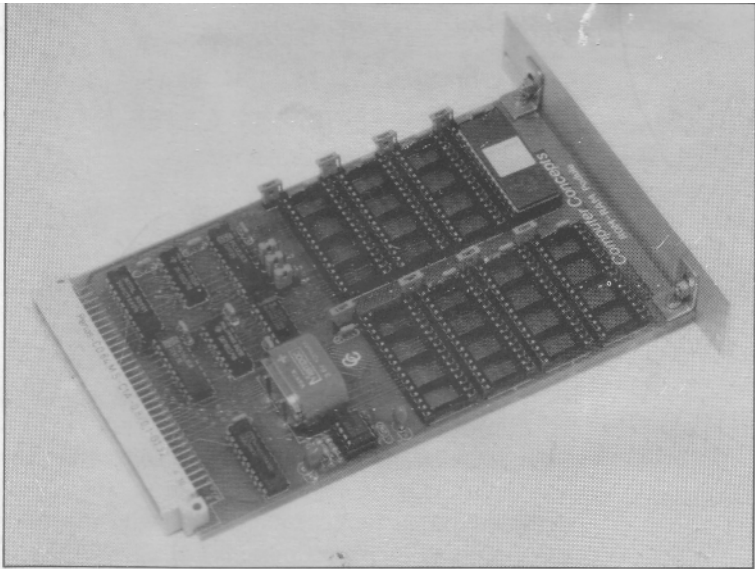


ROM/RAM Podule

For The Archimedes



Computer Concepts Ltd

Contents

1	Introduction	1
2	Installing the ROM Podule	2
2.1	Installing ROM or RAM	7
3	The ROM / RAM Filing System	11
3.1	RFS Commands	13
4	Practical Usage	17
4.1	Handling ROM Files	18
4.2	Using Podule RAM	21
5	Fitting Battery Backup circuitry	30
6	Creating an EPROM 'image file'	34
7	Technical Information	40
7.1	Error messages	41
8	Index	48

© 1988 Computer Concepts Ltd.

All rights reserved. No part of this documentation, software or circuitry may be copied in part or in whole, stored or transmitted in any form or by any means without prior written consent from Computer Concepts Ltd.

Computer Concepts Ltd. have no direct control over the use to which this product is put and therefore offer no warranty, implied or otherwise, relating to its use for any purpose other than that for which it was intended. No responsibility is accepted for the installation of the ROM Podule or for any consequential damage which results from its incorrect use

Trademarks: Archimedes is a trademark of Acorn Computers Ltd., Cambridge, England.

1 Introduction

The Computer Concepts ROM/RAM podule is an internally installed add-on board which accepts a combination of RAM and EPROM chips. In addition to commercial software supplied in ROM form, the user may also create EPROM images with an in-built utility program. The on-board RFS (ROM/RAM filing system) provides a complete interface to files on the podule. whether in ROM (read only) or RAM (read/write).

ROM files load more quickly and are always conveniently available. With RAM fitted to the board, files can be saved and loaded more quickly and conveniently than when floppy discs are in use.

The RAM facility becomes even more useful in conjunction with the optional battery backup circuitry. All files saved in RAM are protected against loss when the machine is switched off. In this way, the user may create an auto-start procedure which is executed as soon as the machine is turned on.

This manual describes the installation and subsequent use of the Computer Concepts ROM/RAM Podule for the Archimedes 305, 310, 410 and 440 models. It is assumed that an operating system version 1.2 or later is installed.

Installation and usage of the optional RAM chips and optional battery backup kit is also described.

The next section deals with the installation and testing procedures.

IMPORTANT NOTE: If an Acorn ROM Podule is already installed, its control software is superseded by the Computer Concepts RFS and should be disabled with the following commands: -

*UNPLUG ROMFS Return

*UNPLUG ROMBOARD Return



2 Installing The ROM Podule

Caution!

Whilst no special skills are required to fit the ROM Podule, the suppliers can accept no responsibility for any damage which may result from incorrect installation. If in doubt, consult your local dealer or supplier.

Tools

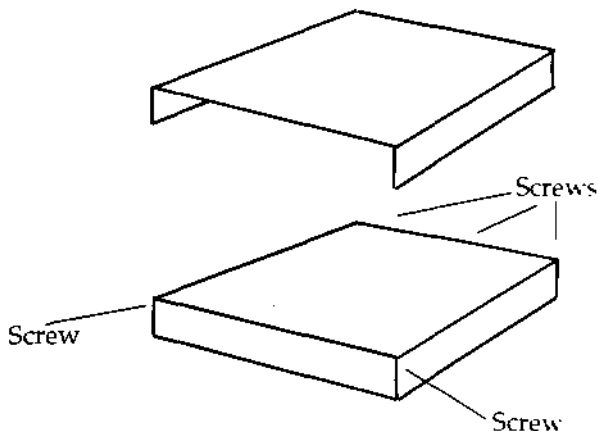
The only tool which you will require is a No.2 cross-head screwdriver.

Disconnect power

Before proceeding, disconnect the power lead and *all* peripherals. Remove anything which may be on top of the computer unit, such as a monitor.

Remove cover from case

The cover is secured to the main unit by five screws, positioned as indicated in the diagram below.



Remove all five screws. The cover does not lift vertically off the main unit. Instead, slide it backward then lift it away

Ensure backplane is fitted

It is first necessary to ensure that a Podule Backplane is installed in your machine. This is a vertically mounted circuit board, positioned approximately half-way back across the main circuit board. It provides two or four sockets into which podules can be fitted.

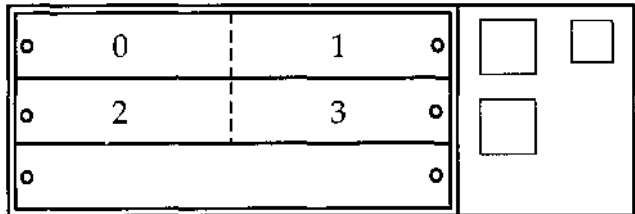
Podule positioning

A podule backplane is factory-fitted to 410 and 440 models, but must be fitted as an optional extra to the 305 and 310 models. If the backplane is not fitted, install it now before continuing.

With no podules fitted, the rear panel of the main unit consists of three metal blanking plates, each secured to the case with a screw at both ends.

The lowest plate (containing various sockets) is secured to the main circuit board and should not be removed during podule installation. The upper two plates are temporary fitments which are replaced when podules are installed.

The Archimedes rear panel, indicating possible podule positions



One 'standard width' podule occupies half the width of one blanking plate. The numbers 0...3 on the diagram above indicate the positions in which podules may be fitted. With a two-socket backplane, as used in the 305 and 310, only positions 0 and 2 are available.

If the ROM podule is fitted adjacent to another podule, their rear panels should be bolted together after installation. However, if no adjacent podule is fitted, the half-width blanking plate supplied should be bolted to the ROM podule rear panel.

Remove rear blanking plate

Unscrew and remove the chosen blanking plate from the rear of the case. The choice of position is up to the individual, unless instructions for another podule give specific instructions. However, position 0 (the top plate) is the logical first choice if there are no other governing factors.

Fit the ROM Podule

The ROM podule may now be inserted (usually from the rear) and, whilst held horizontally, pushed into position in a corresponding backplane socket. The backplane is supported by a strong metal bar; however it is advisable to support it with one hand whilst pushing the podule into place with the other hand.

Secure the podule

Replace any screws which may have been removed from the back panel, ensuring that the ROM podule is connected to an adjacent podule or the half-width blanking plate supplied.

Replace the cover

The podule is now fitted and ready for initial testing. Replace the cover, but leave it unscrewed. It will be necessary to re-open the case in order to install chips in the ROM podule.

Initial testing

Re-connect the power and any peripherals removed at the outset. Turn on the power in the usual way. It is normal for the machine now to take a second or two longer before giving the initial start-up beep.

If the computer fails to start in the normal manner, see the section entitled 'If problems occur...' at the end of this chapter.

Assuming the machine starts correctly, list the details of podules fitted by entering BASIC and typing:-

***STATUS ROMBOARD** and press Return

This should display a result similar to the following:-

```
Podule 0 (Computer Concepts)
  Socket 1 = 27256,   32K EPROM
  Socket 2 = Empty
  Socket 3 = Empty
  Socket 4 = Empty
  Socket 5 = Empty
  Socket 6 = Empty
  Socket 7 = Empty
  Socket 8 = Empty
```

This indicates that a 32K Manager ROM has been detected in socket 1. Even if other chips have been inserted, they will not yet be recognised (see 'Configuration' below).

