

INTEGRA-β

User Guide



The information contained in this document is subject to change without notice. Computech reserve the right to make any improvements or alterations at any time without notice.

Computech makes no warranty beyond merchantability and fitness for purpose of the content of this package. Whilst every precaution has been taken to avoid the possibility of any errors in this document the publisher assumes no responsibility for the consequences of such errors or omissions. Neither is liability assumed for damages resulting from the use of material contained in this package.

All rights reserved. No part of this manual may be copied or stored in any manner whatsoever except for private or study use as defined in the Copyright Act.

Acknowledgements

All references within this manual to the BBC Model 'B' Microcomputer, the terms Tube, and Econet, refer strictly to the products bearing those designations produced by Acorn Computers Limited. Reference to AMX* means a product of Advanced Memory Systems Limited. Reference to PMS Genie* means the product of Permanent Memory Systems Limited.

-Copyright Computech 1988

Issue - Preliminary-10 with ROM 1.07

Will be automatically replaced

- Disc shortly due for issue
- Revised 9 December, 1988

1 Introduction

This expansion introduces a new range of capabilities for the BBC Model 'B' microcomputer, providing in one fully integrated package all of the conventional hardware expansion modules, that is Shadow RAM, Sideways RAM, Sideways ROM and Real-Time Clock. The particular benefits of the INTEGRA-β expansion lie in the provision of an integrated management system and inclusion of battery back-up for all RAM locations on the expansion board.

You should very carefully read the instructions relating to installation and configuration of the system to gain the full benefit of all of the features.

1-1 What you have got

This package contains the following items, which should be checked for freedom from any damage before installation or use. Instructions for installation are given in Section 9 of this guide.

INTEGRA-β Expansion Board
 INTEGRA-β OS ROM Module
 Board Connector
 Support Strips
 User Guide
 Integra Windows Disc

Warning: As with any open printed circuit board care should be taken to avoid the possibility of damage due to electrostatic discharge by ensuring you effectively discharge any high potentials before and whilst handling the board.

1-2 Numbers and Memory

Before delving into the new facilities provided by the INTEGRA-β it would perhaps be useful to gain a better understanding of the organisation of memory within your standard computer. This section is intended to provide just that.

1-2.1 Binary and Hexadecimal numbers

The CPU at the heart of your computer has signal lines connected which are used for telling the rest of your computer which memory location it wishes to access and for passing data to and from memory locations. Like the wires connected to a simple electrical switch, these signal lines can be only off or on, which in computer terms is referred to as low and high.

Since each line can be only in one of two states it was natural to use a counting system in which each digit in a number can be only 0 or 1 to represent the internal operations in a computer. This form of counting is known as the Binary system.

Unfortunately as we have all been brought up to count in a Decimal system where a digit in a number can have any one of ten values, we find counting in the Binary system very difficult and confusing, due to the very much greater number of digits required to represent most numbers. A computer however finds counting in Binary to be quite natural.

1-2.1.1 Binary numbers

In the Binary system each successive digit position in a number represents a value twice that of the digit position to its right, just as our Decimal system represents a value ten times greater.

In a computer it is customary to use groups of eight Binary digits as the most basic building block. This block is known as a Byte and is composed of eight Binary digITs or BITs, with that of the highest value on the left - as in our counting system.

For example consider the number eleven. To represent a decimal number in Binary it is simplest to set up a row of numbers in the series 1,2,4,8 etc. but in reverse order with the highest on the left. The first number should be one which is at least one half of the number to be converted. In this case it would be the number 8, so here is the row

8 4 2 1 These numbers show the decimal value of a bit
 1 0 1 1 = (1x8) + (0x4) + (1x2) + (1x1) = 11

It is customary when writing Binary numbers to include all of the digits in the basic block, in our case a byte of eight bits, so the actual binary representation of eleven would appear as 00001011. For ease in reading the byte is sometimes split into two blocks of four bits, as in 0000 1011. These blocks of four bits are called a Nibble, in other words a small byte. Actually Nibbles are the basic block for most pocket calculators.

If you haven't already worked it out the maximum value for a byte (in decimal) comes out to be 255, that is 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1. But if computers work in multiples of two, why do we have an odd number in a byte. The answer is that the least value of a byte is zero, thus there are 256 possible different values for a byte.

Going back to the early days of computers, the programmers had a difficult time as they had to use the Binary system for all of their activities, no modern aids such as compilers or even assembly language. Understandably they looked for ways to minimise the then high probability of programming errors and firstly instructed their computers to accept input in a numbering system based upon the decimal value of eight, known as Octal. This meant they were able to input three bits at a time instead of just one bit, obviously reducing their problems. As computers became available with greater capabilities it again became expedient to change the numbering system to one of greater range. This numbering system is known as Hexadecimal, having a ratio of 16 (Decimal) between adjacent digit positions. This has remained the most popular numbering system even for smaller computers such as the BBC range, the Octal system now being little used.

1-2.1.2 Hexadecimal numbers

In Hexadecimal the digits used comprise the usual 0 to 9 together with the letter A to F, A representing ten, B representing 11 and so on to F representing 15. With that method two digits can represent a Byte of 8 Bits, thus FF represents $(15 \times 16) + (15 \times 1) = 255$. It is the practice to add a prefix to a Hexadecimal number to avoid confusion with the Decimal numbering system. The prefix is '&' directly followed by the Hexadecimal number as in '&A8D7'.

One further convention is used in computer mathematics, that of using a suffix such as 'Kbytes' or 'Mbytes' to condense large numbers. However, unlike the 'K' or 'M' in Kilo and Mega, which normally means 1,000 or 1,000,000 times that used in computer practice means 1024 times or 1,048,576 times. This is to support the system of multiples of two, 1024 is two multiplied by itself ten times.

1-2.2 CPU and Memory Map

As mentioned earlier, the 6502 CPU has a number of signal lines connected, of which sixteen, known collectively as the 'address bus', are used for telling the rest of the computer which memory location it wishes to access. These sixteen lines enable the CPU to address up to 65,536 (2¹⁶) memory locations or 64K. Each of these locations may be part of one of three types of device all of which are required in a working computer system. These are as follows:

1. Input and Output (I/O) devices such as the keyboard, video display, disc interfaces etc., that allow the computer to take information from the outside world or user and return results back to the outside world or user.
2. Read-only memory or Rom. This is as its name implies is memory within the computer that can contain programmes and data, but is not alterable by the computer. There will be at least two such devices inside your computer, one being the computer Operating System Rom which is responsible for all resource management within the computer. The other an application language such as Basic or perhaps a word-processor.
3. Random access memory or Ram. This memory is probably more appropriately called Read and Write Memory. This is memory that is available to the user in support of programmes and data, the content of which can be altered by the processor.

In your BBC Model 'B' computer, half of the 65,536 memory locations are occupied by Ram just under half contain the Operating System and Language Rom's whilst the remainder is occupied by I/O devices, including the Filing System. As you can see, there is no space left over for any extensions to your computer, including Disc storage systems and other applications.

Fortunately it is possible to get around this apparent problem by the use of bank switching techniques where one of up to sixteen 16K blocks of memory can be switched into one particular 16K section of the computer memory map. Bank switching also applies to the switching of so-called Shadow Memory in place of video memory so as to retain a reasonable amount of User Ram for working space. These aspects are covered in the following section dealing particularly with the INTEGRA-β.

1-3 The INTEGRA System

The operating system in the BBC Model 'B' computer is capable of controlling up to 16 banks of sideways Rom, allowing a number of Rom-based applications and other extensions to your computer to be instantly available. However, the hardware inside your computer only provides for four sockets into which these Roms may be inserted.

The INTEGRA-β provides the remaining twelve banks in the form of eight Rom sockets and four banks of sideways Ram.

Use of sideways Ram in place of sideways Rom allows images or copies of a Rom to be stored on a floppy or hard disc and be loaded into a sideways Ram bank when required, saving the user the trouble of opening up the computer for the awkward job of inserting a new Rom. Unfortunately many sideways Ram expansions suffer from the problem that when you switch your computer off, all the contents of the Ram are lost. To get around this problem, the INTEGRA-β has a small rechargeable battery to maintain a holding voltage on the Ram so that the contents will remain intact when the computer power is off.

In this way you gain the advantages of Rom in that the software in sideways Ram is immediately available when you switch on your computer, and like normal Ram, the contents may be changed at any time.

The INTEGRA-β also features another type of bank switched Ram, commonly known as 'Shadow Ram' This Ram occupies the same area of memory as the video display Ram. The purpose of this memory is to replace the memory lost to the video display, thus giving the user a constant amount of normal user Ram for use by applications, the size of which is not dependant upon the current display mode. This shadow Ram is controlled by some software within the INTEGRA-β OS Rom, which briefly works as follows. When an application is running, this shadow Ram is switched into the computer's memory map and is then available for use by the application.

Whenever the application or the machine operating system needs to write some text or graphics on the screen, the shadow Ram is switched out and the screen memory is switched back in, the information is then written to the screen Ram and the screen ram is then switched out and the shadow memory is switched back in again. This process is completely transparent to the user and most applications. Note however that the initiative for use of shadow Ram must be contained within the application.

The final area of Ram in the INTEGRA-β is a 12Kbyte block called Private Ram. This Ram is used to provide the workspace required by the INTEGRA-β OS Rom. It resides next to the sideways Rom and Ram, but is controlled separately allowing parts of the private Ram to overlay areas of sideways Rom and Ram.

As well as the additional memory areas, the INTEGRA-β also includes a real time clock with a hundred-year calendar. This clock will continuously tick over even when the computer is switched off, allowing software such as accounting packages to be able to read the correct date.

The clock also contains fifty Ram registers. These in conjunction with seventy-eight bytes in the Private Ram are used for storage of some Operating System variables employed in configuring the way in which your computer will initialise.

To summarise, the additional memory areas are as follows

Sideways Memory	used for additional ROM software, or using RAM, for storage of programmes and / or data or as printer buffer.
Shadow Memory	replaces working RAM used for screen memory where software provides support.
Private Memory	used for INTEGRA OS workspace and partially allocatable as printer buffer.
RTC Memory	used for storage of configuration data.

1-4 Syntax

Syntax employed within this guide follows the practice associated with the principal BBC microcomputer manuals and guides and is set out below for reference.

[legend]	indicates a keypress of the named key
<????>	indicates required user input, alpha or numeric
(<????>)	indicates optional user input
(,<????>)	indicates additional optional parameters which are delimited by commas, (spaces if no comma inside brackets)
(X/Y/Z)	the slash indicates OR, (X OR Y OR Z) in this example, which may appear in relation to an input or a result

1-5 Getting started

The INTEGRA-β provides brings an extensive range of additional facilities to your computer. The purpose of this section is to give you a quick run through many of the facilities provided without getting bogged down in all the why's and how's.

