

DUMP OUT 3

Printer Dump ROM for BBC Micro



WE Watford Electronics

CONTENTS

Section 1	INTRODUCTION.....	2
	TECHNICAL NOTES.....	2
	PRINTER INITIALISATION	3
Section 2	DUMPOUT PARAMETERS.....	3
Section 3	DUMPOUT *COMMANDS	
	GIMAGE.....	4
	GWINDOW	11
	TIMAGE	12
	TWINDOW.....	13
	HELP DUMPOUT	13
	HELP PRINTERS	13
Section 4	MODE 7 AND 8 NOTES.....	14
Section 5	MODE 7 OSWORD CALLS.....	14
	MODE 7 GRAPHICS.....	15
	OSWORD &89 READ PIXEL.....	16
	OSWORD &8A PLOT PIXEL.....	18
Section 6	GRAPHIC DUMP RESOLUTION	21
Section 7	GENERAL *COMMAND NOTES.....	22
Section 8	DUMPOUT PARAMETER SUMMARY	23
Section 9	PRINTER TYPES	24
Section 10	ERRORS	25

(C) WATFORD ELECTRONICS 1984

Source code text P.N.Trevethick. No part of this publication or the accompanying software may be reproduced, stored in a retrieval system or transmitted in any form or by any means without the prior written permission of Watford Electronics.

DUMPOUT 3

Section 1 INTRODUCTION

Machine code utility ROM for BBC model B +1.2 OS

- i. Graphic screen dumps of all graphics modes (modes 0,1,2,4,5,7 and 8), for printers from Seikosha, NEC, Shinwa, Epson, Star and Tandy (see detailed list).
- ii. Fast, text only screen dump for all printers.
- iii. Interactive text and graphic window setting utilities. These enable you to visually set up windows much more quickly than is otherwise possible.
- iv. Two new OSWORD calls that allow you to use the beeb graphic coordinate system for testing or plotting mode 7 'pixels'.

1.1 TECHNICAL NOTES

DUMPOUT is an 8k chip that locates in one of the BBC computer paged ROM sockets (or on a paged ROM expansion board). You can put DUMPOUT in any paged socket number. It does not have a language entry point so will never be selected at power on. Putting DUMPOUT in a high priority socket will improve the response times of the new OSWORD calls, though this would only be significant if using these calls frequently.

MEMORY REQUIREMENTS

While processing * commands, DUMPOUT uses the allocated zero-page * command workspace (&A8 - &AF) and some stack memory.

While processing the two new mode 7 OSWORD calls, the OSWORD parameter block and some stack memory are used.

1.2 PRINTER INITIALISATION

Any FX calls you use for selecting printer type, baud rate etc, before normal text printing, are also required at some time before you start doing screen dumps.

Should your printer be switchable between CR and 'new line' operation, then 'new line' should be selected (ie auto LF after CR). With certain printers, if this is not done, the printer will not scroll vertically correctly during dumps.

If you have the Watford PRINTER-ROM, and this has been initialised, then you should bypass it by using *FX5,1 for parallel printers (or FX5,2 for serial) before initiating a graphics dump. PRINTER-ROM can be reselected after the dump by *FX5,3 (or *FX5,6 for serial).

Section 2 DUMP/OUT PARAMETERS

The * command parameter system, that is commonly used in other ROMs, requires you to include any optional parameters in a fixed order. As a result of this, if say you wanted to specify the 5th optional parameter, then you are forced to also include the first 4 parameters, even though you may just want the default settings for these. This is bad enough where there is only a few optional parameters, but is totally impracticable for large numbers of parameters.

For this reason a new system for specifying optional parameters is used in Dumpout 3. All numeric parameters, given in the * command, must follow the appropriate parameter prefix letter. You only need specify the parameters you want, and in general, they may be given in any order.

eg *GIMAGE T X100 500 I/40

and *GIMAGE I40,X100,500,T would both do a two tone graphics dump of the screen X ordinate range 100 to 500 starting 4 inches in from the left edge of the page. This command and the parameters shown are explained in more detail later.

Numeric parameters may be given in 4 ways.

@ % to Z%	Uses current value of system integer variables.	
Prefix %	Binary	eg. %11110000
Prefix /	Decimal	eg /9810
Prefix &	Hexadecimal	eg &01FF

If you enter numbers with no base prefix, the default radix (number base) particular to the parameter type is used.

