



# EXTBASIC

EXTENDED BASIC COMMAND  
PROGRAM

FOR THE AMSTRAD PCW8256 /8512

## HEISENSTEIN

COMPUTER SYSTEMS  
UNITY PASSAGE  
LOWER BRIDGE STREET  
CHESTER CH1 1RY  
TEL 0244 312866

EXTBASIC is source copyright of David Rathbone, 1986, 1987, 1988  
EXTBASIC.COM is copyright of Heisenstein Computer Systems 1988  
AMSTRAD is a registered trademark of Amstrad Plc.  
MALLARD Basic is copyright of Locomotive Software Ltd.

## PREFACE

When Locomotive Software wrote \*Mallard Basic they gave it speed and powerful functions, but because it was designed to talk to computers that had little need for graphic commands the graphic keywords were omitted.

**EXTBASiC** is intended to assist writers of \*Mallard Basic software on the AMSTRAD PCW. By adding new simple easy to use graphic keywords to the language.

### CONTENTS

Introduction	page 1
Starting Extbasic	page 1
Programming with Extbasic	page 1
Parameters	page 2
Ranges	page 2
Syntax of Parameters	page 2
Functions	page 3
General Syntax	page 3
Extra Command Easy Finder	page 4
The Extra Function Names and Use	page 5-23
Extbasic Loading Error Messages	page 24
Error Messages with Extbasic	page 24
Switching Extbasic On/Off	page 24

\* Mallard Basic is Copyright of Locomotive Software

First printed 1988

Copyright 1986,1987,1988 David Rathbone.

With thanks to P.Swift.

Printed in England

Neither the whole nor any part of the information contained herein, nor the software described in this manual may be adapted or reproduced in any form except with prior approval.

**KITSENSTEIN**

## INTRODUCTION

The **EXTRa** functions included in **EXTBASiC** 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
- EXTBASiC** 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 EXTBASiC

Normally when **EXTBASiC** is used you should have a copy of **EXTBASiC.COM** together with a copy of **BASIC.COM** (Mallard Basic) on the same disc. Please refer to **APPENDIX I** should you need help.

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

**EXTBASiC** and pressing **RETURN**.

**EXTBASiC** will load itself into your computer and will automatically load **BASIC** in also. If **Mallard BASIC.COM** is not present on the default disk drive, a menu will appear to prompt you for a choice of disk drive.

Note: Standard **BASIC** loading rules still apply. For example at the **CP/M** prompt (or within a **Profile.sub**) **EXTBASiC** can be used instead of **BASIC**.

Eg: A > **EXTBASiC A:DEMO.BAS**

This will load **EXTBASiC + MALLARD BASIC** and run "**DEMO.BAS**" if all three programs were on the same disc. (See **APPENDIX II** for help).

### PROGRAMMING WITH EXTBASiC

To use an **EXTBASiC** function, the function name must be printed via a **BASIC PRINT** statement followed by any necessary parameters. For instance the simple draw a circle function would be:

**PRINT "CIRCLE.," x-location,y-location,radius;**

There are several important things to notice about that last function which applies to all **EXTBASiC** 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 (32*90) X = 0-89 Y = 0-31
TEXT (24*80) 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 "CIRCLE. 360 180 150 ";
or using a string in the text form:-
10 A$ = " 360 180 150 "
20 PRINT "CIRCLE.",A$;
```

## FUNCTIONS

All **EXTBASIC** functions start with an "!" character; an upside down exclamation mark (*found by pressing the EXTRA and 1 keys together on the PCW keyboard*), followed by the functions name and end with a full stop.

Leaving off the ! character will cause **EXTBASIC** 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$ = "CLS.";PRINT A$;
20 A$ = A$ + "CIRCLE."
30 INPUT X,Y,R
40 PRINT A$,X,Y,R;
50 GOTO 30
```