If this fails to show a reasonable result then there could be any one of a number of causes...

The most likely error to occur is "Bad status option" in response to the command given above. First ensure that the machine's operating system is a recent enough version. Type the command:-

***FX0** and press Return

This should display a result such as:-

Arthur 1.2 (25 Sep 1987)

If the version number or date are prior to this, the operating system needs to be updated.

Next, ensure that the podule controlling software is present in the machine and not 'unplugged' (i.e. disabled). Type the command:-

***ROMMODULES** and press Return

This will list the position, module name and status of each operating system software module. The module entitled 'Podule' (No.23 on Operating System version 1.2) is the important one. Its status must show as Active, e.g.:-

No	. Position	Module name	Status
1	System ROM	UtilityModule	Active

23	System ROM Podule		Active
----	-------------------	--	--------

If the status shows as "Unplugged" then it has previously been software disabled with a ***UNPLUG Podule** command. The Podule module must be active for any podule to operate. Re-initialise it by typing the command:-

***RMREINIT Podule** and press Return

Now switch the computer off, then on again and return to the stage of typing the ***STATUS** command.

If the Podule module is active, the list of ROM Modules and their status should show two modules present on the ROM/ RAM podule. If the ROM/RAM podule is the only one installed, and is positioned in podule slot 0, the command:-

***ROMMODULES Return**

should show the following entries at the end of the list:-

No	. Position	Module name	Status
23	System	ROM Podule	Active
1	Podule 0	RFS	Active
2	Podule 0	ROMPodule	Active

If either of these two modules is inactive, re-initialise them with one or both of the commands:-

***RMREINIT ROMPodule**

***RMREINIT RFS**

If an error is given in response to either command (for any reason other than incorrect typing of the command) there is probably some kind of software fault; contact Computer Concepts, providing details of the tests and results from above.

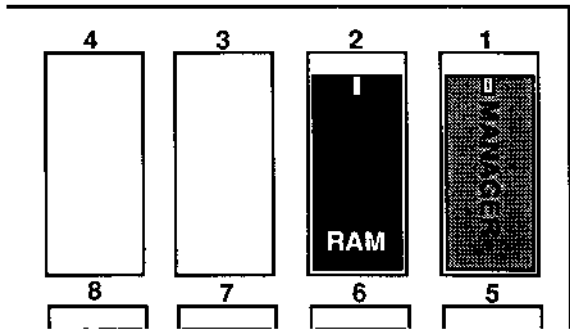
If the RFS and ROMPodule modules fail to appear in the list then there is probably a hardware fault - refer to "If problems occur..." at the end of this chapter.

2.1 Installing ROMs or RAM

Now that the ROM Podule appears to be working, the next stage is to install the required ROM and/or RAM chips. You may find that this has already been done by your supplier, in which case you can omit this section.

Once again, disconnect the power and peripherals, then slide off the cover.

Each ROM or RAM chip will have either 28 or 32 pins. All of the sockets will accommodate 32 pin devices, but 28 pin devices can be positioned in the lower portion. Secondly, each chip will have a small notch or similar indication at one end, which must be pointing toward the top of the board (the left hand side when installed in the computer, looking from the front), as in the diagram below. Use the pre-installed manager ROM as a guide.



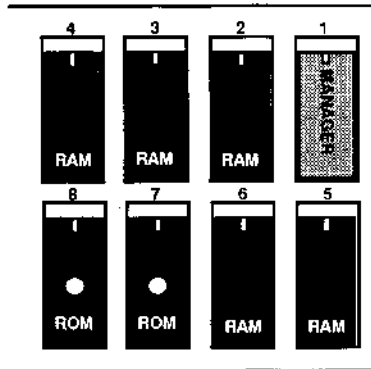
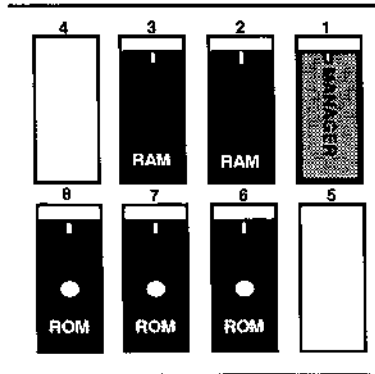
When installing ROMs, it will be necessary to exert a small amount of force until it 'clicks' into place. Always provide sufficient support to the underside of the board to prevent any force being placed on the board itself. Failure to do so may flex the circuit board and damage it.

The ROM Podule has eight sockets, numbered 1...8. Socket N21 is always occupied by the 'Manager' ROM. *The ROM podule will no longer work if the Manager ROM is removed from socket No.1.* This leaves seven sockets available for either ROM (EPROM) or RAM.

The sockets have a pre-defined order of priority. Socket 1 has the highest priority and socket 8 has the lowest.

RAM chips should be inserted starting at the highest priority socket available, number 2, with additional RAM chips following in numbered sequence.

ROM chips should be inserted starting at the lowest priority socket, number 8, with additional ROMs following in order down the numbered sockets. No sockets should be left empty except in the range between RAM and ROM chips. The diagram below illustrates some example layouts.



Re-fit cover

Make one final check that the chips have been inserted in the correct order, that they are the right way round (notch to the top, using the manager ROM as a guide) and that any 28-pin chips are positioned at the right of their sockets (also as the manager ROM). If in doubt, *consult your supplier*. *Inserting* chips the wrong way round or in the wrong part of the socket is likely to damage them.

The cover on the main unit can now be replaced. Slide it on from the back and replace the screws. Replace all power leads and peripherals and re-connect the power. Switch on the machine and ensure that it starts correctly - if not, see 'If problems occur...' at the end of this chapter.

Configuration

The ROM podule manager software does not initially know that any chips have been installed. It is necessary to 'configure' the board whenever a chip is inserted or removed. After configuration, the chip type and position details are retained even when the machine is switched off.

To configure the ROM podule, enter BASIC and type the command:-

AUTOCONFIG *n and press Return ...where *n* is the podule slot number, 0...3.

This performs an 'automatic' configuration of the podule, in which the manager software determines the type of chip installed in each socket. The alternative is that the user must give exact details for each socket number. AUTOCONFIG will, as a side effect, delete the contents of on-board RAM. A confirmation prompt is therefore issued when the command is given. Next, enter BASIC and type:-

***STATUS ROMBOARD** and press Return

As described earlier, this will show the sockets and their contents. Any sockets which are not as expected should be configured individually. If this fails as well, then either the chip has been fitted incorrectly or there is a fault elsewhere.

If problems occur...

Fitting a ROM podule is usually very straightforward.

However, it is possible for a problem to arise at any one of the stages during installation. In each case, start by disconnecting the power and checking that all instructions for podule installation have been followed up to that point . If not, go back and perform that part again.

Sometimes an additional sheet of instructions will be supplied, either with the ROM podule or with a ROM/RAM chip purchased separately. Check carefully that the problem which has arisen is not covered therein. If instructions supplied with an add-on chip conflict with those herein, it is likely that the additional instructions are more recent and therefore correct.

If no cause for the fault can be found, or it cannot be resolved, consult your supplier. If the podule itself appears to function, but one particular chip does not, then it is advisable to contact the supplier of that chip; otherwise contact the supplier of the ROM podule.

IMPORTANT: Do not attempt to modify any part of the podule or computer.

When reporting a fault, to the supplier of a chip, board, computer, or Computer Concepts Ltd. direct, certain information should be provided in order to save time.

Each podule has a serial number. This indicates a production batch number as well as a unit number. Always quote your serial number.

The first step is to describe the current problem, a particular error message for example. This may enable an immediate diagnosis to be made.

Next, if necessary, describe with adequate detail the steps taken which have led to the problem. Even if this appears to be re-stating the installation instructions, it allows an engineer to see your interpretation of the instructions.

Only return the podule itself if requested to do so.

Reporting faults



3 The ROM/RAM Filing System

All filing systems for the Archimedes are designed to act in a similar manner to each other. All of the general commands are the same, acting upon whichever filing system is active at the time, or specified explicitly in the command itself.

Familiar commands such as LOAD used in BASIC will operate in the same way when using the RFS. *CAT or *, which normally gives a catalogue of the disc, will similarly give a catalogue of files present in the ROM podule.

Usually, a machine will automatically start in the disc filing system (ADFS), though any machine can be configured to start in any available filing system, including RFS.