The first thing you will want to do after installing your INTEGRA-β will be to set the time and date. Enter the following commands:

```
*TIME = 13:33:35 [RETURN]
```

```
*DATE = 15/9/88 [RETURN]
```

You will of course enter the time and date by your watch or an appropriate time signal such as the radio or television or the British Telecom talking clock, the number of which will be in your telephone directory.

If the system responds with 'Invalid time' or 'Invalid date' then simply retype the command, ensuring that you enter the time and date correctly.

You may now view the current time and date by entering:

```
*TIME [RETURN]
```

This will display the current time and date as follows:

```
Thu, 15 Sept 1988.13:35:02
```

The next thing to do is to change some of the configuration settings to allow the use of Shadow RAM. Enter the following commands:

```
*CONFIGURE OSMODE 4 [RETURN]
```

```
*CONFIGURE MODE 128 [RETURN]
```

```
*ROMS [RETURN]
```

This last command will produce a list of the ROMs installed in your computer. Find the BASIC ROM and note the number to the left of it. This is the bank number in which it currently resides. Enter this number in place of the 'n' in the command line below.

***CONFIGURE LANG n [RETURN]**

Now press [CTRL] + [BREAK] to reset your computer. You should now see a start up message similar to that shown below. The actual message may vary according to the ROMs and amount of RAM you have in your computer.

```
Computech INTEGRA-B 128K
Acorn DFS
BASIC
```

You will notice that the display is now in mode 0, however if you enter:

PRINT HIMEM-PAGE [RETURN]

You will see that BASIC now has 26368 bytes available to it with the DFS only installed. If you have the Acorn ADFS then this number will be 24832. Normally you would only have 5888 bytes with the DFS or 4352 with the ADFS. What you have actually done with the above commands is to tell the system to select Shadow mode 0 and select the BASIC ROM as the default language.

Now insert a formatted disc in drive 0 and enter the following command to save a copy of the BASIC ROM to disc. Note that you should replace the 'n' in the command with the bank number of the BASIC ROM. This number can be found using the *ROMS command as above.

***SRSAVE BASIC 8000 +4000 n Q [RETURN]**

If you now enter *CAT [RETURN], you should see the filename 'BASIC' in the disc catalogue. When a ROM is stored on disc, it is called a 'ROM image'.

To load the BASIC ROM image into Sideways RAM bank number 4, enter:

***SRLOAD BASIC 8000 4 QI [RETURN]**

The letter 'I' tells the INTEGRA-β OS to install the ROM image so that the computer's Operating System will recognise it.

If you now enter *ROMS [RETURN], you will find that you have two BASIC ROMs in your computer!

Enter *UNPLUG n I [RETURN] where n is the bank number of the original BASIC ROM, followed by *ROMS [RETURN].

You will now see that the original BASIC ROM has two asterisks in the brackets instead of the letter 'L'. This means that the ROM is 'unplugged'.

When a ROM is 'unplugged' in this manner, the computer's Operating System will no longer recognise it's presence.

A Sideways ROM image can be quickly removed from Sideways RAM using the *SRWIPE command. If you enter *SRWIPE 4 [RETURN], followed by *ROMS [RETURN], you will see that the Sideways RAM copy of the BASIC ROM has disappeared.

To reinstate the unplugged BASIC ROM, enter:

INSERT n I [RETURN]

where 'n' is the bank number of the ROM. The I parameter tells the INTEGRA-β OS to do the job immediately. If omitted, then the ROM will be reinstated when you next press BREAK.

Having worked through these simple examples it is now time to turn to the following sections of the User Guide and explore the large range of options which are available with the INTEGRA-β.

2 Configuration

The INTEGRA-β Operating System provides a facility to allow certain aspects of the host computer to be permanently configured. This minimizes the need to enter a series of *FX commands every time the computer is switched on, especially when a particular non-standard start-up configuration is desired. In all cases these options set the default following power-up or a hard reset.

Two operating system commands are used to control this facility, one to input required options and the other to non-destructively read the configured set-up.

*CONFIGURE <parameter1> (<parameter2>)....

*STATUS < parameter

*CONFIGURE is used to set a configuration option whilst *STATUS displays the configured status of a single option.

As an example, to set the default screen mode to Shadow Mode 0, the following entry should be made.

*CONFIGURE MODE 128 [RETURN]

To read the Default screen mode, enter

*STATUS MODE [RETURN]

2-1 Configuration Options

There are twenty-four separate configuration options provided, some of which are equivalent to operating system commands provided in the BBC Model 'B' Operating System (MOS) whilst the majority are new commands providing considerable simplification of the start-up process.

The configuration options are of two general types, those requiring numeric parameters and those whose action is entirely defined by the text entry.

2-1.1 Parameter Options

This section includes support for alarms, data communications, keyboard actions, display, disc drive and printer. Each command requires at least one defining parameter and is used following '*CONFIGURE'.

ALARM <0 1>

Entering ALARM 0 sets the alarm system, after alarm acknowledgement, to reset to 'OFF' status, whilst entering ALARM 1 sets the alarm system, after alarm acknowledgement to reset to 'ON' status.

BAUD <1-8>

Sets the default Baud Rate as follows,

1	75	baud
2	150	baud
3	300	baud
4	1200	baud
5	2400	baud
6	4800	baud
7	9600	baud
8	19200	baud

BEEP <vol>,< pitch >,<max>,<min>

<vol>	Controls alarm volume in the range 0 to 1
0	selects half volume, equivalent to SOUND n,-10,p,d
1	selects full volume, equivalent to SOUND n,-15,p,d
<pitch>	sets the alarm tone pitch in the range 0 to 3.
0	selects a tone equivalent to SOUND n,v,90,d
1	selects a tone equivalent to SOUND n,v,130,d
2	selects a tone equivalent to SOUND n,v,176,d
3	selects a tone equivalent to SOUND n,v,210,d
<max>	sets the maximum time for alarm sounding, if not acknowledged, in the range 0 - 3.
0	selects 7.5 seconds
1	selects 15 seconds
2	selects 30 seconds
3	selects 60 seconds
<min>	sets the minimum time for alarm sounding although acknowledged. This is included to cover situations where use of the alarm acknowledgement key pair may coincide in time with an 'alarm-due' event, resulting in the user not realising an '*alarm due' had occurred. Requires a value in the range 0 - 3.
0	selects 0.5 seconds
1	selects 1 second
2	selects 2 seconds
3	selects 4 seconds

DATA <0-7>

Sets the default serial data format as follows,

	Bits	Parity	Stop Bits
1	7	even	2
2	7	odd	2
3	7	even	1
4	7	odd	1
5	8	none	2
6	8	none	1
7	8	even	1
8	8	odd	1

DELAY <0-255 >

Set Keyboard Repeat initial delay, (cf *FX11,n)

FDRIVE <0-3>

Sets the Floppy Disc drive Stepping Rate. Refer to your Disc Drive and Disc Interface Manuals. 0 represents a higher rate than 3.

FILE <0-15>

Assigns the Default Filing System by ROM bank location number.

IGNORE <0-255>

Set the default character to be ignored by the printer, (cf *FX6,n)

LANG <0-15>

Sets the default language to be entered following a hard reset. If the specified ROM, (by Bank location), does not contain a 'language then the 'No Language Environment', (NLE), is entered.

MODE <0-7/128-135>

Sets the default screen mode. Parameters over 127 are shadow modes.

OSMODE <0-4>

Sets the default Operating System emulation mode. There are five operating system modes available. (Rom id's W...Z = = 4...7, except for Osmode 2 where 12...15 applies).

0	Standard BBC Model B
1	BBC Model B with Shadow RAM
2	BBC Model B+ 128 emulation
3	BBC Master 128 emulation
4	BBC Model B+ emulation
5	BBC Master Compact emulation

PRINTER <0-4>

Sets the default printer type. (cf *FX5,n)

REPEAT <0-255>

Sets keyboard auto-repeat rate. (cf *FX12,n)

TV <0-255>,<0-1>

Sets the default Vertical Display displacement for the monitor and Interlace Off/On mode. (cf *TV n,i)

¹Language can be Basic, a wordprocessor, database or similar.

2-2.2 _____ Text Options

The configuration commands in this section require only a text entry to define the action. All except for SHCAPS have opposite variants.

BOOT

!BOOT always executes unless [SHIFT] + [BREAK] pressed.

BST

This option enables automatic clock time changes at the beginning and end of British Summer Time (BST). These will act as follows, during the last Sunday in April the time will increment from 01:59:59 to 03:00:00, whilst during the last Sunday in October the time will decrement from 01:59:59 to 01:00:00. Disabled by NOBST.

CAPS

Sets [CAPS LOCK] initially to ON.

NOCAPS

Sets [CAPS LOCK] initially to OFF.

NOBOOT

!BOOT only executes when [SHIFT] + [BREAK] are pressed.

NOBST

Disables the BST configuration.

NOSHX

Prevents the copying of memory content as described under SHX.

NOTUBE

Inhibits initialisation of and communication with connected Co-Processor.

SHCAPS

Sets [CAPS LOCK] initially to ON but permits [SHIFT] to change case.

SHX

This command enables the copying of the contents of Main memory into Shadow memory when the screen mode is changed from a non-Shadow mode to a Shadow mode and vice-versa.

Note: Programmes cannot be Auto-Run with SHX enabled.

TUBE

Allows communication with connected Co-Processor.

3 Utility Commands

The INTEGRA-β Operating System provides a range of Utility Commands to facilitate the use of enhancements included in the system. Some of these are similar to commands provided in the BBC range of computers, others are new commands specifically for this expansion.

Those commands which have a direct counterpart in the Configuration Settings permit immediate change of the Configuration Setting. Upon a 'hard break' or computer 'power-up' the actual setting will revert to the Configuration Setting for that command.

***APPEND <fsp>**

This command is similar to the *BUILD command. The only difference between *BUILD and *APPEND is that *APPEND will add text onto the end of an existing ASCII file, whereas *BUILD always create a new file.

As an example to create an EXEC file to initialise the computer for Auto-running a BASIC programme on power-up, enter

```
*BUILD !run [RETURN]
1: *BASIC [RETURN]
2: MODE 128 [RETURN]
3: PAGE=&3000 [RETURN]
4: *BOOT *BASIC | MMODE128 | MPAGE=&3000 | MOLD | MRUN | M
[RETURN]
5: [ESCAPE]
```

All you would have to do now is to LOAD the required BASIC program and it will run when you next switch your computer on.

An alternative, is to change this file into a dedicated EXEC file to initialise your program, called 'myprog'. To do this, you will need to add an extra line to your EXEC file. This is done by typing

```
*APPEND !run [RETURN]
```

The list of commands on lines 1 to 4 will now be displayed, and you are prompted for line 5 as follows

```
5: LOAD 'myprog' [RETURN]
6: [ESCAPE]
```

If you now enter

```
*TYPE !run [RETURN]
```

you will find that line 5 has been added to the file.

***BOOT (<text>)**

This command is similar to a *KEY10 command, the difference being, that it effectively programs the mains power switch instead of the break key consequently the commands in <text> will only function after the computer is switched on, or after a simulated power-up.

For example, to Auto-Run a BASIC program on power-up, enter the following commands:

```
*BASIC [RETURN]
MODE 128 [RETURN]
PAGE=&3000 [RETURN]
LOAD myprog [RETURN]
```

Where 'myprog' is the name of your programme, then enter:

```
*BOOT *BASIC | MMODE128 | MPAGE=&3000 | MOLD | MRUN | M
[RETURN]
```

When the computer is next powered-up, your program will be executed.

***BUFFER (<size> / <#> (<id>)(,<id>...)**

This command is used to set the size of the printer buffer, which may be placed in Private RAM if sized at 8K or must be located in Sideways RAM for sizes from 16K to 64K.

Two forms of selection of the buffer are available, one providing automatic selection whilst the alternate method allows specific RAM banks to be chosen.

The first method requires specification of only one parameter and results as follows:

```
*BUFFER 0      8K buffer in Private RAM
*BUFFER 1      16K buffer in Sideways RAM
*BUFFER 2      32K buffer in Sideways RAM
*BUFFER 3      48K buffer in Sideways RAM
*BUFFER 4      64K buffer in Sideways RAM
```

When using this method for 16K and above buffers the Sideways RAM banks allocated will be the lowest priority banks available which do not contain a ROM image.

The alternative method which allows specification of the Sideways RAM banks to be used for the buffer location requires a (#, to be input in place of the size specifier and to be followed by up to four Sideways RAM bank 'id's delineated by commas. For example,

```
*BUFFER # 4,5,6,7
```

will set up RAM banks 4, 5, 6 & 7 as a 64K printer buffer.

If this method is used you should ensure the particular banks do not contain a ROM image by clearing the banks with *SRWIPE.

***BUFFER ?**

This command should be used to determine the current allocation of the printer buffer. The result returned will identify the size and location(s). For example, after entering *BUFFER 3 [RETURN], to obtain a 48K buffer, *BUFFER ? [RETURN], will return the following message

```
48K in Sideways Ram Banks 4, 5, 6
```


***CLOAD < filename >**

This command is used to load a previously saved configuration set-up file. Multiple configuration files can be kept on disc and loaded as required.

***CSAVE <filename>**

This command is used to save a configuration setup to a named file for subsequent use with particular applications.

***GOIO <addr>**

This command is used to execute machine code at the address specified.

***GOIO (<addr>)**

This command, which requires the brackets to be included, is used to execute a machine call via the specified vector, located at address.

***INSERT <id> (<id>...) (I)**

This command performs the opposite function to the 'UNPLUG' command. Multiple parameters can be included and use of the optional 'I' parameter will result in immediate effect, excepting for some ROMs which require to organise workspace.

For example, to enable the ROMs in banks numbers 4 and 7, enter

```
*INSERT 4,7 I [RETURN]
```

Note: The INSERT and UNPLUG commands (used without the 'I' parameter) only come into effect after pressing [BREAK], also they are non-volatile, this means that a disabled ROM will remain disabled even if you switch the computer off, then on again.

***NLE**

This command is used to enter a 'no language environment'. All commands input will be sent to the Operating System.

***OSMODE (<0-5> / ?)**

This command allows the OS emulation mode to be instantly changed. It is intended to cater for a change to an application which will run only in a particular Osmode. Most usefully incorporated within an 'Exec' file used to initialise an application. The Osmode setting will be maintained until the next hard reset.

The OS emulations provided are, (Rom id's W, X, Y, Z are 4, 5, 6, 7)

- | | |
|---|--|
| 0 | standard BBC model B without Shadow Ram |
| 1 | standard BBC model B with Shadow Ram |
| 2 | BBC models B+ and B+ 128. (Rom id's W...Z are 12...15) |
| 3 | BBC Master 128 |
| 4 | BBC Models B+ and B+ 128. |
| 5 | BBC Model Master Compact |

To select the BBC B+ emulation mode, enter

```
*OSMODE 2 [RETURN]
```

To display the current Osmode, enter

```
*OSMODE ? [RETURN]
```

Note: It is not possible to switch between Osmode 0 and Osmodes 1, 2 or 3 when the printer buffer is not empty.

***PRINT <fsp>**

This command is similar to the *TYPE command, the difference being that text is sent to the printer only, and not the screen.

***PURGE (<ON>/<OFF>/<?>)**

These commands are employed to purge the printer buffer directly, or via BBC Operating system action or calls after enabling the command.

*PURGE [RETURN]	always purges the printer buffer
*PURGE ON [RETURN]	allows operating system to purge the buffer
*PURGE OFF [RETURN]	prevents MOS from purging the buffer
*PURGE ? [RETURN]	returns PURGE status

The BBC Operating System when enabled as above can purge the buffer either by use of the [ESCAPE] key or by typing <*FX21,3>.

Note that [BREAK] will always purge the buffer.

***ROMS**

This command displays a list of active ROMs installed in your computer. The entries in the list are displayed in the form

```
XX (S/L**) Title
```

Where XX is the Sideways ROM or RAM bank number, S indicates the ROM has a service entry, L indicates the ROM is a language, whilst ** indicates the ROM is disabled. Title gives the ROM name.

***SHADOW (<0-1>) and *FX114,<0-1>**

This command is used to force Shadow mode at the next display mode change. Normally when you type MODE 0 in BASIC, you will find that you have just under 5 kbytes available for your program and its data. By using an equivalent Shadow mode, you will have memory available up to address &7FFF, which leaves you with about 25.5 kbytes, and still be able to use high resolution graphics. There are two methods for selecting Shadow modes, one is to add 128 to the mode number, the other is to type in any of the following commands,

```
*SHADOW [RETURN] or
*SHADOW 0 [RETURN] or
*FX114 [RETURN] or
*FX114,0 [RETURN]
```

before selecting the display mode you wish to use. After using one of the above commands, selecting display Mode 0 will select Mode 128. To restore mode selection back to normal, enter

```
*SHADOW 1 [RETURN] or
*FX114,1 [RETURN]
```

*SHADOW ?

Used to determine the current setting of *SHADOW, returning a value of zero or non-zero according to the last setting.

*SHX (ON/OFF)

This command when set to 'ON' enables copying of the content of Main memory to Shadow memory when the screen mode is changed from a non-Shadow mode to a Shadow mode and *vice-versa*. When set to OFF the copying is disabled.

*SPOOLON <fsp>

This command is similar to the *SPOOL command, the only difference being that *SPOOLON appends to the end of an existing file, rather than creating a new file.

*TUBE (ON/OFF)

This command may be used on computers which have a Co-Processor attached. It is used to temporarily over-ride the TUBE / NOTUBE configuration option, allowing non-tube applications to be started from an 'exec' file when the Co-processor is active and *vice-versa*. The effect of this command is maintained until the next hard reset.

To suspend operation of the Co-Processor, enter

```
*TUBE OFF [RETURN]
```

To allow the Co-Processor to continue, enter

```
*TUBE ON [RETURN]
```

Note: If no Co-Processor is connected, the error message 'No Co-Processor!' is displayed.

*UNPLUG <id> (<id>...) (1)

This command is provided to 'remove' a Sideways ROM from your computer, without physically unplugging it. You may wish to do this in the case of *<command> clashes, or in the case of a ROM proving to be incompatible with other software. Multiple parameters are allowed and use of the optional 'I' parameter provides immediate effect. (Certain ROMs, for example filing systems may not respond to the 'I' parameter and require the use of [BREAK]).

For example to Disable the ROM in bank number 4, enter

```
*UNPLUG 4 [RETURN]
```

X <command>

This new command prefix is provided to allow the operation of utility commands and 'pop-up' programmes which do not function in Shadow mode. It operates by unconfiguring Shadow mode, performing the stated command, then reconfiguring Shadow mode. Typical usage is as the examples,

```
*X*SAVE <filename> FFFF3000 + 5000 [RETURN]
```

will save the contents of mode 128 screen memory.

```
*X*LOAD <filename> FFFF3000 [RETURN]
```

will load <filename> into screen memory.

```
*X*GENIE [RETURN]
```

will allow running of PMS Genie from Shadow mode.

4 Sideways Ram Utilities

Sideways Ram is perhaps the most useful and at the same time under-utilised facility provided with memory expansion boards. This is partly due to the general omission on such boards of provision for both an adequate amount of Sideways Ram, together with battery back-up and a comprehensive system of management.

The INTEGRA-β has the capability of meeting all these requirements and provides management which will be familiar to users of the Models 'B+' and Master computers.

As this area is, generally, unfamiliar ground to those accustomed only to the Model 'B' computer this section includes introductory material covering Sideways Ram bank identification and addressing of Sideways Ram. See Section 8 - Technical Reference for information on hexadecimal numbers.

Following the introductory section, which may be skipped by those familiar with the subject, the commands provided for management of Sideways Ram are referenced under four headings, Allocation, Loading & Saving, Copying, and Clearing. Six of these are equivalent to the commands of the same name provided in the Master series and two additional commands permit clearing of the data header information or total content of a Ram bank.

Allocation of a bank or banks of Sideways Ram as a printer buffer is separately controlled - see reference to *BUFFER on page 3-2.

4-1 Introduction

One of the most useful features of the standard BBC Model 'B' computer is the implementation of a system of paged banks of memory permitting Rom based utilities, filing systems and applications to be present in your computer in excess of the normal directly addressable memory limit of 64k and allowing them to be almost instantly available for use as and when required. The process of switching between the Roms is not only transparent to the user, but also to most applications. The computers' Operating System is capable of controlling sixteen such banks of memory, however only four sockets are provided inside the standard computer for installation of Roms. The INTEGRA-β provides a considerable enhancement to this facility in a number of ways, both with hardware and additional software.

On the hardware side, the INTEGRA-β provides an additional eight sockets for Roms, however, unlike the Rom sockets in the standard BBC model B, four of these sockets can accept 32k Ram chips as an alternative, allowing the system to accommodate a total of 192 Kbytes of Sideways Ram. The remaining four banks are occupied by Ram.

One major advantage of using Sideways Ram in place of Sideways Rom is the facility to maintain programme material with included Rom header information

on disc-based files which can then be loaded into the Ram when required, avoiding all the problems and risks of replacing Rom chips. However, with most Sideways Ram systems, the contents of the Ram are lost as soon as you switch your computer off. This is not the case with the INTEGRA-β as it uses a small battery to keep the Ram chips supplied with sufficient power for them to hold their contents when the computer is switched off.

On the software side, facilities are provided via '*' commands for loading and saving the contents of Sideways Rom or Ram to or from disc and using Sideways Ram for data storage or as a large printer buffer. Additional Sideways Rom management is provided by further '*' commands for defining the default language and filing system Roms and disabling a Sideways Rom to get round software clashes.

4-1.1 Bank and Memory Addressing

Three forms of addressing can be used for Sideways memory banks under the INTEGRA-β system. These are, respectively, Physical bank numbers in the range 0 - 15, Logical bank letters in the range 'W' to 'Z' and Address references in the range &0000 to &FFFF. The first two forms of addressing are often known as 'Direct' or, preferably, 'Absolute' addressing, whilst the third method is known as 'Psuedo' addressing.

