

Installing & Using RAMagic!

Caution : All semiconductor devices can be damaged by static electricity and the user must therefore take appropriate care when fitting the RAMagic! assembly. As supplied, there is a piece of black conducting foam protecting the NVRAM module and this should not be removed until RAMagic! is to be fitted to the computer. To reduce the risk of static damage during fitting, the BBC Micro should be left plugged in to the mains but switched off. The user should then regularly touch the silver 'Video' BNC connector at the rear of the micro or the PSU Assembly metal case in order to discharge any static.

RAMagic! is not especially sensitive to static, no more than any other IC, but care should still be taken since the module is not cheap!

General

The RAMagic! (RM) assembly requires one free Sideways Rom socket which may be one of the (initially) three empty sockets on the BBC motherboard (lower right) or any free rom socket on a third party Rom expansion board.

Note 1: The RM module is a 28 pin device but has no pin at #1 and #27. These pins are connected to the two leads attached to the module.

Note 2 : BBC computers, but more particularly ROM expansion boards, sometimes have rom sockets which are 'over used' having being subjected to multiple Rom swaps. This sometimes causes the pin receptacles to become spread and this can result in poor pin contact. Please check that the socket you intend to use is in good, clean condition and has no such spread receptacles.

Note 3 : As soon as possible, I recommend that you make a backup copy of the RM disc just in case. The individual files can be simply copied to another disc (there are no hidden files) or a disc backup utility can be used.

Note 4 : Throughout these instructions, <cr> is used to represent pressing the <RETURN> key.

Fitting

- When ready to fit, discharge yourself (see above) and remove the black conductive foam from the base of the module. With the module correctly orientated with respect to Pin 1, carefully align the pins of the module with the receiving socket and push down gently and evenly until fully home. If the module appears to be a 'loose' fit, try a different socket – the one you are trying may be mechanically degraded through over-use. (Note that Pin 1 on RM is highlighted and this should be located at the top left of any socket assumed to be facing north. The top of roms and sockets are normally marked with a notch between Pin1 and its opposite pin.)
- Connect the Analogue (Joystick) Port pass-through connector.

- If the Sideways Ram mode is to be used ('SWR' on the switch), connect the red Hook Probe to one of the optional IC locations (See Appendix 1) by extending the probe copper hook and carefully 'threading' it round the relevant pin of the target IC. Release the probe and visually check that a good connection has been made and that the hook is not touching any adjacent pins.

(Not applicable on Master, leave disconnected)

- If required, drill an appropriate hole in the case to accommodate the three-position switch or simply allow it to hang free at the back of the case. The RM wires are flexible enough to allow the top cover to be refitted with the wires passing through the rear vent slot.

Software Installation

Set the RM switch to 'RTC' (you will also be prompted during installation) and switch on the Beeb noting that your normal start-up message is displayed. At this stage, RM is 'empty' and inactive and cannot be 'seen' by the Beeb OS. Place the RM disc into your default drive and type ***RMLOAD** <cr>. The installation software will now identify your computer, locate the RM hardware, report in which socket it has been found (00-0F Hex) and load the applicable RM code image. Appropriate prompts are displayed where needed.

If the installer reports that the hardware cannot be found, please switch off and re-check your installation. (Note that the red probe connection is not needed whilst RM is in the 'RTC' or 'ROM' positions and it's connection or otherwise will not affect the installation procedure.)

After installation, and having pressed <Break> when prompted (<Ctrl><Break> on the Master), you will see the Day, Date and Time stamped at the top of the screen. (This feature can be turned off via *TBRK – see later.) Type *HELP <cr> and you will see RM present amongst your sideways roms reporting as **RAMagic! 2.0B** on any 'B' series computer or **RAMagic! 2.1M** if you are installing on a Master 128.

The software installation need only be carried out once and the software will be retained even if the module is removed from the computer or fitted to another Beeb. The RM upper 16K program and data area is always write protected and cannot be written to by you or other software and so corruption is virtually impossible. However, in the unlikely event of a corruption occurring, or for installation of a software upgrade, a re-load can be carried out at any time. If you transfer RM From a 'B' to a Master 128 or vice-versa then the installation must be repeated to load the matching software.

