation : inz est utilisé pour transférer le contenu d'une zone quand toutes les données de cette zone sont correctes.

Exemple : inz 2,a\$

Si vous avez essayé les deux exemples précédents, cet exemple va mettre le contenu de la zone dans a\$. Pour vérifier, tapez :

out a\$,3,5

10 decz 1,25,10,5,r
20 reqz 1:inz 1,a\$
30 print zo, a\$
40 goto 20

Notes

La chaîne de caractères reçoit le contenu de la zone sur la même longueur que la zone, même si cette dernière n'est pas entièrement remplie. La variable est complétée avec des blancs.

outz

Fonction : outz affiche une chaîne de caractères dans une zone.

Syntaxe : outz n,a\$

avec n : numéro d'identification de la zone
a\$: une chaîne de caractères

Utilisation : outz est l'inverse de l'instruction inz. Les données vont de la variable vers la zone. Cette instruction est semblable à l'instruction out mais elle est plus facile à utiliser, grâce au numéro de zone qui remplace les coordonnées.

De plus, cette instruction est plus précise grâce aux contrôles qui définissent la zone.

Exemple : Utilisons les exemples précédents :

a\$ = "11111"
outz 2,a\$

Notes :

- pour une zone définie en printusing
si il y a un débordement de la partie décimale, cette partie est tronquée.
- si il y a un débordement de capacité, la zone est remplie avec des ".....".

clearz

Fonction : clearz efface une zone à l'écran.

Syntaxe : clearz n1 [to n2]
avec n1,n2 comme numéro d'identification.Utilisation : clearz peut être utilisé pour effacer le contenu
d'un tableau sans effacer le tableau lui-même, et ceci
avant une nouvelle saisie.

Exemple : clearz 2

carget

Fonction : carget attend l'appui d'un caractère au clavier.

Syntaxe : carget a\$ [, l, c]

avec a\$: chaîne contenant les caractères permis.
l,c : (facultatif) position dans l'écran où le
curseur doit apparaître.

Utilisation : carget remplace la fameuse ligne BASIC :

100 get a\$: if a\$="" then 100

et ajoute beaucoup de caractéristiques.

100 carget "abc"

arrêtera l'exécution du programme tant que l'une des touches
a , b , c ne sera pas appuyées.Après un "carget", les variables ok et zo prennent les
valeurs suivantes.ok - pointeur de position dans la chaîne du caractère
choisi : a = ok=1, b = ok=2, c = ok=3
zo - Valeur ASCII du caractère choisi.

a = zo=65, b = zo=66, c = zo=67)

Exemple : BASIC

100 get a\$:if a\$="" then 100
110 if a\$="A" then gosub 1000:goto200
120 if a\$="B" then gosub 2000:goto200
130 if a\$="C" then gosub 3000:goto200
140 if a\$="Z" then gosub 4000:goto200
150 goto 100
200 rem :::::the program goes on:::

THE TOOL

100 carget"abcz"
110 on ok then gosub 1000,2000,3000,4000
200 rem :::::the program goes on:::

Note:

Si a\$ est vide (a\$=""), tous les caractères sont acceptés.

100 carget ""

Si vous utilisez les paramètres l et c , vous pouvez
régler la vitesse de flash du curseur en faisant un
poke 703,n ou n est la vitesse (normal = 100)

INTRODUCTION

Dans les applications de gestion, les saisies sont souvent sem- blables et sont utilisées plusieurs fois avec des informations différentes. Dans un programme BASIC normal, une nouvelle saisie est à créer à chaque fois, ce qui prend du temps et de la mémoire.

THE TOOL utilise le concept de page écran, représentant un certain masque de saisie, avec des zones associées. Ces pages- écrans peuvent être sauvegardées sur disquette, ce qui évitent aux applications d'avoir à les réécrire.

La page créée peut ensuite être chargée dans l'écran. Il est donc possible de permuer deux pages écrans, en sauvegardant celle sur écran et chargeant une page du disque.