Direct addressing involves the use of rom id's as shown in the table below for four banks of Sideways Ram which may be paged into the main memory between addresses &8000 and &BFFF. The separate 16K Ram banks are treated as distinct entities.

Psuedo addressing allows one or more banks of Sideways Ram to be treated as a single contiguous block of memory, referenced by means of the particular 'psuedo' addresses tabled below. A memory address range of &0000 to &FFBF is utilised. In this implementation, rom id's must not be specified in the psuedo mode.

Rom id's can be given in two separate forms, as a non-machine-specific id or as a machine-specific id. References for addressing in the machine-specific manner are different for the Model 'B+ 128' and Master series as shown in the table.

ROM ID	B+ 128	MASTER	¹ Psuedo Address
Z	15	7	&FFBF &BFD0
Y	14	6	&BFCF &7FE0
X	13	5	&7FDF &3FF0
W	12	4	&3FEF &0000

¹The actual addresses will be assigned in the order of allocation decided by the user. For example if allocation is ordered 13, 7, 12, then addresses used will be &3FEF-&0000, &7FDF-&3FF0, &BFCF-&7FE0.

When using the 'psuedo' addressing method, 16 bytes at the start of each bank of Sideways Ram are used for Rom Header information. The effective content of each bank is therefore 16,368 bytes, not 16,384 bytes.

Although the blocks of header information will interrupt the continuity of data storage this is transparent to the user as the INTEGRA-β OS automatically makes the necessary adjustments during transfers, reads, or writes.

As well as specifying a Ram bank by its physical number, i.e it's socket number, it is possible to specify it by a logical bank number. The logical bank numbers are specified by the letters 'W', 'X', 'Y' and 'Z' and can be used anywhere that a physical bank number is required. Use of logical bank numbers allows software to use Sideways Ram without knowing the physical bank numbers that it may be using and thus be able to run on the BBC B+ 128 and Master series computers, as well as the INTEGRA-β, without alteration.

On the BBC B + 128 and Master series computers, the logical to physical bank number relationship is fixed, however the INTEGRA-β OS extends the concept to allow any four Ram banks to be accessed via logical bank numbers. This has the advantage that Software is then able to use any four banks that are free and is not restricted to using banks 4 to 7 as on the Master series computers or 12 to 15 as on the BBC B + 128.

The relationship between logical bank numbers and physical bank numbers is set by the '*SRSET' command. This command requires a list of physical bank numbers to be specified, from which, up to four banks of free Ram are automatically selected and allocated logical numbers.

For example, suppose you have Ram installed in banks 12 to 15 as well as banks 4 to 7, but banks 4, 5, 12 and 14 contain Rom images and that for the sake of compatibility with the BBC B+ and Master series computers, you had chosen to use logical bank numbers as parameters to the *SRDATA command. If you now entered:

```
*SRSET 4,5,6,7,12,13,14,15 [RETURN]
```

The INTEGRA-β OS will search through this list and allocate 'W' to bank 6, 'X' to bank 7, 'Y' to bank 13 and 'Z' to bank 15. This would allow you to use *SRDATA W,X,Y,Z' to allocate 64 Kbytes of Sideways Ram for data storage. Normally, this command would be used in a '!BOOT' file.

Sideways Rom or Ram bank numbers may be specified in a number of ways. The following are all valid for specifying bank number 13: '13' 'D' '&D'. Also in the case of the above Ram setup, following the '*SRSET' command, you can use the letter 'Y'.

Rom id lists are specified by separating each Rom id with a comma or a space. In addition a '*' symbol may be given after the list, which effectively causes an operation to be performed on all banks, except those specified in the list, thus

'*' on it's own causes an operation to be carried out on all banks if possible. For example, to 'unplug' all Roms in the '*ROMS' list given above except the INTEGRA-β OS, BASIC and the DFS, the command '*UNPLUG 1, 3, 14 * T' could be used. Equally the above '*SRSET' command could have been entered as '*SRSET *' and would have achieved the same effect.

4-1.2 Sideways Memory Management

A list of the Roms and Rom images installed in your computer may be obtained by entering,

```
*ROMS [RETURN]
```

This will produce a list similar to the following, depending upon Roms installed in your computer,

```
14 ( L ) BASIC
13 (SL) AMPLE Nucleus V1 .00
12 ( S ) Stop Press support 1.9
11 ( S ) AMX STOP PRESS 1.9
10 ( S ) The Publisher
9 (SL) INTER-WORD 1.02
8 (SL) INTER-BASE 1.00a
7 (**) INTEGRA Windows 1.00
6 ( ) ROM
5 ( ) RAM
4 ( ) RAM
3 (SL) INTEGRA-B OS 1.07
2 ( S ) Acorn ADFS 1.30
1 ( S ) DFS 2.26
0 ( S ) PMS GENIE v1.02
```

The first column contains the bank or socket number in which a Rom is installed. The next column contains 'S' if the Rom responds to service calls such as *command etc, and / or 'L' if the Rom is a language or application. You will notice that bank 7 contains '**' in this column. This indicates the bank contains an 'UNPLUGged' Rom. Banks with blanks in this column usually contain some kind of special header to indicate data storage of some kind. The final column contains the Rom titles.

You may have noticed that number 15 is missing. This means that this bank does not contain any type of Rom header that will be recognised by the computer's Operating System or the INTEGRA-β OS. There are two special titles in the list for banks 4, 5 and 6. The title 'RAM' is inserted in Sideways Ram banks following the *SRDATA command and indicates that this bank is currently being used for data storage. The title 'ROM' is inserted in Sideways Ram following the *SRROM command which releases a bank from data storage.

During the course of using your computer you may have come across situations where more than one Rom is capable of responding to an entered

*command, sometime with the result that you may enter a command only to get a totally unexpected response. In more severe cases you may find that the presence of a particular Rom causes strange and often undesirable effects with other software.

Recently, many Rom software producers are incorporating special optional prefix characters into their commands to ensure that a command is sent to a specific Rom. In the case of the INTEGRA-β OS the character 'I' may be used to prefix any *command to ensure that it is recognised by the INTEGRA-β OS only, so the above '*ROMS' command could have been entered as '*IROMS'.

When the use of prefix characters is not available, the only solution has usually been to physically remove the offending Rom from your computer which is a time consuming and for some people, a rather nerve shattering process. The INTEGRA-β OS provides a software equivalent of removing Roms from your computer which functions by causing the Operating System to completely ignore the presence of a Rom. The commands associated with this facility are '*UNPLUG' and '*INSERT'

With reference to the above *ROMS list, if you enter,

```
*UNPLUG 1 I [RETURN]
```

and then enter,

```
*ROMS [RETURN]
```

you would see that the entry in the list for the DFS Rom would be:

```
1 (**) DFS 2.26
```

If you now entered,

```
*DISC [RETURN]
```

you would get the error message 'Bad command'.

The I in the above '*UNPLUG' command is an optional parameter which tells the INTEGRA-β OS to 'unplug' the Rom immediately. If you had omitted the letter 'I', then the DFS Rom would not be 'unplugged' until the next time you pressed the [BREAK] key.

To reinstate an 'unplugged' Rom you use the '*INSERT' command. So to reinstate the DFS Rom you would enter,

```
*INSERT 1 I [RETURN]
```

Again, the 'I' parameter tells the INTEGRA-β OS to 'insert' the Rom immediately. The use of the 'I' parameter with '*INSERT' is also useful for initialising Rom images that are transferred to Sideways Ram using '*SRWRITE' as in that context it provides an equivalent to the 'I' parameter following the '*SRLOAD' command.

It should be noted that filing systems and other Roms which claim workspace by raising the value of BASIC'S 'PAGE' variable or OSHWM require that the [BREAK] key be pressed in order to 'insert' or 'unplug' them correctly. This is in order to ensure that relevant workspace is claimed and released correctly.

Both the '*UNPLUG' and '*INSERT' commands can accept a list of bank numbers on which they are to operate as described in Section 4-1.6.

4-2 _____ Using Sideways Ram

Having described the addressing and identification of Sideways Ram the next area to be considered is the usage to which this region of memory can be ascribed.

Probably the most common usage is the holding of Rom Images followed by permanent or temporary storage of data. A small number of available commercial programmes will use the latter facility but perhaps the best example is INTEGRA Windows.

4-2.1 _____ Rom Images

A Rom image is a copy of the contents of a Sideways Rom that is stored on disc, allowing it to be loaded into the Sideways Ram whenever the facilities contained within the Rom are required. Unlike machine code programmes that are loaded into main Ram, Rom images loaded into Sideways Ram become semi-permanently resident, with all their provided facilities remaining available for as long as the Rom image is present in the Sideways Ram, even after the computer has been switched off.

Rom images are created using the '*SRSAVE' command. For example to create a Rom image of the BASIC Rom in socket 14 in the above Rom list, enter,

```
*SRSAVE BASIC 8000 +4000 14 Q [RETURN]
```

```
*SRSAVE BASIC 8000 BFFF 14 Q [RETURN]
```

8000 is the start address in hexadecimal of the data to be saved. This is the same for all Rom images.

In the first example 4000 is the number of bytes to be saved in hexadecimal. We know the number of bytes because BASIC is a 16Kbyte Rom which is 16384 bytes. Some Roms contain only 8Kbytes, in which case the length would be specified as 2000. If you are unsure about the length of a Rom then use 4000. Note that in the first example, a '+' symbol separates the start address from the length. This tells the INTEGRA-β OS that the length parameter is the actual number of bytes rather than the end address ('BFFF') as in the second example.

The '14' is the bank number or 'ROMid' from which the Rom image is to be created.

The final parameter, 'Q' tells the INTEGRA-β OS to use main memory as a transfer buffer. Unfortunately, filing systems are not capable of reading the contents of Sideways Rom or Ram, therefore it is necessary to copy the contents of Sideways Rom or Ram into main memory so that the filing system can access it. The alternative method is to read the contents of the Sideways Rom a single byte at a time, saving each byte individually to disc. Unfortunately this method takes rather a long time, but should it be necessary to avoid corruption of main memory, then this method can be used by omitting the letter 'Q' from the end of the command line.

It should be noted that it is not possible to create usable Rom images of PAL-PROM devices such as INTER-WORD' from Computer Concepts as these contain special switching mechanisms that allow a number of 16Kbyte blocks to occupy a single Rom bank.

4-1.3 Loading Rom images

Once you have created a Rom image, you will need to load the image into Sideways Ram and initialise it before it can be used. This is done using the '*SRLOAD' command. For example, to load the 'INTEGRA Windows' Rom image into Sideways Ram bank number 7 and initialise it, you would enter:

```
*SRLOAD WINDOWS 8000 7 QI [RETURN]
```

'WINDOWS' is the filename of the INTEGRA Windows Rom image. '8000' is the address to which the Rom image is to be loaded, which will be &8000 for all Rom images. '7' is the Sideways Ram bank number. 'Q' tells the INTEGRA-β OS to load the image as quickly as possible, using main memory as a transfer buffer in the same way as the '*SRSAVE' command.