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

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

Eg:

```
10 PRINT "DRAW.";
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

<b>GRAPHICS PIXEL MODE</b>	<b>NORMAL</b> NORMAL GRAPHICS MODE 5 <b>INVERT</b> INVERT GRAPHICS MODE 5 <b>IRUBBER</b> ERASE GRAPHICS MODE 6 <b>ISAVE</b> SAVE SCREEN TO DISK 7 <b>ILOAD</b> LOAD SCREEN FROM DISK 7 <b>IMOVE</b> MOVE GRAPHICS CURSOR 8 <b>IPLOT</b> PLOT A PIXEL X,Y 8 <b>IDRAW</b> DRAW A LINE X,Y 9 <b>ICIRCLE</b> DRAW A CIRCLE X,Y,R 9
<b>DISK / SCREEN</b>	
<b>DRAWING</b>	
<b>TEXT CURSOR MOVEMENT</b>	<b>IUP</b> MOVE TEXT CURSOR UP 10 <b>IDOWN</b> MOVE TEXT CURSOR DOWN 10 <b>ILEFT</b> MOVE TEXT CURSOR LEFT 11 <b>IRIGHT</b> MOVE TEXT CURSOR RIGHT 11 <b>IAT</b> LOCATE CURSOR AT R,C 12 <b>IHOME</b> LOCATE CURSOR AT 0,0 12 <b>ISCP</b> SAVE CURSOR POSITION 13 <b>IRCP</b> RESTORE CURSOR POSITION 13 <b>I-CUR</b> CURSOR BLOB OFF 14 <b>I+CUR</b> CURSOR BLOB ON 14
<b>LOCATION</b>	
<b>CONTROL</b>	
<b>TEXT MODE</b>	
<b>UNDERLINE</b>	<b>I-UL</b> UNDERLINE OFF 15 <b>I+UL</b> UNDERLINE ON 15 <b>I+REV</b> REVERSE VIDEO ON 16 <b>I-REV</b> REVERSE VIDEO OFF 16
<b>REVERSE VIDEO</b>	
<b>SCREEN MODE</b>	
<b>MAX ROW/COL</b>	<b>I32*90</b> SELECT 32*90 SCREEN SIZE 17 <b>I24*80</b> SELECT 24*80 SCREEN SIZE 17
<b>BACKGROUND</b>	<b>I-PAPER</b> BACKGROUND GREEN 18 <b>I+PAPER</b> BACKGROUND BLACK 18
<b>SCREEN CONTROL</b>	
<b>CLEAR SCREEN</b>	<b>ICLS</b> CLEAR SCREEN AND HOME 19 <b>ICLS2</b> CLEAR & PROTECT CURSOR 19
<b>BOTTOM LINE</b>	<b>I-SL</b> DISABLE STATUS LINE 20 <b>I+SL</b> ENABLE STATUS LINE 20
<b>BIT WINDOW</b>	<b>I+STORE</b> STORE SCREEN AREA 21 <b>I-STORE</b> RESTORE SCREEN AREA 21
<b>PIXEL GET</b>	<b>IPOINT</b> RETURN BIT FROM SCREEN 22
<b>PCW8000's only</b>	<b>ILDUMP</b> COPY SCREEN TO PRINTER 22
<b>MISCELLANEOUS</b>	<b>IPAUSE</b> PAUSE FOR A DELAY OR KEY 23 <b>IBEEP</b> SOUND A BEEP 23
<b>ERROR MESSAGES</b>	<b>EXTBASIC</b> ERROR MESSAGES 24 <b>EXTBASIC ON/OFF</b> EXTBASIC ON/OFF MODE 24

**EXTRA FUNCTION NAME:** INORMAL  
**GRAPHICS PIXEL MODE:** Reset graphic style to normal.

### Use

Returns the graphics line style to the normal default of overwriting screen locations with green pixels for IPLOT, IDRAW, and ICIRCLE. The normal graphics style is the complete opposite to the style entered using IRUBBER.

### Form

**PRINT "INORMAL";**

### Notes

This command has no effect when the graphics style is already normal  
*Associated Keywords*  
**INVERT. IRUBBER. IPOINT.**

**EXTRA FUNCTION NAME:** !INVERT.  
**GRAPHICS PIXCEL MODE:** Sets the graphic style to invert.

### Use

With this style in effect IPLOT, IDRAW, and ICIRCLE 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

**INORMAL. IRUBBER. IPOINT.**

**EXTRA FUNCTION NAME:** IRUBBER.  
**GRAPHICS PIXEL MODE:** Sets the graphic style to erase.

Use

With this style in effect (PLOT, DRAW, and CIRCLE 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*

**INVERT. INORMAL. IPOINT.**

**EXTRA FUNCTION NAME:** ISAVE.  
**GRAPHICS DISK / SCREEN:** Save screen to disk.

Use

To copy the screen display to disk.

Form

**PRINT "ISAVE.(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*

**ILOAD.**

**EXTRA FUNCTION NAME:** ILOAD.  
**GRAPHICS DISK / SCREEN:** Load screen from disk.

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*

**ISAVE.**

**EXTRA FUNCTION NAME:** **IMOVE.**  
**GRAPHICS DRAWING:** Move graphics cursor.

**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.**

**EXTRA FUNCTION NAME:** **IPLOT.**  
**GRAPHICS DRAWING:** Plots a pixel on the screen.

**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. INVERT.** or **IRUBBER.** with the default being **INORMAL** style.

The graphics cursor is set to new plotted coordinates