The lower 16k data area, with the switch in the 'RTC' and 'ROM' positions, is also write protected other than for access by RAMagic! itself. With the switch in the 'SWR' position the lower 16K data area can be written to (and read) as if it were normal sideways ram (except it's non-volatile!) and so an RM saved memory image can also be overwritten. Be careful not to select the 'SWR' position if you have saved valuable data to NVRAM with *MSAVE. Re-load any such saved image to Beeb memory first using *MLOAD and transfer to disc.

For RAMagic! * commands to be recognised and actioned the switch should be left in the 'RTC' position. However, the unit can be re-rolled to become a non-volatile 16K Sideways Ram or Rom module by selecting 'SWR' or 'ROM' respectively. For now, leave the switch in the 'RTC' position, the other settings will be explained later.

Getting started

Before looking at the commands in detail, let's just try a couple of things so you can confirm RAMagic! is working.

Type *TBRK <cr> and you will get an 'OK' response. Press <Break> (<Ctrl><Break on the Master) and you will see that the Day, Date and Time stamp at the top of the display has gone. *TBRK is a 'toggle' so each time you type the command the internal 'Time_on_Break' flag is inverted. Since RM uses NVRAM, the setting will be retained through power off.

I will have set the time and date during testing but not necessarily very accurately so you can now set these yourself using *TSET and *DSET. These require specific and exact formats and all fields must be fully specified. Type *TSET <cr> and *DSET <cr> (i.e. with no data) and you will be shown the required format.

As with all hardware-only Real Time Clock/Calendars, the day of the week is not actually linked to the date so it is up to the user to set the day correctly. The day will then be automatically incremented every midnight as you would expect. Internally, the weekdays are numbered 1 to 7 representing Sunday to Saturday which is how 'day' will be returned if you use OSWORD #14 and is as per the Acorn specified implementation on the Master series computers.

If you saw my demo on 'You Tube', you will have seen a short Basic program being saved in NVRAM and then automatically loaded and run at switch on so let's quickly try that. (*No auto-boot across the Tube with a 2P*)

Type the following program :

```
10 FOR X=1 TO 100
20 PRINT"HELLO FROM RAMAGIC!      ";
30 NEXT
```

Save to NVRAM by typing *MSAVE TEST <cr> and press 'Y' when the erase confirmation prompt is displayed.

The auto-run commands are preset to load and run a Basic program from NVRAM so simply type *AUTON <cr> to switch on auto-boot.

Now, switch off your Beeb (for several years if you want!), switch back on and watch the program 'magically' reappear and run!

To cancel the auto-boot, type *AUTOFF <cr>.

We will now look at all the commands in more detail.

Using RAMagic!

For specific syntax help on RM commands, type *HELP MAGIC. (Note 'MAGIC' and not 'RAMAGIC!' – this is simply to reduce typing!)

***TIME**

Prints the time at the current cursor position followed by a newline.

***DATE**

Prints the day and date at the current cursor position followed by a newline.

***NOW**

Prints the day, date and time at the current cursor position followed by a newline.

***TBRK**

Toggles the display of Date and Time stamp after <Break> or switch on. Each time *TBRK is entered, the internal controlling flag is toggled so ON becomes OFF and vice-versa. Setting is preserved in NVRAM through power off.

(Time stamp is only displayed after Power On or <Ctrl><Break> on the Master)

***TSET <hh:mm:ss>**

Sets the time to the supplied values. The fields and colon delimiters must be exactly as shown and any errors will result in the required format being displayed as per *HELP. Correct entry will result in the time being echoed as per *TIME

***DSET <day> <dd-mm-yy>**

Sets the date to the supplied values. The fields and hyphen delimiters must be exactly as shown and any errors will result in the required format being displayed as per *HELP. Correct entry will result in the date being echoed as per *DATE (Note that <day> is the first three characters of the day of the week. e.g. SAT)

***MCLR**

Clears the entire NVRAM memory image storage area. A cautionary confirmation prompt is displayed where 'Y' must be typed or the command will be aborted. Confirmation is sought since any existing saved memory image will be lost and cannot be subsequently recovered. Take note!