The 'I' tells the INTEGRA-β OS to inform the computers Operating System that this Rom exists in the computer so that it may be used immediately. If this parameter was omitted then the Rom would not be recognised until the [BREAK] key was next pressed. In the case of Roms which claim workspace by raising the value of BASIC'S PAGE or OSHWM, the [BREAK] key will always need to be pressed to initialise them.

An alternative method of loading and saving Rom images is also available, using the '*SRREAD' and '*SRWRITE' commands which transfer data between Sideways Ram and main memory. In the case of transferring Rom images between the current Filing system and Sideways Ram, the general principle is to transfer to the image to a specified area of main memory using '*LOAD' or '*SRREAD' and then transfer the image to Sideways Ram and the filing system using '*SRWRITE' or '*SAVE' respectively.

For example, the BASIC Rom image could have been created with the following commands:

```
*SRREAD FFFF4000 +4000 8000 14 [RETURN]
```

```
*SAVE BASIC FFFF4000 +4000 [RETURN]
```

Likewise, the INTEGRA Windows Rom image could be loaded and installed with the following commands:

```
*LOAD WINDOWS FFFF4000 [RETURN]
```

```
*SRWRITE FFFF4000 +4000 8000 6 [RETURN]
```

```
*INSERT 14 I [RETURN]
```

With the '*SRREAD' and '*SRWRITE' commands the first parameter is the start address in main memory to or from which data is transferred. The second parameter following the '+' symbol is the number of bytes to be transferred. Like the '*SRSAVE' command, this parameter may alternatively be specified as the end address of the block in main memory, which in the above examples would be &FFFF7FFF. The final two parameters make up an absolute Sideways Ram address with the '6' being the Ram bank number.

The '*SRWRITE' command can be a very useful aid to machine code programmers who write Sideways Roms using the BASIC assembler as it is possible to write a set of source code files, each CHAINing the next file, with the object code from each file being transferred to a Sideways Ram bank using '*SRWRITE'.

4-2.2 Data Storage

The INTEGRA-β OS provides facilities to aid in the use of Sideways Ram for storage of data in support of an application. Up to four Ram banks may be allocated for this purpose allowing just under 64Kbytes of storage space. Unlike the equivalent facilities on the BBC B+ 128 and Master series computers, data may be retained in Sideways Ram whilst the computer is switched off. A useful example application of this is given at the end of this section.

When using Sideways Ram for data storage a new method of specifying Sideways Ram addresses is used known as 'Psuedo Addressing'. Normal methods of referencing Sideways Ram addresses are known as 'Absolute addressing'. The differences between Psuedo and Absolute addressing are as follows:

Absolute addressing requires two parameters. The first is the actual physical address of a location in the Sideways memory area which resides between locations &8000 and &BFFF inclusive. Thus the physical address will be a hexadecimal number from &8000 to &BFFF. The second parameter required to specify the address is the Ram bank number.

Psuedo addressing allows up to four 16K banks of Sideways Ram to be regarded as a continuous block of memory with addresses in the range &0000

to &FFBF. The INTEGRA-β OS has the responsibility of determining the Physical bank that is accessed by a given address. It should be noted that maximum address is &FFBF and not &FFFF. The reason for this is that a 16 byte header is placed at the start of each Ram bank that is allocated for Psuedo addressing, however the INTEGRA-β OS adjusts the addresses during a transfer to skip the headers and so the address range is continuous.

Allocation of Sideways Ram for data storage is carried out using the 'SRDATA' command. This command requires a list of bank numbers to be specified from which upto four banks will be allocated for data storage.

If you look at the '*ROMS' list above, you will see that bank numbers 4 and 5 contains the title 'Ram'. This is part of a special header which is used to indicate that a Ram bank is currently allocated for data storage. This allocation would have been setup with the following command:

```
*SRDATA 4,5 [RETURN]
```

With the above allocation, 32736 bytes will be available as follows:

Addresses &0000 to &3FEF will access bank 4,

Addresses &3FF0 to &7FDF will access bank 5.

Once allocated in this way, data may transferred between Sideways Ram and the computers main memory, including that in a co-processor, using the '*SRREAD' and '*SRWRITE' commands with psuedo addressing. It is also possible to save or load data to or from the current filing using the '*SRSAVE' and '*SRLOAD' commands, again with psuedo addressing.

For example, in conjunction with the '*X*' command, it is possible to copy a shadow mode 0 screen into Sideways Ram using the '*SRWRITE' command as follows:

```
*X*SRWRITE FFFF3000 +5000 0 [RETURN] or
```

```
*X*SRWRITE FFFF3000 FFFF7FFF 0 [RETURN]
```

The screen image can be transferred from Sideways Ram back to screen memory using the '*SRREAD' command:

```
*X*SRREAD FFFF3000 +5000 0 [RETURN] or
```

```
*X*SRREAD FFFF3000 FFFF7FFF 0 [RETURN]
```

As with the '*SRSAVE' command, there are two methods of specifying the address ranges for the transfer. The first example shows the use of a start address (FFFF3000) followed by a '+' symbol and the number of bytes to be transferred (5000). The second example shows the use of an end address (FFFF7FFF). The first four digits (FFFF) in the addresses are only required when a co-processor is active to indicate that I/O processor is to be accessed.

The start and end addresses always refer to main memory and not Sideways Ram so it is only possible to transfer between main memory and Sideways Ram, and not between Sideways Ram areas.

The final parameter '0' is the psuedo address in Sideways Ram to or from which data is to be transferred.

As mentioned above, it is also possible to transfer the data to or from you current filing system using '*SRSAVE' and '*SRLOAD'. The screen image copied to Sideways Ram with the above '*SRWRITE' command could now be saved to disc with the following command:

```
*SRSAVE SCREEN 0 +5000 [RETURN]
```

Comparing this example with others given previously for saving Rom images, you will see that the bank number has been omitted. This is because the bank number is implied by the start address used (0). As with normal use of the '*SRSAVE' command, an end address could have been specified in place of the length and the Q option could have been used if corruption of main memory is not a problem.

One application of this would be to load a standard screen template into Sideways Ram, using '*SRLOAD' with psuedo addressing after the Sideways Ram has been allocated with '*SRDATA', and then to use '*X*SRREAD' as above for downloading the screen, thus a collection of routines for building up a screen template could be replaced by a single '*X*SRREAD' command. This would be especially useful with menus for mode 7 screens as they only occupy 1000 bytes, and so 16 mode 7 screens could be stored in a single Ram bank.

When Sideways Ram is no longer required for data storage, the '*SRCLEAR' command is used to free the banks for other uses. Like the '*SRDATA' command, a list of bank numbers needs to be given. For example, to clear the above '*SRDATA 4,5' allocation, the following command is used:

```
*SRCLEAR 4,5 [RETURN]
```

To concluded this section, here are a couple of function key definitions that may prove to be useful to those prone to power failures, or who insist upon working through thunderstorms:

For DFS only users:

```
*KEY0*SRWR.0+80 0|M*SRWR.400+400 80|M*SRWR.1900 7FFF 480|M
```

```
*KEY1*SRRE.0+80 0|M*SRRE.400+400 80|M*SRRE.1900 7FFF 480|M
```

For ADFS and DFS users:

```
*KEY0*SRWR.0+80 0|M*SRWR.400+400 80|M*SRWR.1F00 7FFF 480|M
```

```
*KEY1*SRRE.0+80 0|M*SRRE.400+400 80|M*SRRE.1F00 7FFF 480|M
```

For 65C02 co-processor users

```
*KEY0*SRWR.0+80 0|M*SRWR.400 F7FF 480|M
```

*KEY1*SRRE.0+80 0|M*SRRE.400 F7FF 480|M

Before using these definitions, 32Kbytes, or in the case of 65C02 co-processor users, 64Kbytes, of Sideways Ram needs to be allocated for data storage using the *SRDATA command. Once this has been done, and the appropriate definitions for system have been entered, pressing function key 0 at any time that you may enter a * command will save your working memory to Battery-backed Sideways Ram. Pressing function key I will restore the previously saved data. These definitions have been tried successfully with a number of applications, including: INTER series Roms from Computer Concepts, View 3, View sheet, View professional and BASIC.

4-3 Sideways Ram Commands

As mentioned above the commands provided include all those which users of the Master series will have encountered and additionally allow for clearing header information or entire banks.

Some of these commands and examples of usage have been covered in the introductory section therefore few examples are included below.

Note that these commands utilise Cassette Filing System workspace and therefore must not be used whilst a Cassette Filing System file is open.

4-3.1 Allocation

The three commands provided for allocation of Sideways Ram banks allow full control over the allocation process and permit use of 'absolute' and 'psuedo' addressing modes.

*SRDATA <id> (,<id>)...

This command is used in the psuedo-addressing mode for allocation of a bank or banks of Sideways Ram for block storage. (16 bytes will be used for header information).

*SRROM <id> (,<id>)...

This command is used in the 'absolute' addressing mode for allocation of a bank or banks of Sideways Ram for rom images storage.

SRSET (< id >)(, <id>)...()/?

This command is used to change the meanings of the letters 'W', 'X', 'Y' and 'Z' when used to represent RAM bank numbers. Use of this command will allow the actual RAM banks used by an application to be easily altered, so the user will not have to clear ROM images out of banks 4 through 7 which are the default banks for 'W' to 'Z'.

To set 'W' to 'Z' as banks 7, 10, 11 and 13, enter

*SRSET 7,10,11,13 [RETURN]

The most common usage of this command will be to allow the INTEGRA-β OS to allocate the banks itself. This may be done by entering

*SRSET [RETURN]

This is most useful prior to entering an application which makes use of sideways RAM as data storage. To find the current allocation of 'W' to 'Z' enter

*SRSET ? [RETURN]

4-3.2 Loading and Saving

These commands will load or save a block of Sideways RAM under the specified filename, to the current filing system. Both 'absolute' and 'psuedo' addressing are supported for these commands.

*SRLOAD <filename> <loadaddress> (<id>) (Q) (I)

Loads the named file from current filing system, into Sideways RAM. Load address, if 'absolute', must be within the range &8000-&BFFF and the Rom id must be specified, alternatively if 'psuedo', must be within the range &0000-&FFBF and the rom id must be omitted.

The 'Q' parameter, if specified, will cause the transfer to be done as fast as possible, but uses main memory from OSHWM (PAGE) to HIMEM, and so will overwrite any data or programmes in main I/O processor memory.

The 'T' parameter, if specified, will inform the Operating System of the presence of the ROM image so that it will be offered service calls, this allows it to respond to commands without the need for pressing break. However, if the ROM needs to claim work space (as in the case of filing systems), you will still need to press the BREAK key.

*SRSAVE <filename> <savestart> <saveend> (<id>) (Q)

*SRSAVE <filename> <savestart> +<length> (<id>) (Q)

This command is used to save a nominated bank of Sideways Ram. Save start and save end, (or save start +length), if 'direct', are within the range &8000-&BFFF and the Rom id must be specified, alternatively if 'psuedo', are within the range &0000-&FFBF and the rom id is omitted.

The 'Q' parameter, if specified, will cause the transfer to be done as fast as possible, but uses main memory from OSHWM (PAGE) to HIMEM, and so will overwrite any data or programmes in main I/O processor memory.

4-3.3 Copying

As with the commands included in the previous section the option of using 'absolute' or 'psuedo' addressing modes is available.

*SRREAD <startaddress> <endaddress> <s-start> (<id>)

*SRREAD <startaddress> +<length> <s-start> (<id>)

Copies an area of Sideways Ram into main memory. Start address and end address refer to main memory, as does length, which is an alternative form of definition of the end address providing '+' is included. S-start refers to the start address in Sideways Ram and may be either an 'absolute' or direct address in which case the bank id must be specified, or s-start may be a 'psuedo' address in which case the bank id is implied from the address and should not be included.

*SRWRITE <startaddress> <endaddress> <s-start> (<id>)

*SRWRITE <startaddress> + <length> <s-start> (<id>)

Copies an area of normal user memory into Sideways Ram. Start address and end address refer to normal user memory, as does length, which is an alternative form of definition of the end address providing '+' is included. S-start refers to the start address in Sideways Ram and may be either an 'absolute' or direct address in which case the bank id must be specified, or s-start may be a 'psuedo' address in which case the bank id is implied from the address and should not be included.

4-3.4 Clearing

The following commands allow you to remove data header information or clear the contents of a Sideways RAM bank.

*SRCLEAR (<id>) (, <id>)...

Used without bank id parameters, "SRCLEAR will overwrite the sixteen bytes of header information with zeros, for all banks allocated for data storage by use of *SRDATA. If bank id parameters are used, only those banks identified will be affected.

*SRWIPE <id> (, <id>)...

This command completely clears the contents of the banks specified, overwriting all contents with 'nulls', or &00.

5 Time and Date Commands

This section covers the extensive range of computations possible with TIME, ALARM and DATE Commands. Inevitably the major application lies in the computation of dates and the presentation of calendar information. An increasing amount of software is able to read a system clock and utilise the data in it's own activities.

5-1 Setting the Time and Date

To set the time, you must use the *TIME = command, followed by the time in 24 hour format.

To set the Time to 1:45pm, enter

```
*TIME = 13:45 [RETURN]
```

The time can also be set to number of seconds past the minute by specifying the seconds as follows:

```
*TIME = 13:45:30 [RETURN]
```

If you type in an invalid time, eg 25:69:87, the system will respond with the error message, 'Invalid time'.

To set the date, you must use the *DATE = command, followed by the DATE in the European numeric format (dd/mm/yy).

For example to set the date to 21st December 1987, enter

```
*DATE = 21/12/87 [RETURN]
```

Note that the day of the week is not given. This will be calculated for you. When setting the date the system does not perform a full date validity check, and so it will only generate the 'Invalid Date' error if you enter something like *DATE = 32/13/87.

To display the time and date, enter

```
*TIME [RETURN]
```

```
Mon, 21 Dec 1988. 13:48:57
```

5-2 Time-of-Day Alarm

This facility provides the equivalence of an ordinary alarm clock but with several enhancements allowing volume, tone and duration of alarm to be configured, also to be cancelled or remain set following acknowledgement.

When the alarm is triggered, or becomes an 'alarm-due', two signals are generated, an audio and a visual signal. The audio signal continues for the period set in the configuration option or until acknowledged, whilst the visual signal continues until acknowledgement. Acknowledgement is performed by pressing the [SHIFT] and [CONTROL] keys simultaneously and holding them down until the signals cease. The audio signal utilises the BBC speaker device

and the visual signal consists of alternate flashing of the 'CAPS LOCK' and 'SHIFT LOCK' Led indicators on the keyboard.

The main operations of the alarm system are controlled by the *ALARM command, (see section 3 - Utility Commands), whilst inputs for settings utilise a common 24-hour format with the TIME commands. The alarm system is automatically enabled upon entry or change of an alarm setting.

To set the alarm time to 12:45, enter

*ALARM = 12:45 [RETURN] or

*ALARM = 12:45:00 [RETURN]

The '=' sign is optional, being included for consistency.

To disable the alarm, enter

*ALARM OFF [RETURN]

To enable the alarm, enter

*ALARM ON [RETURN]

To display the alarm time and status, enter

*ALARM ? [RETURN]

A message, similar to the following, will be displayed

12:45:00 / ON

The '?' entry is optional, included for consistency.

5-3 Date and Calendar Commands

The CALENDAR command is similar to the DATE command, the only difference being that the DATE command displays the DATE on which a specified date occurs, and the CALENDAR command displays the months calendar for the month in which the specified date occurs.

To display the current date, enter

*DATE [RETURN]

Mon, 21 Dec 1987.

To display the current calendar, enter

*CALENDAR [RETURN]

```

                December 1987
Sun           6   13   20   27
Mon           7   14   21   28
Tue           1   8   15   22   29
Wed           2   9   16   23   30
Thu           3  10  17   24   31
Fri           4  11  18   25
Sat           5  12  19   26

```

5-3.1 Date Computations

By entering only part of a date, it is possible to find the day of the week a certain day falls on, or the date of the first Monday in March for example. When using this facility, the date is made up from four parts, as shown below, and is entered in the form day, d/m/y where <day> is the Day of the week (*eg Sunday, Sun, Su etc)

<d> is the Day number in month (1-28/29/30/31)

<m> is the Month number in year (1-12)

<y> is the Year number (0-99)

Examples

- To find the day of the week on which Christmas day falls in 1988 occurs, enter
*DATE 25/12/88 [RETURN]
You will get the response:
Sun, 25 Dec 1988.
- To find the next year in which the 29th February occurs on a Sunday, enter
*DATE SUN,29/2 [RETURN]
You will get the response:
Sun, 29 Feb 2004
- To find the date of the first Monday in March 1988, enter
*DATE MON,/3/88 [RETURN]
You will get the response
Mon, 07 Mar 1988

The following tables show the result displayed for given entries following *DATE and terminated by [RETURN].

Without <day> parameter.

Enter	Displayed.....
<null>	Current date returned.
d	Date on next occurrence of <d>
/m	Date of next occurrence of 1 st day of month <m>
d/m	Date of next occurrence of day of month <d> in month <m>
//y	Date of New Years day in year <y>
d//y	Date on day <d> of first month (January) in year <y>
/m/y	Date on 1 st day of month <m> in year <y>
d/m/y	Date on day of month <d> in month <m> in year <y>

With <day> parameter.

Enter	Displayed.....
day	Date on next occurrence of <day>
day,d	Next date that day number <d> occurs on day of week <day>
day,/m	First occurrence of day of week <day> in next occurrence of month <m>
day,d/m	Date of next occurrence of day of month <d> in month <m> that occurs on day of week <day>
day,//y	First occurrence of day of week <day> in year <y>
day,d//y	First occurrence of day <d> that occurs in day of week <day> in year <y>
day,/m/y	Date of first occurrence of day of week <day> in month <m> in year <y>

Date validity check.

Enter	Displayed.....
day,d/m/y	If entry was valid displays date entered. If date entered was not valid causes 'Invalid date' error message.

It is also possible to calculate offsets from a date by giving an offset specifier in place of the day of week parameter. Two types of offset facility are available. One allows you to find the date <n> occurrences of <day> after a specified date, whilst the other determines the date on a number of days before or after a specified date. The rest of the date specification is as listed above without the <day> parameter.

Whilst the examples given are based upon the *DATE command, they can equally be utilised with the *CALENDAR command.

Enter	Displayed.....
<day>+	Date on day of week <day> after a specified date.
<day> -	Date on day of week <day> before a specified date
<day> <n>	Date on day of week <day>, <n> weeks after the specified date.
+n (0-99)	Date < n > days after the specified date.
-n (0-63)	Date <n> days before the specified date.

Examples of the use of offsets.

1. To find the date on the Tuesday following 23rd March 1988, enter
 *DATE TU+,23/3/88 [RETURN]
 You will get the response:
 Tue,29 Mar 1988.
2. To find the date on the Friday before 23rd March 1988, enter
 *DATE FR-,23/3/88 [RETURN]
 You will get the response
 Fri,18 Mar 1988.
3. To find the date on the third Thursday after 23rd March 1988, enter
 *DATE TH3,23/3/88 [RETURN]
 You will get the response
 Thu,07 Apr 1988.
4. To find the date 30 days before 23rd March 1988, enter
 *DATE -30,23/3/88 [RETURN]
 You will get the response
 Mon,22 Feb 1988.

5. To find the date 60 days after 23rd March 1988, enter

```
*DATE +60,23/3/88 [RETURN]
```

You will get the response

```
Sat, 07 May 1988.
```

Other examples.

Enter	Displayed.....
*DATE+	Date tomorrow
*DATE-	Date yesterday
*DATE TODAY+	Date in one week
*DATE TODAY-	Date one week ago
*DATE+30	Date in 30 days time
*DATE Tuesday	Date next Tuesday
*DATE SA3	Date on third Saturday from today

This latter set of examples is an indication of the potential value in programmes covering commercial and financial areas, such as accounting, spreadsheets and databases.

6 Integra Windows

The Integra Windows is complex system for controlling up to sixteen separate windows and is presented as the basis for development of software to utilise these facilities.

All of the system support to enable control of the windows is included in order that programmes utilising the Integra Windows are only required to initiate built-in sequences and monitor status. This means such programmes can be relatively simple and efficient.

This facility runs in conjunction with an AMX-compatible Mouse and a new interrupt-based mouse driver is incorporated as an integral part of the system. It is hoped to provide an extension to allow keyboard manipulation of Windows as an upgrade.

Included in the system is provision for opening, sizing moving, pausing and closing windows, each of which can have vertical and horizontal scrolling with scroll bars, title block and the necessary icons for sizing, moving, pausing and closing.

The windows may be overlaid or separately positioned but if a window is overlaid it cannot be accessed for commands. In practice the number and size of windows is dependent upon the amount of sideways RAM memory available.

Windows operates in MODE 0, which with the useful amount of programme and data memory available to the INTEGRA-β allows for the use of word processing and spreadsheet applications without limiting the display possibilities.

6-1 Installing WINDOWS

Full instructions will accompany the disc, which will be shortly issued. Installation will be automated from a defined keypress combination and the programme together with data storage will require a minimum of three 16k banks of sideways Ram. Additional free sideways Ram will obviously allow more and / or larger windows to be employed.

6-2 Using WINDOWS

The issued disc will include information related to usage of Windows and programming information.

7 Applications

This section is devoted to notes on use of particular software applications which have in the past occasioned some difficulties in use. Obviously the included programmes represent only a few of those available to run on the INTEGRA-β.

