

Q-emuLator for Windows

Version 3.2 - November 2017

Users' Manual

Edited by Daniele Terdina

and Phoebus Dokos

Q-emuLator is © 2017 by Daniele Terdina
The Sinclair block logo, SuperBASIC and QL are © 1983 - 1986 Amstrad plc
QDOS is © 1983 - 1986 Amstrad plc except North America where it is © Paul Holmgren - Frank Davis
SMSQ/e, The Pointer Environment and Toolkit II are © 1983-2007 by Tony Tebby
Minerva is © 1985-2000 by Q-View/Laurence Reeves
Microsoft® Windows™ is © 1981 - 2017 Microsoft Corporation
All other brands mentioned within this manual are © of their respective owners
This manual is © 2017 by Daniele Terdina

Table Of Contents

1. Getting Started	4
1.1 Introduction	4
1.2 Requirements	4
1.3 Installation	4
1.4 Running the emulator	4
1.5 Overview	5
2. Emulator Features	6
2.1 Main Features	6
2.2 "Expanded QL" mode Features	6
3. Configuring Q-emuLator	7
3.1 The QL Configuration window	7
3.2 Speed menu	8
3.3 Memory menu	8
3.4 Configuration files	9
3.5 Q-emuLator preferences	9
4. File Handling	10
4.1 QDOS files	10
4.2 Virtual microdrive slots and access to Windows files	10
4.3 QDOS formatted floppy disks	10
4.4 Format QL floppy disks	10
4.5 Access to QXL.WIN files	11
4.6 Microdrive images	11
4.7 Floppy disk images	12
4.8 Mounting ZIP files as drives	12
4.9 QLPAK archives	12
5. Supported Devices	13
5.1 Keyboard	13
5.2 Video and full screen modes	14
5.3 Screen magnification filter	14
5.4 Screen hardware flashing	14
5.5 Ram disk	15
5.6 Sound	15
5.7 Mouse support	16
5.8 Joystick	16
5.9 Real time clock	16
5.10 Serial ports	17
5.11 Parallel ports	18
5.12 TCP/IP	18
6 Printer emulation	19
6.1 Virtual printer	19
6.2 Printer emulation window	19
7 Advanced topics	20
7.1 Copy-protected microdrives	20
7.2 Using Q-emuLator files on the QL	20
7.3 Using different QL ROMs	20
7.4 Creating a ROM image on a QL	21
7.5 Using the Tooklit II ROM	21

7.6 Using other 16KB QL expansion ROMs	21
7.7 Running SMSQ/e for the Gold Card.	21
7.8 Hardware add-on API.	22
Appendix I - Format of .QCF files	23
Appendix II - Format of QL files stored in the PC file system	26
Appendix III - Included ROMs and copyrights	27
Appendix IV - QL ROMs supported by Q-emulator	28
Appendix V - Unizip code license	29
Appendix VI - Debug_68k Users' manual	30
1. Installation	30
2. The debug_68k window	30
3. The disassembly window	31
4. The register display	31
5. The stack display	31
6. Debugger commands	32
7. Stepping through code	32
8. QL specific commands	32
9. Using debug_68k.dll in emulators	32
Appendix VII - Contact information	33
Q-emulator support site	33
Q-emulator blog	33
E-mail address for feedback	33

1. Getting Started

1.1 Introduction

Q-emuLator is a software emulator of the Sinclair QL home computer. The QL was first released in Europe by Sinclair in 1984.

Originally written in 1994/95 for the Mac OS, Q-emuLator was ported to Windows three years later, and is still being improved. A version for Mac OS X is also available.

Q-emuLator emulates the QL processor (the Motorola MC68008) and redirects the QL I/O (video, keyboard, mouse, mass storage, sound, joystick, parallel and serial ports) to the PC hardware. Putting a QL ROM in this virtual environment makes it possible to run QDOS without its noticing that it's not running in a QL black case.

Q-emuLator runs like any other applications in the Windows environment, without taking over all the machine. You can switch to other active applications, and Q-emuLator can also run (more slowly) in the background.

Q-emuLator (Expanded registration) includes a RAM disk and a TCP/IP driver so that you can use QDOS-compatible Internet applications. Partial Gold Card emulation and Aurora and Q60 video card emulation allow running the SMSQ/E operating system (available as a separate download).

1.2 Requirements

A 486 or better system with Windows 95, Windows 98, Windows ME, Windows NT 4, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10 or later.

Sound emulation and full screen graphics are available if DirectX is installed on your system. (All Windows installations already include DirectX except for the original release of Windows 95).