Note : The NVRAM integer variable storage area (@%-Z%) is not cleared by this command. (See *VSAVE and *VLOAD)

***MROM <filename>**

(Not implemented on Master)

Used to load a standard Sideways Rom image from disc into the lower 16K area. After loading, setting the switch to 'ROM' and pressing <Break> will effectively install the Sideways Rom which will now behave exactly like any ROM or EPROM in that it cannot be corrupted and will be preserved through power off. The image is specified by <filename> which follows the standard DFS protocols. A cautionary confirmation prompt is displayed where 'Y' must be typed or the command will be aborted. Confirmation is sought since any existing saved memory image will be overwritten and cannot be subsequently recovered. Take note!

Once the switch has been set to 'ROM', RAMagic! is now effectively disabled and none of it's commands will be available. However, the RTC is maintained at all times regardless of switch setting and RAMagic! can be reselected at any time by setting the switch back to 'RTC' and pressing <Break>. The loaded ROM image will be maintained in 'RTC' unless deliberately overwritten or cleared. *MDIR (see later) will show the currently loaded Sideways Rom image.

Note from the above that the use of the RAMagic! mode and the use of the 'ROM' or 'SWR' modes are mutually exclusive – i.e. the two modes cannot be used simultaneously.

***MSAVE (<name>)**

Saves the entire contents of Beeb user Ram to NVRAM starting at PAGE (as defined by BASIC) and records a header for the saved image with optional <name> and the current date and time. The save continues until the NVRAM is full which is typically at address **&7CFF** for a Tape based Beeb or **&87FF** for a Disc based Beeb. On a Beeb running a Second Processor (2P) across the Tube, the saved image is always from **&0800** to **&76FF**. Hence, the saved image encompasses all user Ram on a non-2P Beeb but falls slightly short of a complete image on a 2P. However, in the case of a 2P, a program would have to be unusually large to not be fully saved to NVRAM.

A cautionary confirmation prompt is displayed where 'Y' must be typed or the command will be aborted. Confirmation is sought since any existing saved memory image will be overwritten and cannot be subsequently recovered. Take note!

The parameter <name> is optional but can be any string of characters up to 14 in length.

*(See *MLOAD for RAMagic!'s definition of the value of PAGE. Master values for the three boundaries shown in bold are one Page less i.e. &7BFF, &86FF and &75FF)*

***MLOAD**

Copies a saved memory image from NVRAM to Beeb or 2P user Ram. On a non-2P Beeb, the image is loaded starting at PAGE (regardless of where the image was originally saved from) and continues until HIMEM is reached. On a 2P, the memory is always loaded to **&0800** and continues to the end of NVRAM which equates to 2P address **&76FF**.

On a Beeb, since the NVRAM Save and Load both work from the current value of PAGE, RAMagic! can be used to move programs up and down in memory as required. For example, if a program is designed (and will only) run from tape with

PAGE=&E00, it can be loaded from disc to &1900 (the disc based default PAGE value), *MSAVE'd to NVRAM and then be re-loaded at &E00 by setting PAGE=&E00 prior to the *MLOAD.

RAMagic! actually determines PAGE by reading the memory page value from location &0018 which holds the Hi-Byte of PAGE in all BASIC versions. Thus, this location could be modified as required to force non-standard NVRAM saves and loads although care should be taken if BASIC is the current language as it would become confused if &0018 is changed without using the 'PAGE=' method. However, for assembly language use where *MSAVE and *MLOAD can be used via OSCLI, this information may prove useful.

If there is no valid image saved to NVRAM, the *MLOAD command will be ignored. (e.g. following a *MCLR or if *MDIR reports nothing)

***MDIR**

Displays the header information for the saved memory image as :

<saddr> <name> <date> <time>

where <saddr> is the 16 bit hex source address from which the save began, <name> is the name entered for the saved image and <date> & <time> are the date and time stamp at the point of save.

Typically, <saddr> will be &1900 for disc based systems and &0E00 for tape based systems or on the Master 128. For a Tube 2P saved image, <saddr> will always be &0800. If the last NVRAM write action was *MROM <filename>, <saddr> will always be set to &8000 and <name> will be the rom image loaded.