Inevitably during development of an intelligent expansion board problems become apparent with some software applications which may not always restrict their reading and writing activities to areas conventionally assigned for those purposes. This can particularly be the case where the expansion has the capability of emulating some aspects of various models of a computer family.

Development and usage of the INTEGRA-β has revealed some such instances which have now been largely resolved by alterations to the management ROM rather than as is often the case, by leaving the software house to effect patches to solve such difficulties.

As might be expected most of the difficulties concerned use of Shadow RAM, for which on the BBC Model 'B' no protocols were laid down by the manufacturer, leaving each supplier to establish their own approach and standards.

For the guidance of users the following listing and remarks gives information upon the approach to be taken with the particular software so as to obtain correct usage and results. This list is by no means exhaustive and we shall welcome any further reports of experiences with other software.

ADE Plus 1.00 (SYS)

Works correctly in all Osmodes. Reading of time and date from the real time clock is only performed in Osmode 3.

AMX SuperArt 3.61 (AMS)

Works correctly in Osmodes 0, 1 & 2. In Osmode 3 the drawing create program does not work.

AMX Stop Press! 1.9 (AMS)

All features except page preview work in all Osmodes. The page preview option however, works only in Osmode 0. It appears no benefit is obtained from the use of Shadow Ram and so this package should be used in Osmode 0.

Bank Manager (C)

Contact us for details of programme modification.

Genie Junior 1.10u, 2.00u (PMS)

Works correctly in all Osmodes if not in Shadow mode when entered, including reading of time and date from real time clock. If entered from a Shadow mode Genie Junior will not function.

The INTEGRA-β OS provides a solution to this problem. By calling the *X* prefix command, e.g., *X*GENIE', Genie can be entered from Shadow mode.

The above should also apply to Genie.

Inter-Base (CC)

Functions correctly in Osmode 2.

Inter-Sheet (CC)

Functions correctly in Osmode 2.

Inter-Word 1.02 (CC)

Works correctly only in Osmodes 0 and 2. Settings of '*CONFIGURE SHX' and '*SHADOW' have no effect.

If on entering edit mode the normal blank page and default ruler do not appear press [ESCAPE] to return to main menu. Enter the following command sequence, ':KILL [RETURN]', then respond 'Y', then respond '(*)IWORD'. Do not exit Inter-Word using [CTRL] + [BREAK]. (If you do the remedy is to switch off the computer, or enter *FX200,3 followed by [BREAK] or [CTRL]+[BREAK] and then re-enter Inter-Word. This is not associated with the INTEGRA-β)

It is recommended that ':KILL' be used before exiting and a '*<command>' be used to exit by entering another 'language'.

Music 5000 & Ample Nucleus (HT)

To ensure correct functioning insert the following three lines in your working copy of the system disc !BOOT file immediately before the command *BASIC.

```
*OSMODE 4 \Select OSMODE 4
*SHADOW \Force Shadow mode at next mode change
*SHX ON \Ensure programmes end up in correct memory bank!
```

View Professional 1.0 (AS)

Works correctly in all Osmodes.

Wordwise 1.20 (CC)

Works correctly in all Osmodes, but makes no use of Shadow RAM either for editing or for review. If in Osmodes 1, 2 or 3 and '*SHADOW 0' or '*SHADOW' has been entered, then editing will not be possible.

Wordwise Plus 1.4F (CC)

Works correctly in all Osmodes but makes no use of Shadow RAM for editing. Shadow RAM is only used for the Preview option from the main menu.

In Osmodes 1, 2 and 3, '*CONFIGURE SHX' must be set, otherwise text will be lost following preview. Also '*SHADOW I', (default state) must be set.

These remarks should also apply to Wordwise Plus 2 1.58.

General Games

Most games software generally runs only in Osmode 0. It is recommended that a collection of frequently used configuration settings be kept on disc (use *CSAVE). This will simplify reconfiguration of the computer.

Suppliers

AS	Acornsoft
AMS	Advanced Memory Systems Ltd.
C	Contex Computing
CC	Computer Concepts Ltd.
HT	Hybrid Technology Ltd.
PMS	Permanent Memory Systems Ltd.
SYS	System Software Ltd.

8 Technical Reference

This section of the User Guide includes information useful to those requiring particulars of the principal register usage of the INTEGRA-β system and also includes explanations of the error messages generated by the INTEGRA OS. Any other error messages which may be displayed are generated either by the Machine Operating System, a Filing System or running application.

8-1 New Osbyte/Osword calls

A number of new Osbyte/FX and Osword calls have been implemented, of which the following are the most generally useful.

OSBYTE &00 (0)

Identify Operating System version

Entry parameters:

- X = 0 Execute BRK with message indicating OS version.
- X <> 0 Return OS type in X.

On exit:

X contains OS ID number:

- 1 OS 1.20 (OSMODEs 0 and 1)
- 2 BBC B+128/INTEGRA-B native mode (OSMODEs 2 and 4)
- 3 BBC Master 128 (OSMODE 3)
- 5 BBC Master Compact

OSBYTE &6C (108)

Select Shadow/Screen memory for direct access

On entry:

- X = 0 Select Shadow RAM
- X = 1 Select Screen/main memory

OSWORD &OE (14) Read real time dock

This call provides three functions that are implemented on the BBC Master 128 computer:

1. Read time and date in string format
2. Read time and date in binary coded decimal (BCD) format
3. Convert BCD values into string format

1. Read time and date in string format

On entry:

- XY+0 0

On exit:

XY+0,24		Contains the time / date string formatted as: ddd,nn mmm yyyy.hh:mm:ss
XY+0	ddd	three character abbreviation for the day of the week.
XY+3	' '	
XY+4	nn	day number in the month.
XY+6	''	
XY+7	mmm	three character abbreviation for the month.
XY+10	''	
XY+11	yyyy	year
XY+15	' '	
XY+16	hh	hour in 24 hour format.
XY+18	:'	
XY+19	mm	minutes
XY+21	:'	
XY+22	ss	seconds
XY+24	&0D	

2. Read time and date as BCD values

On entry:

XY+0	1
------	---

On exit:

XY+0,6		Contains the time / date string formatted as: ddd,nn mmm yyyy.hh:mm:ss
XY+0	Year	
XY+1	Month	
XY+2	Day of Month	
XY+3	Day of week	
XY+4	Hours	
XY+5	Minutes	
XY+6	Seconds	

3. Convert BCD values unto time / date string

On entry:

XY+0	2	
XY+1	Year	&00-&99
XY+2	Month	&01-&12
XY+3	Day of Month	&01-&31
XY+4	Day of week	&01-&07 (&01=Sunday)
XY+5	Hours	&00-&23
XY+6	Minutes	&00-&59
XY+7	Seconds	&00-&59

On exit:

XY+0,24		Contains the time / date string as returned for option 1.
---------	--	---

8-2 **Memory Registers**

There are two registers to support memory allocations under INTEGRA OS control. These are however duplicated to facilitate bit-changing and return to a prior state following an interrupt service routine. The registers and their locations are as follows,

ROMSEL	at \$FE30	ROM Select latch
ROMID	at \$00F4	Ram copy of ROMSEL
RAMSEL	at \$FE34	RAM select latch
RAMID	at \$037F	Ram copy of RAMSEL

Internal organisation of these registers is,

ROMSEL & ROMID

Bits 0,3	ROMNUM	Sideways ROM/RAM select bits
Bits 4,5		Not used
Bit 6	PRVEN	Private RAM enable
Bit 7	MEMSEL	Shadow/Main RAM toggle

RAMSEL & RAMID

Bits 0,1	AUX0,1	Not used but must be preserved
Bits 2,3		Not used
Bit 4	PRVS1	Private RAM 1K area select (&8000-&83FF)
Bit 5	PRVS4	Private RAM 4K area select (&8400-&8FFF)
Bit 6	PRVS8	Private RAM 8K area select (&9000-&AFFF)
Bit 7	SHEN	Shadow RAM enable bit

ROMNUM is a 4 bit binary number which determines the currently selected Sideways ROM or RAM bank.

PRVEN and PRVS1, 4 and 8 are used together to overlay sections of the private RAM over areas of Sideways ROM or RAM. This means that Software in Sideways ROM or RAM can have direct access to the Private RAM. These bits are best described in the form of a truth table.

Control bits				Memory pages and selected area				
PRVEN				from	&8000	&8400	&9000	&B000
	PRVS1	PRVS4	PRVS8	to	&83FF	&8FFF	&AFFF	&BFFF
0	X	X	X	SWR	SWR	SWR	SWR	SWR
1	1	0	0	PRV	SWR	SWR	SWR	SWR
1	0	1	0	PRV	PRV	SWR	SWR	SWR
1	0	0	1	SWR	SWR	PRV	SWR	SWR
1	0	1	1	PRV	PRV	PRV	SWR	SWR

SWR is the currently selected Sideways ROM or RAM bank
PRV is Private RAM

The arrangement of the Private RAM selection may at first appear to be unduly complex, but the use of an additional master enable bit in the ROMSEL register (PRVEN) ensures that the MOS will always select Sideways ROM/RAM when issuing service calls as it always sets this bit to zero. If this were not the case, the computer would frequently crash as a result of attempts to execute non-existent machine code in Private RAM. SHEN and MEMSEL are used together to determine whether Main/Video memory or the Shadow RAM is paged into the memory area &3000 to &7FFF.

Control bits		Memory area selected
SHEN	MEMSEL	&3000-&7FFF
0	X	Main / Video memory
1	0	Shadow memory
1	1	Main / Video memory

8-4 Error Messages

Error messages generated by the INTEGRA OS Rom are listed below with an indication of the possible reason for the message. Error messages not listed here have been generated either from the MOS, a filing system or from a running language or application.

Syntax:	One or more parameters missing from a command line.
Invalid Time:	The computer was unable to accept the time information entered with *TIME or *ALARM. This is usually caused by one or more parameters being out of the acceptable range, i.e. 0-59 for seconds and minutes or 0-23 for hours.
Invalid Date:	<ol style="list-style-type: none"> 1. The computer was unable to accept the date information entered with the *DATE or *CALENDAR commands. This is usually caused by one or more of the numeric parts of the date input being out of the acceptable range, i.e. 1 - 31 for day of month, 1-12 for the month or 0-99 for the year. 2. This message is generated if a date does not exist, for example 29th February 1989, as 1989 is not a leap year. If the message is generated following input of *DATE without parameters then the date stored in the clock is invalid and should be reset.
Day, date mismatch:	Generated following a *DATE or *CALENDAR command where the specified day cannot be matched with the specified date. For example, TUE.03/10/88, which was a Monday.
Bad number:	A numeric parameter for a command was invalid, or out of range.
Osmode range:	OSMODE number greater than 5.
Not found:	A specified file could not be found.
No Co-Processor!	*TUBE ON was entered when the computer could not detect the presence of a Second Processor.
Printer in use!	Occurs when switching between OSMODE 0 and any other OSMODE when the printer buffer is not empty.
String too long:	Caused by string following the *BOOT command being too long to fit in the function key buffer.
Too Big:	Buffer size specifier when using *BUFFER command was greater than 4.
Bad parameter:	Unrecognised *CONFIGURE or *STATUS command.
Bad id:	ROM id missing from a command line or was invalid.

