

# 4 OSWORD CALLS

The OSWORD routines are very similar in concept to the OSBYTE routines. The major difference arises in the way of passing parameters. Instead of being passed in the X and Y registers, they are placed in a parameter block. The address of this parameter block is sent to the routine in the X (for the low byte) and Y (for the high byte) registers.

## **OSWORD      OS call specified by contents of A taking parameters in a parameter block.**

Call address &FFF1 Indirected through &20C

On entry,

    A selects an OSWORD routine.

    X contains low byte of the parameter block address.

    Y contains high byte of the parameter block address.

OSWORDS which are called with accumulator values in the range &EO (224) to &FF (255) are passed to the USERV (&200). The routine indirected through the USERV is entered with the register contents unchanged from the original OSWORD call.

Other unrecognised OSWORD calls are offered to the paged ROMs (see service ROMs section 10.1, reason code 8).

## **OSWORD summary**

A=0 Read line from currently selected input into memory.

A=1 Read system clock.

A=2 Write system clock.

A=3 Read interval timer.

A=4 Write interval timer.

A=5 Read byte of I/O processor memory.

A=6 Write byte of I/O processor memory.

A=7 Perform a SOUND command.

A=8 Define an ENVELOPE.

A=9 Read pixel value.  
A=&A Read character definition.  
A=&B Read palette value for a given logical colour.  
A=&C Write palette value for a given logical colour.  
A=&D Read previous and current graphics cursor positions.

## **OSWORD call with A=&0      Read line from input**

This routine takes a specified number of characters from the currently selected input stream. Input is terminated following a RETURN or an ESCAPE. DELETE (&7F/127) deletes the previous character and CTRL U (&15/21) deletes the entire line. If characters are presented after the maximum line length has been reached the characters are ignored and a BEL (ASCII 7) character is output.

The parameter block :-

XY+	0	Buffer address for input	LSB
	1		MSB
	2	Maximum line length	
	3	Minimum acceptable ASCII value	
	4	Maximum acceptable ASCII value	

Only characters greater or equal to XY+3 and less than or equal to XY+4 will be accepted.

On exit:

C=0 if a carriage return terminated input.

C=1 if an ESCAPE condition terminated input.

Y contains line length, excluding carriage return if used.

## **OSWORD call with A=&1      Read system clock**

This routine may be used to read the system clock (used for the TIME function in BASIC). The five byte clock value is written to the address contained in the X and Y registers. This clock is incremented every hundredth of a second and is set to 0 by a hard BREAK.

## **OSWORD call with A=&2      Write System Clock**

This routine may be used to set the system clock to a five byte value contained in memory at the address contained in the X and Y registers.

## **OSWORD call with A=&3      Read interval timer**

This routine may be used to read the interval timer (Used for events, see section 6.4). The five byte clock value is written to the address contained in the X and Y registers.

## **OSWORD call with A=&4 Write interval timer**

This routine may be used to set the interval timer to a five byte value contained in memory at the address in the X and Y registers.

## **OSWORD call with A=&5 Read I/O processor memory**

A byte of I/O processor memory may be read across the Tube using this call. A 32 bit address should be contained in memory at the address contained in the X and Y registers.

XY+	0	LSB of address to be read
	1	
	2	
	3	MSB of address to be read

If the I/O processor uses 16 bit memory addressing only least significant two bytes need to be specified.

On exit:

The byte read will be contained in location XY+4.

## **OSWORD call with A=&6      Write I/O processor memory**

This call permits I/O processor memory to be written across the Tube. A 32-bit address is contained in the parameter block addressed by the X and Y registers and the byte to be written should be placed in XY+4. For compatibility with future products it is recommended that XY+2 and XY+3 be set to zero.

## **OSWORD call with A=&7      SOUND command**

This routine takes an 8 byte parameter block addressed by the X and Y registers. The 8 bytes of the parameter block may be considered as the four parameters used for the SOUND command in BASIC.

e.g. To perform a SOUND 1,-15,200,20

XY+	0	Channel	LSB	1	&01
	1		MSB		&00
	2	Amplitude	LSB	-15	&F1
	3		MSB		&FF
	4	Pitch	LSB	200	&C8
	5		MSB		&00
	6	Duration	LSB	20	&14
	7		MSB		&00

This call has exactly the same effect as the SOUND command.

## **OSWORD call with A=&8      Define an ENVELOPE**

The ENVELOPE parameter block should contain 14 bytes of data which correspond to the 14 parameters described in the ENVELOPE command. This call should be entered with the parameter block address contained in the X and Y registers.

## **OSWORD call with A=&9      Read pixel value**

This routine returns the status of a screen pixel at a given pair of X and Y co-ordinates. A four byte parameter block is required and the result is contained in a fifth byte.

XY+	0	LSB of the X co-ordinate
	1	MSB of the X co-ordinate
	2	LSB of the Y co-ordinate
	3	MSB of the Y co-ordinate

On exit:

XY+4 contains the logical colour at the point or &FF if the point specified was outside the window.

## **OSWORD call with A=&A      Read character definition**

The 8 bytes which define the 8 by 8 matrix of each character which can be displayed on the screen may be read using this call. The ASCII value of the character definition to be read should be placed in memory at the address stored in the X and Y registers. After the call the 8 byte definition is contained in the following 8 bytes.

XY+	0	Character required
	1	Top row of character definition
	2	Second row of character definition
	.	
	.	
	.	
	8	Bottom row of character definition

## **OSWORD call with A=&B      Read palette**

The physical colour associated with each logical colour may be read using this routine. On entry the logical colour is placed in the location at XY and the call returns with 4 bytes stored in the following four locations corresponding to a VDU 19 statement.

e.g. Assuming that a VDU 19,2,3,0,0,0 had previously been issued then OS WORD &B with 1 at XY would yield :-

XY+	0	2	logical colour
	1	3	physical colour
	2	0	padding for future expansion
	3	0	
	4	0	

## **OSWORD call with A=&C Write palette**

This call performs the same task as a VDU 19 command (which can be used from machine code using OSWRCH). The advantage of using this OSWORD call rather than the conventional VDU route is that there is a significant saving in time. Another advantage is that OSWORD calls can be used in interrupt routines while VDU routines cannot. This call works in the same way as OSWORD &B (see above); a parameter block should be set up with the logical colour being defined at XY, the physical colour being assigned to it in XY+1 and XY+2 to XY+4 containing padding 0s.

**OSWORD call with A=&D****Read last two graphics cursor positions**

The operating sytem keeps a record of the last two graphics cursor, positions in order to perform triangle filling if requested. These cursor postions may be read using this call. X and Y should provide the address of 8 bytes of memory into which the data may be written.

XY+	0	previous X co-ordinate,	low byte
	1		high byte
	2	previous Y co-ordinate,	low byte
	3		high byte
	4	current X co-ordinate,	low byte
	5		high byte
	6	current Y co-ordinate,	low byte
	7		high byte