Note that if the lower 16K of NVRAM is written to directly as sideways ram (switch in the 'SWR' position) and not via RAMagic! then the header will not be updated and may be misleading. It is better in these circumstances to erase the NVRAM via *MCLR before using 'SWR' as this will erase the header and *MDIR will return a null string. Alternatively, a dummy *MSAVE can be carried out setting <name> to say 'SIDEWAYS RAM' before selecting 'SWR'

***VSAVE**

(Ignored if Tube active)

Saves to NVRAM the fixed area of Beeb ram where the system integer variables, A% to Z% and @%, are stored. These variables can thus be preserved through power off and restored at any time using *VLOAD. This area of NVRAM is not cleared by *MCLR and can only be overwritten by a further *VSAVE. (The NVRAM storage area for the integer variables is separate from the main memory image storage area.)

Note that the BBC OS and/or BASIC do not initialise the % area of ram and hence the initial values of the integer variables after switch on can be somewhat random. Only @% (the print formatting variable) is initialised by the OS. I mention this in case the user thinks that RAMagic! is doing something odd with non-initialised integer variables.

***VLOAD**

(Ignored if Tube active)

Restores from NVRAM a previously saved set of values for the system integer variables, A% to Z% and @%.

***AUTON**

(Ignored if Tube active)

Switches the auto-boot feature to ON. When selected to on and immediately after the Beeb has finished initialising following power on or Break, three command lines, ACOM1 to ACOM3, are inserted into the keyboard input buffer for sequential execution. These command lines are simple text strings and are acted upon as if they had been typed from the keyboard and so must be recognisable by either BASIC or the OS. The auto-boot flag is stored in NVRAM and its state is preserved through power off.

Note : If you have configured the Beeb to auto run and can't break in to stop the process, simply hold down <Escape> as you press <Break> or switch on and at the prompt type *AUTOFF.

*(See also *AUTOFF, *ACOM and *ACOMn)*

***AUTOFF**

(Ignored if Tube active)

Switches the auto-boot feature to OFF. (See also *AUTON)

***ACOM**

(Command is accepted and actioned with Tube active but since auto-boot does not work with a Second Processor the ACOM command lines will not be executed at boot)

Displays the three current auto-boot command lines as stored in NVRAM. These are displayed as a 1,2,3 list but the numeric is not part of the stored string. Empty (null) lines are still shown as the numeric only.

*(See also *ACOMn, *AUTON and *AUTOFF)*

***ACOMn**

(Command is accepted and actioned with Tube active but since auto-boot does not work with a Second Processor the ACOM command lines will not be executed at boot)

Allows the user to define three auto-boot command lines and thus **n** is 1, 2 or 3 (no <space> between *ACOM and n). The command lines can be up to 15 characters long but if a longer line is typed no error will be issued and the line will be truncated at 15 characters. Multiple statement lines can be used such as **OLD:RUN** but the standard BASIC/OS rules apply – e.g. a star (*) command can follow a colon but cannot be followed by a colon and 'dot' abbreviations may be used such as **O.:RUN**. (The use of abbreviations, where valid, allows more than three individual auto-boot commands to be issued.)

*Typing *ACOMn <cr> (i.e. with no string) will delete (null) that command line.*

Null lines will not be passed for execution at auto-boot.

The command lines are stored in NVRAM and are preserved through power off.

After installation, the command lines are set to the following (shown as would be displayed by *ACOM) but these are just initial defaults to get you started :

```
1 *MLOAD
2 OLD
3 RUN
```

(See also *ACOM, *AUTON and *AUTOFF)

***SLEEP**

This command simply disables IRQ interrupts and puts the 6502 processor into a tight loop. This is useful to ensure that no spurious NVRAM (or disc etc.) writes can occur as power fails at switch off.

The command is intended to be used where the Beeb is to be switched off at a known time via a mains time switch or similar. For example, the time switch would be set as close as possible to the Beeb RTC and set to switch power off at say 2215 hours and the Beeb is programmed to stop processing at 2200 hours. Within the software, the program would monitor for 2200 hours and then enter a safe shutdown sequence. For example, close any open files if a disc system is in use, save any integer variable data via *VSAVE and finally wait for power off in a 'safe' condition by executing *SLEEP.

If you type this command to try it out you will have to press <BREAK> to recover the Beeb as it will have effectively crashed!

***DOSDFS n**

This command converts an IMG, SSD, DSD or ROM file on a Windows or MS-DOS floppy disc to a disc image or ROM file in true Acorn compatible DFS format.