Les 3 instructions permettent les fonctions suivantes :

- | | | |
|-------------------------------|--------|--------|
| - sauvegarde d'une page écran | ssave | 3 - 18 |
| - chargement d'une page écran | sload | 3 - 19 |
| - effacement d'une page écran | sclser | 3 - 20 |

NOTE:

Comme nous le verrons en détail dans le prochain chapitre, sload et ssave peuvent être utilisés en mode GRAPHIC et en mode TEXT. Une page écran texte peut faire 0 ou 9 blocks suivant le nombre de zones sauvegardées.

Une page écran graphique fera 37 blocks. Le type de page écran sauvegardée correspond au mode courant (page texte en mode TEXT et page graphique en mode GRAPHIC). Par contre le chargement d'une page, ne s'occupe pas du mode courant : ce qui permet de charger une page graphique dans l'écran graphique en mode TEXT et vice versa.

THE TOOL

ssave

Fonction : ssave sauvegarde une page écran sur disquette.

Syntaxe : ssave du, "name" [,n1] [,n2 to n3] [...]

avec
du : numéro du lecteur 1541 (8)
n1,n2,n3 : zones associées

Utilisation : La page écran courante sera sauvegardée. Les zones (decz) peuvent aussi être sauvegardées avec les pages texte, mais cette commande est optionnelle. Dans tous les cas, la page entière sera sauvegardée.

Exemple : ssave 8,"prg1",1,12,30 to 70

Note : 1) ssave 8,"name",0 to 127

sauvegardera toutes les zones associées.
Il n'y aura pas d'erreurs si il y a des zones non déclarées.

2) ssave 8,"@0:prg1" peut être utilisé.

3) ssave sauvegarde l'écran courant (graphique ou texte).
Dans le cas d'une page graphique, sauvegarder des zones générera une erreur de syntaxe.

sload

Fonction : sload charge en mémoire une page sauvegardée sur disque.

Syntaxe : sload du,"name"

avec
du : numéro du lecteur 1541 (8)

Utilisation : Même principe que save.

La page sera directement chargée dans la mémoire écran correspondante. Les zones seront aussi chargées en mémoire.

Exemple : sload 0,"prog1"

sclear

Fonction : sclear efface l'écran en cours.

Syntaxe : sclear

sans paramètres.

Utilisation : Cette instruction efface tout ce qui aura été affiché sur l'écran y compris le contenu des zones. Par contre les déclarations de zones ne seront pas détruites par cette instruction.

Exemple : sclear

Note : sclear efface l'écran texte en mode TEXT et l'écran graphique en mode graphique.

screen

Fonction : permet de modifier la couleur de l'écran, la couleur du bord ainsi que la couleur du caractère à afficher.

Syntaxe : screen sc,br[,cr]

avec sc : couleur de l'écran (N<=ac<15)
br : couleur du bord (0<=br<15)
cr : couleur des caractères (0<=cr<15)(optional)

Utilisation :

L'utilisation de l'instruction screen vous permet de modifier rapidement les couleurs de l'écran ainsi que la couleur des caractères affichés par l'HE IDOL (tline,tcol,out,...).

exemples :

10 rem ::: quelques couleurs nationales:::.....
20 sclear:screen 1,6,2:out"U.S.A",12,20:carget"
30 sclear:screen 1,5,2:out"ITALY",12,20:carget"
40 sclear:screen 0,2,7:out"GERMANY",11,7:carget"
50 sclear:screen 1,2,2:out"JAPAN",11,9:carget"

Note : 1) Table des couleurs:

Parametres	Couleurs	Code
Pour	Noir	0
	Blanc	1
	Rouge	2
	Turquoise	3
	Pourpre	4
	Vert	5
	Bleu	6
SC	Jaune	7
	Orange	8
	Marron	9
	Rouge Clair	10
	Gris 1	11
	Vert Clair	12
	Bleu Clair	13
BR	Gris 3	14
		15
et		
CR		

2) STOP et RESTORE redonne l'écran et le bord noir ainsi que les caractères blancs.

