

EXBASIC

NABITCHI



EXBASIC

BASIC EXTENSION PROGRAMME

REAL GRAPHICS FOR THE AMSTRAD
PCW8256/8512

MERSEYSIDE INNOVATION CENTRE
131 MOUNT PLEASANT
LIVERPOOL L3 5TF
051-708-0123/708-8775

INTRODUCTION

The extra functions included in EXBASIC are drawn from the following criteria:

1. Facilities previously unavailable from BASIC.
2. Facilities that could be programmed from BASIC but would be too slow, inefficient and difficult for the novice programmer.
3. Facilities available to BASIC that needed to be made easier to use

EXBASIC with all its extra facilities takes up only 4k of your computer memory. This memory usage is usually made up for by you now being able to write better, more compact and easier to read programs. On top of this you also have the ability to use facilities previously unavailable from BASIC such as graphics, without having to try and understand GSX !!

STARTING EXBASIC

EXBASIC is started from CP/M in a similar manner to BASIC by typing:-

EXBASIC and pressing RETURN.

This program will load into your computer and will automatically load BASIC in also.

If BASIC is not present on the default disk drive, a menu will appear to prompt you for a choice of disk drive.

Note: Standard BASIC rules still apply.

Eg: EXBASIC A:MYPROG.BAS

will load EXBASIC/MALLARD BASIC and run "MYPROG" from BASIC.

PROGRAMMING WITH EXBASIC

To use an EXBASIC function, the function name must be printed via a BASIC PRINT statement followed by any necessary parameters. The command itself may thus be regarded as text and may therefore be stored in a string variable.

For instance the simple draw a circle function would be:

```
PRINT "iCIRCLE.",x-location,y-location,radius;
```

EXBASIC FUNCTION SYNTAX

There are several important things to notice about that last function which applies to all EXBASIC functions. These are detailed as follows:

PARAMETERS

These are numbers supplied to a function to direct it in performing a task. In the case of CIRCLE these are the x and y (column and row) location on the screen of the centre of the circle and the radius. There may be no parameters or many but where one is required it must be given otherwise an error could result.

Parameters are always numeric and so may be stored in variables.

The following forms are both equivalent:

```
10 X=360
20 Y=180
30 R=150
40 PRINT "CIRCLE.",X,Y,R;
```

and

```
10 PRINT "CIRCLE.",360,180,150;
```

Parameter numbers must be positive integers and so the following are invalid:

```
10 PRINT "DRAW.",-10,5;
20 PRINT "DRAW.",1,3,1,2E2;
```

RANGES

All parameters have ranges within which they must lie and are detailed individually for each command later in this document.

In general these are:

```
GRAPHICS      X = 0-719
               Y = 0-255
```

```
TEXT (32x90)  X = 0-89
               Y = 0-31
(24x80)       X = 0-79
               Y = 0-23
```

Most other parameters have ranges of 0-65535 except where otherwise specified.

SYNTAX OF PARAMETERS

Parameters must be separated by commas or semicolons except when in textual form.

TEXT FORM

A parameter list may be stored as text so long as there is a space character between each number. The following forms are valid and equivalent:


```
10 PRINT "iCIRCLE, 360 100 150 ";
```

or

```
10 A$=" 360 100 150 "  
20 PRINT "iCIRCLE.",A$;
```

FUNCTIONS

All EXBASIC functions start with an i character; an upside down exclamation (EXTRA I), and end with a full stop.

Leaving off the i character will cause EXBASIC to regard the command as text and so will ignore it and print it on the screen.

Leaving off the . character will cause an error and will abort program execution.

GENERAL SYNTAX

All functions may appear in multi-line statements with other commands. They may also be in textual form and so be concatenated or stored as string variables.

Eg:

```
10 A$="iCLS,":PRINT A$;  
20 A$=A$+"iCIRCLE."  
30 INPUT X,Y,R  
40 PRINT A$,X,Y,R;  
50 GOTO 30
```

An EXBASIC print line must usually end with a semicolon to maintain the cursor position.

Printing an EXBASIC function may be split over several lines so long as no irrelevant printing or input occurs between.

Eg:

```
10 PRINT "iDRAW,";  
20 X=10:Y=20  
30 PRINT X,Y;
```

The above program would draw to locations 10,20 on the screen. The following is also valid:

```
10 PRINT "i";  
20 PRINT "C";  
30 PRINT "L";  
40 PRINT "S";  
50 PRINT ".";
```

EXTRA COMMAND EASY FINDER

GRAPHICS : STYLE	iNORMAL, NORMAL GRAPHICS MODE	5
	iINVERT, INVERT GRAPHICS MODE	5
	iRUBBER, ERASE GRAPHICS MODE	6
GRAPHICS : DISK	iSAVE, SAVE SCREEN TO DISK	7
	iLOAD, LOAD SCREEN FROM DISK	7
GRAPHICS : DRAWING	iMOVE, MOVE GRAPHICS CURSOR	8
	iPLOT, PLOT A PIXEL	8
	iDRAW, DRAW A LINE	9
	iCIRCLE, DRAW A CIRCLE	9
TEXT : CURSOR MOVEMENT	iUP, MOVE TEXT CURSOR UP	10
	iDOWN, MOVE TEXT CURSOR DOWN	10
	iLEFT, MOVE TEXT CURSOR LEFT	11
	iRIGHT, MOVE TEXT CURSOR RIGHT	11
TEXT : CURSOR LOCATION	iAT, LOCATE CURSOR AT X,Y COORDINATES	12
	iHOME, LOCATE CURSOR AT TOP LEFT OF SCREEN	12
TEXT : CURSOR CONTROL	iSCP, SAVE CURSOR POSITION	13
	iRCP, RESTORE CURSOR POSITION	13
	i-CUR, CURSOR BLOB OFF	14
	i+CUR, CURSOR BLOB ON	14
TEXT : STYLE	i-UL, UNDERLINE OFF	15
	i+UL, UNDERLINE ON	15
	i+REV, REVERSE VIDEO ON	16
	i-REV, REVERSE VIDEO OFF	16
SCREEN : STYLE	i32*90, SELECT 32*90 SCREEN SIZE	17
	i24*80, SELECT 24*80 SCREEN SIZE	17
	i-PAPER, BACKGROUND GREEN, TEXT BLACK	18
	i+PAPER, BACKGROUND BLACK, TEXT GREEN	18
SCREEN : CONTROL	iCLS, CLEAR SCREEN AND HOME	19
	iCLS2, CLEAR SCREEN AND PROTECT CURSOR	19
	i-SL, DISABLE STATUS LINE	20
	i+SL, ENABLE STATUS LINE	20
	i+STORE, STORE SCREEN AREA	21
	i-STORE, RESTORE SCREEN AREA	21
	iPOINT, RETURN BYTE FROM SCREEN	22
SCREEN : PRINTING	iLDUMP, COPY SCREEN TO PRINTER	22
MISCELLANEOUS :	iPAUSE, PAUSE FOR A DELAY OR KEY	23
	iBEEP, SOUND A BEEP	23
	ERROR MESSAGES	24
	EXBASIC ON/OFF	24

!NORMAL

GRAPHICS : STYLE

Reset graphic style to normal.

EXTRA FUNCTION

Use

Returns the graphics line style to the normal default of overwriting screen locations with green pixels for !PLOT, !DRAW, and !CIRCLE. The normal graphics style is the complete opposite to the style entered using !RUBBER.

Form

PRINT "!NORMAL,";

Notes

This command has no effect when the graphics style is already normal

Associated Keywords

!INVERT, !RUBBER, !POINT.

!INVERT

GRAPHICS : STYLE

Sets the graphic style to invert.

EXTRA FUNCTION

Use

With this style in effect !PLOT, !DRAW, and !CIRCLE, commands use an inverting style. Pixels already set on the screen are erased and the blank areas of screen are set with green pixels. Thus drawing with this style creates a 'negative' of what is already on the screen.

Form

PRINT "!INVERT,";

Notes

This is useful in graphics programs where a moving object is needed which does not destroy the information displayed on the screen.

Associated Keywords

!NORMAL, !RUBBER, !POINT.

Sets the graphic style to erase.

EXTRA FUNCTION

Use

With this style in effect iPLOT, iDRAW, and iCIRCLE, functions use blank pixels and so erase set or unset pixels on the screen.

Form

```
PRINT "iRUBBER,";
```

Notes

It is important to remember when this style is in effect, as often nothing appears to happen when drawing over a blank area of the screen. To draw after erasing something, first set one of the other graphic styles.

Associated Keywords

```
iINVERT, iNORMAL, iPOINT,
```


!SAVE.

GRAPHICS : DISK

Save screen to disk.

EXTRA FUNCTION

Use

To copy the screen display to disk.

Form

PRINT "!SAVE.(file-name-expression)";

The file-name-expression gives the name of the file which will contain the screen display. If no disk identifier is used, the default disk drive is assumed. If no file extension is used, .PIC is assumed. If a file of that name already exists it is deleted.

Notes

A screen saved on the disk may contain characters and graphics and occupies about 23k maximum. This feature may be used to save screen pictures for use in rolling picture demonstrations, drawing programs, or where program space is scarce - a screen display need only be generated once by a program and then displayed by use of this one small command in the program where space is restricted.

Normal FILE-NAME rules apply with the exception of wild-cards.
Brackets must be around file name or it will cause an error.
Associated Keywords

!LOAD.

!LOAD.

GRAPHICS : DISK

Load screen from disk.

EXTRA FUNCTION

Use

To read a previously saved screen file from disk onto the display.

Form

PRINT "!LOAD.(file-name-extension)";

The file-name-expression gives the name of the file containing the screen display on the disk to be loaded onto the screen. If no disk identifier is used, the default disk drive is assumed. If no file extension is used, .PIC is assumed.

Notes

Loading a screen from disk will overwrite all information currently on the screen.
Normal FILE-NAME rules apply with the exception of wild-cards.
Brackets must be around file name or it will cause an error.
Associated Keywords

!SAVE.

iMOVE

GRAPHICS : DRAWING

Move graphics cursor.

Extra Function

Use

This command moves the graphics cursor to the given x and y location absolute, in preparation for a iDRAW.

Form

PRINT "iMOVE.",x-location,y-location;

x-location is a positive integer in the form of a number or variable within the range of 0-719.

y-location is a positive integer in the form of a number or variable within the range of 0-255.

Associated Keywords

iPLOT, iDRAW, iCIRCLE.

iPLOT

GRAPHICS : DRAWING

Plots a pixel on the screen.

EXTRA FUNCTION

Use

This function places a single pixel onto the screen in the selected graphics style at the indicated x and y location in absolute form.

Form

PRINT "iPLOT.",x-location,y-location;

x-location is a positive integer in the form of a number or variable within the range of 0-719.

y-location is a positive integer in the form of a number or variable within the range of 0-255.

Notes

The graphics style may be selected by using iNORMAL, iINVER, or iRUBBER, with the default being NORMAL style.

The graphics cursor is set to new plotted coordinates

Associated Keywords

iDRAW, iCIRCLE, iMOVE.

!DRAW

GRAPHICS : DRAWING

Draw a line of pixels on the screen.

EXTRA FUNCTION

Use

This function places a line of pixels onto the screen in the selected graphics style from the graphics cursor to the supplied x and y coordinates in absolute form.

Form

PRINT "!DRAW",x-location,y-location;

X-location is a positive integer in the form of a number or variable within the range of 0-719 and is the end of draw coordinate.

Y-location is a positive integer in the form of a number or variable within the range of 0-255 and is the end of draw coordinate.

Graphics cursor is set to the last drawn coordinate.