This function can only be used on a Beeb (or derivative) with a 1770 Double Density disc interface and associated DFS which supports or emulates OSWORD \$7F.

The utility is very simple to use, every bit as easy as the listing (advert?) says, but there are some 'rules' concerning the use of the Windows transfer discs. Don't worry, there's nothing sinister or complicated, but the utility has been based on certain assumptions in order to allow the utility to be as user-friendly, fast and reliable as possible whilst fitting in to the measly 4K of code space I allowed myself for the whole RAMagic! package.

The Windows transfer source disc

The physical disc must be true Double Density media (**not stamped HD!**) and must be formatted to 720k under an authentic FAT 12 format utility. Upto Win 98, you could drop down a select box to request a 720k format but at XP and beyond this option has gradually vanished. However, all is not lost since on most machines you can still format a floppy to 720k through the pseudo-DOS Command Prompt window as follows :

If you have a 'Command Prompt' shortcut somewhere then select this otherwise select **start > Run..** , type **cmd** and click **OK**. At the command prompt (doesn't matter what directory you are dumped in) type the following with a floppy in your A: floppy drive :

format a: /t:80 /n:9 <cr>

and Windows will format the floppy to 720k. You have two ready-to-use discs so you may not need any more unless your PC is a long way from your Beeb – see next!

Very Important : You can only transfer one image per DOS disc at a time and any existing file must be deleted from the disc before the next file is sent to the floppy. Therefore, get into the habit of opening drive A: and deleting any file that's already there before copying a new file onto the disc. If you do inadvertently copy a second file to the disc it is not sufficient to just erase the first - you must then erase both files and re-copy the one you are transferring next to the floppy. You won't do any harm to the disc by having two or more files on there but if you try a transfer on the Beeb it's anybody's guess what the final DFS disc might contain! Final rules, no directories (folders), hidden or system files on the DOS floppy – you will be warned if such files are detected.

Easy summary – The single DOS file to be transferred from the PC must always be copied to an already blank (all files erased) 720k DD transfer disc.

(Hence the earlier comment about your PC being a long way from your Beeb – if you want to transfer lots of images without running backwards and forwards then you might want to make a few more transfer discs!)

The BBC destination disc

**The following largely applies to disc image transfers but please read it all and see later notes specific to ROM image details.*

The receiving Beeb disc needs to be pre-formatted by your DFS. It doesn't need to be blank (empty of files) because the image transfer will overwrite whatever is on there sector by sector including the control/directory sectors. **(Take that as a warning too because you'll completely lose anything that's already on the disc!)**

As far as RAMagic! is concerned the Beeb destination drive can be 40 or 80 track, double or single sided but the actual requirement will be dictated by the type of image being transferred. SSD and IMG files are single sided images and hence a single sided disc drive is fine but for DSD files the drive must be double sided.

SSD and IMG files will always produce an image on the top side of the disc (0 or 1) and you can't direct it to the bottom. However, if you want to use both sides of a disc you can, after the transfer, use a DFS disc copy utility to transfer the top to the bottom (0 to 2 or 1 to 3) before transferring a second image to the top. Do bear in mind though that most games discs will be run by <Shift><Break> and hence will always default to drive 0.

With regard to using a 40 or 80 track drive, it is best that you determine what is required from the size of the extracted DOS image file according to the following maximum capacities :

40T/SS : 100k 40T/DS : 200K 80T/SS : 200K 80T/DS : 400K

Thus, DSD images must use either 40T/DS or 80T/DS and IMG & SSD will be either 40 or 80 track since the double-sided doesn't apply!

Easy summary – The final Beeb disc must be pre-formatted under DFS. It doesn't have to empty but remember you will lose anything already on the disc. Make sure the drive/disc have the capacity to record the image. If transferring a .dsd image, you must pre-format both sides of the destination disc, i.e. 0 and 2 or 1 and 3.

For ROM image files, the final 'product' is an 8k or 16k hex image file in standard Acorn DFS format. This image can be loaded into Sideways RAM, blown into an EPROM as a Sideways ROM or loaded into RAMagic! as an effective mixture of the two. The same rules apply to the Windows disc as for sector images but since the final file is a standard DFS file, any formatted disc is fine for the destination disc and the only requirement is that there is enough room on the disc for the file.