Présentation

Nous vous avons présenté dans le précédent chapitre la différence entre le mode TEXT et le mode GRAPHIC. Ce dernier chapitre décrirait le générateur d'écran avec la gestion des affichages, des zones et des pages écrans, qui fonctionnent avec un écran de 40 lignes par 25 colonnes correspondant dans THE TOOL au mode TEXT.

Nous allons vous présenter maintenant les instructions qui permettent de dessiner dans un écran haute-résolution de 320 points par 200 points. Cet écran est défini comme une table à dessin dont l'origine se trouve en bas et à gauche. Un point sera donc défini dans cet écran par une coordonnée horizontale et une coordonnée verticale.

graphic

Fonction : graphic indique au 64 que le mode courant est le mode graphique et qu'il doit afficher l'écran graphique.

Syntaxe : graphic

sans paramètres.

Utilisation : Cette instruction sera utilisée dès que l'on voudra visualiser l'écran graphique. Mais en mode TEXT, les instructions graphiques fonctionnent.

Exemple: 10 tline 40,1,1 : tline 40,25,1
20 tcol 25,1,1 : tcol 25,1,40
30 save 8,"frame"
40 graphic
50 rem une instruction graphique
60 save 8,"graphic"
70 rem nous sommes encore en mode graphique.

Note : Le passage d'un mode à l'autre n'efface pas le contenu des différents écrans. Chaque instruction peut-être utilisée à tout moment et fonctionnera dans l'écran correspondant.

Deux instructions fonctionnent dans les deux modes : ce sont les instructions save qui sauvegarde la page courante et clear qui efface la page courante.

move

Fonction : move déplace le curseur graphique.

Syntaxe : move x,y

avec : x - coordonnée horizontale ($0 \leq x \leq 319$)
y - coordonnée verticale ($0 \leq y \leq 199$)

Utilisation : Move permet de positionner le curseur graphique dans un point de l'écran avant un plot ou un draw. Cette instruction modifie les coordonnées du curseur et dessine rien.

Exemple: 10 graphic

20 move 0,0 : rem déplacement à l'origine.
30 i=160 : j=100 : move i,j
40 text
50 rem la suite dans les prochaines pages.
60 ...

Note : L'origine est différente entre le mode graphique et le mode TEXT. Cela est dû à l'usage en saisie de données, ainsi dans le mode TEXT, l'origine est en haut et à gauche. A l'inverse, en mode GRAPHIC, il est plus logique de mettre l'origine en bas et à gauche : similaire au papier millimétré.

draw

Fonction : draw trace ou efface une ligne depuis la coordonnée courante jusqu'aux coordonnées x,y

Syntaxe : draw x,y,ty

avec : x - coordonnée horizontale ($0 \leq x \leq 319$)
 y - coordonnée verticale ($0 \leq y \leq 199$)
 ty- une variable ayant pour valeur :
 1 : trace une ligne
 0 : efface une ligne

Utilisation : draw est utilisé chaque fois que l'on veut dessiner ou effacer un ligne droite sur l'écran. graphice.

Exemple: 05 rem ::::dessin d'un cube:::::::::::
 10 graphic:clear
 20 a=0;b=20;c=40;d=100
 20 move a,a :rem déplacement à l'origine
 30 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
 40 draw c,b,1:draw c+d,b,1:draw c+d,b+d,1
 50 draw c,b+d,1:draw a,d,1
 60 move d,a:draw d+c,b,1
 65 move c,b:draw c,b+d,1
 70 move d,d:draw d+c,d+b,1
 80 caget "" : rem attend un caractère
 90 text

Note : 1) Si vous remplacez le troisième paramètre 1 par 0, vous effacerez le cube.

2) L'instruction caget sans paramètre fonctionne aussi en mode graphique.

3) Vous pouvez aussi utiliser

draw a,d,dr

avec dr=1 - on trace
 dr=0 - on efface

plot

Fonct.-on : plot met ou éteint un point en x,y

Syntaxe : plot x,y,ty

avec : x - coordonnée horizontale ($0 \leq x \leq 319$)
 y - coordonnée verticale ($0 \leq y \leq 199$)
 ty- une variable ayant pour valeur
 1 : trace un point.
 0 : efface un point.