Associated Keywords

!PLOT, !CIRCLE, !MOVE.

!CIRCLE

GRAPHICS : DRAWING

Draws a circle on screen.

EXTRA FUNCTION

Use

Draws a circle in the currently selected graphics style at the given x and y coordinates to a given radius.

Form

PRINT "!CIRCLE",x-location,y-location,radius;

X-location is a positive integer in the form of a number or variable in the range of 0-719.

Y-location is a positive integer in the form of a number or variable in the range of 0-255.

Radius is a positive integer in the form of a number or variable in the range of 1-255. Numbers greater than 255 have 255 subtracted repeatedly until the result is <256 which then becomes the RADIUS.

Notes

The graphics style may be selected by using !NORMAL, !INVERT or !RUBBER, with the default being NORMAL style.

The graphics cursor is set to the centre of the circle.

Drawing a circle beyond the boundaries of the screen may give unusual results.

Associated Keywords

!PLOT, !DRAW, !MOVE.

iUP

TEXT : CURSOR MOVEMENT

Cursor Up,

EXTRA FUNCTION

Use

This command moves the cursor up one line.

Form

PRINT "iUP";

Notes

If the command is not followed by other characters to display then the print expression must be followed by a semicolon.

This directly replaces use of the character code sequence 27 "A" to move the cursor up.

Associated Keywords

iDOWN, iLEFT, iRIGHT, iAT,

iDOWN

TEXT : CURSOR MOVEMENT

Cursor Down,

EXTRA FUNCTION

Use

This command moves the cursor down one line.

Form

PRINT "iDOWN,";

Notes

Using this command on the bottom line of the screen causes an upward scroll.

If the command is not followed by other characters to display then the print expression must be followed by a semicolon.

This directly replaces use of the character sequence 27 "B" to move the cursor DOWN.

Associated Keywords

iUP, iLEFT, iRIGHT, iAT,

iLEFT.

TEXT : CURSOR MOVEMENT

Cursor left,

EXTRA FUNCTION,

Use

This command moves the cursor left one character position.

Form

PRINT "iLEFT,";

Notes

Using this command on the extreme LEFT column of the screen will have no effect on the cursor position.

If the command is not followed by other characters to display then the print expression must be followed by a semicolon.

This directly replaces use of the character sequence 27 "D" to move the cursor LEFT.

Associated Keywords

iUP, iDOWN, iRIGHT, iAT

iRIGHT.

TEXT : CURSOR MOVEMENT

Cursor right,

EXTRA FUNCTION,

Use

This command moves the cursor right one character position.

Form

PRINT "iRIGHT,";

Notes

Using this command on the extreme RIGHT column of the screen will have no effect on the cursor position.

If the command is not followed by other characters to display then the print expression must be followed by a semicolon.

This directly replaces use of the character sequence 27 "C" to move the cursor RIGHT.

iAT

TEXT : CURSOR LOCATION

Move cursor to a given screen position.

EXTRA FUNCTION

Use

In a print command iAT, moves the cursor to a given print position.

Form

PRINT "iAT.",row,column;

Row and Column are integer expressions within the width and height (in lines) of the screen.

X-position and Y-position may be variables or integers.

Associated Keywords

iUP, iDOWN, iLEFT, iRIGHT,

iHOME

TEXT : CURSOR LOCATION

Move cursor home.

EXTRA FUNCTION

Use

Moves the cursor home to the top left corner of the screen.

Form

PRINT "iHOME,";

Notes

This command directly replaces the character code sequence 27 "H" to home the cursor.

Associated Keywords

iCLS2.

iSCP.

TEXT : CURSOR CONTROL

Store current position.

EXTRA FUNCTION

Use

Stores the cursor location in memory for returning to later.

Form

PRINT "iSCP.;"

Notes

The stored location is not accessible to BASIC variables and is only used in conjunction with the RETURN to CURSOR POSITION function.

Replaces use of the control code sequence 27 "j".

Associated Keywords

iRCP.

iRCP.

TEXT : CURSOR CONTROL

Restore cursor position.

EXTRA FUNCTION

Use

Returns cursor to the location previously saved by iSCP.

Form

PRINT "iRCP.;"

Notes

This command is ineffective unless the cursor position has been previously saved using the iSCP function.

Replaces use of the control code sequence 27 "k"

Associated Keywords

iSCP.

i-CUR.

TEXT : CURSOR CONTROL

Disable cursor blob.

EXTRA FUNCTION

Use

Removes the cursor blob from the display although the text cursor is still present in invisible form.

Form

PRINT "i-CUR.";

Notes

Directly replaces use of the control code sequence 27 "f".

This is a useful function when screen dumping as the cursor also appears in the printout and this is rarely desirable when printing a picture.

Associated Keywords

i+CUR.

i+CUR.

TEXT : CURSOR CONTROL

Enable cursor blob.

EXTRA FUNCTION

Use

Returns the cursor to normal display after it has been disabled with i-CUR.

Form

PRINT "i+CUR.";

Notes

This function has no effect unless the cursor is first disabled with i-CUR. Directly replaces use of the control code sequence 27 "e".

Associated Keywords

i-CUR.

i-UL.

TEXT : STYLE

Underline off.

EXTRA FUNCTION

Use

Reverts text printing to normal non-underlined after underline has previously been set to on.

Form

PRINT "i-UL.";

Notes

This function replaces the use of the control code sequence 27 "u".

Associated Keywords

i+UL.

i+UL.

TEXT : STYLE

Underline On.

EXTRA FUNCTION

Use

Selects a text printing mode where all text is underlined.

Form

PRINT "i+UL.";

Notes

The effect of this function is reversed using i-UL and replaces the use of the control code sequence 27 "r".

Associated Keywords

i-UL.

i+REV.

TEXT : STYLE

Reverse Video On.

EXTRA FUNCTION

Use

Selects a printing mode with all text in reverse video (Black on Green).

Form

PRINT "i+REV.";

Notes

The effect of this function is reversed using i-REV and replaces the use of the control code sequence 27 "p".

Associated Keywords

i-REV.

i-REV.

TEXT : STYLE

Reverse Video Off

EXTRA FUNCTION

Use

Reverts text printing mode to normal where all text in reverse video (Black on Green) has been set to on.

Form

PRINT "i-REV.";

Notes

This function replaces the use of the control code sequence 27 "q".

Associated Keywords

i-REV.

32x90

SCREEN : STYLE

Selects 32x90 character mode.

EXTRA FUNCTION

Use

Selects the available screen size in rows and columns to the maximum available (Default used by BASIC).

Form

PRINT "32x90,";

Notes

This command also clears the screen of all information and replaces use of the control code sequence 27 "y".

Associated Keywords

24x80,

24x80

SCREEN : STYLE

Selects 24x80 character mode.

EXTRA FUNCTION

Use

Selects the maximum useable screen size in rows and columns to 24x80.

Form

PRINT "24x80,";

Notes

This command will also clear the screen of all information except within the excess space at the bottom and left side of the screen.

This replaces use of the control code sequence 27 "x".

Associated Keywords

32x90,

i-PAPER

SCREEN : STYLE

Selects dark background and light foreground.

EXTRA FUNCTION

Use

Sets the whole screen to normal settings of background colour as black and text colour in green.

Form

PRINT "i-PAPER.";

Notes

This replaces use of the control code sequence 27 "b" 0 27 "c" 0.

Associated Keywords

i+PAPER.

i+PAPER

SCREEN : STYLE

Selects light background and dark foreground.

EXTRA FUNCTION

Use

Sets the whole screen to inverse video with the background colour green and the text colour black.

Form

PRINT "i+PAPER.";

Notes

This replaces use of the control code sequence 27 "b" 0 27 "c" 1.

Associated Keywords

i-PAPER.

iCLS.