If the DOS source file is upto (and including) 8192 bytes in length, an 8k (&2000) image will be created otherwise a 16k (&4000) image will be created.

RM can happily handle Windows long filenames but since the Beeb is restricted to seven character filenames, the final image is named as follows :

If the DOS filename is upto 8 characters long (excluding the 'dot' extension) then the Beeb rom image filename will be the first 7 characters of that name. If the DOS filename exceeds 8 characters then the Beeb rom image filename will be the first 6 characters of that name.

Using the utility

You now know most of it from the preceding disc guidelines, so here's the last few bits of information.

Valid image files are IMG, SSD, DSD and ROM and hence the DOS filename will be such as elite.ssd, manygames.dsd or toolkit.rom. In practice, RM only looks at the first character of the extension after the '.' to determine the image type. Since all the image types are raw hex data, RM has no way of validating the file so it's up to you to make sure you're not asking it to convert nonsense. Remember, the old computing adage applies well here – Garbage in, garbage out.....

So, you've downloaded the zip file, extracted the image, wiped any file(s) from the transfer disc and copied the image to the transfer disc. Now what?.....

Take the PC floppy and put it in your 3.5" Beeb drive. Note whether this drive is 0/2 or 1/3 and we'll refer to this drive as just 0 or 1 and it's the source drive. Put a disc in your other drive, the destination drive and if it's not formatted, do so now with whatever utility you normally use. **Note that for a dsd image you must format both sides of the destination disc.**

If the destination is already formatted but not blank and you are about to transfer a disc image then you are about to lose everything on the disc! If you are transferring a ROM image then all you will do is add an 8k or 16k file to the disc and everything else is safe. For any type of transfer, any program in Beeb memory will be corrupted so if it's not saved, do so first - You have been warned!

Now type *DOSDFS n where n is the number of the source drive and will be 0 or 1 only. Press return and the transfer will begin. In all cases, you will see the name of the file being processed (name derived as explained earlier) but the name is only actually used for ROM image transfer and will be the name of the created image file on the destination disc. For disc images, the name displayed just acts as a check that RM has found the file you expected! Additionally, you will see the file type displayed as 'SSD/IMG', 'DSD' or 'ROM' followed by the file size in Hex. This file size will, if converted to decimal (Windows Calculator – Advanced Mode), be exactly the same as the Windows file size (Right Click on file – Properties). After some experience you will be able to use this to estimate how long the transfer will take, somewhere between approximately 15 seconds and 3.5 minutes. You will see the source and destination drive lights alternately flashing as the transfer takes place and when finished, you just get an 'OK' prompt.

If the drive lights should stop flashing and the drives go quiet without any error message, hit <Escape> and you should get the prompt back with an error message. This can only happen if there is a weird read error from the source disc but for most read or write errors you will get a suitable error message and the transfer will be aborted.

After the 'OK', if the destination drive is the Beeb default drive (0), you can immediately hit <Shift><Break> and start playing the game! If you have copied 0 to 1 then you will have to move the newly created disc into drive 0 before booting.

It's as easy as that!

'ROM' & 'SWR' switch positions

(Not usable on Master)

If the switch is set to either of these positions, <Break> should be pressed before proceeding and subsequently none of the RM commands will be available until 'RTC' is reselected and <Break> pressed again. However, the clock is unaffected by switch position and will happily maintain the time and date but cannot be accessed unless 'RTC' is selected.

The 'ROM' position simply 'locks' 16k of the NVRAM and effectively turns RM into an Rom or Eprom in that the memory can be read but not written to. This allows RM to act as a Sideways Rom if the 16k has been previously loaded with a rom image via *MROM or with any data whilst in the 'SWR' position. This mode of operation is identical to other SWR modules when the Write Protect switch has been set on but RM has the advantage that the contents will be maintained even when power is switched off.

The 'SWR' position converts RM into a standard 16K Sideways Ram module with Read/Write access. Again, the contents of the 16k will be maintained through power off.

Master 128 differences & limitations

- Due to the more complex architecture of the Master, it is not possible to use the 'ROM' and 'SWR' modes of RAMagic!. The technical issues are surmountable but more RM components would be necessary with more connections to the Master and this would defeat the 'simple fit' objective of RM.