Selecting the RFS

Type the command:-

***RFS** and press Return

in order to select the RFS as the current filing system. All filing system commands now issued will operate on the RFS. The command:-

* . Return

will produce a catalogue list of all files present in ROM or RAM on the podule. The list is sorted alphabetically.

The command *FREE will display the number of bytes available for *storing* files in the RFS. This will be zero if no RAM is fitted or if all the space is currently occupied.

Exiting from RFS

The RFS is left by issuing a command to enter a different filing system. For example, use the command *ADFS to select the disc filing system.

Specifying an RFS file from another filing system

Conventions exist which enable files in filing systems other than the current one to be specified. With the RFS selected, a file named "FRED" is specified simply as "FRED". However, with some other filing system selected, the same file can be specified as "RFS:FRED" or as "-RFS-FRED". Both alternatives are equally acceptable. In the same way, an ADFS file can be specified from RFS by "ADFS:\$AFILE".

* Command Compatibility

All 'Fileswitch' and 'Operating System' star commands are implemented, together with a few ADFS equivalents. The following list provides a brief summary.

*ACCESS	*DUMP	*NOLIB
*ADFS	*ENUMDIR	*NOURD
*APPEND	*EX	*OPT
*BACK	*EXEC	*PRINT
*BUILD	*FREE	*REMOVE
*CAT	*INFO	*RENAME
*CDIR	*LCAT	*RUN
*CLOSE	*LEX	*SAVE
*COMPACT	*LIB	*SHUT
*COPY	*LIST	*SPOOL
*CREATE	*LOAD	*SPOOLON
*DELETE	*MAP	*TYPE
*DIR	*NODIR	*URD

Unavailable ADFS commands

The following commands available in ADFS are inappropriate to RFS but, rather than generate an error if software issues one of these commands from within RFS, each command is implemented with a null operation. *MOUNT is an exception; in RFS this will reset current directories etc. as in ADFS.

*BACKUP	*DRIVE	*NAMEDISC
*BYE	*FORMAT	*TITLE
*DISMOUNT	*MOUNT	*VERIFY

Note: If appropriate, the ADFS may be specified as a prefix to the command, for example:-

***-ADFS-MOUNT**

will explicitly call the ADFS *MOUNT command.

Data Files

Data file handling is not performed by star commands but is provided by individual programs which themselves call a number of low-level filing system routines. The RFS has been designed to support the 'standard' complement of commands usually available in ADFS. From BASIC, this means that all of the commands: OPENIN, OPENOUT, OPENUP, CLOSE, BGET, BPUT, EXT, EOF, INPUT# and PRINT# will work in the usual manner. Obviously, write operations will only work if RAM is fitted to the podule.

3.1 RFS Commands

In addition to the general filing system command already described, a number of additional star commands are provided. These can be issued whilst outside RFS.

Automatic configuration

***AUTOCONFIG <podule N9>**

This command automatically configures the ROM/RAM podule. An attempt is made to determine what is installed in each of the sockets in the specified podule. Two pieces of information are required, (a) whether a chip is ROM or RAM, (b) the memory capacity. With standard devices it is unlikely that the automatic configuration will produce incorrect information, but the determination methods are not infallible.

The automatic configuration command stores the device map in a dedicated area of battery-backed CMOS RAM in the Archimedes itself. The RFS will re-initialise itself following a *AUTOCONFIG command.

As a side-effect of AUTOCONFIG, all RAM on the podule is cleared. Any files stored in podule RAM at that time will be lost. Because this is potentially an unpleasant side-effect, a safety-net prompt is first issued when the command is used. The action is only performed upon confirmation.

ROM Podule information

***ROMPOD**

With no following parameters, this command lists details of all the ROM podules fitted. With only one podule installed, fitted in slot 0, the command would show the following:-

ROM Podules:

Podule 0 (Computer Concepts)

***ROMPOD <Podule N2>**

Followed by a podule N2 (in the range 0...3), this command will list details of the devices fitted to that podule. The listing shown is the same as the *STATUS ROMBOARD command when only one podule is fitted, for example:-

Podule 0 (Computer Concepts)

Socket 1 = 27256, 32K EPROM

Socket 2 = 62256, 32K RAM

Socket 3 = 62256, 32K RAM

Socket 4 = Empty

Socket 5 = Empty

Socket 6 = 27512, 64K EPROM

Socket 7 = 27128, 16K EPROM

Socket 8 = 27512, 64K EPROM

***ROMPOD <Podule N2> <Socket N2> <Type>**

By specifying socket and type parameters, the ROMPOD command can be used for direct configuration of chip types in podule sockets.

The <Type> parameter may be any of:

0,27128, 27256, 27512, 271024, 6264, 62256, 621024, or 0, 128, 256, 512, 1024, 20_65, 20_256, 20_1024 or 0,1,2,3,4, 5,6,7, corresponding to: 0=Empty, 16K, 32K, 64K ROM/EPROM, 128K EPROM, 8K, 32K, 128K RAM chip.

Whilst the *AUTOCONFIG command is generally much easier to use for straightforward configuration, the *ROMPOD command does have advantages in certain circumstances. For example, given that the state of a podule after auto configuration is as listed above, the device shown

in socket 7 is a 16K EPROM. This chip can be software 'unplugged' by indicating that the socket is empty. The command would be:-

***ROMPOD 0 7 0** Return

After issuing this command, cataloguing the RFS will show that files contained in the ROM in socket 7 have disappeared. As far as the RFS is concerned, there is no ROM in socket 7. Because this information is stored in battery-backed CMOS RAM, the state will remain the same even if the machine is switched off and back on again.

Issuing the command:-

***ROMPOD 0 7 27128** Return

will re-insert the ROM. Cataloguing the RFS will confirm that the files are once again present. Note that when *ROMPOD is used to configure a socket the RFS is automatically re-initialised in order to update the catalogue.

Warning: Careless use of configuration can have undesirable effects. For example, files stored in RAM nearly always spread across chips (assuming more than one RAM chip is installed); removing a RAM chip either by configuration or physically will destroy the linked file structure and almost certainly corrupt all RAM files.

ROMBOARD STATUS

***STATUS**

Once the ROM Podule is installed (and the RFS modules are present) the Operating System *STATUS command will show the status of each ROM podule at the end of the usual status list. The status displayed is the same as that given by the *ROMPOD <n> command.

***STATUS ROMBOARD**

By specifying ROMBOARD as a parameter following *STATUS, the status for all ROM Podules is listed. Note that the word "ROMBOARD" is used, rather than "ROMPOD", in order to offer compatibility with Acorn's official syntax.

Acorn Compatible Configure command

`*CONFIGURE ROMBOARD <Pod No><Skt><Type>`

The `*CONFIGURE` command is implemented in the Archimedes itself, but the "ROMBOARD" option is implemented in the RFS. This particular command is provided merely for compatibility with the Acorn ROMFS.

The parameters are more complicated than for the `*ROMPOD` command, so there is little point in using the `*CONFIGURE` method. However, for completeness, the syntax is as given above, but the Type parameter must be one of: 10_128, 10256, 10512, 10_1024, 2064, 20256, 20_1024. The 10_ prefix indicates EPROM and the 20_ prefix indicates RAM. Note that "_" is an underline character, not a hyphen. These prefixes are optional, but if omitted, the device is assumed to be an EPROM.

Creating an EPROM Image file

`*MAKEROM`

This is the command used to invoke the utility program for creating EPROM image files. See section 6 for further details.

4 Practical Usage

The Computer Concepts ROM/RAM Filing System is very similar to the ADFS in many respects, especially when RAM is installed in the podule, allowing saving operations as well as loading. The majority of commonly used commands for dealing with disc files are also available within the RFS. Cataloguing, selecting directories, etc. are all supported.

For a ROM podule without RAM installed, its use is limited to execution of commercially available ROM software and 'home brewed' EPROMs.

A ROM Podule fitted with RAM provides a fast RAM disc for saving and loading programs and data. One of the most useful facilities is the creation of auto-start-up files (!BOOT) which can be made to execute a series of commands as soon as the computer is turned on. At the simplest level this can be used to set preferred values in operating system variables and, at a higher level, to ensure immediate execution of a particular program.

This section of the manual describes some commands and facilities which may prove useful in conjunction with the ROM podule, though the more interesting facilities are only available with RAM, and preferably battery-backup as well.

ROM based software packages

Some software is only available in ROM form, though other packages available on disc may also be available as a ROM version. Contact your supplier for details of price and availability. The advantage of having software on ROM is that it is always instantly available as part of the machine - no need to load it from disc each time, with the added advantage that it will load very much faster as well.

Each package which provides software on ROM should also provide the necessary start-up instructions, configuration size, etc. since these will vary from one program to another.



Putting files in EPROM

A comprehensive utility program is provided in the Manager ROM, enabling one or more files to be taken from disc to produce an 'EPROM image file'. This can then be 'blown' into an appropriate EPROM, using a proprietary EPROM programmer, and installed in the podule. The "MAKEROM" utility is described in chapter 6.

Note that auto-start files described later in this section may also be included within an EPROM and will work the same in EPROM as in RAM - the difference is that making changes is not so easy!

4.1 Handling ROM Files

By their nature, ROM files may not be modified in any way by RFS commands. It would also be reasonable to expect that their catalogue entries could not be changed, but in fact they can.

When the RFS first starts up, usually at switch-on, it examines the files stored on the podules and creates a catalogue in main RAM. The catalogue entries for ROM files therefore exist in main RAM, rather than in the ROMs themselves. This enables ROM files to be temporarily renamed, deleted, etc.

The point to remember about ROM file catalogue entry changes is that the effects last only until the next time the catalogue is re-constructed. This occurs explicitly upon a hard reset (Ctrl-RESET), switch-on, and when the RFS is reinitialised (*RMREINIT RFS). The catalogue is also reconstructed when one or more sockets are re-configured, for example when *AUTOCONFIG or *ROMPOD n n n is used.

ROM File Access attributes

ROM files are automatically assigned an access type of LR, meaning that they are Locked and may be Read or loaded. Using the `*ACCESS` command will change these attributes, even for ROM files. However, it is impossible to write to ROM files, so an attempt to set the `W` attribute will simply be ignored.

Note that RFS makes an exception for ROM files, ignoring the `L` attribute, so that they can be deleted without first changing the access.

ROM File Deletion

No matter which commands are used either by accident or design, ROM files cannot be damaged. However, it is sometimes useful for them to appear to be deleted, which the RFS allows, by use of standard file deletion commands such as `*DELETE`.

As stated above, all ROM files are 'Locked' but the file attributes may not be changed. In normal circumstances these two facts would prevent deletion, since a file must be unlocked before it can be deleted. However, RFS deliberately *ignores* the file attributes when an attempt is made to delete a ROM file.

When a ROM file is deleted, all that happens is that its entry is removed from the RFS catalogue held in normal Archimedes RAM. Therefore, if a file named "ANYFILE" exists in ROM, it can be temporarily deleted by the command:-

***DELETE ANYFILE** Return

The file will then have disappeared from the RFS catalogue, until the catalogue is re-constructed (see above).

When using `*WIPE` to delete files, it will be necessary to specify the `F` option in order to Force deletion of locked ROM files. For example, the command:-

***WIPE * F** Return

would delete all files from both ROM and RAM (assuming that the RFS is selected of course). Unless the ***WIPE** default options have been changed, a safety-net question prompt will be issued prior to deletion of all files. In order to force confirmation, use the C option, for example:-

***WIPE * F C** Return

To cancel this confirmation use **-C**, which means cancel the C option. BUT, be warned, whilst occasionally a useful fast way to clear all files from RAM and ROM, the following command will delete everything! Remember that RAM files cannot be retrieved once deleted.

***WIPE * F ~C** Return

ROM File Renaming

The standard ***RENAME** command may be used to rename ROM files. As described above, the change will only stay in effect until the next time the RFS catalogue is re-built.

If two files exist with the same name, only the first one in the catalogue list will be accessible. However, by renaming it, the duplicate named file becomes accessible. If necessary, a little juggling with ***RENAME** can achieve renaming of just the second file, i.e. rename first file, rename second file, then rename first file back to its original name - only the second has changed.

Rather than go through a series of renaming commands manually each time the catalogue is re-built, such commands can be installed within a **!BOOT** file to be executed whenever the machine is initialised.

ROM File Copying

The standard ***COPY** command may be employed to copy files from podule RAM, but not necessarily from ROM. Some files in ROM will only work in ROM, not in RAM (or disc) form. This may be because of specific anti-piracy checks made during execution of commercially available software or, with the same result, because of memory-saving

program overlay methods. In either case, it will be found that either the files will simply not copy from ROM to RAM (or disc), or that having copied them, the copies will not execute. This is not a problem because files in ROM cannot accidentally be deleted and are quite safe. The usual need to make regular backups does not apply.

4.2 Using Podule RAM

The cost of putting a small amount of RAM on a podule is very little, and very worthwhile. The provision for !BOOT files (see below) alone warrants its use.

Unfortunately, unless the battery backup option is installed, any files stored in podule RAM will be lost when the machine is switched off. This can be inconvenient, but not insurmountable, since the entire RFS contents may easily be copied to disc.

Unlike ROM files (described above), catalogue entries and attributes for RAM files are stored in the podule RAM itself. Once deleted, they stay deleted. Access attributes, copying, deleting and renaming are the same as for disc. However, compaction is a little different.

Podule RAM Compaction

ADFS arranges its files on disc in a certain manner, as a result of which it is sometimes necessary to 'compact' them. This is a process by which all the valid files which have become scattered around the disc surface are brought together in as small a space as possible, eliminating wasted space between files.

RFS files are arranged in podule RAM in a similar manner to ADFS files on disc. This means that at any particular time the total amount of free space may consist of a series of small areas between files. When a file is saved, it must fit within one of these gaps. For example, assuming that the total free

space is shown (by *FREE) as 10,000 bytes. If this happens to consist of two non-adjacent areas, one of 4,000 bytes and one of 6,000 bytes, the largest file which could be saved would be 6,000 bytes long. Attempting to save a file which is 10,000 bytes long would result in an error message "Compaction required". When this message is given, it indicates specifically that there is room for the requested file to be saved, but that it will only be possible after compaction has taken place. (If there isn't enough room for a file even after compaction then a "RAM full" error is generated, corresponding to an ADFS "Disc full" error.)

***COMPACT**

The command *COMPACT will initiate compaction of podule RAM, assuming that RFS is the current filing system when it is issued. In order to direct the command specifically to RFS, use the usual convention, for example:-

*** -RFS -COMPACT**

Note that although the ADFS *COMPACT command accepts various parameters, (disc and Quick option) the RFS equivalent requires none. However, for the purpose of compatibility, the RFS equivalent does not generate an error if one or two parameters are specified, although they are completely ignored.

Copying files between RFS and ADFS

The standard *COPY command may be used for copying files between filing systems as well as within the same filing system. The *BACKUP command is specific to ADFS and cannot be used in conjunction with the RFS.

Copying all RFS files to ADFS disc

It is necessary when podule RAM is not battery backed (and advisable even when it is) to make a backup copy of all RFS files to an ADFS disc. This may be achieved with the command:-

***COPY RFS:\$.* ADFS:\$.*R Return**

(the R option forces Recursive copying of sub-directories).

If the *COPY options (controlled by the Copy\$Options system variable) are in their default state, the above command will prompt with a question for every file, asking whether or not it should be copied. Whilst this is useful to avoid copying unnecessary files, there are likely to be so few in RFS compared with disc capacity that it is usually worthwhile removing the Confirm option. The following command will copy all RFS files to ADFS, giving information about all files copied, but without prompting for each:-

***COPY RFS:\$.* ADFS:\$.* ~C V R** Return

It is the V option (Verbose) which controls the printing of information about each file copied. Using —V instead of V will avoid printing this information.

Copying all ADFS files to RFS

If there is sufficient podule RAM space available, all files from ADFS can be copied to RFS in one operation. This is achieved with the command:-

***COPY ADFS:\$.* RFS:\$.* ~C R** Return

However, the Confirm option is often useful when copying this way round, allowing just selected files to be copied. Use the command:-

***COPY ADFS:\$.* RFS:\$.* C R** Return

This will prompt before copying each file, so that just the necessary ones can be chosen during the copy operation.

Copying with wildcards

Sometimes the files to be copied will have similar names or be in a specific directory. In this case, wildcard characters may be used to specify all files matching a certain criteria. For example, to copy all files from a directory called "Modules" on ADFS to an existing directory of the same name in RFS, use the command:-

COPY ADFS:\$.Modules.* RFS:\$.Modules.

