



# BBC MEMORY MAPPING

**In the first part of this series we looked at how an operating system is arranged in general terms and examined, in particular, how the BBC Micro's OS is arranged. In this instalment, we'll look at how the BBC OS actually uses its memory, and how vectors are used in a computer operating system.**

The BBC Micro makes use of various areas of memory, many of which have multiple functions. A typical example of this is page &9 of the memory — the area of RAM between locations &900 and &9FF — which is used, at different times, as the RS232 output buffer, a cassette output buffer, speech workspace and extended sound workspace! No one can accuse Acorn of not getting the most from its memory.

A further example is the block from &0E00 to &1900; in a BBC Micro with a cassette recorder as its filing system, this area of memory is free for use as BASIC program space. However, when a disk system is being used, this area is used as disk filing system workspace, reducing the amount of memory left for BASIC programs by over 2.5 Kbytes. This is extremely annoying in screen Modes 0 to 2, where memory is restricted anyway.

The final example of multiple use of memory is the area between &8000 and &BFFF. If a DFS ROM, a word processor or utility ROM is fitted in the machine it will occupy this block. This is also the area of memory that the BASIC interpreter ROM resides in. These are called 'paged ROMs'; only one of them can be active at any one time. The ROM required is selected by the operating system, and 'switched in', or 'paged in' — hence the name. Under normal circumstances, the BASIC ROM is paged in, and the BASIC interpreter is running, thus enabling us to type in and execute BASIC programs. However, when we execute a DFS command, the OS pages the DFS ROM, executes the command, and then pages the BASIC ROM back in. During the execution of the DFS commands, the BASIC ROM is simply ignored. Other ROMs, such as the Telesoftware Filing System or the Wordwise word processor chip, also occupy this memory space and are also paged in and out as required. It's probably just as well that the BBC Micro uses the same area of memory for several purposes, as otherwise there'd be little memory left for us to work with.

## USER MEMORY

Once the OS has taken its share of the computer memory, what remains is the available user memory. The BASIC system variable PAGE contains

the address of the Start of BASIC program area, and the variable HIMEM points to the start of the screen display memory; the space between is available for BASIC programs, variables and machine code programs. Type in this instruction to find these addresses, and the memory available for BASIC programs:

```
PRINT PAGE, HIMEM, (HIMEM-PAGE) "BYTES FREE"
```

In the following table, you'll find a brief description of the relative merits of different storage areas for machine code in the BBC Micro.

When Area is Suitable For Machine Code Programs	Type of Filing System	
	Tape	Disk
<b>&amp;0000 — &amp;00FF</b> Page &D—only if no paged ROMs are in the machine	(✓)	X
<b>&amp;0800 — &amp;0CFF</b> Page &B and &C — if no user defined characters or user function keys are used in the program	(✓)	(✓)
<b>&amp;0A00 — &amp;0AFF</b> Page &A — if serial interface is not in use	(✓)	(✓)
Space saved in BASIC program area in the DIM command		
✓ = OK (✓) = OK, with Reservations X Not used		

For most routines that are to be used in conjunction with BASIC programs, storing the code in the BASIC program area is the best method.

Once the machine code is in memory, it generally needs some workspace to allow it to run. Most 6502 machine code programs require some zero page locations as workspace, mainly because certain indexed addressing instructions will need zero page locations. Acorn have made extensive use of this page for the OS, but some locations have been set aside for machine code programming. A byte by byte account of what each of the OS locations does is little use to the programmer, and directly accessing these locations is not encouraged — the useful locations should be accessed by calling the appropriate OS versions, thus creating problems when a machine code program written under one OS version is run under a different version.

## ENTERING THE OS

The two major routes we can take to use the routines resident in the BBC OS are the OSBYTE and OSWORD routines: