Support Group Application Note
Number: 043
Issue: 3
Author: D J Bell

Master Series Operating System Application Note (OS 3.20)

Applicable Hardware :

BBC Master 128
BBC Master Turbo
BBC Master 512
BBC Master ET
BBC Master Compact

Related Application Notes:

Support Group
Acorn Computers Limited
Acorn House
Vision Park
Histon
Cambridge CB4 4AE

OSBYTE Calls

&0 (0) Enter with X<>0, Returns X=3 for Master 128/Turbo/512/Sc and X=4 for Master ET.
Entering with X=0, (or *FX0) displays OS version.

&14 (20) Parameters for *FX20 are now ignored and *FX20 resets standard exploded font. Software writers must add parameters as required for Model B/B+.

&16 (22) Increment ROM polling semaphore.
Used to request MOS polling with service call 21 every 10mS (polling with semaphore non-zero).

&17 (23) Decrement ROM polling semaphore.
Used to stop MOS polling with service call 21 every 10mS.

&44 (68) Test Sideways RAM presence. Report presence of each page of Sideways RAM. Detects state of links for ROM/RAM selection.

&45 (69) Test pseudo/absolute use of Sideways RAM page.

&6B (107) Switch Internal/External 1MHz Bus.
*FX107,0 - select external bus (default)
*FX107,1 - select internal bus.

&6C (108) Switch Main/Shadow memory into main map.
*FX108,0 - switch shadow memory into main map area, &3000-&7FFF (immediate).
*FX108,1 - switch main memory into main map area (immediate).

&6D (109) Make temporary filing system permanent.

&70 (112) Write to Main/Shadow memory:
*FX112,0 - write to memory specified by mode change.
*FX112,1 - write to main memory (immediate).
*FX112,2 - write to shadow memory (immediate).

&71 (113) Display Main/Shadow memory:
*FX113,0 - display memory specified by mode change.
*FX113,1 - display main memory (immediate).
*FX113,2 - display shadow memory (immediate).

&72 (114) Write to/Display Main/Shadow memory (*SHADOW)
*FX114,0 - use shadow memory at next mode change.
*FX114,n - use mode defined at next mode change. (where n is 1-255).

&81 (129) Now extended to return new OS version:
Enter with X<>0 & Y=255
Returns with X=3 for OS 3.00

Enter with X=0
Returns X=253 for Master Series.

&84 (132) Read top of user RAM for Master Series.

&85 (133) Read top of user RAM for a given mode (was display RAM start for a given mode).

&A1 (161) Read CMOS RAM
Enter with A=0, X=n, where n is the RAM location number (30=49).
Returns with result in Y.
Use *STATUS for locations 0-29.

&A2 (162) Write CMOS RAM
Enter with A=1, X=n, where n is the RAM location number (30-49). *FX162 can be use.
Use *CONFIGURE for locations 1-29. Location 0 is protected.

&A4 (164) Check processor type.

&A5 (165) Read output cursor position.

&B3 (179) Read/Write ROM polling semaphore.
(was Read/Write OSHWM).
A=179, X=n, Y=0 reads semaphore into X and sets state to n.
Setting state directly with this call will interfere with OSBYTE 22 & 23 use.
A=179, X=0, Y=255 reads semaphore into X.

&B6 (182) Read NOIGNORE state (was Read font explosion).

&EE (238) Change numeric pad base:
"0" key is based at 48, *FX238, <base> with base from 0-255 will alter key characters.

&FA (250) Read memory area used for writing to.

&FB (251) Read memory area used for reading from.

&FE (254) Controls effect of SHIFT on numeric pad.
*FX254,0 - makes SHIFT have effect.
*FX254,<1-255> - deletes effect of SHIFT.
(NOTE: This call returned RAM size in Model B/B+).

OSWORD Calls

&E (14) Read CMOS clock.

&F (15) Write to CMOS clock.

&42 (66) Block transfer to/from sideways RAM.

&43 (67) Load or Save to/from sideways RAM.

New Service calls to Sideways ROMs

&15 (21)	Polling interrupt. Made 100 times per second if OSBYTE 22 issued.
&18 (24)	Interactive HELP. Made by MOS when it executes a *HELP command, after service call 9. MOS offer CLI text following a *HELP to a ROM participating in the interactive help system.
&21 (33)	Offer Static Workspace in Hidden RAM. Call is made on a Reset. Workspace starts at &C000 in Hidden RAM and can only be used by a Filing System, and only one at a time. Workspace has an upper limit of &DBFF. Call analogous to &01, but uses hidden RAM.
&22 (34)	Offer Dynamic Workspace in Hidden RAM. ROMs should ideally ignore Call &02, which takes workspace in main memory.
&23 (35)	This is top of Static Workspace. Tells ROMs where top of static workspace in hidden RAM is.
&24 (36)	Dynamic Workspace requirements. ROMs should indicate how much memory they will each claim through Call 34. Y contains current bottom of dynamic allocation and should be decremented by required number of pages.
&25 (37)	Inform MOS of Filing System name and info. (See Reference Manual 1 for detailed information on this call).
&26 (38)	Close all files. Issued at a Reset. Filing systems should select themselves, close open files and then de-select. Used by *SHUT command.
&27 (39)	Reset has occurred. Call made after hard reset. Mainly for Econet Filing System so that it can claim NMI's. This call is now required since the MOS no longer offers workspace on a soft BREAK. A Sideways ROM should therefore re-initialise itself.
&28 (40)	Unknown CONFIGURE option. Used to extend range of commands. A Sideways ROM having a claim on CMOS RAM may use this command to update its configuration information.
&29 (41)	Unknown STATUS option. Used to provide extra commands. See &28.
&2A (42)	ROM based language starting up. This enables languages, such as the TERMINAL, to remove their interception of buffering functions etc. prior to the next language taking control.

New VDU commands

VDU 18,m,c - Define graphics colour.

m = 0 to 4 same as as 1.2 MOS.
m = 5 Leave screen colour unchanged.
For each of n=1,2,3,4 (ecf pattern numbers):
m = 16n Overwrite the colour on the screen.
m = 16n+1 OR the colour of the screen.
m = 16+2 AND the colour of the screen

m = 16+3 EOR the colour of the screen.
m = 16+4 Invert the colour of the screen.
m = 16+5 Leave screen colour unchanged.

VDU 22,m - Select screen mode.
m = 0 to 8 As 1.2 MOS.
m = 128 to 135 covers shadow screen modes.

VDU 23,0,r,v,0,0,0,0,0,0 - Control 6845 CRTC directly.
As 1.2 MOS (but with additions of n=2 & n=3) ie:
n = 0 Stops cursor appearing.
n = 1 Cursor appears on screen (default case).
n = 2 Cursor is steady.
n = 3 Cursor flashes at approx 1.5 times/sec (Default case).
Flash rate is doubled in cursor edit mode.

VDU 23,2-5,a,b,c,d,e,f,g,h - Set ecf pattern.
ecf patterns can be set to pixel groups of 8*8, 4*8 or 2*8 if mode has 2, 4 or 16 colours respectively. VDU 23,2 thru VDU 23,5 sets patterns 1 thru 4 respectively.
Integers a thru h define pattern rows from top to bottom. If the integer is derived from stuvwxyz in binary then:
For 2 colour mode, logical colours from left to right are: s, t, u, v, w, x, y, z
For 4 colour mode, logical colours from left to right are: sw, tx, uy, vz
For 16 colour mode, logical colours from left to right are: suwy, tvxz

VDU 23,6,n,0,0,0,0,0,0,0 - Set dotted lines pattern.
n = &FF Solid line as in 1.2 MOS.
n = &AA Dotted line as in 1.2 MOS (Default-Reset every mode change).
n = &EE Dashed line (dot-dot-dot-space repeated).
n = &E4 Dash-dotted line (dot-dot-dot-space-dot-space-space repeated).

VDU 23,7,m,d,z,0,0,0,0,0 - Scroll window directly.
Allows text window or arbitrary rectangle to be scrolled without cursor movement:
m = 0 Scroll text window.
m = 1 Scroll entire window.
d = 0 Scroll right.   d = 1 Scroll left.
d = 2 Scroll down.   d = 3 Scroll up.
d = 4 Scroll in positive X direction (defined by VDU 23, 16, etc).
d = 5 Scroll in negative X direction (" " " " " ").
d = 6 Scroll in positive Y direction (" " " " " ").
d = 7 Scroll in negative Y direction (" " " " " ").
z = 0 Scroll by 1 character cell.
z = 1 Scroll by 1 character cell vertically, 1 byte horizontally.
(ie 8 pixels in 2 colour modes, 4 in 4 colour modes, 2 in 16 colour modes, and 1 character in mode 7). This is the minimum distance that can be scrolled and still be able to do a hardware scroll if the full screen is scrolled.

VDU 23,8,t1,t2,x1,y1,x2,y2,0,0 - Clear block of text window.
This causes a block of the text window to be cleared to the text background colour. The parameters indicate where the two ends of the block (ie string start and string finish) are, with t1, x1 and y1 relating to the start of the block and t2, x2 and y2 to the end of the block.

In each case, ti indicates a base position, to which (xi,yi) is added to get the true position. The character position at the start of the block is generally included in the clear, that at the end is not.

ti = 0 Base position is "top left of window".
ti = 1 Base position is "top of cursor column".
ti = 2 Base position is "off top right of window".
ti = 4 Base position is "left end of cursor line".
ti = 5 Base position is cursor position.
ti = 6 Base position is "off right end of cursor line".
ti = 8 Base position is "bottom left of window".
ti = 9 Base position is "bottom of cursor column".
ti = 10 Base position is "off bottom right of window".

Other values of ti have undefined effects. (The quotes are to indicate that all of these positions are calculated taking the cursor movement controlsset by VDU 23,16 into account - eg after VDU 23,16,2,0,0,0,0,0,0,0 "left" above right etc.).

The results of this function are undefined if the absolute values of the coordinates of the two ends go outside the range -128 to 127. This is best avoided by not using value of xi and yi outside the range -128 to 47. Should the end point of the block lie before the start point, no clearing will be done.

VDU 23,9,n,0,0,0,0,0,0,0 - Set 1st flash time.
Same spec as 1.2 MOS.

VDU 23,10,n,0,0,0,0,0,0,0 - Set 2nd flash time.
Same spec as 1.2 MOS.

VDU 23,11,0,0,0,0,0,0,0,0 - Set default ecf patterns.

Mode	Pattern	Colour	VDU 23,2-5 type definition
0	1	Dark Grey	&CC,&00,&CC,&00,&CC,&00,&CC,&00
	2	Grey	&CC,&33,&CC,&33,&CC,&33,&CC,&33
	3	Light Grey	&FF,&33,&FF,&33,&FF,&33,&FF,&33
	4	Hatching	&03,&0C,&30,&C0,&03,&0C,&30,&C0
1,5	1	Red-Orange	&A5,&0F,&A5,&0F,&A5,&0F,&A5,&0F
	2	Orange	&A5,&5A,&A5,&5A,&A5,&5A,&A5,&5A
	3	Yellow-Orange	&F0,&5A,&F0,&5A,&F0,&5A,&F0,&5A
	4	Cream	&5F,&5A,&5F,&5A,&5F,&5A,&5F,&5A
2	1	Orange	&0B,&07,&0B,&07,&0B,&07,&0B,&07
	2	Pink	&23,&13,&23,&13,&23,&13,&23,&13
	3	Yellow-Green	&0E,&0D,&0E,&0D,&0E,&0D,&0E,&0D
	4	Cream	&1F,&2F,&1F,&2F,&1F,&2F,&1F,&2F
4	1	Dark Grey	&AA,&00,&AA,&00,&AA,&00,&AA,&00
	2	Grey	&AA,&55,&AA,&55,&AA,&55,&AA,&55
	3	Light Grey	&FF,&55,&FF,&55,&FF,&55,&FF,&55
	4	Hatching	&11,&22,&44,&88,&11,&22,&44,&88

Mode 0 patterns are different from 4 to avoid TV effects.