Further details of the *COPY command and its options may be found by typing:-

***HELP COPY** Return

!BOOT Auto-start files

As is the case with ADFS, a file in the RFS can be made to automatically execute as soon as the machine is switched on. The name of the file to execute automatically *must be* " !BOOT" although this may simply be a program which executes another program in turn.

The simplest way to create a !BOOT file in podule RAM, is to make it a BASIC program. The following will serve as an example of the steps necessary to execute a file whenever the machine is turned on.

Enter BASIC and type the following, pressing RETURN after each line:-

```
NEW
10 PRINT"Auto-starting from RFS!"
*RFS
SAVE" !BOOT"
*OPT 4,2
*CONFIGURE BOOT
*CONFIGURE FILESYSTEM RFS
```

The one-line BASIC program simply prints a message to confirm that the program has been executed at the appropriate moment. *OPT 4,2 specifies that the !BOOT file should be executed by *RUN, which works fine for a BASIC program.

The first *CONFIGURE command specifies that a !BOOT file should be executed as soon as the machine is switched on. The second is used to select the RFS as the default filing system, so that it is used as soon as the machine is switched on. These are ignored until Ctrl-Break is pressed or the machine is switched off and on again.

Try the program by switching off for a few seconds then switch on and the message from line 10 should appear.

This is a simple program and takes no special steps to leave the machine in a reasonable state upon completion. However, for actual use it is advisable to perform a few additional commands in order to leave the machine in a position ready for use.

In order to make the above file execute at switch-on, it has been necessary to change configured options. After the above program finishes, no language is selected, the current filing system is RFS-and the actions of RESET and Shift-Break are reversed (which is an unwanted side effect of the *CONFIGURE BOOT option). Each of these points can be put right by ending the !BOOT program with a few additional commands: -

```
20 *ADFS
30 *FX255,8
40 *BASIC
```

Line 20 selects ADFS as the current filing system. Line 30 restores the actions of RESET and Shift-Break to their 'normal' state. Line 40 is used to select the start-up language, effectively overriding the configuration option provided for the same purpose. To select the desktop at the end of the program, line 40 could be replaced with:-

```
40 *DESKTOP
```

BASIC allows the SAVE command to be given without a filename, provided that the filename is present in the first line of the program, following "REM>". If line 5 is added to the program in the form:-

```
5 REM> !BOOT
```

then there is a high risk that a !BOOT file on disc will accidentally be overwritten, just by issuing the SAVE command when ADFS is the current filing system. It is far safer to be explicit, so that the file will always save in RFS, regardless of the current filing system. Use:-

```
5 REM> RFS: !BOOT
```

Sometimes it would be desirable to cause an ADFS disc to execute its !BOOT file after the RFS has completed its own startup. This is easily achieved, by issuing an operating system command to perform an auto-boot on the current filing system, having selected ADFS at the end of the RFS !BOOT program. This command could be added into the **RFS: !BOOT program as:-**

```
32 PRINT"Executing ADFS !BOOT..."
34 SYS"OS_FSControl",15
```

Note that the upper and lower case characters in line 34 must be as shown above. With these lines installed and the RFS !BOOT file operative, the ADFS is selected and an attempt is made to perform a !BOOT file from the default drive. However, the *OPT4 may not be set, there may not be a !BOOT file, or there may not even be a disc in the drive! The latter two will generate an error, whilst a disc with *OPT4,0 as the current setting will be ignored and the RFS !BOOT program will continue with line 40.

Unfortunately, it is not possible to select a language such as *BASIC before the end of the program, since the command will itself terminate the current program. Therefore, if a disc !BOOT program is executed, it must itself select a language on exit, otherwise the user will have to enter *BASIC or whatever manually. To cope with language selection when no !BOOT file or no disc is present, the following line could be inserted:-

```
31 ON ERROR GOTO 40
```

This would cause a language to be selected, effectively skipping the disc boot operation, if any error occurred whilst booting the disc.

To summarise the RFS !BOOT program following the additions discussed above, the complete program is listed below. The same line numbers as above have been retained to avoid confusion, although the sequence is now somewhat untidy. Lines 31, 32, 34 will very often be omitted for normal

use, but could be inserted and simply commented out. Remember that in every machine there are a few locations in CMOS RAM that a user may apply to whatever they wish. One BIT of this RAM could be configured appropriately and tested within the !BOOT file, executing optional instructions according to its setting. However, that goes beyond the scope of this manual.

Any additional instructions required should be inserted between lines 10 and 20, renumbering if necessary.

```
5 REM> RFS:!BOOT
10 PRINT"Auto-starting from RFS!"
20 *ADFS
30 *FX255,8
31 ON ERROR GOTO 40
32 PRINT"Executing ADFS !BOOT..."
34 SYS"OS_FSControl",15
40 *BASIC
```

Disabling auto-start from RFS

In order to prevent the RFS:!BOOT file from auto-starting, use the commands:-

***CONFIGURE NOBOOT** Return

***CONFIGURE FILE ADFS** Return

...then press Ctrl-Break. This will select ADFS as the default filing system and prevent !BOOT from being used at switch-on.

As a slightly faster way to prevent booting temporarily, the RFS boot option could be set to zero:-

***OPT 4, 0** Return

However, this would still leave RFS selected as the default filing system.

Setting the *RUN search path (Run\$Path)

Although not necessarily obvious, whenever a *RUN command is executed one or more specific filing system directories are searched for a file of the given name. The names of directories which are to be searched are held in a system variable named Run\$Path. Typing the command:-

***SHOW Run\$Path** Return

will usually give the result:-

```
Run$Path : Type String, value : ,%.
```

This shows that by default only the directory "%." (which represents 'the currently selected library directory') is to be examined for a file of the required name. For example, the command ***RUN FRED** causes a search for "% . FRED".

The Run\$Path is particularly useful in relation to RFS. For example, by setting it to RFS:%. all *RUN files will be searched for in RFS only. More usefully, the command:-

***SET Run\$Path ,RFS:%. ,ADFS:%.** Return

will cause automatic searching first in the RFS library, then in the ADFS library.

With this setting in force, if a BASIC program named "PROG" exists in RFS, the command:-

***RUN PROG** Return

will execute it, even if RFS is not the currently selected filing system.

The search order through a list of directories corresponds to the order in the Run\$Path variable. Therefore, if the file named PROG existed both in RFS and ADFS, the above Run\$Path would cause the RFS file to be found first. However, the result can be reversed by changing the Run\$Path to ,ADFS:%.,RFS:%. By searching RFS as the first priority, files can be copied to RFS and used in preference to those on disc, saving on loading time without adding to the commands required.

The RFS: !BOOT file given on the previous page may be extended to contain the line:-

```
11 *SET Run$Path ,RFS:%. ,ADFS:%.
```

In this way, the Run\$Path will always be set to search RFS then ADFS, unless explicitly changed. Alternatively, this command could be typed at the keyboard or included in a disc ! BOOT file if no RFS:!BOOT file is being used.

Resetting the computer or changing batteries

Information about the ROM Podule configuration and boot option are stored in the Archimedes internal CMOS RAM. Information in CMOS RAM is battery-backed, but if batteries run low or need to be changed (or if the machine is switched in whilst holding the R key), all of these settings are lost. In such a case, the ROM podule will appear to be empty, showing only the manager ROM present.

Using the AUTOCONFIG command is the simplest way to reconfigure the board, but as a side effect it will clear the contents of any on-board RAM! If this is not convenient then a *ROMPOD *n,n,n* command will have to be used to specifically configure each socket.

5 Fitting Battery Backup

The battery backup circuitry is an optional extra available for the Computer Concepts ROM podule. If this is already fitted to the board, then these instructions may be ignored. The following provides details of how to fit the battery backup option.

WARNING:- Although described as a 'user upgrade', fitting battery backup involves a small amount of soldering and the handling of electronic components. This upgrade should not be attempted unless the procedure is fully understood. User installation of the battery backup circuitry is not covered by the guarantee offered on the ROM podule.

Tools required

- Small soldering iron
- Solder suitable for electronic components
- Pair of fine side cutters or similar, capable of cutting in a confined space. Pliers or a knife are not suitable.

Components required

One "Computer Concepts ROM podule battery backup upgrade kit"

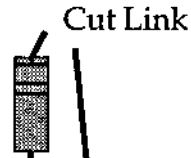
This consists of just two items:-

- IC16 ICL 7673
- B1 3V6 100mA/h PCB Battery (Varta Mempac)

Installation procedure

1) Completely remove the ROM / RAM podule from the computer.