Q60 and Aurora display emulation are only available on Windows XP or later. Emulation of the standard QL display is of course available on all versions of Windows.

Formatting of QL disks is available only when running on Windows NT 4 or later (including Windows 10, Windows 8, Windows 7, Windows Vista, Windows XP and Windows 2000).

1.3 Installation

To install Q-emuLator, run the `qemuLator.msi` installer that you downloaded from the Q-emuLator web site. Installation requires about 4 MB of free disk space.

If you wish to uninstall Q-emuLator, open the 'Add or Remove Programs' Windows control panel, select 'Q-emuLator' from the list of installed programs and click the 'Remove' button.

To purchase a registration code, follow the link from the Q-emuLator's web site.

1.4 Running the emulator

After installing Q-emuLator, you can start it by using the shortcut in the Windows Start menu.

Before starting the emulated QL, you may want to click on one of the microdrive slots (the gray rectangles in the lower part of the Q-emuLator window) to attach it to a location containing some QL software, like a QDOS floppy disk, a windows directory or a ZIP file. For example, attach it to the 'QL Software\QL demo' subdirectory found in the Q-emuLator installation folder.

Start emulation by selecting 'Start' from the 'QL' menu, or simply by clicking on the picture of the QL keyboard in the main window.

If the first microdrive slot is attached to a mass storage location containing a SuperBASIC program named 'boot', that program will automatically start after pressing F1 or F2 at the initial QL prompt.

1.5 Overview

The program main window has two parts: the upper part shows the QL display, and the lower part the virtual microdrives (Figure 1).



Figure 1 - The Q-emuLator main window (QL emulation stopped)

You can attach a Windows directory, a floppy disk drive or a QXL.WIN file to a microdrive slot by clicking on it (in the microdrive slot gray rectangle) and selecting a command from a pop-up menu (Figure 2). You can also attach ZIP files, enabling you to read their contents as if they were read-only disks. Finally, you can access microdrive and floppy disk physical images.

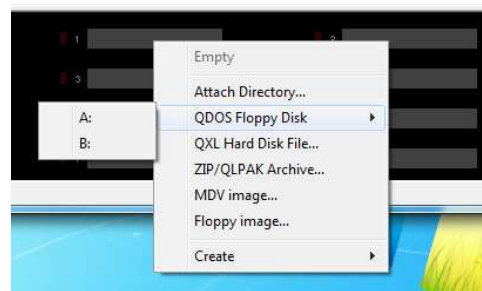


Figure 2 - Slot assignment pop-up menu

Regardless of the type of mass storage attached to a particular slot, you can access it with any of the MDV, FLP or WIN devices names. In other words, all of MDV1_, FLP1_ and WIN1_ refer to slot number 1. This way, you can for example copy your microdrive programs to a Windows directory on your PC hard disk, and it will automatically work despite the program trying to access its files by using MDV1_ instead of WIN1_ (WIN is the device name traditionally used to access the hard disk in QDOS systems expanded with a hard disk interface).

To transfer files from your QL system to the PC you can use QDOS formatted floppy disks and read them from Q-emuLator. If your PC has problems reading your old QL floppy disks and you still have

your QL, try using Q-emuLator to format an empty floppy disk, then copy your old floppy disks to the new one on your QL.

To switch between window mode and full screen, press F12.

2. Emulator Features

2.1 Main Features

- Original QL with up to 384KB of RAM
- Supports all Sinclair and Minerva ROMs as well as many experimental ROM editions
- Up to two 16KB expansion ROMs
- Up to 8 virtual microdrives, each mapped to a Windows directory of QDOS floppy disk
- Access to QDOS formatted floppy disks
- Format QL floppy disks (Windows XP/2000/NT only)
- Read QLPAK QL software packages
- Access microdrive images
- Original QL screen (mode 4, mode 8) and second screen (used by some games and by the Minerva ROM)
- Display hardware flashing in 8 colour mode (Windows 7 only)
- Windowed or Full Screen modes of operation
- Compatible with all PC keyboard layouts
- Sound (BEEP) emulation
- Joystick (CTRL port) emulation
- Serial ports
- Ability to store QL and Q-emuLator settings for easy retrieval and fast changes of emulation mode
- Easily editable QL and Q-emuLator settings through a user-friendly interface
- Ability to paste text from the Windows clipboard
- Speed comparable to that of a real QL.