Section 3 DUMPOUT * COMMANDS

*GIMAGE <printer> (P) (T) (G) (M<mask>)
(F) (L) (H<scale>) (V<scale>) (R<0-3>)
(X<min><max>) (Y<rain><max>) (I<indent>) (C) (E)

Initiates a multi tone screen to printer graphics dump of an area of mode 0, 1, 2, 4, 5, 7, or S screens. The number of tones produced is the same as the number of logical colours of the current screen mode (except for mode 2, 7 and 8 where colours 8 to 15 (mode 7 flashing colours) just repeat the tones used for colours 0 to 7).

The simplest command only needs to include the <printer> parameter, eg. *GIMAGE EPSON, this produces a dump having all the default characteristics. In most instances you will probably only need to include a few parameters in the command.

If included, the parameters F,L,H,V,R should be in this order, though not necessarily together. H and V may follow R but if the dump is 90 or 270 degrees, H would then control the printed length, and V the printed width.

Wherever F occurs it resets L, H and V to their default settings.

Wherever L occurs it resets H and V to their default values.

To anyone who has not used a machine code graphics dump program before, the dump speed may seem rather slow. However, this is quite normal, for where as printing a line of normal text only involves sending 80 bytes to the printer, printing one line of dot graphics may mean sending up to 1280 bytes to the printer, and require 10,240 screen pixel read operations.

<printer> This is where you specify the printer type. A list of printer types recognised is produced by *HELP PRINTERS

P Physical colour values used when dumping. This has no effect on mode 7 dumps as physical colour values cannot be assigned in this mode.

By default logical colour values are used.

When dumping 4 colour screen modes, if logical colours are used for dumping (P not selected), then DUNPOUT automatically uses the 4 shades that have the widest possible brightness separation. If you select P (physical colours) then, depending on the colours assigned, the 4 shades could be of similar density.

T Two tone dump for higher resolution.
By default the dumps are multi-tone.
When doing two tone dumps, the default <mask> setting will cause all colours except colour 0 (physical colours if P parameter included), to print black. Or, if G parameter included, then all colours except colour 7 will print black.

G Positive grey scale reproduction.
By default dumps use a negative grey scale (printing white darkest and black lightest).
*GIMAGE EPSON G P Would produce a dump that has the same brightness scale as that you would see when viewing the display on a monochrome monitor.

M<mask> Default: base binary; value 11111110.
Range 0 to &FF but only 1 to &FE usefull.
8 bits controlling colour masking.
In two-tone dumps, where the mask bit corresponding to a particular colour is set, then that colour will be printed in black. If the bit is clear, the colour will be printed blank (ie white or your paper colour).
In multi-tone dumps, where a <mask> bit is set the corresponding colours tone may be printed, otherwise that colours image is blank. The bits are arranged in order of print tone density.
If P for physical colours is selected then the following table relating mask bits to colours applies.

Mask-bit		Steady		Flashing	
neg	<G>	Number	Colour	Number	Colour
0	7	0	black	8	black/white
1	6	4	blue	12	blue/yellow
2	5	1	red	9	red/cyan
3	4	8	magenta	13	magenta/green
4	3	2	green	10	green/magenta
5	2	6	cyan	14	cyan/red
6	1	3	yellow	11	yellow/blue
7	0	7	white	15	white/black

If <G> for positive grey scale is being used, then read the required bit from the <G> column, otherwise read it from the neg column.

If doing a logical colours dump (ie P not selected), then use the following table instead.

Mask-bit		
neg	<G>	Logical colour number
0	7	0/8
2	5	1/9
4	3	2/10
6	1	3/11
1	6	4/12
3	4	5/13
5	2	6/14
7	0	7/15

The value to use for mask, is the number with bits set for each of the colours 0 to 7 that you want printed black (or as a tone). If mode 2 or 8 is in use then colours 8 to 15 just follow what you have selected for colours 0 to 7.

eg.

neg mask value (binary) colours printed (black)

00000001	0	and	8
00010100	1 and 2	and	9 and 10
10000000	7	and	15
11111110	1 to 7	and	9 to 15

The first example shown would cause a two-tone dump of a normal screen of text to produce white letters with a black surround. The last example is the default setting.

It would be silly selecting either <mask>=0 as then nothing would be printed at all, or <mask>=&FF in a two-tone dump as then you would get a whole black page. For two tone dumps, it is simpler not to bother using the <G> parameter as exactly the same effect can be achieved by changing the mask value.

Selects a faster dump.

This is only available on certain printers. If not available, then including F will have no effect.

F dumps have less dots than normal speed ones, and so are of less high resolution.

Never include F after the L, R, H or V parameters.

Linear step.