9 Installation

Carefully read through these instructions before attempting to install the INTEGRA-β Expansion. The tools required for installation are a cross-point screwdriver, to open the BBC case, (two screws at rear and two below the keyboard), a chip extractor or two small narrow-bladed flat screwdrivers to remove the 6502 processor chip and any existing ROM chips to be relocated also narrow-nosed pliers to steady the existing power connector blades on the BBC board when removing the slide-on receptacles.

9.1 Preparing the Computer

Several steps are necessary before the INTEGRA-β can be inserted. Firstly switch off the computer by the main switch at rear, also the main A.C. power supply and disconnect the main power lead from the supply socket. It may not be essential to remove plug-in ribbon or other cables to connected peripherals such as monitor, printers, etc. but it will be very much more convenient to do so to access the case screws.

Remove the BBC case screws and take off the upper half of the casing. For the purposes of locating various components we shall consider the exposed printed circuit board to have North, East, South and West directions. North being towards the rear when looking from over the keyboard, East lying to the right, South towards you and West to the left. The INTEGRA-β Expansion fits in the corner bounded by the North and West sides of the BBC board.

We shall assume that no other expansions are fitted which would prevent installation of this unit. Firstly the internal plastic walls on the West side adjoining the power supply module and on the North side should be thoroughly cleaned to ensure no greasy deposits remain. Next you should remove the 6502 processor chip. This is located just to the East of the board centre, socket reference ICI is marked on the BBC board. This operation should, in the absence of a chip extractor, be performed very carefully using the small screwdrivers to gradually lever up each end of the chip in turn, about a millimetre at a time and keeping both blades in place between the chip and the chip socket.

You should also consider in which ROM socket the INTEGRA-β ROM is to be inserted. Providing it is in a higher priority socket than any Filing System ROM such as DFS or ADFS then all features of the expansion will function as required. It is recommended to insert the ROM in socket IC101 at the extreme South-East Corner of the BBC Board so as to allow maximum RAM capacity. It is however possible to use the ROM in one of the vacant sockets on the

INTEGRA-β Board. These are of higher priority than the sockets on the BBC Board. Accordingly extract and relocate any existing ROM's in the BBC Board to meet your requirements. Replacement of ROM's in the BBC board will also require removal of the keyboard. This is held by two machine screws and nuts, the screw heads being accessed from below the keyboard.

Removal of the keyboard also requires disconnection of the ribbon cable jumper and loudspeaker connectors. Ensure these are replaced before closing the casing.

Next remove the two power connectors leads, (Red and Black), from the mid-West area of the BBC board adjoining position of chips IC89 and IC98.

Finally remove three of the four black or blue jumper sockets from the vertical pins located between IC21 and IC22, roughly central about 55 millimetres from South edge of the BBC Board. These sets of pins are marked S22, S21 and S20 from North to South. The socket to be left in place is the one lying East-West at the North of the four-pin set marked S21. (Jumpers fit on North pair of each set of selector pins).

9.2 Unpacking the INTEGRA

The Board is contained in the anti-static bag between the halves of the polystyrene blocks. This bag is provided in case of high static levels when customary handling precautions will be required. No problems should arise in the normal environment. However un-necessary handling should be avoided. The adapter plug should be removed from the foamed plastic and examined to ensure all contact pins are straight. If any have been bent use a straightening block or very carefully reset with small pliers. The INTEGRA-β ROM should be left in its packing for the moment. Also included are two support strips to be mounted as described below.

9.3 Installing the INTEGRA Board

The INTEGRA-β board is supported both by insertion into the processor socket and by the supplied self-adhesive strips. Two types of strip are provided, to fit on the West wall and the North wall of the BBC casing. Both strips are 'L' shaped, strip 'A' having two horizontal extensions, whilst strip 'B' has only one and additionally has a cut-away to suit the Analogue Port. Strip 'B' is used only as a ledge. First mark a line along the West wall at 15mm from the top of the wall, then mark a line along the North wall at 31.5mm from the top of the higher part of the wall. Remove the backing paper from the 'A' strip and attach it to the West wall with it's top edge in line with the mark previously

made. Then similarly attach the 'B' strip to the North wall. Press both strips firmly into contact with the walls. The INTEGRA-β board is supported at 23/24mm above the BBC board. See diagram inside rear cover. (Some boards may be supplied with an alternative support strip with a central slot for strip 'A'. With these mark a line at 18mm from top of West wall and proceed as before. If the strip is tight on the board edge push a round object of about 3mm diameter along the length of the slot before fixing).

Remove the INTEGRA-β board from its packing and check that all of the nineteen jumper connectors are fitted to the board at all selector pin sets. Take the supplied 40-pin connector, remove the loose pin guard frame, turn the board upside down and offer the connector (frame away from board) up to the two rows of sockets below the board, ensuring all pins are entered into their respective sockets before inserting fully home.

Next connect the captive Red and Black power leads to the power connector blades on the BBC board. The red wire goes to the North (Vcc) of the two blades and the black wire to the South blade (0V).

Now take the INTEGRA-β Board and place it's West end against the entrance to the slot in the support strip 'A' sliding the board North or South to align the connector pins below the board with the BBC processor socket. When North-South alignment is correct slightly lift the East end and press the board into the slot. Lower the board again to engage the pins in the socket and when correctly aligned press the contact pins home into the socket holes. Use of a small mirror during this operation will aid checking of pin alignment.

Locate the three control leads with single connectors fitted on the INTEGRA-β board. Letters C, B, and A are marked on the board identifying the connection of the yellow, blue and green wires respectively. Fit the Yellow lead (C) to the South-West pin of S21, the Blue lead (B) to the central pin of S20 and the Green lead (A) to the central pin of S22.

The three removed jumper sockets should be secured with tape to some open part of the computer casing. (For Issue 1 board A, B, C are Yellow, Blue, Green).

Take the 6502-type CPU chip and insert it into the 40-pin socket ensuring the rebate in the chip end is at the North. All chips on the INTEGRA-β Board are located with their index mark (usually a semi-circular recess) to the North.

Finally connect the BBC's detached Red and Black power leads to the horizontal connectors at South-West corner of the INTEGRA-β board. The Red lead goes to the West connector and the Black lead to the East connector of the pair, marked + 5V and 0V respectively.

This completes board installation.

9.4 Inserting RAM or ROM's

Refer to the board diagram, Figure 6-3, and note the numbering of the eight vacant 28-pin sockets. These sockets are arranged to be able to accept Type 62256 Low Power 10/12 ns 32kb RAM chips or 8kb, 16kb or 32kb ROM chips. If the designated (or equivalent) RAM chips are used only four may be added, one in each vertical pair of sockets, i.e. in Blocks 9, 11, 13 or 15.

If 32kb ROM chips are used, again they can only fit in the same sockets, whereas if 16kb or 8kb ROM chips are used they can be fitted in both sockets of all pairs. It is NOT permissible to fit a ROM into any socket South of a RAM chip or a 32kb ROM.

Next remove the INTEGRA-β ROM from its packing and insert into the socket selected for installation ensuring the indent in the moulded case is located at the North end of the socket. To maximise possible RAM installation it is recommended this ROM is fitted in socket IC101 located at the South-East corner of the BBC board. It is necessary for it to be installed in a socket having higher priority than any Filing System Rom, such as DFS or ADFS, to permit operation as intended. If advantage is to be taken of the *CONFIGURE LANG command the INTEGRA-β OS ROM should be installed in the highest priority socket.

Having selected sockets for the various ROM or RAM chips used and inserted the chips it may be necessary to make alterations in the position of the grey, blue or black jumper sockets at the North and South of the occupied RAM/ROM sockets. These jumper sockets are used to select line or battery-backed 5 volt supply, write control and addressing of 8kb, 16kb or 32kb chips. For each pair of RAM/ROM sockets there is a group of twelve pins at the North and three pins at the South. Arrangements of the jumpers for various chip selections are shown on the INTEGRA-β board.

The Issue 3 board diagram for 8/16K ROM jumper configuration is not correct. Use the diagram in the User Guide. One additional jumper at East end of North edge of the board is used for temporary disconnection of the battery-backed supply. This is intended for killing the content of Sideways Ram in the event of a very severe crash. With the computer switched OFF temporarily remove the jumper from the ON setting and fit to the OFF setting for about a minute, then replace on the ON setting. Do NOT switch on the computer with that jumper on the OFF setting. System reset and reconfiguration will then be necessary.

9.5 **Operational Check**

Before finally replacing the keyboard and closing the BBC case you should check to ensure all connections have been made as described and that no loose items or tools have been left in the case. Also check that the BBC OS ROM has not been removed from its designated location in socket IC51. Replace and secure the keyboard checking that the ribbon cable jumper connector and loudspeaker connector have been correctly inserted (if previously removed). Replace the video monitor connection and the mains power connection and switch on at mains socket. To enable the computer to acknowledge the INTEGRA-β Expansion on the first switch-on you must hold down the '@' key whilst switching on at the computer.

The display should then show

```
Full System Reset.
Are you sure?
```

Confirm by entering

```
<yes> [RETURN]
```

To check the system is functional enter

```
*ROMS [RETURN]
```

The screen will then display a listing of the active ROM's. Other peripherals can then be connected with the computer switched off.

9.6 **Installation of WP Switch**

This is an optional accessory and may not be included.

Commence with BBC Model 'B' cover removed and INTEGRA-P Board installed in the computer. Position the Write Protect Switch Assembly so that the dished washers fit inside the left-hand lower part of the rear wall, the rubber blocks resting on the wall and the switch assembly located outside the rear wall.

Remove the two jumpers marked WP 4/5 and WP 6/7. Fit the yellow lead marked 1 onto the right-hand pin and the associated green lead onto the left pin of the WP 4/5 pair. Repeat for the leads marked 2 at WP 6/7.

If more than two switches are fitted, so as to control RAM fitted in blocks 8/9, 10/11, 12/13 and 14/15 then remove the existing jumper from the centre pins of the block of 12 pins North of the RAM chip(s). Fit the yellow and green leads as above.

Remove the jumper marked Battery Link on the INTEGRA-β board and fit the red lead and jumper in its place. Save the removed jumpers.

When the BBC cover is replaced the switch assembly will be locked in the 8mm high slot at the rear of the case. If it is necessary to alter the locking effect the screw (and nut, if fitted) through the attachment should be adjusted and then locked with a dab of paint or nail varnish.

If additional switches are subsequently ordered it will be necessary to extend the connection from the Battery Link to the additional switches inside the switch assembly by soldered connection.

To use the switches for Write Protection the switch toggle should be moved away from the BBC casing. Moving the toggle towards the casing allows normal Write Enabled use of RAM.

Mounting the INTEGRA-β Board