VDU 23,12-15,a,b,c,d,e,f,g,h - Set simple ecf pattern.

This sets a simple 2*4 (or double for mode 0) pattern. Patterns1 thru 4 are set by VDU 23,15 respectively. The logical colours from left to right are:

Top row - a,b
next row - c,d
next row - e,f
last row - g,h

Mode 0 has double pixels to avoid TV patterning.

VDU 23,16,x,y,0,0,0,0,0,0 - Cursor movement control.

Allows control of cursor after a character has been printed. This control sequence replaces the current flag byte as follows:

(Current byte) AND X) EOR Y

If the byte of flags is abcdefgh in binary then:

a = 0 Normal
a = 1 Undefined
b = 0 In VDU 5 mode, cursor movement outside of a window cause special actions ie, carriage returns generated.
b = 1 In VDU 5 mode, cursor movement outside of a window does not cause special actions
c = 0 Cursor moves in positive direction. d & h define action if cursor move outside of window.
c = 1 Cursor does not move.
d = 0 If Y movement would go outside of window, window is scrolled in VDU 4 mode. In VDU 5 mode it moves to opposite edge of the window.
d = 1 As above but cursor always moves to opposite edge.
efg = 000 Text X direction is right, Y direction is down.
efg = 001 Text X direction is left, Y direction is down.
efg = 010 Text X direction is right, Y direction is right.
efg = 011 Text X direction is left, Y direction is up.
efg = 100 Text X direction is down, Y direction is right.
efg = 101 Text X direction is down, Y direction is left.
efg = 110 Text X direction is up, Y direction is right.
efg = 111 Text X direction is up, Y direction is left.
h = 0 If movement would go outside of window, cursor moves to negative edge and one step in positive Y direction. If this goes outside of window, d defines behaviour. This is '80' column mode.
h = 1 If movement would go outside of window, a 'pending cursor movement' is generated. It is released before next character is printed (or another control code). This is '81' column mode.

VDU 23,17-27a,b,c,d,e,f,g,h - Unassigned (but reserved).

VDU 23,27,a,b,c,d,e,f,g,h - Acornsoft sprites.

VDU 23,28-31,a,b,c,d,e,f,g,h - Unassigned (for user application progs).

Reserved for use by application programs. Results in a call to the unknown Plot codes vector &226,&227. Call can be recognise as follows:

* C = 1 on entry to the vector.
* A contains the VDU 23 code (ie the first number following 23).
  All of the sequence except the 23 can be found in ascending order starting at the location: (Start of VDU variables) + &1B, ie at &31B in MOS version 1.2.

VDU 23,32-255,a,b,c,d,e,f,g,h - Define character.
Spec as 1.2 MOS.

VDU 24,11,1h,b1,bh,r1,rh,t1,th - Set graphics window.
Spec as 1.2 MOS.

VDU 25,p,xl,yl,yh - Plot

VDU 25,0-63 - Plot line.
Spec as 1.2 MOS but some improvements.

VDU 25,64-71 - Plot point.
Spec as 1.2 MOS.

VDU 25,72-29 - Horizontal line fill.
Spec as 1.2 MOS.

VDU 25,80-87 - Plot triangle.
Spec as 1.2 MOS.

VDU 25,88-95 - Horizontal line fill.
Spec as 1.2 MOS.

VDU 25,96-103 - Plot rectangle.
Plots a filled axis aligned rectangle with opposite corners at the current graphics cursor and the new point.

VDU 25,104-11 - Horizontal line fill.
Similar to VDU 25,72-79....., with the differences that the word "non-background" should be replaced by "foreground".