*Associated Keywords*

**IDRAW. ICIRCLE. IMOVE.**

**EXTRA FUNCTION NAME:** **IDRAW.**  
**GRAPHICS DRAWING:** Draw a line of pixels on the screen.

**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 "IDRAW.," 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*

**IPLOT. ICIRCLE. IMOVE.**

**EXTRA FUNCTION NAME:** **ICIRCLE.**  
**GRAPHICS DRAWING:** Draws a circle on screen.

**Use**

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

**Form**

**PRINT "ICIRCLE.," 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 **INORMAL. INVERT.** or **IRUBBER.** with the default being **INORMAL** 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*

**IPLOT. IDRAW. IMOVE.**

**EXTRA FUNCTION NAME:** **↑UP.**  
**TEXT CURSOR MOVEMENT:** **Cursor Up**

**Use**

This command moves the cursor up one line.

**Form**

**PRINT "↑UP";**

**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*

**↓DOWN. ↓LEFT. ↓RIGHT. ↓AT.**

**EXTRA FUNCTION NAME:** **↓DOWN**  
**TEXT CURSOR MOVEMENT:** **Cursor Down**

**Use**

This command moves the cursor down one line.

**Form**

**PRINT "↓DOWN";**

**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*

**↑UP. ↓LEFT. ↓RIGHT. ↓AT.**

**EXTRA FUNCTION NAME:** **←LEFT.**  
**TEXT CURSOR MOVEMENT:** **Cursor left.**

**Use**

This command moves the cursor left one character position.

**Form**

**PRINT "←LEFT.";**

**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*

**↑UP. ↓DOWN. ↓RIGHT. ↓AT.**

**EXTRA FUNCTION NAME:** **→RIGHT.**  
**TEXT CURSOR MOVEMENT:** **Cursor right.**

**Use**

This command moves the cursor right one character position.

**Form**

**PRINT "→RIGHT.";**

**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.

*Associated Keywords*

**↑UP. ↓DOWN. ↓RIGHT. ↓AT.**

**EXTRA FUNCTION NAME:** IAT.  
**TEXT CURSOR LOCATION:** Move cursor to new screen position.

**Use**  
In a print command IAT. moves the cursor to a given print position.

**Form**  
**PRINT "IAT",row,column;**

**Notes**  
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.

**EXTRA FUNCTION NAME:** IHOME.  
**TEXT CURSOR LOCATION:** Move cursor home.

**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**  
IUP. IDOWN. ILEFT. IRIGHT.

**EXTRA FUNCTION NAME:** ISCP.  
**TEXT CURSOR CONTROL:** Store current position.

**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.

**EXTRA FUNCTION NAME:** IRCP.  
**TEXT CURSOR CONTROL:** Restore cursor position.

**Use**  
Returns cursor to the location previously saved by SCP.

**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.



EXTRA FUNCTION NAME: f-CUR.  
TEXT CURSOR CONTROL: Disable cursor blob.

Use

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

Form

PRINT "f-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*

f + CUR.

EXTRA FUNCTION NAME: f + CUR.  
TEXT CURSOR CONTROL: Enable cursor blob.

Use

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

Form

PRINT "f + CUR.";

Notes

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

Associated *Keywords*

f-CUR.

EXTRA FUNCTION NAME: f-UL  
TEXT MODE UNDERLINE: Underline off.

Use

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

Form

PRINT "f-UL";

Notes

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

+ UL.

EXTRA FUNCTION NAME: f + UL  
TEXT MODE UNDERLINE: Underline On.

Use

Selects a text printing mode where all text is underlined.

