New COMMANDS

NEW — Erases all memory. This is the same as a reset.

ZERO — Zeroes all single letter variables (A thru Z)

DEFAULT — Resets all device variables (character windows CF, NT etc)

DATA — Initializes variables (i.e. DATA @(5), 10, 20, 30 will give you @(5)=10 @(6)=20 @(7)=30). Very powerful, saves lots of memory. You merely place starting variable first; we could have used and starting variable including single letter variables such as A, B or C, etc.

CIRCLE — Format X, Y, RADIUS, New Mode Example CI. Ø, Ø, 2Ø, 1 (See NEW MODE later in article) will put a one pixel thick circle in the center of the screen 20 pixels wide or FOR R=1 to 20; CI. Ø, Ø, R, 1; NEXT R will put up a solid circle in the same location.

SCROLL — Format = X, Y, X size, Y size
Number of Lines (Pixels)

SNAP — Format = SNAP X, Y, Xsize, Ysize, starting string location. Imagine you had built a 10 pixel by 10 pixel character in the center of the screen - to save this entire character plus a small border around him so he could move around the screen without blinking and erasing any trail you would key in SNAP 0, 0, 12, 12, @(0). This character would then reside in strings @(0) thru @(23) [12+12=24].

SHOW — FORMAT = SHOW X, Y, Showmode, @(#) After you have SNAPPED a character, you can now clear the screen and SHOW him any where you want virtually instantaneously (in the same or different colors)!!

SHOWMODE — Ø = Plop ; 1 = OR ; 2 = XOR

NEW MODES — Ø = Nothing ; 1 = XOR 1 ; 2 = XOR 2 ; 3 = XOR 3 ; 4 = PLOP Ø ; 5 = PLOP 1 ; 6 = PLOP 2 ; 7 = PLOP 3

NEW VARIABLES

-79 XL — XL = Left boundary graphics window.

80 XR — XR = Right boundary graphics window.

51 YT — YT = Top boundary graphics window.

-50 YB — YB = Bottom boundary graphics window.

By using these four variables you can set up a graphics window that will display graphics. Even though the size of your graphic may exceed the window boundaries, the only graphics that will show on screen will be within the window you have designed.

-79 CL — CL = Left character window boundary.

80 CR — CR = Right character window boundary.

51 CT — CT = Top character window boundary.

-50 CB — CB = Bottom character window boundary.

The use of these four variables allows you to pull some extremely sharp tricks with on-screen printing: example

DEFAULT VALUES

FA = 165
FB = 91
FC = 8
BC = 7
CC = 7
NT = 1
CF = 29468 Small
CF = -25438 Large

Default Value — Setting CL = 0; CR = 4; CT = 50; CB = -50; List will allow you to list a program vertically one letter at a time instead of horizontally one line at a time with scrolling.

CF — CF = Character Font
CF = LA Large
CF = SM Small

white 7 CC = CC Character Color

Green 165 FA = Foreground Color #1
Red 91 FB = " " 2
FC = " " 3
BC = Background Color

LC = Last character printed on screen. Will return ASCII value of last character printed.

NB = Number Base
= 2 Binary
= 10 Decimal
= 16 Hexidecimal

Example: Decimal to Hexidecimal Converter Program

```
10 INPUT "?", D; NB=16; PRINT D; NB=10; GOTO 10
```

BC		Default
7	white	7
126 or 134	Yellow	
205 or 221	Cyan	
172	Green	165
43	Magenta	
82 or 90	Red	91
249	Blue	
0	Black	

CHE	Character Color		7	white
FA	Foreground Color	#1	165	Green
FB		#2	91	Red
FC		#3	8	
BC	Background Col.		7	white

"SHOW" X, Y, SHOWMODE, @ (#). This command will display the snapped image with XY as the center. The string # (@(#)) then must be the same as the snap @(#). Special showmodes are:

0 : PLOP

1 OR

2 XOR

NEW MODES !

The following grafix commands use the new mode table below:

CIRCLE,
POINT,
LINE,
BOX.

MODE:

0-nothing	4-PLOP 0
1- XOR 1	5-PLOP 1
2- XOR 2	6-PLOP 2
3- XOR 3	7-PLOP 3

The XOR mode [illegible] is used mix the color with whatever color is already on the screen. This means that if you were to use a box in mode 1 the color would be equal to the FA value and when it passed over an area of a different color the two would mixwith each other rather than erase one with the other.

The Plop mode will place the box on the screen in the color you select anderase anything that was there before.

! = Converts Hex to Decimal when placed in front of a Hex value such as, PRINT !2FF will yield 767 printed on screen.

← = Placing a ← in front of any variable except (SZ) will return with it's memory location. Such as; Print ←C will yield "-29522".