Each printed dot represents a certain range of X and Y ordinate values. As dots are printed horizontally and vertically, the screen ordinate values are stepped by the amount covered by one dot.

By default the horizontal and vertical steps are chosen so that the default scale values produce a 0 or 180 degree dump which is 8 inches wide by 6 inches high. For many printers the step sizes are not whole numbers, and this can lead to some distortion of picture detail. This tends to be most noticeable when dumping mode 7 separated graphics.

L selects steps where each printed dot covers a whole number of screen pixels. This gives less distortion of picture detail, but the aspect ratio and dump size will then vary between printers.

If using L it is pointless selecting scale values which are not integer multiples or sub-multiples of 100. ie you should only use V/H values of 400, 300, 200, 100, 80, 40, 20, & 10 etc.

In mode 7, because vertically 10 pixels cover 41 Y ordinate values, the only vertical dump reduction that can produce satisfactory separated graphics is V&500. Never include L after the R, H or V parameters.

V<scale> Default base hexadecimal.
H<scale> Default value R100.

These are both 2 byte numbers giving you very fine control over the dump size from minute to enormous. The dump size is inversely proportional to scale. eg H&200 (or H/512) would halve the horizontal size, and H&80 (or H/128) would double it.

Where scale is specified prior to the R parameter, H and V control the printed width and height. Placed after the R (rotation) parameter H and V control size relative to the screen X and Y axes, this will be the printed height and width respectively if R1 or R3 (for a 90 or 270 degree dump) is selected.

Scale does not vary with screen mode, and, if the L parameter is not selected, the same scale will give the same size dumps on all printers.

Here are some examples of the (H, V and R) parameters required to give usefull full screen dump sizes.

H V and R		SIZE (inches)		
		width	height	
	R0	8	6	Normal 4x3
	R1	6.4	7.5	
	H&CE V&B4 R1			
or	R1 H&B4 V&CE	8	10.67	Largest 4x3
	V&EF	8	6.4	Plotter scaling
	V&EF R1			
or	R1 H&EF	6.4	8	Plotter scaling

In most cases the best scaling to use for a dump is that giving a 4 by 3 aspect ratio (this is the same as a TV screen). Some applications may require plotter scaling, ie where a true circle does not become an ellipse when dumped. (For a full screen plotter dump, where the X axis covers 8 inches, the Y axis should cover 8 x 1024/1280 = 6.4 inches.)

The scale used in each direction is the product of all scale parameters acting in that direction.

eg *GIMAGE EPSON V&180 H&200 V&200 H&80 gives a horizontal scale of unity (&100) and a vertical scale of &300 (one-third normal size).

This means that you can include the example parameters above to give the ratios required, followed by equal H and V parameters to produce the size that you want. eg For a half-size plotter 90 degree dump use

*GIMAGE EPSON V&EF R1 H&200 V&200

- R<0-3> Default: base decimal; value 0.
Including R0, R1, R2 or R3 rotates the dump through 0,90,180 or 270 degrees respectively.
- X<min><max> Graphic coordinates of area to be dumped.
Y<min><max> Default base decimal. By default the area of the screen dumped is that in the graphics window.
These parameters may be used to select a different area, but note that if this area is larger than the graphics window then only that part within the window will be printed.
Should the origin have been altered, then you should either take account of this when calculating the X and Y ranges, or restore the default graphic origin by using VDU26 or <CTRL><Z> before initiating the dump.
- I<indent> Default: base decimal; value 0.
Set gap between left edge of paper and start of dump in 1/10ths of an inch.
By moving the paper back between dumps and using indent, you can produce small dumps, side by side on the page.
- E Contrast expansion.
Makes mode 7 text characters and separated graphics stand out more clearly from the background.
Should your printer have insufficient dots available to accurately reproduce coloured mode 7 screens (this can be the case where size reduction is in use, and on some printers also at the default size), then text and separated graphics may not be reproduced in great detail. This effect is most noticeable where the foreground and background colours are of similar brightness.
Contrast expansion works by changing the foreground printing tone, of text and separated graphics, to either white or black depending on whether the foreground colour is lighter or darker than the background. The background and contiguous graphic foreground colours are still reproduced in multi-tone.

C

All mode 7 graphics are printed as contiguous.

This improves the shading at any separated graphics, especially in reduced sized dumps.

When mode 7 separated graphics are dumped, if the printer has insufficient dots available to accurately reproduce the separated graphics, the dump image may have bands running through the separated graphics areas. Should this occur selecting C will generally give a more pleasing result.

If C and E are both selected then separated graphics are printed contiguous, but not expanded, and text is printed expanded.