2.2 “Expanded QL” mode Features

Registering Q-emuLator activates more features:

- Full emulation speed (exact speed depends on your PC speed), many times faster than a real QL
- Up to 16MB of QL RAM
- Support for subdirectories (“Level 2” file system)
- Map microdrive slots to QXL.WIN files
- Mount ZIP files as read-only drives
- Access floppy disk images
- Compatibility with recent versions of SMSQ/e for the Gold/Super Gold Cards¹
- Built-in ram disk
- Mouse support (when the Pointer Environment or SMSQ/E is installed)
- Printer emulation
- Parallel ports
- Acceleration of some QDOS graphics routines
- Fast screen mode for legacy systems (Shift+F12 - only available on Windows XP or earlier)

¹ So far, SMSQ/e versions from 2.70 to 3.13 (**with or without** colour drivers) have been tested.

3. Configuring Q-emuLator

3.1 The QL Configuration window

The term *configuration* denotes a set of Q-emuLator's preferences and of emulated QL's settings. Only one configuration can be active at any time. You can change some configuration settings through menus. For example you can change the QL's RAM amount through the 'Memory' menu (Figure 3). All the other settings that can't be changed through menus are editable in the 'QL Configuration' window.

To open the QL Configuration window, choose the 'QL Configuration' command from the 'QL' menu (Fig. 4).

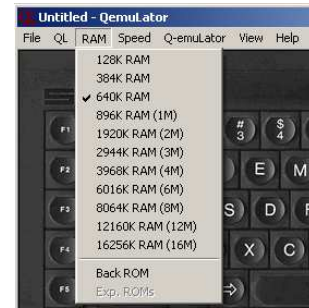


Figure 3 - Memory menu

The QL Configuration window has five pages (tabs), named 'ROMs' (Fig. 5), 'Devices' (Fig. 6), 'Serial ports' (Fig. 7), 'Graphics' (Fig. 8) and 'Start-up' (Fig. 9). You access each page by clicking on its label in the top part of the window. Some of the options in the window may be disabled if you have not registered.

The 'ROMs' page allows setting the names and locations of the QL ROM images to be used by the emulator. File paths can be absolute or relative. In the latter case they are relative to the directory containing the Q-emuLator application.

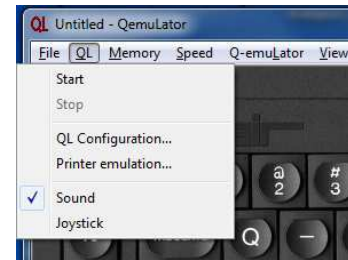


Figure 4 - QL menu

The 'Devices' page allows to enable or disable devices not present in the original QL (for example, the ram disk) and to change their names. Device names can be 1 to 9 characters long.

You can also enable or disable file system support for QL subdirectories (called "level 2"). When "QDOS MDV image driver" is checked, the original Sinclair driver is used to access microdrive images instead of the Q-emuLator driver. This results in increased compatibility, but slower access time.

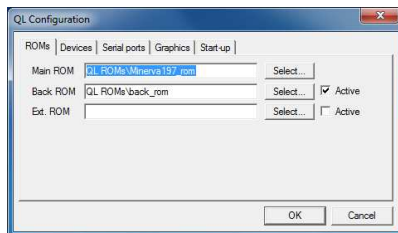


Figure 5 - ROMs

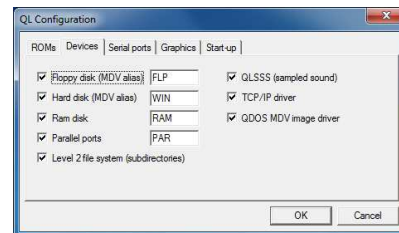


Figure 6 - Devices

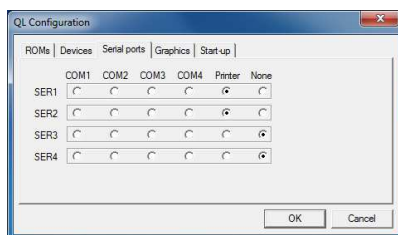


Figure 7 - Serial Ports

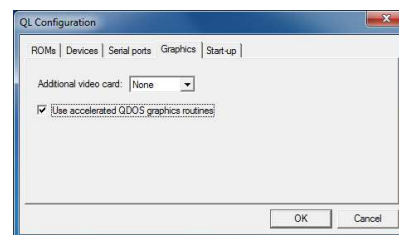


Figure 8 - Graphics

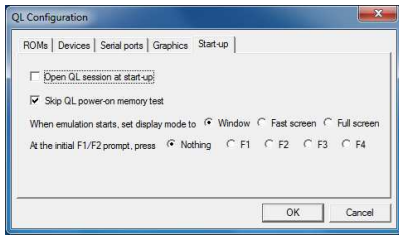


Figure 9 - Startup

BLOCK, MODE). Enable graphics acceleration for maximum speed.