Utilisation : plot est utilisé pour tracer en point par point autre chose qu'une droite.

Exemple: 05 rem ::::dessin d'un cercle dans un carré:::::
 10 graphic
 20 a=0;d=100:e=50:pi=3.14159:pr=pi/12:dr=1
 20 move a,a :rem déplacement à l'origine
 30 draw a,d,dr:draw d,d,dr:draw d,a,dr:draw a,a,dr
 40 for i=0 to 2*pi step pi
 50 plot e*(1+cos(i)),e*(1+sin(i)),dr
 60 next i
 80 caget "" : rem attend un caractère
 90 text

Note : Si vous utilisez le même programme avec dr=0, vous effacerez le cercle et le carré.

point

Fonction : point teste si un point est allumé ou éteint sur l'écran graphique.

Syntaxe : point x,v,tv

avec : x - coordonnée horizontale (0 <= x <= 319)
 y - coordonnée verticale (0 <= y <= 199)
 tv - une variable qui aura pour valeur :
 1 : le point est allumé
 0 : le point est éteint

Utilisation : point est utilisé pour vérifier la présence de points dans l'écran graphique. Ceci est très utile pour la gestion des intersections.

Exemple: 05 rem :::::dessin d'un cercle dans un carré:::::
 10 graphic
 20 a=0:d=100:e=50:pl=x:pr=y/12:dim a(100)
 20 move a,a :rem déplacement à l'origine
 30 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
 40 for i=0 to 100
 50 point 25,i,a(i)
 60 next i
 80 caget "" : rem attente un caractère
 90 text
 100 for i=0 to 100: print a(i):next i
 110 rem a(0)=1 : a(100)=1 : lignes verticales

Note :

display

Fonction : display affiche une chaîne de caractères dans l'écran graphique, en utilisant les conventions du générateur d'écran (1 <= l <= 25 & 1 <= c <= 40).

Syntaxe : display a\$, l ,c

avec : a\$ - une chaîne de caractères
 l - numéro de ligne de l'origine (1 <= l <= 25)
 c - numéro de colonne de l'origine (1 <= c <= 40)

Utilisation : display vous permet d'afficher du texte ou des caractères semi-graphiques dans l'écran graphique. Il fonctionne comme l'instruction out.

Exemple: 05 rem :::::rajout d'un texte :::::::::::::::::::::
 10 graphic
 20 a=0:d=100:e=50:pl=x:pr=y/12
 20 move a,a :rem déplacement à l'origine
 30 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
 40 for i=0 to 2*pi step pi
 50 plot e*(1+cos(i)),e*(1+sin(i)),1
 60 next i
 70 display "ceci est un cercle",5,10
 75 display "-----",6,10
 80 caget "" : rem attend un caractère
 90 text

Note :

text

color

Fonction : text revient en mode texte et affiche l'écran texte.

Fonction : color est utilisé pour mettre de la couleur dans l'écran graphique par blocs de 8 x 8 points.

Syntaxe : text
sans paramètres

Syntaxe : color [-] l,c,l9,ln,co

Utilisation : permet de revenir la page texte

Exemple: 05 rem ::::idem à l'exemple précédent + ligne 70 & 75::
10 graphic
20 a=0:d=100:e=50:pl=N:pr=N/12
30 move a,a :rem déplacement à l'origine
40 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
50 for i=0 to 2*pi step pi
60 plot a*(1+cos(i)),e*(1+sin(i)),1
70 next i
70 display "ceci est un cercle",5,10
75 display "-----",6,10
80 carget " " : rem attend un caractère
85 a=ave8," 0:cercle.grf"
90 text
95 a=clear
96 tline 40,1,1:tline 40,25,1:tcol 25,1,1:tcol 25,1,40
98 a=ave8," 0:frame.txt"
100 end
11at

Note : 1) Le mode TEXT est le mode par défaut du système THE TOOL.

