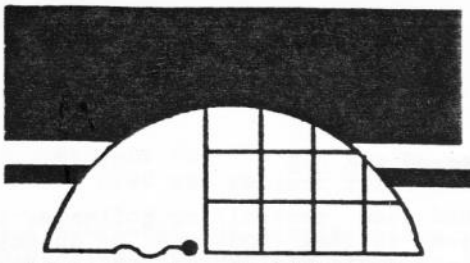


control systems

MULTI-ROM

MULTI-ROM INSTALLATION AND USE

MULTI-ROM contains 16K of RAM and some controlling software. It may be used to simulate ROM cartridges so that ROMS may be quickly interchanged between Disk/Microdrive and MULTI-ROM. MULTI-ROM is ideal for developing ROM based software as code may be quickly downloaded for testing without blowing Eproms.

> PLEASE NOTE: It is quite possible for MULTI-ROM to load and use Pirate copies of ROM images. The Manufacturers cannot, in any way, support such actions. It is in every QL enthusiasts interest that new software is continually developed. This will not happen if pirate copies are circulated.

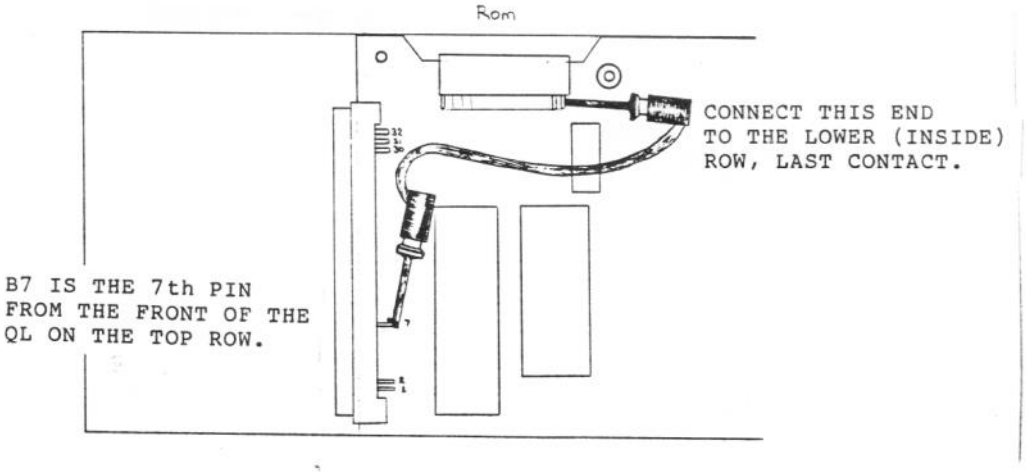
MULTI-ROM is quickly and easily installed and may be removed at any time to allow standard ROM cartridges to be used. Installation is as follows and includes a small modification to the QL's internals which is easily fitted with the aid of a cross point screwdriver.

1) Lay the QL onto its keyboard and remove the eight outermost screws. One screw may be only accessible by removing the warranty sticker.

2) Holding the QL together, turn it the right way up and lay it on its feet facing the normal direction.

3) Lift the lid of the QL sufficiently to clear the base and move it towards the rear of the QL until the ROM connector is exposed.

4) Take the supplied wire and connect the two points shown on the diagram. One end goes to B7 on the Expansion connector and the other to the lower connection at the right hand end of the ROM socket.



5) Replace the lid of the QL and holding the two halves together, turn it over and replace all eight screws.

6) MULTI-ROM now simply plugs into the ROM port, but before doing this it is necessary to take the images of required ROMS.

MAKING A ROM IMAGE

This is a quite simple process using the SBYTES command.

1) Turn off the QL. Fit the ROM cartridge into the ROM socket. Re-power the QL.

2) With the correct disk or microdrive loaded, simply type.

```
SBYTES filename,49152,16384
```

filename is any valid device_filename and is the destination for the image.
49152 is the start of the ROM socket in the QL's memory map.
16384 is the maximum size of ROM.

The above is valid for any ROM and will save it in exactly the same format as it is run in.

Now insert MULTI-ROM in the ROM socket and power up the QL. The following message will be displayed at the top of the screen.

```
Micro Control Systems present
QL Multi-Rom  v1.x  (c)M.Snape 1987
```

To use MULTI-ROM the following Superbasic extensions have been supplied.

****

MULTI-ROM BASIC EXTENSIONS

LOAD_ROM filespec

With this procedure a ROM image is loaded from a device with the filename specified into the MULTI-ROM's RAM. The ROM image may be any length up the 16384 (16K) bytes. When the ROM is copied to the RAM, if it is different to the RAM's previous contents then the QL is WARMSTARTed (see seperate description of WARMSTART). This is necessary because if the previous image was linked into the system and then removed the QL would crash. Note that as in WARMSTART, any program in basic will be lost and any ramdisk contents destroyed.

To load a ROM image, type,

LOAD_ROM filename <enter>

As stated previously if the contents of the MULTI-ROM's RAM is different to the new image being loaded then the QL is warmstarted. If the contents are the same then a return is made to SUPERBASIC. There is an important reason for this, consider the following;

A user wishes to have one or two disks on which a particular program will be run. Before every session with these disks a certain ROM must be loaded into the MULTI-ROM. As an example a boot file is listed that loads the PASCAL ROM.

```
100 LOAD_ROM flp1_pascal
110 MODE 4
120 PRINT "ROM loaded"

etc.
```

First time the program is run it will only get as far as line 100 then will warmstart - since it has just loaded the new image. If this were to happen everytime then we would be in an infinate loop. What actually happens is that MULTI-ROM sees that we are loading the same ROM as is already there and proceeds on to line 110.