In the 'Graphics' page you can also enable emulation of the Aurora or Q60 video cards.

The 'Start-up' page allows to specify an optional key to be automatically pressed at the initial QL F1/F2 prompt, which display mode to use when starting emulation and to enable or disable two more Q-emuLator settings:

- If the 'Skip QL power-on memory test' option is checked, the QL will not execute the memory test at start-up. The memory test is the first thing that you see when you power on or reset a real QL and is responsible for the coloured dots that fill the screen before the F1/F2 prompt. Checking this option skips this test, so you go directly to the F1/F2 prompt. As a result, the QL's start-up is quite faster, especially when using large amounts of QL memory.
- The 'Open QL session at start-up' option instructs Q-emuLator to immediately start a QL emulation session when starting or when loading the configuration.

3.2 Speed menu

Q-emuLator's emulation speed is determined by the 'Speed' menu (Fig. 10). You can choose between "Full Speed", "Original QL" and "Gold Card".



Figure 10 - Speed Menu

Only "Original QL" speed is available in unregistered copies.

When running at "Original QL" speed, the emulator simulates the precise timing of each instruction to allow games and animations to run at the same speed as on a QL.

The main exceptions are that file operations and reading SuperBASIC programs are much faster. This avoids having to wait minutes to load complex SuperBASIC programs like on a real QL, but a side effect is that sometimes you may not see programs' splash screens for long (QL software would often show splash screens only for the duration of loading the main program from microdrive).

3.3 Memory menu

The 'Memory' menu allows setting the emulated QL's RAM amount and to enable or disable the add-on ROMs.

Note that some QL software may only work with specific amounts of RAM. Early software often fails unless the RAM is set to 128 KB as it expects data and programs to be at fixed locations in memory.

If you enable Aurora or Q60 video emulation, you can only use up to 4 MB or 8 MB of RAM because the rest of the memory address space is used by the video display memory.

3.4 Configuration files

Configuration files store the emulated QL's configuration and some other Q-emuLator settings.

Configuration files have a ".QCF" suffix (file extension).

To start Q-emuLator with the settings specified in a particular configuration file, double-click the icon of the .QCF file, or drag it onto the Q-emuLator.exe icon. If you start Q-emuLator without opening any configuration file, the program uses a default configuration. If you want to change the default configuration, you can do this by saving the desired settings into a configuration file named "default.qcf" and placing it in Q-emuLator's installation directory.

You load and save configuration files through commands in the Q-emuLator's 'File' menu (Fig. 11).

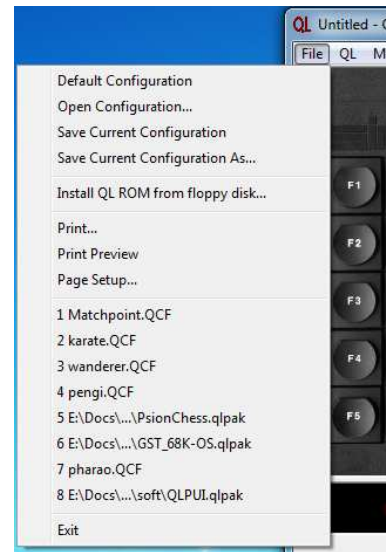


Figure 11 - File Menu

3.5 Q-emuLator preferences

Preferences are special settings that are global to all emulation sessions. They are automatically saved as program preferences rather than stored in ".QCF" files.

The following settings are saved as preferences:

- QL display magnification factor (windowed mode)
- Magnification filter on/off (can be changed through the 'View' menu)
- Printer emulation settings (see chapter 6)

4. File Handling

4.1 QDOS files

QL executable files (as well as some other kinds of files) are stored on the PC in a special format containing QDOS-specific information.

Most non-executable files do not need to store extra QDOS information; therefore they have the same format both in the QDOS and in the Windows environment.

Files are transparently converted between the different formats by the emulator. The user and QL software just use them like standard QL files.

The QL file system limits the maximum file name length to 36 characters.

4.2 Virtual microdrive slots and access to Windows files

When you click on a slot, you can choose to attach it to a Windows directory or to a floppy disk drive with a QDOS-formatted disk in it, or you can empty the slot if it was already filled.

If you registered your copy of Q-emuLator, you can also attach virtual microdrive slots to QXL.WIN files, ZIP files and microdrive or floppy disk images.