ESCAPING FROM A GIMAGE DUMP

On some printers, if dot graphic data is terminated part way through a line, the printer remains stuck waiting for further graphic data (usually until the power is turned off, and back on, to clear it).

To prevent the ESCAPE key causing this effect, normal escape operation is disabled while dumping is in progress. To allow you to terminate a dump, the program polls the ESCAPE key. With certain printers, polling stops while the printer is actually printing a graphics line (the 'caps lock' and 'shift lock' indicators will both be illuminated). Should this be the case, you must hold down ESCAPE until the printer has finished the current line, and the escape message appears.

Even if the program initiating the dump had disabled escape, the ESCAPE key will still operate while GIMAGE is running.

When the dump ends, or is terminated, the previous escape key effect (see FX 200 description) will be restored.

If you trap errors (using ON ERROR) so that pressing ESCAPE initiates the screen dump, then you must be careful to release the key quickly or else the dump will be immediately terminated, and will generate another escape condition.

Note you should not use BREAK to terminate a GIMAGE dump unless the wrong printer has been accidentally specified.

*GWINDOW

	Draws the graphic window on screen, its size and position can then be altered using the cursor keys.
Cursor	The 4 cursor keys alone move the windows left and bottom edges.
TAB	Simultaneously holding down <TAB> with cursor keys moves the right and top edges.
SHIFT	Simultaneously holding down <SHIFT> with cursor keys, makes movement 8 times faster.
Key <D>	Sets the default (full screen) graphics window, while leaving the graphics origin unaltered.
Key <P>	Prints the VDU 24 values that would set up the same sized window. If the graphics origin has been altered this is taken account of in the values printed. If you want to know what the origin is set to, press <0> then <P>. The first and second numbers are the inverse of the origin values. eg. If -640;-512;639;511; was printed, then the origin X,Y coordinate is 640,512
Key <V>	Same as <P> but in hi-res modes it views the values at the graphics cursor. These numbers will disappear as soon as another key is pressed.
Keys <RETURN> and <ESCAPE>	Both terminate GWINDOW leaving the graphics window set to the last one that was shown. Using <ESCAPE> will generate an escape error.

Providing that key P is not used (or key V in mode 7), then the screen contents are unaffected by this command. GWINDOW can be usefully included within programs, just prior to where *GIMAGE is used, in order to allow selection of the part of the screen to be dumped. The X and Y range parameters would not be specified in the *GIMAGE command. When the window box appears, you would simply adjust it as required then press <RETURN> to allow program execution to proceed to the *GIMAGE command.

In hi-res screen modes, GWINDOW draws flickering lines around, and just inside, the window borders.

In mode 7, GWINDOW only draws the window corners.

If a corner lies on a character that is not one of the block graphic character codes, then that corner is not displayed.

If a corner falls in part of a character line where graphics is selected, then one of the 6 'chunky' pixels in the corner character will flicker. If in a text area, then the corner character will flicker between the original and a second one. For each original character there are 6 possible flicker characters. Which one is displayed depends on the corners position within that character.

Note that GWINDOW will not work if software which rotates or scales the screen axes is in use.

Mode 8 : If problems, see notes in later section.

*TIMAGE (X<min><max>) (Y<min><max>) (I<indent>)

Does a fast, text only dump, of the text window in any mode. Graphics will appear as "*".

X<min><max> Text coordinates of area to be dumped.

Y<min><max> Default base decimal. By default the area of the screen dumped is that in the text window.

These parameters may be used to select a different area, but note that if a text window has been defined, then these parameters give positions relative to the current window (as TAB x,y and VDU31,x,y do). If you want to specify the area in absolute text coordinates, then restore the default windows by using VDU26 or <CTRL><Z> before initiating the dump.

I<indent> Default: base decimal; value 0.

Set gap between left edge of paper and start of dump in printer space characters. The actual size will depend on the print style in use.

Note that since the total number of characters per line varies with print style, no error checking is done on the total of indent + window width. Thus if say 140 was selected when in mode 3, and with an 80 column print style, then each line of text would wrap around 2 print lines.

***TWINDOW**

As GWINDOW but for setting text window.

While in use the 4 characters just inside the corners of the text window will flicker.

Use of TWINDOW restores the default graphics window.

Key <D>

Sets the default (full screen) text window. But note that it will still use the slower software scrolling. Use <CTRL><Z> or VDU26 to restore hardware scrolling.

Key <P>

Prints the VDiJ28 values that would set up the same sized text window. These appear near the top left corner of the screen. Use the command *TV255 to move the display down if you cannot see the figures.