Selecting these positions will not cause any harm but corruptions of the lower 16k (not the RM code area) will occur. These corruptions make it impossible to load the NVRAM for use as a Rom and would clearly prevent the use of RM as a Sideways Ram module. As a consequence, the *MROM command is not present in the command set for the Master version of RAMagic!

(As a consequence of the above, the red Hook Probe is left disconnected on the Master)

- With RM fitted, you now have two real-time clocks in your Master! This is not a problem and RM will use it's own clock for Time-on-Break (see *TBRK) and for time stamping *MSAVE'd memory images. However, to avoid conflict with the Master's existing *TIME and *DATE commands, the equivalent RM commands have been changed to *RTIME and *RDATE. The time & date set commands, *TSET & *DSET (and all other RM commands), remain the same. The Master already has an OSWORD 14 Type 1 call to return the time and date from it's own internal clock and so the RM implementation of this call (designed for the 'B' series) has been removed.
- The 'Time-on-Break' function only displays the time stamp after Power On or <Ctrl><Break>. It is not displayed after a simple <Break>.
- The Master does accept internal Sideways Roms but only one can be fitted (ROM number 8, IC 27) if all the internal default Sideways Ram is to be utilised. Additional sockets may be utilised (ROM numbers 4/5 and 6/7, IC 41 and IC 37) but an associated link must be moved which disables two 16k SWR banks and allocates the selection logic to one 16k or 32k Sideways Rom.

Thus, it is advisable to use ROM 8, IC 27 for RAMagic! and use external cartridges for additional Sideways Roms if all the four banks of internal Sideways Ram are to be retained. RAMagic! will function correctly in a Master cartridge but unfortunately is physically too large to allow the cartridge to be fitted with the top cover in place.

The table below details the operation of the Master ROM/SWR and associated links :

ROM ID	Link for SWR	Link for ROM..	..at IC
6/7	LK19 West	LK19 East	IC37
4/5	LK18 West	LK18 East	IC41

Appendix 1

The red hook probe is used to pick up the system Read/Not Write signal which is widely available across the main board. Some suggested attachment points are :

BBC Model B

IC73 (D7002C) Pin 24 - The 28 pin ADC chip for, and close to, the analogue port. Usually the best option, most accessible and the cleanest version of 'Read/Not Write'. If using an ATPL Rom board, this chip is lifted up onto the Rom board and still easy to get at.

IC14 (LS245) Pin 1 - Very accessible being a Pin 1 connection but chip often buried under expansion boards. To lower right of 6502 CPU.

IC26 (LS139) Pin 3 - Another option, again often hidden by expansions. Lower left of CPU.

IC33 (LS04) Pin 10 - Out of the way at the top left and rarely hidden but slightly less room for probe.

There are more options if you can't get at any of these but usually at least one will be accessible. Contact me if you need any help with this connection.

BBC Model B+

On this model, there are usually no significant expansion modules which interfere with attachment points and so I recommend using the IC84 (D7002C) Pin 24. This is the large 28 pin ADC chip and is just behind the analogue (joystick) port. As you face the machine from the keyboard, Pin 28 is top right of the chip and the pin numbers count down towards the keyboard.

If you have a Sidewise rom expansion board, IC84 is covered and in which case you can either remove the Sidewise and attach the probe before refitting Sidewise, or use a different attachment point. If you require the latter, please contact me and I will advise on alternative connections.

Notes :

1. If you have a Master 128 which is fitted with **Retroclinic's** dual MOS board, there is a problem in that this board completely covers the rom socket farthest from the keyboard and partly obscures the centre socket, the latter being the one I recommend for RM use. However, this board then prevents RM from being fitted in either of these sockets and the socket nearest the keyboard cannot be used for RM since it fouls the speaker/cartridge housing. I am currently experimenting with the fitting of additional empty sockets under the Dual MOS to see if it can be raised sufficiently to clear RM whilst not itself fouling the Master top cover.
2. If you have a B+ which is fitted with **Retroclinic's** dual OS board, care is needed to select a suitable rom position for RM and other roms since the Dual OS board obscures some of the sockets.