Note that you can access each of the eight virtual MDV slots with all of the MDV, FLP and WIN device names, regardless of what type of medium is attached to a slot.

4.3 QDOS formatted floppy disks

Before changing a QDOS floppy disk in the drive, or removing it from the drive, it is recommended that the corresponding virtual slot is emptied. This ensures consistency of the data written to the disk. It's not necessary to empty the floppy disk slots before closing the emulator.

Note: FLP1 and FLP2 refer to slot 1 and slot 2, not to drive A: and B:. Of course, if you like you can attach drive A: to slot 1 and drive B: to slot 2, but you are not required to do so.

4.4 Format QL floppy disks

(not available on Windows 95/98/ME)

The **FORMAT** command is available only on Windows NT4, Windows 2000 or Windows XP, Windows Vista, Windows 7 or later.

For example, if you want to format a floppy disk in QL format you could proceed as follows:

1. Start Q-emuLator, and start emulation by clicking on the QL picture and then pressing F1 at the QL F1/F2 prompt;
2. Click on the first microdrive slot and select QDOS Floppy Disk and then A: from the menu;
3. Insert the disk in drive A: and finally enter the SuperBASIC command **FORMAT FLP1_**.

When you **FORMAT** a disk, Q-emuLator automatically detects the disk density and formats the disk with the appropriate number of sectors, resulting in 720 KBytes (1440 sectors) of free space for a DD (double density) disk and 1.44 MBytes (2880 sectors) of free space for a HD (high density) disk.

A window pops up asking you to choose between Full Format and Fast Format modes, or you can cancel the operation. Full Format completely erases the disk, while Fast Format just empties an already formatted disk, but is extremely fast.

IMPORTANT: Use the Fast Format option only to quickly erase all the files from an already formatted disk (either in PC or QL format). If the disk is unformatted and you use Fast Format, you may lose all the data that you subsequently store on the disk. With both the Full and Fast types of format, any data stored on the disk is completely erased and there's no way to recover it after the format operation!

You can use the Fast Format option to quickly erase a formatted PC disk and convert it to the QL disk format.

During a Full Format operation, a window shows its progress and offers the option to abort it by pressing the Cancel button. However, even if you press Cancel at this point, any data previously present on the disk are lost.

4.5 Access to QXL.WIN files

(available only with the "Expanded QL" registration)

In this manual the term 'QXL.WIN file' refers to the Windows/DOS files used by the QXL² (a hardware QL emulator for DOS) to store the emulated QL filesystem. These files are similar to images of SMSQ/e hard disks stored in a single file.

Q-emuLator allows you to attach QXL.WIN files to microdrive slots to access their content: right-click on a microdrive slot and select 'QXL Hard Disk File...' from the menu, then choose a QXL.WIN file to open.

You can read and modify existing QXL.WIN files, but creating new ones is not supported, yet. However, some free tools (wxqt2 and qxltools) are available on the Internet (<http://www.daria.co.uk/>) that will provide you with that capability.

Creating new directories in QXL.WIN files is not currently supported.

4.6 Microdrive images

Microdrive images contain a dump of some or all sectors of a microdrive cartridge. They can also contain other optional information to emulate bad sectors and special microdrive copy-protection information.

You can mount two types of microdrive images:

- Images obtained from a real microdrive by running the mdump_task utility on a QL.
- Virtual microdrives used by the QLay emulator. Mounting QLay images is only available with the "Expanded QL" registration.

To create microdrive images from real microdrives, you need to be able to transfer files between the QL and the PC, usually through floppy disks or a serial link. Copy the mdump_exe utility (found on

² Also used by uQLx - the "sister" emulator of Q-emuLator for UN*X systems-, its derivatives and the QPC II SMSQ/e emulator.

the 'Software' page of the Q-emuLator web site) to your QL and use it to create a microdrive image from the microdrive. Transfer it to the PC and attach it to a Q-emuLator virtual microdrive slot.

4.7 Floppy disk images

(available only with the "Expanded QL" registration)

Floppy disk images contain all the raw sectors from a floppy disk. Third party utilities exist to copy images from/to floppy disks. Supported sizes are 360 KB, 720 KB and 1.4 MB.

You can create a new QDOS-formatted floppy image file by clicking on a microdrive slot and selecting 'Create' and 'Floppy Image...'.

Some programs can copy floppy disks from/to compressed floppy image files, or you may compress floppy images yourself with the ZIP program so they use less disk space. Q-emuLator can directly access compressed floppy images as a read-only drive.

4.8 Mounting ZIP files as drives

(available only with the "Expanded QL" registration)

ZIP files can be mounted in the virtual microdrive slots to allow accessing their contents as if they were a read-only drive.