All other keys are the same as for GWINDOW except that key <V> is not used.

TWINDOW can be usefully included within programs, just prior to where *TIMAGE is used, in order to select the part of the screen to be dumped.

In mode 7, if a corner falls on a teletext control character (128-159), then part of the line to the right of that corner may flicker.

In other modes :-

If the logical background colour of a corner character is not the same as the last background selected (using COLOUR 128-155), then that corner will not flicker.

As you change the window, the new characters under the corners will become the same colour as the last foreground colour selected (using COLOUR 0-128). Thus if the current foreground and background colours are the same, the characters under the corners may disappear.

Note: Current mode 8 implementations do not allow text windows.

***HELP DUMPOUT**

Lists the syntax of all Dumpout 3 *commands.

***HELP PRINTERS**

This produces a list of what Dumpout 3 recognises for the <printer> parameter of GIMAGE.

Section 4 MODE 7 AND 8 NOTES

MODE 7

In mode 7 the OS allows you to define graphics windows, using VDU24 command in the same way as for hi-res graphics modes. The only point to watch is that no error checking on the left and right edge ordinates is done, so you have to be carefull not to define a window with edges off the screen. The *GWINDOW command prevents you going off the left edge of the screen, but does not check the right edge.

The graphics origin in mode 7 can also be changed, but Dumpout 3 is only designed to work using the default origin position at 0,0.

MODE 8

To get screen mode 8, you must either have a suitable graphics ROM, or type in a short program to set up mode 8 yourself. An example program is given in the Advanced User Guide book.

When in mode 8 you should not use the *TIMAGE command or VDU26, and should only set up text windows (by using VDU28) with caution. You can alter the graphics origin and define graphics windows, though horizontal error checking is inoperative.

The above mentioned program for setting mode 8 leaves the graphics window set incorrectly. If a full screen reduced size graphics dump is done, before setting a correct window, more than one copy of the screen will be produced (and any lines you DRAW that go off the right edge of the screen will re-appear at the left).

To set a correct full screen window either do VDU28,0;0;1279;1023; or use *GWINDOW and press <D> .

Section 5 MODE 7 OSWORD CALLS

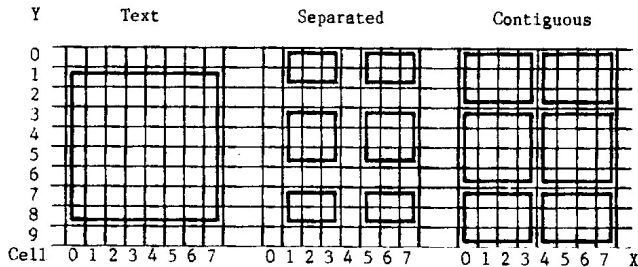
When using OSWORD calls you must set the processors accumulator to the OSWORD number, and the X and Y registers to the low and high bytes respectively of the parameter block address. You then call OSWORD (address &FFF1).

Sometimes information is sent in the parameter block, and sometimes the call returns information in this block. Both OSWORD calls in Dumpout pass information in both directions.

5.1 Mode 7 Graphics

The mode 7 graphics coordinates cover the X range 0 to 1279 and the Y range 0 to 1024. The top Y value is 1 greater than that of the hi-res graphic system, this is because each mode 7 text line is allocated 41 Y ordinate values and mode 7 has 25 text lines. In the X direction each character column covers 32 X ordinate values.

The new mode 7 OSWORD calls consider the characters to be defined on a 8 wide by 10 high matrix. Each of these matrix cells is 4 X ordinate values wide by 4 Y ordinate values high, except for the bottom cell row which is 5 high.



TEXT CHARACTERS

These all lie within the 8 x 8 pixel matrix shown. The ASCII (mode 0-6) character definitions are used, except for character codes 91-94 and 123-126 (and the equivalent characters above 128).

CONTIGUOUS AND SEPARATED GRAPHICS CHARACTERS

The diagrams show the matrix pixel cells that are covered by the 6 block graphic cells. These cells are sometimes referred to as 'chunky' pixels.

5.2 OSWORD &89 Read a mode 7 pixel

This call returns the status of a mode 7 screen pixel at any given graphics co-ordinate.

A parameter block of 16 bytes is required.