2) Using a pair of fine cutters, cut the two wire links labelled LKO and LK1. Each is a wire loop, positioned below and slightly to the right of the battery position. If there is a chance that the cut links may re-contact, make two cuts so that a section of wire is removed.



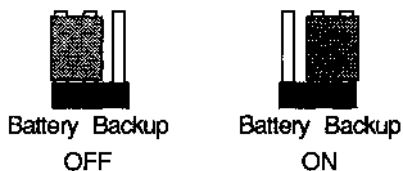
- 3) Gently push the three wire leads of the battery into the corresponding holes on the podule. The battery will only fit in one orientation on the component side of the board.
- 4) Using a soldering iron and solder, solder the three battery leads into place. Be careful not to apply the soldering iron for longer than necessary since this will cause excessive heating of other delicate components.
- 5) Insert IC16 into the socket adjacent to the battery and links. This is the only 8-pin socket on the board. Ensure that the orientation mark or notch is at the correct end, towards the battery.
- 6) Check that the procedure has been followed, then re-install the podule.

Link selection

At the top (notch) end of each ROM/RAM socket is a small shorting link, as illustrated below.



Each link acts like a switch, making an electrical connection between a common central point and one of two outer points. In this way the link provides selection whether or not power from the on-board battery will supply the corresponding chip. With the link set to the left, the corresponding chip will only be powered whilst the computer is switched on; set to the right, the chip will be powered by the on-board battery when the computer is turned off.



For each socket fitted with a RAM chip, remove the link and fit it to the right, sliding it over the rightmost two pins. Failing to set the link for a RAM chip will cause it to lose its contents when the computer is switched off. Mistakenly setting the link to the right for a ROM will cause excessive drain of the battery which may then run flat and cease to supply the RAM!

Charging the battery

When the battery is supplied it might be in a discharged state. The battery charges all the time while the computer is switched on, but charging from flat will take a long time. It is advisable to leave the computer switched on for about ten hours before relying on the battery backup (short intervals with the power off will not matter).

Testing the battery backup circuitry

When the computer has been switched on for an hour or two with the podule fitted, it will be possible to test the battery backup circuitry (the battery having charged sufficiently over this period for testing).

First ensure that one or more RAM chips are installed, and that their corresponding links (see above) are set to the right hand position.

Use the *FREE command to ensure that the number of bytes free corresponds approximately to the amount of RAM installed. Save a file - a BASIC program or a text document will do. Catalogue the RFS to make sure that the file has been saved, then switch the machine off for a minute or two. Switch the machine on again and catalogue the RFS to check that the file is still present.

If it does not appear to operate correctly, leave the unit longer to charge, then try again. Also check that the links have been selected correctly. Check over the instructions again and, if all else fails, contact the supplier with specific details of the problem. Do not return the board unless instructed to.

Battery discharge period

The battery fitted to the ROM Podule is re-chargeable. It automatically charges very slowly whilst the computer is switched on. Whilst the computer is switched off, the battery is supplying selected RAM chips and therefore gradually discharging. Eventually the battery will become so discharged that it can no longer supply the RAM chips and their contents will be lost. Exactly how long this will take is very difficult to estimate; it depends upon how well the battery is charged (which depends upon how long the computer has been switched on over the recent period) and the number, size and make of the RAM chips being powered. As a rough guide, it is unlikely that the battery will run flat in less than six months, even if all seven chips are inserted, provided that the computer has been on for about ten hours. However, depending upon the type and manufacturer of the RAM chips (some draw 100 times more power than others!), the battery may remain charged for well over one year.

To be on the safe side, if the computer might be turned off for more than a few weeks, copy the RAM contents to disc for safety. Since it is advisable to make a disc copy more regularly than this in case of accidental deletion etc., this is unlikely to need special attention.

Note that it is important to use only low power CMOS static RAM chips, usually denoted by the letter L or the letters LP in the part number. The access speed of RAM chips must be 120ns or faster. A typical part number for a 32K byte RAM chip would be 256C-12L.

6 Creating an EPROM 'image'

Within the Podule Manager is a utility program called "MAKEROM", used for creating an 'EPROM image file'. This allows a selection of disc files to be combined within one or more 'image' files.

Once an image file has been generated, it can be programmed into an EPROM, using an appropriate EPROM programming device. If access to an EPROM programmer is not available, Computer Concepts will take an image file and program it into an EPROM. Anyone interested in this service should contact Computer Concepts direct. Prices and availability are liable to vary from time to time. (Local dealers or national user groups may also be able to help).

Entering the MAKEROM utility

MakeROM is written in a form known technically as a *transient* program. When executed, it is loaded into RMA workspace, then its memory requirement is relinquished upon exit. However, it is quite possible that there will be insufficient RMA workspace available if MakeROM is used from within an application such as BASIC. Type the command:-

***MAKEROM** Return

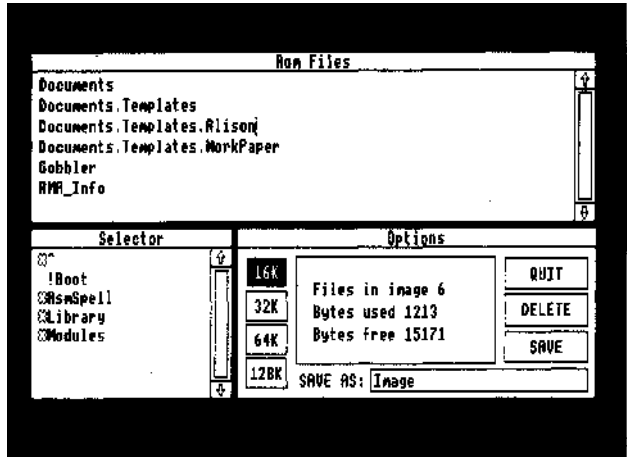
This should enter the WIMP environment and show the MakeROM program on the screen. However, if instead an error message:-

"MakeROM unable to claim workspace"
is displayed, it will be necessary to enter the operating system command mode in order to execute MakeROM. From BASIC, the commands to do this are:-

***QUIT** Return

***MAKEROM** Return

The MakeROM program should now be displayed on the screen, as shown below.

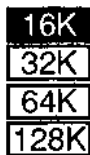


The MakeROM display is divided into three main areas; a list of files which have been selected for inclusion in the image file, titled "EPROM files" (top half), a file selector (bottom left) and a general status and options area (bottom right).

All operations in MakeROM are performed by mouse. The usual Acorn conventions are used for left and centre mouse buttons: the left button is used for all selecting, the middle button shows a menu. The right hand button is not required for any specific purpose by MakeROM and so acts as another select button.

Selecting the image size

Four buttons in the status area are provided for selection of the final image file size. Initially the smallest size, 16K Bytes, is selected. This will suit creation of a 27128 EPROM. Alternatively any one of the other three buttons may be selected by pointing the mouse cursor and clicking the select button.



The image size can be re-selected at any time. However, MakeROM will only allow a suitable size to be selected. For example, if the current image size is 32K and the total number of bytes used by files selected for inclusion is 20K, it would not be possible to select a 16K image size.

Selecting files for inclusion in the image

Files for inclusion within the image can be taken from ADFS or RFS. Initially MakeROM will show a list of files and directories from the currently selected disc/filing system in the file selector.

The scroll bar may be used in the normal way to move up or down the list of files in the selector window. Click twice in quick succession (double-click) in order to select one of the files. Immediately the selected file will appear in the EPROM files list and the status information will be updated accordingly, showing bytes used, bytes free and the number of files selected.

Additional files for inclusion are selected in the same manner. A warning dialogue will be displayed if a file being selected is too large to fit in the currently selected image size.

Pressing the centre mouse button whilst the pointer is over the file selector will display a menu. Whilst in ADFS, this menu allows a disc to be mounted, selection of an alternate disc drive, or selection of the RFS filing system.

Only drives which are available (according to the configuration settings) will be listed as selectable options.

Note that the Mount option may be used as a quick way to return to the top level directory of the currently selected disc.

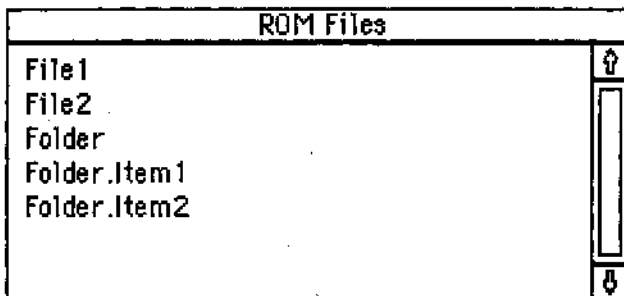


When RFS is the current filing system, either because it was current when MakeROM started or because it was selected from the menu above, the Options menu shows only one item - ADFS. The Options menu therefore allows an easy change back and forth between RFS and ADFS if necessary.