This is especially useful because QDOS ZIP files often contain special QDOS file information that would be lost if you unzipped the file with the Windows version of ZIP/UNZIP, but that are preserved by Q-emuLator. (When the special information is lost, the most common symptom is receiving a 'bad parameter' error when trying to EXEC or EXEC_W a program.)

Mounting ZIP files often allows executing the contained QL software directly without having to expand the archive first.

To expand a ZIP archive, rather than using the QDOS version of the UNZIP program, you can simply mount the ZIP file and copy its contents from there to a different microdrive slot associated with a Windows directory (for example by using the WCOPY command).

4.9 QLPAK archives

Files with extension QLPAK contain a piece of QL software together with the optimal emulation settings to run it.

You can start a QLPAK by double clicking on it or by attaching it to a QL slot. If a web site links to a QLPAK, just clicking the link should download it and start Q-emuLator automatically.

Once loaded, the QLPAK can override some or all of the configuration settings and usually makes its contents available as a read-only drive attached to one or more microdrive slots.

5. Supported Devices

5.1 Keyboard

Q-emuLator is compatible with all types of PC keyboards and with most keyboard layouts. Windows key combinations known as “dead keys” are also supported.

The keyboard type-ahead buffer has been expanded from the QL's 7 characters to over 1000 characters.

Some PC keys not available originally on the QL are used as a shortcut for common combinations of QL keys (Table 1):



































PC (Q-emuLator) Key	QL Key combination	Notes
	 	-
	 	-
 ... 	  ...  	-
	 	-
	 	-
		-
		-
	 	Both  and   can be used to freeze/unfreeze the QL display output. The Scroll Lock Led on the keyboard is on when the QL screen is frozen.

Table 1 - Q-emuLator - QL Key equivalents

Three other PC keys have special functions :

-  and  : Toggle between window and full screen modes.
- : Paste up to 1KB of text from the Windows clipboard.
- : Invoke the built-in debugger. (Press  again to exit and continue emulation.)

5.2 Video and full screen modes

Normally the QL display is shown in the main area of the Q-emuLator window.

The window can be resized to enlarge the QL display and to show more or less of the 8 available microdrive slots.

When the emulation is running, you can enter Q-emuLator's full screen mode by pressing the F12 key. To return to window mode, press F12 again.

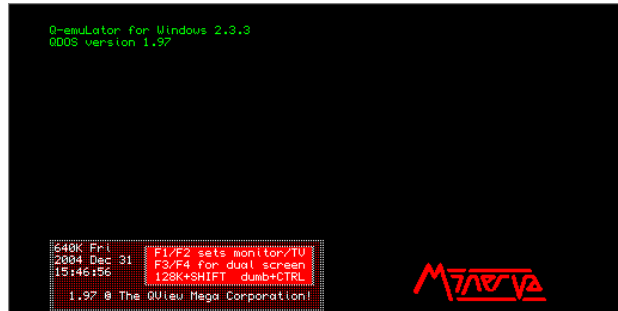


Figure 12 - Full Screen mode Q-emuLator booting Minerva 1.97

In full screen mode, menus and microdrive slots are not visible. To access them, return to window mode first by pressing F12.

If you have the “Expanded QL” registration and a version of Windows up to Windows Vista, press the Shift+F12 key to enter the fast screen mode, where the QL display occupies most of the screen. This is the fastest graphics mode. Press F12 again to return to window mode.

5.3 Screen magnification filter

When the QL screen is enlarged, a special filter adds extra pixels to the image. The goal is to smooth lines and text while keeping the image sharp, and to avoid the jagged look you would get by just duplicating the original pixels.

The filter can be turned on and off from the ‘View’ menu. It only affects the QL standard graphics resolutions and is always disabled for the Aurora and Q60 display modes.

When running on Windows 7 or later, the magnification filter is available both in full screen and window modes. On previous versions of Windows, it's only available in full screen mode. The filter may not be available on computers with very old CPUs.