RUN_ROM filespec

This procedure will take the same ROM images as for LOAD_ROM but will load them into the QL's Ram instead then boot them from there. This is extremely useful where a toolkit and a language need to be run together. For example:

A boot file could contain;

```
100 LOAD_ROM flp1_qjump_toolkit
110 RUN_ROM  flp1_qflash
120 RUN_ROM  flp1_ProPascal
```

This will load the toolkit into MULTI-ROM and reboot. Second time around it will load into the QL's RAM Qflash ramdisk and ProPascal. These will be initialised and print their signon messages in channel 0. Any number of ROM's may be run internally, the only limitation is the 16K each one takes.

Note that some, position dependent ROMS will not run in RAM as they expect to run at address $C000. Also some ROMS may check that they are actually in ROM by writing to themselves. Known ROMS that will not run in QL internal RAM are;

Eidersoft ICE
MetaComCo C

These ROM's will of course still work in the MULTI-ROM.

WARMSTART

As its name implies this procedure will re-start the QL. It is called a warm start because the QL's memory test is completely bypassed and thus takes only a couple of seconds to re-boot. Warmstart automatically takes into account exactly how much memory is fitted. Note that WARMSTART is as destructive as reset - all memory contents are cleared including any ramdisk - so treat with respect.

WARM-KEY

In addition the combination of keys CTRL-ALT-7 are programmed as a 'Warm-key' and when pressed together perform the same function as the WARMSTART command. This means that the QL may be restarted at any time and in any program (makes the reset button redundant!).

WARM128

This procedure is the same as warmstart except that it fools the QL into thinking that there is only 128K of RAM fitted. This is useful for programs that will not run in expanded machines.

RESET

This procedure completely resets the QL and is identical to pressing the reset button at the end of the QL. The warnings for WARMSTART of course apply.

****

DEFAULTS DRIVER

Included in the MULTI-ROM software is a utility to provide a defaults device driver. The default specified will be added to any filename that does not start with one of the recognised device names - a list of which is available via DEVLIST. It will work in basic with nearly all the keywords and in the majority of programs.

The only cases in basic that it will not support are DELETE and FORMAT. The latter is not too much of a problem as it is infrequently used and any ambiguity as to the name of the device being formatted could be fatal. Delete is a problem however and the keyword DEF$ is supplied to help here.

As supplied the default is FLP1_ therefore typing ;

LOAD boot will fetch the boot file from FLP1_

COPY boot to MDV1_BOOT will copy the same file to MDV.

Defaults work with: COPY, DIR, LOAD, SAVE, LRUN, EXEC, SEXEC, SBYTES, LBYTES and OPEN, also the MULTI-ROM commands LOAD_ROM and RUN_ROM.

DEVLIST [channel]

This procedure will list all the devices recognised by Multi-Rom to the specified channel. If no channel is specified then it will default to #1. If you add a new device to the QL and its name does not appear in the list then it must be added with the ADD_NAME procedure.

ADD_NAME newname

This procedure will add a new recognised device to the list of devices. Up to four new names may be added. The newname must be 3 characters long and may or may not be in quotes.

DEFAULT defname

This procedure allows the default device name to be changed. If the default is currently MDV1_ then entering; DEFAULT FLP1_ will change the default to FLP1_. The default may be up to 16 characters in length.
Where the defaults are an inconvienience (conflicts with some other toolkit etc) then they may be defeated by specifying a null length default ie;

```
DEFAULT ""
```

DEFDEV$

This function returns the current default setting. It is useful to know this on several occasions;

(i) Where a program requires to alter the default for the duration of the program then return the original value.
Consider the following program;

```
100 old_def$=DEFDEV$:REMark read current default
110 DEFAULT=MDV1_   :REMark change default
 .
main prog here
 .
500 DEFAULT=old_def$  :REMark put back original default
```

(ii) Where a file is to be deleted from within a program. The keyword INSTR can be used to see if the file string includes the default ie;

```
100 INPUT 'Enter the file to delete ';file$
110 IF DEFDEV$ INSTR file$
120   DELETE file$
130   ELSE DELETE DEFDEV$&file$
140 ENDIF
```

Here it is used so that if file$ doesn't include the default then it is added before the DELETE.

(iii) To inform the user of the default, ie;

```
100 PRINT 'Enter file name (default is ';DEFDEV$;')';
110 INPUT file$
```

EXIST (filespec)

This function may be used to test if the specified file exists so that the user can be warned of such without crashing the basic program. If the line; ans=EXIST(boot) is entered then ans will be returned as;

0 = file 'boot' not found.
1 = file 'boot' found.
2 = file 'boot' found but in use.

Using 0 and 1 means that true/false tests can be carried out, ie;

```
170 IF NOT EXIST(source$)     instead of

170 IF EXIST(source$)=0       although either could be used.
```

* note that this function is not restricted to files. It may also be used to test if the serial ports are free etc.

****

FOR TECHNICAL SUPPORT ON THIS
PRODUCT CONTACT:

MICRO CONTROL SYSTEMS,
ELECTRON HOUSE,
BRIDGE STREET,
SANDIACRE,
NOTTINGHAMSHIRE. NG10 5BA.

TELEPHONE 0602 391204.