2) Si une instruction graphique est exécutée en mode TEXT, elle sera exécutée dans l'écran graphique et ne sera visible que dans le mode GRAPHIC.

3) Si une instruction texte est exécutée en mode graphique, elle sera exécutée dans l'écran texte.

Utilisation : color permet de définir un bloc de couleur dans l'écran graphique. Son utilisation est identique à l'instruction de REV en mode TEXT (cf page 3 - 7).

avec : l - numéro de ligne de l'origine (1 <= l <= 25)
c - numéro de colonne de l'origine (1 <= c <= 40)
l9 - hauteur (ou nombre de lignes)
ln - longueur (ou nombre de colonnes)
co - couleur (0 à 15 ~ cf page 3-21)

Exemple: 05 rem ::identique à l'exemple précédent + lignes 70 & 75
10 graphic
20 a=0:d=100:e=50:pl=N:pr=N/12
30 move a,a :rem déplacement à l'origine
40 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
50 for i=0 to 2*pi step pi
60 plot a*(1+cos(i)),e*(1+sin(i)),1
70 next i
70 color-1,1,25,40,7
75 color1,1,25,40,6
80 carget " " : rem attend un caractère
90 text

Note :

L'écran graphique est un ensemble de points qui sont allumés ou éteints. La ligne 70 sert à mettre les points éteints ou le fond de couleur jaune. La ligne 75 sert à mettre les points allumés de couleur bleue.

auto

Fonction : auto numérote automatiquement les lignes d'un programme basic.

Syntaxe : auto n

avec n : incrément (1 <= n <= 255)

Utilisation : Pour utiliser le mode de numérotation automatique des programmes, il suffit de composer la commande auto suivi d'un numéro compris entre 1 et 255 qui représente l'incrément entre deux numéros de lignes.

Ensuite, il vous suffit de composer votre première ligne de basic et dès l'appui de la touche return, le numéro de la ligne suivante s'affichera automatiquement.

Pour sortir de ce mode de numérotation, il suffit de taper la touche return dès que le numéro de la ligne BASIC est affiché.

Exemple : auto 10
10 for i = 1 to 10
20

Note : auto ne peut-être utilisé qu'en mode direct.

delete

Fonction : delete supprime des lignes de programme.

Syntaxe : delete a
delete a-
delete -a
delete a-b

avec a,b : numéros de lignes

Utilisation : delete supprime des lignes de BASIC avec la même syntaxe que la fonction list.

1) delete a supprime la ligne a

2) delete a- supprime depuis la ligne a jusqu'à la fin du programme.

3) delete -a supprime depuis le début du programme jusqu'à la ligne a

4) delete a-b supprime toutes les lignes entre a et b

Exemple : delete 20

Notes : - delete sans argument génère une 'syntax error'
- delete ne peut être utilisé qu'en mode direct.

renu

Fonction : renu renumérote les lignes des programmes.

Syntaxe : renu [a[,b[,c]]]

avec 3 arguments facultatifs

Utilisation : renu renumérote toutes les lignes d'un programme
(y compris GOTO, GOSUB, etc...)

4 possibilités :

- 1) renu
Renumérote tout le programme
La première ligne commence en 100
La valeur de l'incrément est 10
- 2) renu a
Renumérote tout le programme
La première ligne commence en 100
La valeur de l'incrément est a
- 3) renu a,b
Renumérote tout le programme
La première ligne commence en a
La valeur de l'incrément est b
- 4) renu a,b,c
Renumérote à partir de la ligne a jusqu'à la fin
Le nouveau numéro de ligne est b
La valeur de l'incrément est c

Exemple :

renu 20,10

Cette commande renumérote tout le programme
en commençant en 20 avec un incrément de 10

Note : ne peut-etre utilisé qu'en mode direct.

dump

Fonction : dump donne la liate des variables utilisées et
leurs valeurs en cours.

Syntaxe : dump

Utilisation : dump peut être utilisé :

- après l'exécution d'un programme
- après un STOP (et vous pouvez continuer avec:CONT).
- après une erreur de syntaxe.

Dump ne donne pas la valeurs des tableaux screen.