XY+0	X ordinate LSB
XY+1	X ordinate MSB
XY+2	Y ordinate LSB
XY+3	Y ordinate MSB
XY+4	Pixel
XY+5	Y Cell (7 position of pixel in character)
XY+6	Foreground colour of character containing pixel
XY+7	Background colour of character containing pixel
XY+8	Column number of character containing pixel
XY+9	Row number of character containing pixel
XY+10	Character
XY+11	Double height table (rows 1 to 8 in bits 0 to 7)
XY+12	Double height table (rows 9 to 16 in bits 0 to 7)
XY+13	Double height table (rows 17 to 24 in bits 0 to 7)
XY+14	Double height table (row 0 in bit 0 ...always clear) bits 1 to 7 Attributes of character containing pixel
	bit 1 Held separated
	bit 2 Separated graphics mode
	bit 3 Hold graphics mode
	bit 4 Conceal on
	bit 5 Double height on
	bit 6 Graphic
	bit 7 Flash on
XY+15	Current hold character

BEFORE CALL

Place coordinates of the point to be tested in XY+0 to XY+3

If double height is in use anywhere on the screen then you must make the first call with XY+11 set to &FF. Doing this causes the whole screen to be scanned for occurrences of the start double height control character (code 141). A bit in the double height table is set for every line which is the bottom one of a double height pair.

If double height is not in use, then to save time (bypassing the screen scan) make the first call with XY+11 to XY+14 all set =0 and XY+8=&FF.

The OSWORD block, from XY+6 to XY+15 inclusive, should not be corrupted between read pixel calls, if it is then the next call must be done as for the first call above. This must also be done if double height lines are added or removed between calls.

If between calls, either any control codes have been added to or removed from the screen, or a plot pixel OSWORD call has been done using the same parameter block, then the next read pixel call must be done with XY+8 set =&FF .

AFTER READ CALL

Pixel contains the colour value at the coordinates specified, values 0 to 7 indicate colours the same as hi-res modes physical colour numbers.

Flash does not alter the value returned in Pixel, you can test attribute bit 7 to find out if flash is on.

If the point was outside the graphics window, then Pixel=&80 is returned.

If the point lies in the separated gap part of a graphics cell which is on, eg if the top left of the 6 graphic cells was on when X Cell=0 and Y Cell=1 (see diagram), then Pixel bit 6 is set. So, in effect, bit 6 set indicates that Pixel would have been equal to Foreground if contiguous graphics was active at the pixel, instead of separated.

Character Column and Row are the absolute text coordinates of the character containing the pixel, and Y Cell is the vertical position of the cell within the character. (The X Cell position is not returned by this call, but is easy to calculate. In BASIC XCell=(XMOD32)DIV4 . Alternatively you can find X Cell by doing a dummy plot pixel call.)

Character is the code of the character containing the pixel, and Current hold character is the graphics hold character that is displayed if character is a control character.

Foreground and Background are the colour values in use at the character.

5.3 OSWORD &8A Plot a mode 7 chunky graphics pixel

This call sets, inverts or clears one cell of a teletext block graphic character. The area of the screen in which plotting is done must have previously been set to graphics mode. ie All characters in the area must follow a 'graphics foreground' control character. To get the largest plotting area, PRINT CHR\$(145) to (151) at column 0 on every line.

Note that unlike plotting in hi-res modes, you cannot explicitly select the colour to use for plotting. The two colours possible are the foreground and background colours selected in that part of the line where the point lies.

A parameter block of 11 bytes is required.

XY+0	X ordinate LS8
XY+1	X ordinate MSB
XY+2	Y ordinate LSB
XY+3	Y ordinate MSB
XY+4	X Cell (X position of pixel in character)
XY+5	Y Cell (Y position if pixel in character)
XY+6	Plot number
XY+7	unused
XY+8	Column number of character containing pixel
XY+9	Row number of character containing pixel
XY+10	Character

BEFORE CALL

Place coordinate of pixel in XY+0 to XY+4

Put the Plot number in XY+6

Plot Number		Type of plot	
Hex	Dec		
45	69	Set foreground	
46	70	Invert pixel	
47	71	Set background	
C5	197	Set foreground	Overwrite text
C6	198	Invert pixel	Overwrite text
C7	199	Set background	Overwrite text

In parts of the screen where graphics is selected, some text including all upper case letters can still be printed. Where you plot a pixel which falls in a text character location, a graphics chunky pixel cannot be displayed without removing the character first. Where bit 7 of the plot number is set (numbers bigger than 127), then text characters are cleared to one of the space codes, 32 or 160, and the pixel is plotted. Where bit 7 is clear text is not overwritten, and no plot is done in text locations.

AFTER PLOT CALL

Character Column and ROW are the absolute text coordinates of the character containing the pixel.