Form

PRINT "f + UL";

Notes

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

Associated *Keywords*

f-UL

**EXTRA FUNCTION NAME:** I+REV.  
**TEXT MODE REVERSE VIDEO:** Reverse Video On.

**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.

**EXTRA FUNCTION NAME:** I-REV.  
**TEXT MODE REVERSE VIDEO:** Reverse Video On.

**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.

**EXTRA FUNCTION NAME:** I32\*90.  
**SCREEN MODE (ROW/COL):** Selects 32\*90 character mode.

**Use**

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

**Form**

**PRINT "I32\*90.;"**

**Notes**

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

*Associated Keywords*

I24\*80.

**EXTRA FUNCTION NAME:** I24\*80.  
**SCREEN MODE (ROW/COL):** Selects 24\*80 character mode.

**Use**

Selects the maximum useable screen size in rows and columns to 24\*80.

**Form**

**PRINT "I24\*80.;"**

**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*

I32\*90.

**EXTRA FUNCTION NAME:** I-PAPER.  
**SCREEN MODE:** Selects dark background & light foreground.

**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.

**EXTRA FUNCTION NAME:** I + PAPER.  
**SCREEN MODE:** Selects light background and dark foreground.

**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.

**EXTRA FUNCTION NAME:** ICLS.  
**SCREEN CONTROL CLS:** Clear screen and home cursor.

**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.

**EXTRA FUNCTION NAME:** ICLS2.  
**SCREEN CONTROL CLS2:** Clear screen and save cursor.

**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.

**EXTRA FUNCTION NAME:** I-SL  
**SCREEN CONTROL STATUS:** Disable status line.

**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

**EXTRA FUNCTION NAME:** I+SL  
**SCREEN CONTROL STATUS:** Enable status line.

**Use**

Enables the status line previously disabled by the -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

**EXTRA FUNCTION NAME:** I+STORE.  
**SCREEN CONTROL:** Store an area of screen in memory buffer.

**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;**  
X-location and Y-location describe the top left hand corner of the box to be stored. (X ,X2 max 719 and Y ,Y2 max 255)

**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.

**EXTRA FUNCTION NAME:** I-STORE.  
**SCREEN CONTROL:** Replace the screen buffer on the screen.

**Use**

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

**Form**

**PRINT "I-STORE"; x-location, y-location;**

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**

If ILOAD. or ISAVE. are used I-STORE. will give an error as ILOAD., ISAVE. use the same memory buffer. Try the following program;

10 PRINT "ICLS.HELLO THERE I+STORE.",0,0,100,7;

20 PRINT "I-STORE.";1,1;" ISTORE. ";2,2;"I DOWN.IDOWN."

**RUN**

**Associated Keywords**

I+STORE.

**EXTRA FUNCTION NAME:** IPPOINT.  
**SCREEN CONTROL:** Return the value of a pixel from the screen.

**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 "IPPOINT.": 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 or 0.

**Remember to RESET the plot/draw to IRUBBER./INORMAL./INVERT.** before using plot to draw on the screen.

**EXTRA FUNCTION NAME:** ILDUMP.  
**PRINTING (PCW8000's ONLY):** Copy the screen to printer.

**Use**

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

**Form**

**PRINT "ILDUMP.":**

ILDUMP. has no parameters.

**Notes**

This function will cause an error if used on the PCW9512 even if an external matrix printer is connected to the parallel port.