× Directories in the file selector

A directory is indicated in the file selector by a special symbol in the left hand column (×). Double-clicking on a directory name will select that directory and display the files within it. The first item in the file selector may be used to go back one level.

Each selected file is added to the ROM Files list, as illustrated below. A scroll bar is provided for lists which are larger than the standard list size.



Changing the name of a selected file

Once a file is in the ROM Files list its name may be changed. The new name will be the one which appears in the RFS catalogue when the EPROM is later inserted into the podule.

To change a name, click the select button anywhere on the name to be changed. A cursor will appear at the point where the mouse is clicked, then typing, pressing delete etc. will allow editing in the usual way. Press Return when the name has been changed as required. Note that although no error will be issued by MakeROM whilst editing, the filename must be entered according to the usual conventions.

Creating directories



RFS supports a form of directories similar in many ways to ADFS directories. However, because the official Acorn format for ROM files and filenames is different from ADFS, creating a directory is slightly unusual.

Clicking the menu button on the mouse whilst the pointer is over the ROM Files list will display the Options Menu. Once again, the selectable options are different to those available on other areas of MakeROM.

Selecting the 'New Dir' option will create a directory entry in the ROM Files list. This will always be named "Directory" by default, so should be renamed as described above.

Putting files into a new directory

When specifying a complete 'file path' in ADFS, including directory names, the form is:-

<directory>.<directory>.<filename>

Where each directory is separated by a full stop and the last item is the actual filename. The same conventions must be used when creating RFS ROM images (dictated by the official Acorn format).

Having created a directory entry, and renamed it, for example, "Folder", files which are to appear within that directory should be renamed as "Folder.<filename>". For example, two files named "Item1 " and "Item2" should be renamed as "Folder.Item1" and "Folder.Item2" in order that they will appear within the "Folder" directory when the resulting EPROM is plugged into the podule.

Deleting all current ROM files



Deleting one file from the ROM files list



The second item in the Options menu, available whilst the pointer is over the ROM Files list, is "Delete All". When used, this will first prompt for confirmation, then remove all of the files currently in the ROM Files list. It therefore provides a quick means for starting over again, which is particularly useful after completing one image and before starting another.

Note that this option will not have any effect upon files in the source file selector, i.e. disc files are quite safe.

A click-on icon labelled "DELETE" is provided in the status area (bottom right). This allows a single file to be removed from the list of files to be included within the EPROM image.

Point at the filename to be deleted within the ROM Files list and click the select button. This positions the cursor as if ready for editing. Next point at the "DELETE" icon and click the select button once only. The selected file is removed and the status information is updated, showing that the space occupied by the file in the image has been reclaimed.

Saving the EPROM image

Once the list of files for inclusion is complete and their names are correct, the image file should be saved onto disc ready for producing an EPROM.

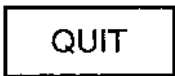
Using the file selector, choose the directory into which the image should be saved. Alternatively, insert a different disc and use the MOUNT menu option.

Bytes used 15631	DELETE
Bytes free 753	SAVE
SAVE AS: Image	

Beneath the status information is a box containing the filename to be used when saving the current EPROM image. By default, the name is "Image". Either use this name as it is or click on it to position the edit cursor and change it to the required name.

When the 'Save As' name is as required, point at the icon labelled SAVE (as illustrated above), and click the select button. If an error occurs, for example if there is insufficient room on the current disc, mount another disc and click once again on the SAVE icon.

Exiting from MakeROM



To end the session and exit from MakeROM, point at the icon labelled QUIT and press the select button.

Quitting from MakeROM by mistake could be very inconvenient, since all file selections, renaming etc. would be lost. Therefore, when the QUIT option is selected, a warning dialogue will appear, asking for confirmation to quit or providing the opportunity to cancel the action and continue. Select the appropriate option with care.

7 Technical Information

The Computer Concepts ROM Podule is designed to conform with Acorn protocols for both hardware and software. All documented calls necessary for implementing a filing system have been adhered to, therefore any software which does not directly reference a filing system specific command or feature should work under RFS.

Current Filing System

In order to determine whether RFS is the currently active filing system, use the "OS_Args" call with R0=0. For example:-

SYS"OS Args", 0 TO FS Return

This will return the current filing system number in the BASIC variable FS. Values for common filing systems are shown below.

NetFS	5
ADFS	8
RFS	22

A complete list of filing system names and numbers can be found in the official Acorn "Programmer's Reference Manual", volume 1.

*ROMLOAD command

A command not documented elsewhere in this manual is *ROMLOAD. This was provided for development purposes and is unlikely to be applicable in normal use. It allows the contents of a file to be loaded at a specified offset, directly into a RAM chip in any one of the ROM/RAM podule sockets. Note that this will destroy the entire catalogue of RAM files if the files currently span across the specified socket. It should not be used in a trivial manner and is documented here purely for completeness. Syntax:-

***ROMLOAD <podule><socket><filename> [<offset>]**

Acceptable EPROM types

Two distinct standards exist for EPROM pin assignments - JEDEC compatible and non-JEDEC compatible. The latter are designed to be compatible with 28-pin ROMs. The difference only arises with 128K Byte (MegaBit) chips.

The Computer Concepts ROM/RAM Podule will accept *only* JEDEC compatible devices, which excludes 128K byte ROMs, but permits certain types of 32-pin EPROMs. In case of doubt, contact the EPROM manufacturer to ensure that the device conforms to the JEDEC standard pin-out.

Access speed: Note that all chips used in the podule must have an access speed of 250ns or faster.

7.1 Error Messages

According to the Acorn convention, filing system errors return a four-byte error number of the form:-

Byte 3	Byte 2	Byte 1	Byte 0
00	01	Filing System number	Error number

ADFS is filing system number 8, and so returns error numbers of the form &000108nn. RFS is filing system number 22 and so returns error numbers of the form &000116nn. BASIC errors, Operating system errors and BBC compatible FileSwitch errors all return single-byte numbers.

The low-byte RFS error number has been kept the same as its equivalent ADFS error. This allows programs to check that byte 2 contains 01, then mask out all except byte 0 and act upon the general meaning of that error number. For example, a "Compaction required" error message is generated by ADFS and RFS, both with a low-byte error number 152. Upon occurrence of this error, a program could issue a *COMPACT command to rectify the situation.

0 "ROModule can't claim workspace"

This error is issued if the "ROModule" module was unable to claim workspace upon initialisation. It is unlikely that this will occur.

17 "Escape"

This error is issued if the Escape key is pressed.

144 "RFS Can't claim workspace"

This error is issued if, for some reason, the RFS cannot claim any workspace upon initialisation or re-initialisation.

145 "Can't add RFS"

This error is issued if, for some reason, the RFS cannot be added as a filing system.

150 "Can't delete Current directory"

This error is issued if an attempt is made to delete the currently selected directory.

151 "Can't delete Library"

This error is issued if an attempt is made to delete the currently selected library directory.

152 "Compaction required"

This error indicates that the sum of the free space on all podules is large enough for the required write operation, but that compaction is required to rearrange it into a single contiguous block. Note that this corresponds to the ADFS error with the same low-byte error number and the same message.

153 "No free space block"

This error is similar to the 'Compaction required' error above, except that in this case software compaction would not be able to combine the available RAM into a single block of the required size. This can occur either when RAM is spread across two podules or when RAM chips have been installed in non-contiguous sockets.

162 "Can't delete User Root directory"

This error is issued if an attempt is made to delete the currently selected user root directory.

168. "Broken directory"

This error is issued when corruption has been detected in a directory. It may be necessary to AutoConfigure the RAM in order to clear this state, though it is unlikely to occur.

168 "Object is a directory"

This is a FileSwitch error, issued when an attempt has been made to perform a byte access upon an object which is a directory.

179 "Directory full"

Unlike ADFS, RFS does not have true directories, since the Acorn structure for ROM files prevents it. Instead, the hierarchical structure is simulated. As a consequence, the total number of catalogue entries is limited to the number available in one directory. A "Directory full" error will therefore be issued by RFS when the *total* number of objects (file or directory names) would exceed 79. Note that the actual number of files allowable is liable to change between releases of the RFS.