VDU 25,112-119 - Plot parallelogram.
Plots a filled parallelogram with vertices at the old graphics cursor, the current graphics cursor, the new point, and at (new point)-(current graphics cursor)+(old graphics cursor) in cyclic order. The 4th point is calculated in terms of internal pixel coordinates to ensure that the sides are parallel.

VDU 25,120-127 - Horizontal line fill.
Similar to VDU 25,88-95...., with the difference that the work "background" should be replaced with "non-foreground".

VDU 25,128-143 - Flood fill.
This flood fills the screen starting from the new point and continuing until non-background (plot codes 128-135) or foreground (plot codes 136-143) pixels are found. These sequences make use of soft-key 11-15 buffers (they will reset soft-keys to empty strings and will fail to do anything if these soft-keys are being expanded). Sequences may fail if:

* The area to be filled is too complicated.
* The colour being used to fill can itself be filled.
* An escape occurs.

VDU 25,144-159 - Plot circle.

Plots a circle outline (plot codes 144-151) or a filled circle (plot codes 152-159) with its centre at the current graphics cursor and the new point on its boundary.

VDU 25,160-183 - Plot circular arc.

Plots a circular arc (plot codes 160-167) the filled chord segment between a circular arc and the chord joining its endpoints (plot codes 168-175) or the filled pie sector between a circular arc and the two radii joining its endpoints to the centre of the circle (plot codes 176-183). In all three cases, the centre of the circle is at the old graphics cursor, the first endpoints of the arc is at the current graphics cursor, the second endpoint of the arc is on the circle an in the same direction from the centre of the circle as the new point is, and the circular arc is taken to be the arc going clockwise from the first endpoint to the second one.

VDU 25,184-191 - Move/copy rectangle.

Causes the axis aligned rectangle with opposite corners at the old and current graphics cursors to be moved (plot codes 185,189) or copies (plot codes 186,187,190,191) so that its new bottom left hand point is at the new point (plot codes 184 and 188 simply move the graphics cursor to the new point, like other plot codes which are 0 mod 4).

Any part of the source rectangle which lies outside the current graphics window is assumed to contain the current graphics background colour for the purposes of the copy or move. The difference between copying and moving is that moving set any part of the source rectangle which lies outside the destination rectangle to background, whereas copying leaves such parts of the source rectangle unchanged.

VDU 25,192-207 - Plot ellipse.

Plots an ellipse outline (plot codes 192-199) or a filled ellipse (plot codes 200-207). The centre of the ellipse is at the old graphics cursor.

VDU 25,208-239 - Unassigned.

Not reserved for application programs. Following assigned:
VDU 25,232-239,x1,xh,y1,yh - Acornsoft sprites.

VDU 25,240-255 - User program calls.

Reserved for application programs. Will result in a call to the unknown plot codes vector (&226,&227). Call recognised by:

* C = 0 on entry
* Computer is in a graphics mode (can test location (start of VDU variables) + &61 ie, &361 on 1.2 MOS. this contains (number of pixels/byte) - 1 (ie 1,3 or 7) in graphics modes, and 0 in non-graphics modes.
* A contains the VDU 25 code (ie the first number following the 25). The coordinates can be found in ascending order starting at the location (start of VDU variables) + &1F ie &31F on MOS 1.2.

VDU 26 - Restore default windows.

Spec as 1.2 MOS.

VDU 27 - Null.

Spec as 1.2 MOS.

VDU 28,1x,by,rx,ty - Define text window.

Spec as 1.2 MOS.

VDU 29,x1,xh,y1,yh - Define graphics origin.
Spec as 1.2 MOS.

VDU 30 - Home cursor.
Spec as 1.2 MOS.

VDU 31,x,y - Tab cursor.
Spec as 1.2 MOS.

VDU 32-126 - Print a character.
Spec as 1.2 MOS.

VDU 127 - Backspace and delete.
Spec as 1.2 MOS.

VDU 128-255 - Print a character.
Prints characters from the extended character set in a similar manner to VDU 32-126.

A full description of the Operating System can be found in the Master Series Reference Manual part 1. The Master Series Welcome Guide contains a summary of the features.