X Cell and Y Cell give the position of the cell within the character.

Character is the code of the character prior to it being altered by the plot.

If you just want to find out what a graphics coordinate is, in terms of Column, Row, XCell and YCell, you can do a dummy write pixel call with plot number =0. This will not plot anything, but returns the same values as a normal plot would.

A BASIC program follows that demonstrates using mode 7 'inverse point' plotting to draw straight lines.

When inverse plotting is used you have to be careful to only plot 1 point in each 'chunky' cell that the line passes through. Inverse plotting in the same cell an even number of times would leave the cell unaltered. This is why x and y step sizes of 16 and 41/3 are given in lines 130 and 140.

The Y starting value of 2 makes sure that the 3 points plotted on each line fall exactly within the 3 chunky cell rows. A start value of 0 would cause the middle cell to be inverted twice, and the top cell to be missed completely. This is due to the middle chunky cell being taller than the top and bottom ones.

When plotting complex shapes it is easier to use only Foreground or Background plotting modes, as then it does not matter how many times a particular chunky cell is plotted. You can then use smaller step sizes, and do not have to worry about the step starting position.

```

10 REM Mode 7 plot pixel OSWORD demo. Draw inverse pixels line.
20
30 MODE 7
40 *HELP
50
60 OSWORD=&FFF1
70 DIM Block% 16
80
90 Write=&8A :REM OSWORD number to plot a pixel
100 PlotNo=70+&80 :REM Plot inverse point, overwriting graphic mode
text characters.
110 PROCsetUpScreen
120
130 FOR x%=100 TO 400 STEP16
140 FOR y=2TO1024 STEP41/3
150 Block%!0=x%
160 Block%!2=y
170 Block%?6=PlotNo
180 PROCosword(Write)
190 NEXT
200 NEXT
210
220 PRINT""Finished "
230 END
240 :
250 DEF PROCosword(A%)
260 REM Pass OSWORD number in call
270
280 X%=Block%MOD256
290 Y%=Block%DIV256
300 CALL OSWORD
310 ENDPROC
320 :
330 DEFPROCsetUpScreen
340 REM Set whole screen to t'text graphics mode
350 FOR y%=0TO24
360 PRINTTAB(0,y%)CHR$(151);:REM Graphic white
370 NEXT
380 ENDPROC
390 :

```

Section 6 GRAPHIC DUMP RESOLUTION

The number of jots per line available on a printer, limits the maximum horizontal dump resolution possible.

Some printers have insufficient dots per line to be able to reproduce a full resolution image of mode 0, or a multi-tone image of modes 1,2 or 7 when the default scale of &100 is used.

A full resolution dump is possible on all printers providing the dump is expanded sufficiently (by using H scale values smaller than &100). When expansion is used, either the dump must be printed at 90 or 270 degrees to allow the whole screen to be dumped, or alternatively dump only a part of the screen.

For most printers, the default vertical scale limits the vertical resolution to achieve a reasonable picture aspect ratio. Resolution reduction here is more acceptable than having a tall thin dump.

The effect of resolution limitation, is that something which is only one screen pixel large, may not be printed at all or may be printed with a tone density error when doing a multi-tone dump. This is most often visible on lettering or mode 7 separated graphics. Groups of the same coloured pixels together, will produce accurate colour tones.

Section 7 GENERAL *COMMAND NOTES

*Commands begin with a word which identifies the command. Many commands are followed by further information, each piece of information being referred to as a parameter. The command syntax displayed using *HELP identifies optional parameters by putting them in round brackets (). Parameters can either be characters selecting an option, or may have one or more numeric arguments. When entering a command, do not include the round and pointed brackets shown in the command syntax, but ensure that spaces or other delimiter characters are included between every parameter and argument.

*commands used to initiate screen dumps can be included within programs, or entered in immediate command mode.

*command words have the same effect whether entered in upper or lower case or a mixture of both. However any hexadecimal arguments should use upper case A to F.

You can abbreviate words by using a "." eg *HELP PRINTERS becomes *H.P.

DUMPOUT allows you to abbreviate words to single letters, but you must give enough letters for the command to be distinguishable from the *commands used by the OS and other ROMs. eg otherwise *T. for *TWINDOW would be taken as *TAPE by the OS.

An ambiguous * command is intercepted by whatever has the higher priority, the priority order being 1st the OS, 2nd the paged ROMs from the highest numbered paged socket downwards, 3rd filing system commands in ROM, 4th filing system commands in the filing system library.