Exemple : a = 2 : b = 3 : a\$ = "test"

dump donnera :

```
a = 2
b = 3
a $="test"
```

Note : peut-être utilisé en mode direct et en mode programme.

error

Fonction : error donne l'endroit où il y a erreur dans une ligne BASIC

Syntaxe : error

sans argument

Utilisation : Quand il y a une erreur dans un programme BASIC, le programme s'arrête et apparaît un message d'erreur sur l'écran. Pour localiser cette erreur dans une lignes BASIC, il suffit de taper :

error

Error affichera la totalité de la ligne BASIC et mettra en inversion vidéo, l'endroit présumé de l'erreur.

Exemple :

```
10 a$ = STR$(a)
20 end
run
error
```

Note : ne peut être utilisé qu'en mode direct.

Aucune autre instruction ne doit être utilisée après l'arrêt du programme. (par exemple un LIST).

find

Fonction : find permet de retrouver une occurrence dans un programme

Syntaxe : find de ch de [A - B]

avec de : un caractère utilisé comme délimiteur et un " si l'on recherche dans une chaîne de caractères.

ch : les caractères que vous cherchez

A-B: (facultatif)

A : ligne de départ de recherche

B : ligne de fin de recherche

Utilisation : Recherche d'une instruction ou d'un texte dans un programme BASIC

Exemple :

```
10 ?"BLANC"
20 ?"ROUGE"
30 ?"BLEU"
40 ?"GLU"
50 END
```

FIND /"GLU"/ : rem ou FIND 0glu0

Note : peut être utilisé en mode direct ou en mode programme.

Exécution en mode pas à pas

trace
off

Fonction : trace permet d'exécuter un programme une instruction à la fois.
off arrête le mode pas - à - pas.

Syntaxe : trace sans argument
off sans argument

Utilisation : Quand le mode pas à pas est demandé, le programme exécute une instruction à la fois en affichant la ligne basic en cours en haut de l'écran

- pour afficher la ligne appuyez sur shift pour visualiser l'instruction.
- pour continuer, relâcher la touche shift
- pour arrêter le pas-à-pas du programme appuyez simultanément sur shift et run et faisant attention de relâcher d'abord la touche shift.

Exemple : En mode programme vous pouvez démarter l'exécution de votre pas-à-pas quand vous le désirez:

100 rem votre programme
110 trace
120 gosub 1000
130 off
140 rem suite de votre programme.

Note : peut être utilisé en mode direct et en mode programme.

Présentation :

Nous avons groupés dans ce chapitre, toutes les fonctions annexes qui ne dépendent pas de l'écran. Ces instructions complètent le BASIC, en facilitant sa programmation, spécialement pour les manipulations de chaînes de caractères.

Ces instructions comprennent :

- | | | |
|--------------------------------------|----------------|-------|
| - création d'un chaîne de caractères | - CREATST | 6 - 2 |
| - recherche dans une chaîne | - HUNT | 6 - 3 |
| - complément else de if ... then | - IF THEN ELSE | 6 - 4 |
| - copie écran texte sur imprimante | - HCOPI | 6 - 5 |
| - lecture des valeurs des joysticks | - JOY(n) | 6 - 6 |

Ces instructions sont toutes indépendantes et peuvent être

être

hunt

Fonction : hunt recherche la position d'un caractère dans une chaîne de caractères.

Syntax : hunt [-] chr\$(ca),a\$,pn

avec - : symbole facultatif

avec : cherche le premier caractère correspondant
sans : cherche le premier caractère différent

et

ca : valeur ASCII du caractère cherché.

a\$: nom de la chaîne de caractères.

pn : pointeur qui indique où commence la recherche dans la chaîne.

Utilisation :

hunt permet des opérations de contrôles sur des valeurs saines et aussi des opérations de test.

Exemple : a\$ = "patricia durand"

hunt chr\$(65),a\$,1:rem 65 = a

? zo

2

hunt-chr\$(69),a\$,1:rem 69 = e

?zo

1

Note : Si le caractère cherché n'existe pas, zo = 0.