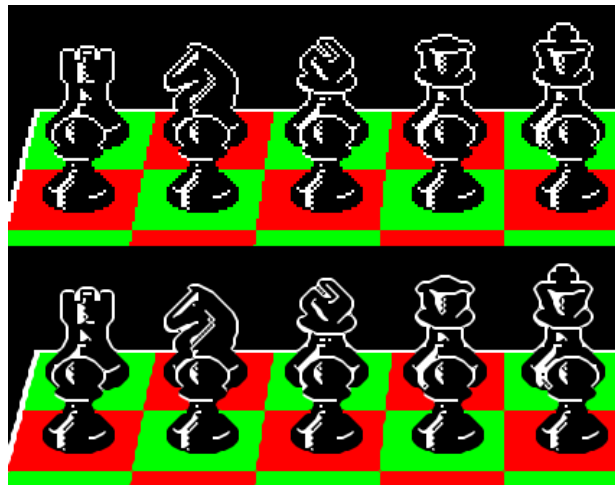


Figure 13 - Part of a QL screenshot enlarged by duplicating pixels (top) and by the Q-emuLator magnification filter (bottom)

5.4 Screen hardware flashing

When running in 8 colours mode, it is possible on a QL to mark pixels so that they appear to flash on and off. The flashing is accomplished by the display hardware. The FLASH command allows to print flashing text from SuperBASIC.

When running on Windows 7 or later, flashing is always emulated.

On previous versions of Windows, flashing is only emulated in full screen mode, and only when the screen magnification filter is disabled.

5.5 Ram disk

(available only with the “Expanded QL” registration)

Q-emuLator provides a built-in ram disk device driver. If you use it instead of loading a QL software ram disk driver (like ‘ramprt’) you will gain in speed.

The device has eight drives, ‘ram1_’ through ‘ram8_’.

You can use the ram disk either in dynamic and static mode. In dynamic mode you can freely copy files to the ram disk and more ram will be allocated for it every time it is needed, up to filling the whole QL’s memory. In static mode you assign a fixed memory amount to each drive, by formatting the drive and specifying the requested number of blocks as the medium name: For example, the command ‘**FORMAT ram1_400**’ assigns a fixed amount of 200KB (as each block is 512 bytes long) to the first ram drive.

To erase a whole ram disk drive at once use the ‘**FORMAT**’ command without specifying a file name: For example ‘**FORMAT ram4_**’.

In static mode all the memory needed by the driver is allocated when you format the drive. This means that you are sure that no other job can take away that memory before the ram driver needs it, but also that the ram driver is not allowed to use more memory than the amount specified.

Dynamic mode is easier to use, while static mode is faster: in dynamic mode the driver has to call QDOS routines every time that it wants to allocate or release a block of memory, whereas in static mode all the memory is allocated at the beginning and QDOS is no longer called when accessing the drive.

The ram disk device driver is not ‘level 2’ (ie. it doesn’t support subdirectories), and there are no SuperBASIC extensions to modify its behaviour (for example there is no **RAM_USE** keyword).

If you prefer to use a different ram disk driver (although it will be slower, as it has to run under emulation), just disable the built-in device driver in the ‘QL Configuration’ window, and install your third party driver by installing an add-on rom containing it or by loading it as an extension from your boot file.

Note: For a ram disk drive used in dynamic mode, the DIR command always reports a full drive, but you can still add more files to it as long you have enough free QL memory.

5.6 Sound

QL ‘**BEEP**’ emulation is enabled by default. You can turn sound off through a switch in the ‘QL’ menu.

There is still a minor problem with sound emulation: a click is sometimes heard if a **BEEP** command is issued while a previous **BEEP** command is still playing.

If you have the “Expanded QL” registration, Q-emuLator also supports playing back sampled sounds by implementing a ‘QLSSS’ (QL Sampled Sound System) driver like the one on the Q40/Q60 computers. The sampling frequency is fixed at 20 kHz. The best way to use it is through a user-friendly driver written by Simon Goodwin that creates a device named SOUND, that plays a sound when it’s copied to it. You can find the driver in the ‘QL Software\QLSSS’ subdirectory of the QemuLator installation directory. The file is named ‘sound4_bin’, read the ‘sound4_text’ document in the same directory for instructions on how to use it.

5.7 Mouse support

(available only with the “Expanded QL” registration)

Mouse emulation is provided under the Pointer Environment through a special mouse driver.

To take advantage of it, install the ‘MacMouse’ mouse driver (included in the Q-emuLator package) JUST AFTER having installed the Pointer Environment (that is, the ‘PTR_GEN’ extension, not included in the Q-emuLator package).

To install the mouse driver use the following SuperBASIC command:

```
a=RESPR(1024):LBYTES mdv1_MacMouseXY,a:CALL a
```

or, if Toolkit II is installed:

```
LRESPR mdv1_MacMouseXY
```

The command assumes that the ‘MacMouseXY’ file (replace X and Y with the version number: for example ‘MacMouse11’ for the mouse driver version 1.1) is in the first Microdrive slot.