NEW VARIABLES

GRAFIX VARIABLES

XL, XR, YT, YB — left, right, top and bottom limits of the graphics window

CHARACTER VARIABLES

CL, CR, CT, CB — left, right, top and bottom limits of character window

CC — character color

FA — foreground color 1
FB — foreground color 2
FC — foreground color 3
BC — background color

LC — last character displayed on screen

NB — number base - ie. changing to 16 will print Hex, 2 binary and so on

BYTE (VARIABLE, byte #) = EXPR/SETS indicated byte (0 - low order, 1 high order) = to EXPR

C = BYTE (EXPR, byte #) returns indicated tyte of EXPR

CHANGING CHARACTER FONTS

To change from the default of SMALL 3x5 to LARGE 5x7, type the following:

CF = LARGE

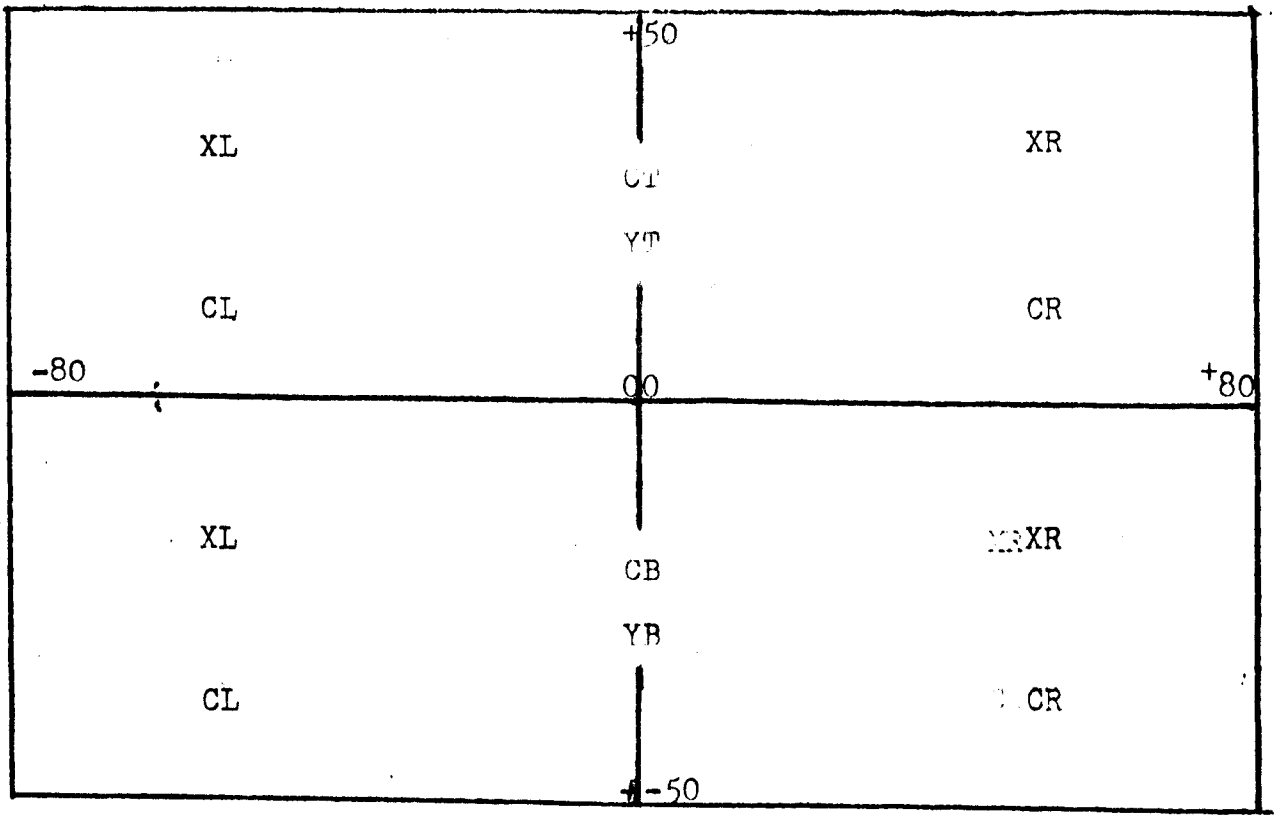
To reset to small characters, tye:

CF = SMALL

HEX # INPUT

By putting a ! in front of a hexidecimal number, it becomes decimal. for example:

PRINT !2FF will type: 767, the decimal equivalent.



GRAFIX WINDOW SCREEN LOCATIONS, XL, XR, YT, YB.

CHARACTER WINDOW SCREEN LOCATIONS AT, CL, CR, CT, CB.

ADDRESS OPERATOR

Putting a ← in front of a variable, will give its address. for example:

PRINT ← A will print -17238, which is the memory location where A's value would be stored.

TOKENS

The following control characters type the corresponding tokens.

A	RND (AMBIGUITY)
B	BOX
C	CLEAR
D	DATA
E	(edit key for line editor)
F	FOR
G	GOSUB
H	RUBOUT or ERASE
I	INPUT
J	GOTO (jump)
K	IF (near I)
L	LIST
M	GO
N	NEXT
O	CIRCLE (there's a circle on the key)
P	PRINT
Q	SNAP (near SHOW on keyboard)
R	RETURN
S	STEP
T	TO
U	POINT (UPDATE A POINT)
V	DEFAULT (VARIABLE DEFAULT)
W	SHOW (WRITE TO SCREEN)
X	RUN (X FOR EXECUTE)
Y	SCROLL (WHO KNOWS)
Z	ZERO

EXAMPLE: To abbreviate PRINT type PR. PRI. or just P.