SCREEN : CONTROL

Clear screen and home cursor,

EXTRA FUNCTION

Use

Removes all displayed information from the screen leaving it blank with the cursor at the top left hand corner.

Form

PRINT "iCLS,";

Notes

May only be used in a print command. This command replaces use of character sequence 27 "E" to clear screen with 27 "H" to home cursor afterwards.

Associated Keywords

iCLS2.

iCLS2.

SCREEN : CONTROL

Clear screen,

EXTRA FUNCTION

Use

Removes all displayed information from the screen leaving it blank with cursor position untouched.

Form

PRINT "iCLS2,"

Notes

May only be used in print command. This command replaces use of character control sequence 27 "E" to clear screen.

Associated Keywords

iCLS.

i-SL.

SCREEN : CONTROL

Disable status line.

EXTRA FUNCTION

Use

Disables the status line at the bottom of the screen allowing the last line to be used.

Form

PRINT "i-SL."

Notes

This function is useful when saving the screen display to disk using iSAVE, to make the screen display clean of unnecessary text on the bottom line of the screen.

Replaces use of the control code sequence 27 "0".

Associated Keywords

i+SL.

i+SL.

SCREEN : CONTROL

Enable status line.

EXTRA FUNCTION

Use

Enables the status line previously disabled by the i-SL. function.

Form

PRINT "i+SL.";

Notes

Enabling the status line will remove information on the bottom line of the screen.

Replaces use of the control code sequence 27 "1".

Associated Keywords

i-SL.

i+STORE.

SCREEN : CONTROL

Store an area of screen in memory buffer.

EXTRA FUNCTION

Use

This command allows you to store an area of screen in a buffer for later replacement on to the screen at any desired location using i-STORE. Graphics blocks may thus be manipulated to interesting effect.

Form

PRINT "i+STORE.", x-location, y-location, x2-location, y2-location;

All parameters are real positive integers ranging from 0-719 in the x axis and 0-255 on the y axis. X-location and Y-location describe the top left hand corner of the box to be stored.

Notes

The storage buffer may hold a maximum of 1024 bits or pixels and so may store up to 16 screen characters.

Supplying parameters which result in a larger buffer requirement will default to the whole buffer being filled.

Associated Keywords

i-STORE.

The screen buffer starts at memory location 58999 (&HE677) and is 128 bytes large. Peeking the contents of this buffer is a fast way to read the screen from BASIC.

i-STORE.

SCREEN : CONTROL

Replace the screen buffer on the screen.

EXTRA FUNCTIONS

Use

This command restores the contents of the screen buffer previously stored using i+STORE. to the screen at any given location.

Form

PRINT "i-STORE.", x-location, y-location;

All parameters are real positive integers ranging from 0-719 in the x axis and 0-255 on the y axis. X-location and Y-location are the x and y axis coordinates on the screen to target the returning information from the screen buffer.

Notes

Note: If iLOAD or iSAVE is used iSTORE will give an error as iLOAD, iSAVE use the memory buffer (This was to keep the size of EXBASIC small so as to give maximum BASIC user area of RAM. Interesting effects may be achieved using this command. Try the following program;

```
10 PRINT "iCLS,HELLO THERE;i+STORE.",0,0,100,7;
20 PRINT "i-STORE.",1,1;" iSTORE.",2,2;" iDOWN,iDOWN."
RUN
```

Associated Keywords

i+STORE.

iPOINTSCREEN : CONTROL

Return the value of a pixel from the screen.

EXTRA FUNCTION

Use

This command interrogates the screen for the value of a pixel set or unset when used in conjunction with a graphics drawing function.

Form

```
PRINT "iPOINT,";  
PRINT "iPLOT,",x-location,y-location;
```

X-location and y-location are positive real integers within the range of 0-719 in the x-axis and 0-255 in the y-axis.

Notes

The value returned from this function is available at memory location 58943 (&HE63F). Also available at location 58944 (&HE640) is a positional marker indicating pixel position within the byte.

This is ideal for reading if a pixel is 1,0.