**EXTRA FUNCTION NAME:** IPAUSE.  
**MISCELLANEOUS:** Pause program.

**Use**

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

**Form**

**PRINT "IPAUSE.":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.

**Eg:**

**10 I\$ = INKEY\$:IF I\$ = "" THEN 10**

*would be replaced by:*

**10 PRINT "IPAUSE.":0;**

The delay may be a variable or integer. The ASCII code of the key pressed is available at memory location 58916 (&HE624).

**EXTRA FUNCTION NAME:** IBEEP.  
**MISCELLANEOUS:** Computer sounds a beep.

**Use**

Try the following program:-

**10 PRINT "ICLS,IPAUSE. 0 PRESS ANY KEY";**

**20 PRINT "IHOME,IBEEP. THE KEY YOU PRESSED WAS ";**

**30 PRINT PEEK(58916)**

Here EXTBASIC is doing most the work by waiting for a key, beeping, then printing out the ASC11 value of the pressed key.

**Form**

**PRINT "IBEEP.":**

**Notes**

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

## EXTBASIC LOADING ERROR MESSAGES

### LOADING EXTBASIC - ERROR MESSAGE:

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

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

Eg. PIP M: = A: BASIC.COM

Change the disc for the disc holding EXTBASIC then type PIP

A: = M: BASIC.COM.

### ERROR MESSAGES WITH EXTBASIC:

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

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

\* ERROR IN NEW PRINT COMMAND \*  
iPLOT.,x?,y?

If during the running of a program a new function error occurs, EXTBASIC 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 EXTBASIC ON/OFF

#### SWITCHING EXTBASIC OFF

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

#### SWITCHING EXTBASIC ON

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

## APPENDIX I

### TO MAKE A WORKING COPY OF EXTBASIC AND MALLARD BASIC

You will need the following:-

Your CP/M disc, \ This EXTBASIC disc, \ A new blank formatted disc, \ This manual and a PCW Computer.

1. Turn your PCW on without any discs in it, then place your original system disc with CP/M into drive "A".
2. LOAD CP/M. (do not load Locoscript!)
3. Next to the "A >" prompt type "PIP" Then press RETURN.
4. Type the following next to the "":  
M: = A: BASIC.COM Then press RETURN
5. Remove your system disc.
6. Place in the same drive this "EXTBASIC" disc.
7. Type the following next to the "":  
M: = A: EXTBASIC.COM Then press RETURN
8. REMOVE the "EXTBASIC" disc.
9. Place into the drive A BLANK FORMATTED DISC (if not formatted use diskkit & start again from "1").
10. Type the following next to the "":  
A: = M: \* Then press RETURN
11. You will now have a copy of EXTBASIC and Basic on the same disc. When the "\*" is back remove the disc.
12. Turn the computer off.
13. To use EXTBASIC first turn your PCW on. Then load CP/M from your system disc, then remove the CP/M disc when the "A >" prompt is displayed. Now insert your new working copy and type EXTBASIC Then press RETURN

## APPENDIX II

### LOADING THE EXTBASIC DEMO PROGRAMS:-

Your EXTBASIC disc comes with a number of demo programs. The above method of making a working copy will not put these on your working disc. (As you will develop your own EXTBASIC programs)

To run the demo programs first load EXTBASIC as outlined above, (No.13) then in EXTBASIC insert this disc (Not your copy) and type DIR then press RETURN. A DIRECTORY list of all the EXTBASIC programs will be shown Eg: DEMO.BAS (any program that ends in .BAS you can Run and LIST). When you have selected the name of the program you would like to run type, RUN "DEMO.BAS" then press RETURN (the name being typed within quotes)

GOOD LUCK AND HAPPY EXTBASICING!

David Rathbone.