It is possible that you have two ROMs that both use the same command word. Should this happen with DUMPOUT then putting DUMPOUT in the lower numbered socket will allow you to access the command in DUMPOUT by prefixing it with a capital "W", and to access the other ROM by typing the command normally. eg. *GWINDOW and *WGWINDOW will both initiate the graphics window utility.

For long term ROM addition compatability, it is best to type *commands in full when used within programs.

All command words and arguments must be separated by some non-alphanumeric character. DUMPOUT 3 allows <space> or "," to be used as delimiters. Where a word has been terminated by a "." the use of a delimiter before the next word or parameter is optional.

Section 8 DUMPOUT PARAMETER SUMMARY

C	All mode 7 graphics printed as contiguous. This improves the shading of separated graphics especially in reduced sized dumps.
E	Contrast expansion. Makes mode 7 text characters and separated graphics stand out more clearly from the background.
P	Selects a faster dump. Only available on certain printers. Never put F after the L, R, H or V parameters.
G	Grey scale reproduction. By default dumps use a negative scale (white prints black).
H<scale>	Default: base hexadecimal; value &100. A 2 byte number controlling the dump horizontal scale. Larger scale gives smaller dump size.
I<indent>	Default: base decimal; value 0. Set gap between 'left edge of paper and start of dump. In 1/10ths of an inch for GIMAGE. In printer space characters for TIMAGE.
L	Linear step size. Gives less distortion of picture detail, but dump size and aspect ratio will then vary between printers. Never put L after the R, H or V parameters.
M<mask>	Default: base binary; value 11111110. Range 0 to &FF but only 1 to &FE usefull. 8 bits controlling colour masking. Where a bit is set a tone may be printed, otherwise the colours' image is white.
P	Physical colour values used when dumping. By default logical colour values are used.
<printer>	This is where you specify the printer type. A list of printer types recognised is produced by *HELP PRINTERS

R<0-3> Default: base decimal; value 0.
Dump rotation 0,90,180,270 degrees.

T Two tone dump for higher resolution.
By default the dumps are multi-tone.

V<scale> Default: base hexadecimal; value &100.
A 2 byte number controlling the dump vertical scale.
Larger scale gives smaller dump size.

X<min><max> Default base decimal.

Y<min><max> By default the whole of the current window is dumped.
These parameters may be included to dump only a part of
the current window area.
Graphics coordinates of area for GWINDOW (Ymin is
the bottom ordinate).
Text coordinates for TWINDOW (Ymin is the top line).

SECTION 9 PRINTER TYPES

<printer>	Printers	Printer state after graphics dump
SEK	Seikosha GP80/100	Normal width text
SEK2	Seikosha GP250	Line spacing 1/6 inch
TND	Tandy LPVII/DMP100	Normal width text
DMP	Tandy DMP120/200/400	Elite, bold off, elongation off
NEC	NEC PC8023	Pica, logic seeking, 1/6 line spacing
EPSON	Epson MX/RX/FX /STAR	Line spacing 1/6 inch
SHINWA	CTI CPSO	Line spacing 1/6 inch

Section 10 ERRORS

NUMBER ERROR TYPE

Hex	Dec	
11	17	Escape
90	144	Bad Number An illegal character was given in a numerical argument.
93	147	Printer spec Not a recognised printer type.
95	149	Text mode Screen mode other than 0,1,2,4,5,7,8
96	150	Too big A number was given that is larger than that allowable. (Note that the maximum allowable H and V scale numbers varies between printers, but the dump would be minute before reaching this limit.)
DC	220	Syntax A parameter is given that does not have a recognised prefix letter. The commands'correct syntax is listed.

There is a known problem that can occur when using DUMPOUT 3, or any other ROM that can produce error messages more than 48 characters long, with certain other ROMs currently on sale. This is due to these other ROMs corrupting the hardware stack when they should be inactive. This only manifests itself when you cause a syntax error in a GIMAGE command, and results in the message being corrupted.

In the case of Disc Doctor (V1.09), it puts a byte from 0 to &F (depending on which paged socket it is in) in the middle of any ROM error messages that are more than 48 characters long. If you have a paged ROM expansion board, put Disc Doctor in socket 0 or 10. This will then just produce a spurious null or line-feed in the error message. If you do not have an expansion board then use socket 15 (the one nearest the side of the case), this will turn off paged-mode when long errors occur.

If you suspect a ROM of causing problems, try turning it off.

First set S% = socket number of suspect ROM.

Then type S%?&2A1 = 0

If the ROM has been successfully turned off, it will no longer respond to *HELP with any message.

BREAK will turn all ROMs back on again.