Remember to RESET the plot/draw to iRUBBER/iNORMAL/iINVERT before using plot to draw on the screen.

iDUMPSCREEN : PRINTING

Copy the screen to printer.

EXTRA FUNCTION

Use

This function performs a screen dump to the printer in the same manner as the use of the EXTRA and PTR keys.

Form

```
PRINT "iLDUMP,"
```


!PAUSE.

MISCELLANEOUS

Pause program.

EXTRA FUNCTION

Use

Pause program operation until time delay expires or key is pressed.

Form

```
PRINT "!PAUSE.", DELAY;
```

The delay may be in the range of 0-65535

Delay 0 gives an indefinite delay until a key is pressed.

Delay 1-65535 give increasing time delays of 1-65535 seconds continuing program execution for up to 9 hours on delay expiring or a key being pressed.

Notes

This command replaces the program structure to wait for a single key input.

Eg:

```
10 I$=INKEY$:IF I$= "" THEN 10
```

would be replaced by:

```
10 PRINT "!PAUSE.", 0;
```

The delay may be a variable or integer.

The ASCII code of the key pressed is available at memory location 58916 (&HE624).

!BEEP.

MISCELLANEOUS

Computer sounds a beep.

EXTRA FUNCTION

Form

```
PRINT "!BEEP.";
```

Notes

This command directly replaces use of the character code 7 to sound a beep.

ERROR MESSAGES

LOADING EXBASIC - ERROR MESSAGE:

"Cannot Find Your Mallard Basic On This Disc !"
"Please Can You Tell Me Which Disc Drive It Is In ?"

Note: PCW8256 owners with one disc drive can load EXBASIC then remove the disc, place the disc with MALLARD BASIC into the drive and press "A", or copy MALLARD BASIC and EXBASIC on to the same disc with CP/M's PIP function.

Eg. PIP M:=A BASIC.COM

Change the disc for the disc holding EXBASIC then type PIP A:=M:BASIC.COM.

ERROR MESSAGES WITH EXBASIC:

All functions with number parameters will give individual error messages if the number range is exceeded.

Eg. PRINT "iPLOT.",5000,1000;

* ERROR IN NEW PRINT COMMAND *
PLOT,,x?,y?

If during the running of a program a new function error occurs, EXBASIC will do the equivalent of CONTROL (ALT) C.

Eg. Break At Line No.

If the error occurred on the command line or last line of program leading to a command line. The next statement entered will lead to a "BREAK OK" message. This is to avoid the same error occurring again. Simply type "List" to remove the held control - C, then type your commands normally.

SWITCHING EXBASIC ON/OFF

SWITCHING EXBASIC OFF

POKE Memory location &HE622 (58914) with 255 to deactivate the trapping of EXBASIC functions. These will now appear as a listing on the screen and may be useful as a diagnostic or when the i character is needed for a purpose other than an EXBASIC function.

SWITCHING EXBASIC ON

POKE Memory location &HE622 (58914) with 0 to activate EXBASIC.

ADDENDUM - COMBINING EXBASIC AND MALLARD BASIC

1. PLACE SYSTEM DISK SIDE 2 INTO TOP ("A") DRIVE.
2. LOAD CP/M.
3. NEXT TO THE "A>" TYPE "PIP" RETURN.
4. TYPE THE FOLLOWING NEXT TO THE "X".
M:=A:BASIC.COM
5. REMOVE SYSTEMS DISK.
6. PLACE IN THE TOP DRIVE YOUR "EXBASIC" DISK.
7. TYPE THE FOLLOWING NEXT TO THE "X".
M:=A:EXBASIC.COM
8. REMOVE THE "EXBASIC" DISK.
9. PLACE INTO THE DRIVE A BLANK FORMATTED DISK (IF NOT FORMATTED USE DISCKIT AND START AGAIN FROM "1").
10. TYPE THE FOLLOWING NEXT TO THE "X".
A:=M:*. *
11. PRESS RETURN.