180 "Directory not empty"

This error is issued when an attempt is made to delete a directory which still contains objects. The directory must be empty before it can be deleted using *DELETE. However, *WIPE is more flexible and can be used to recursively delete directories and sub-directories.

189 "Access violation"

This error is issued when an attempt is made to read a file which does not have read access, or write to a file which does not have write access.

192 "Too many open files"

This error is issued when an attempt is made to open one more than the permitted maximum number of open files (eight are allowed in RFS).

194 "File open"

This error is issued if an attempt is made to delete a file which is currently open for reading or writing.

195 "Locked"

This RFS error is issued if an attempt is made to delete or modify a file for which the 'L' (Locked) attribute is set. Note that all files in ROM are automatically locked, since it is physically impossible to modify them in the same way as RAM files. However, RFS makes an exception in the case of ROM files, allowing their temporary catalogue entries to be renamed or deleted. This error would be issued, for example, when trying to write to a ROM file or delete a locked directory name.

195 " <filename> is locked"

This is a FileSwitch error, similar to the RFS "Locked" error. It is issued on occasions when FileSwitch detects an error before passing it to RFS for action. For example, FileSwitch will issue this error if an attempt is made to create a file with the same name as one which exists and is locked.

196 "Already exists"

This error indicates an attempt to rename a file with the same name as one which already exists.

198 "RAM full"

This error is issued when an attempt is made to write information to podule RAM when there is insufficient free space remaining. Note that this directly corresponds to the ADFS error "Disc full" and has the same low-byte error number. Use the *FREE command to show the amount of free space remaining in RFS.

201 "Protected podule"

This error is issued if an attempt is made to write to a podule on which the 'write protect link' is set. See below for details of the write protect link.

204 "Name too long"

This error is issued if an attempt is made to use a file or directory name which exceeds the allowable maximum length of ten characters. Note that complete paths can be specified upto 63 characters long.

204 "Bad file name"

This is a FileSwitch error which indicates that the filename is unacceptable, e.g. a null string filename will generate this error.

214 "Not found"

This RFS error is issued when a referenced file is not found. For example, using *RENAME when the source filename does not exist.

214 "File not found"

A FileSwitch error, issued when a referenced file cannot be found - usually because it does not exist. This would be issued, for example, when an attempt is made to *DELETE a file which does not exist.

214 "*File<name>* not found"

Another FileSwitch error, as above. This is sometimes issued rather than the above message, though no available Acorn documentation states the difference. It is issued, for example, when an attempt is made to *EXEC a file which does not exist.

222 "Bad RFS channel on file handle *nn*"

This error is issued if the RFS is passed an incorrect channel number, when it has not been trapped by FileSwitch. This error should not occur in the normal course of events. It can occur if the RFS is re-initialised whilst a file is open.

222 "Channel on file handle *nn*"

This is a FileSwitch error which is issued when a file handle is specified which does not correspond to a currently open file. For example, this would be issued in response to the BASIC command BGET#1 when no file is currently open on channel N21.

222 "Channel"

This is also a FileSwitch error, similar to the above. It is issued, for example, when the BASIC command BGET#0 is used, which in turn will usually occur after opening a non-existent file (for which handle No.0 is always returned).

250 "No RAM installed"

This error is issued when an attempt is made to perform an operation upon RAM when none is installed. Note that this does not correspond to any documented ADFS error.

253 "Wild cards"

This error is issued when an attempt is made to use wildcards where they are not permitted. For example, it is **issued when wildcards are used in a *RENAME command.**

253 "Wild cards"

This error is issued by FileSwitch and corresponds to the same RFS message above. It is issued, for example, when wildcards are used in a *DELETE command.

254 "Bad command"

This error is issued by the Operating system when a *command is not recognised. Note that it will occur if any of the modules "Podule", "RFS" or "ROMPodule" are unplugged (or not present for any other reason) when an attempt is made to use one of their commands.

ROMPodule errors

These errors are generated by the ROMPodule module. Because they are Podule related errors, they have been given error numbers in the &500-&6BF range, designated by Acorn as the Podule error range.

&500 "Bad podule number"

This error is issued if a podule number is specified outside the acceptable range of 0 to 3 inclusive. Note that only podule numbers 0 and 2 exist on a two-slot backplane.

&502 "No installed ROM/RAM podule"

This error is given when attempting to act upon a podule which is not a ROM/RAM podule, e.g. a BBC I/O podule.

&508 "Bad socket number"

This error is issued when a socket number outside the acceptable range 1 to 8 is specified in a *ROMPOD, *ROMLOAD or *CONFIGURE ROMBOARD command.

&509 "Bad device type"

This error is issued when an invalid type number is specified in either of the commands *ROMPOD or *CONFIGURE ROMBOARD. The standard acceptable types are shown below:-

0	empty socket
27128	16K EPROM
27256	32K EPROM
27512	64K EPROM
271024	128K EPROM
6264	8K Static RAM
62256	32K Static RAM
621024	128K Static RAM

&50A "Not a Computer Concepts podule"

This error indicates that an attempt was made to act upon a ROM podule which is not a Computer Concepts type. For example, the *AUTOCONFIG command will only operate upon a Computer Concepts podule.

Loader errors

These errors are issued explicitly by the podule loader code, and have been given error numbers within the &500-&6BF range.

&581 Address out of range for podule**&582 Read only device****Podule Write-protect link**

The link labelled LK2 (positioned just to the left of socket No.4) can be used to write protect all sockets. Set to the left, write operations are enabled (the normal state); set to the right, all normal write operations are disabled. Note however that this link will not protect against deletion of RAM contents caused by the *AutoConfig command.

7 Index

- !BOOT 21, 24...
- *ACCESS 19
- *AUTOCONFIG 9, 13...
- *CAT 11
- *Command compatibility 12
- *COMPACT 21
- *CONFIGURE BOOT 24
- *CONFIGURE FILESYSTEM 24
- *CONFIGURE ROMBOARD 16
- *DELETE 19
- *DESKTOP 25
- *FX0 5
- *HELP 24
- *MAKEROM 16, 34...
- *MOUNT 12
- *OPT 4 27
- *RENAME 20
- *RFS 11
- *RMREINIT 5, 6
- *ROMLOAD 40
- *ROMMODULES 5, 6
- *ROMPOD 14
- *STATUS 5, 15
- *STATUS ROMBOARD 4, 9, 15
- *UNPLUG 1
- *WIPE 19, 20
- RFS- 11
- ADFS commands 12
- Arthur 5
- Attributes 19
- Auto-start files 24
- AutoConfig 9, 13...
- Automatic configuration 13...
- Backplane 2
- Battery Backup 30...
- Battery discharge 33
- BGET 13
- Blanking plate 3
- BPUT 13
- Changing filing systems 11
- Charging the battery 32
- CLOSE 13
- CMOS RAM settings 29
- Compaction 21
- Configuration 5, 9...
- Copying 22...
- Current filing system 40
- Cutting links 30
- Data files 13
- Deletion 19
- Directories 36
- Disabling !BOOT 27
- Disabling autostart 27
- Entering MAKEROM
- EPROMs 16, 34..., 41
- Error messages 41
- Exiting 11, 39
- Faults 10
- File handling 13
- Files in ROM 18
- FileSwitch 12
- Filing system number 40
- FS_Control 26
- Handling ROM files 18
- Image files 16, 34...
- Image size 35
- Installation:
 - Battery backup 30...
 - ROM Podule 2...
 - ROMs or RAM 7...
- Links 31, 47
- Loader errors 47
- Locked files 19
- MAKEROM 16, 34...

Manager ROM 5, 7
Numbering (podules) 3
OPEN 13
Options menu 36...
OS Args 40
Podule module 5
Podule numbers 3
Priority 8
Problems 10
RAM 21
Renaming 20
RFS 6, 11
RFS Commands 13
ROM file deletion 19
ROM files 18...
ROM/RAM Filing system 11
ROMBOARD 1
Romboard status 15
ROMFS 1, 16
ROMPodule 6, 46
Run\$Path 28
Saving an image file 39
Sockets 4, 7
Status information 15
System variables 28
Testing 4, 32
Unavailable commands 12
UNPLUG 1, 15
Using podule RAM 21
Wildcards 23



Computer Concepts Ltd.
Gaddesden Place
Hemel Hempstead
Hertfordshire
HP2 6EX

ROM/RAM Podule design and RFS software by Chris Honey
MakeROM by Dave De Vorchik
Manual by Rob Pickering