Exception: You don’t need to follow these steps to install the mouse driver when running SMSQ/e for the Gold Card. SMSQ/e includes the Pointer Environment in it, and Q-emuLator automatically installs a mouse driver when SMSQ/e is loaded.

Once the mouse driver is installed, any active QL mouse cursor will be moved instead of the Windows pointer when it is over the QL display window (provided that the QL display window is the frontmost window).

Note that as the QL cursor follows the Windows’ one, its speed is independent from any settings in the QL environment. On the PC, the mouse speed can be adjusted in the Mouse Control Panel.

Unlike the left and right mouse buttons, the central button is not emulated.

5.8 Joystick

Select ‘Joystick’ in the ‘QL’ menu to enable or disable joystick emulation. When enabled, the PC joystick will be seen by the QL as connected to the CTL1 port.

5.9 Real time clock

The QL had a real time clock, but you had to set it every time you started the computer, because there was no battery to keep it working while the computer was powered off. On Q-emuLator there’s no need to set the QL real time clock, as the current date and time is read from the PC clock every time the program starts.

Note: Adjusting the QL clock doesn’t affect the PC clock, and the change is limited to the duration of the current emulation session.

5.10 Serial ports

If you have the “Expanded QL” registration, the serial ports are by default redirected to the built-in virtual printer (see chapter 6). You can changed this setting in the ‘QL Configuration’ window and either disable them or link them to the PC COM: ports. The unregistered version of Q-emuLator doesn’t have a virtual printer, but can still use the PC serial ports.

To open a serial port I/O channel you specify the device name and options with the following syntax (note: the options are only used for COM: ports, and ignored by the virtual printer):

SERnpftv

n, *p*, *f*, *t* and *v* represent optional characters that you can add to the name to specify the port number and other parameters. The following table (Table 2) shows the possible values for these parameters:

Parameter	Parameter's meaning	Possible Values	Values' meaning	Default Value
<i>n</i>	Serial Port number	1-4	Port 1 - Port 4	1
<i>p</i>	Parity Checking	E O M S	Even Odd Mark Space	None (no parity checking)
<i>f</i>	Flow Control	H I	Hardware Handshake No Handshake	H
<i>t</i>	CR, EOF translation	R Z F C	No translation Use Ctrl-Z as End-Of-File Use Form Feed as End-Of-File Use Ctrl-Z as End-Of-File and Carriage Return as End-Of-Line	R
<i>v</i>	7 bit mode	V	Transmit/receive 7 bits per byte instead of 8	None (8 bits)

Table 2 - SER device settings

You can omit any of the *n*, *p*, *f*, *t* and/or *v* parameters (the driver assumes default values), but if you specify them you can't invert their order. For example you can write **SERHZ**, but not **SERZH**.

The baud rate can be set to most of the usual QDOS values, plus many PC values not originally supported by the QL, up to 256000 baud, if your serial port supports it. For the higher transmission rates, however, it is recommended to use hardware handshake, as otherwise the emulated QL could be too slow to handle the data without losses. A transmission rate of 31250 baud is also available for MIDI applications.

The **TRA** command, present in JS and later ROM's versions, is not implemented. This command allows to specify a character translation table, but usually communication programs don't use this feature.

The Minerva and SMSQ/e extensions to the serial driver are not currently supported.

5.11 Parallel ports

(available only with the “Expanded QL” registration)

You can access to the LPT1: and LPT2: parallel ports of your PC through the PAR1 and PAR2 QL devices.

The driver supports bi-directional ports.

To open a parallel port I/O channel you specify the device name and options with the following syntax:

PARnt

n and *t* represent optional characters that you can add to the name to specify the port number and other parameters. The following table (Table 3) shows the possible values for these parameters:

Parameter	Parameter's meaning	Possible Values	Values' meaning	Default Value
<i>n</i>	Parallel Port number	1-2	Port 1 -2	1
<i>t</i>	CR, EOF translation	R Z F C	No translation Use Ctrl-Z as End-Of-File Use Form Feed as End-Of-File Use Ctrl-Z as End-Of-File and Carriage Return as End-Of-Line	R

Table 3 - PAR device settings

You can omit the *n* and/or *t* parameter (the driver assumes default values), but if you specify them you can't invert their order. For example you can write PAR2Z, but not PARZ2.

You can disable the QL parallel device or change its name in the '*QL Configuration*' window.

5.12 TCP/IP

(available only with the “Expanded QL” registration)

Q-emuLator implements **SCK_** and **TCP_** device drivers to allow use of the Windows TCP/IP stack. Only a subset of the socket functions are implemented, but there is enough functionality to run the QDOS port of the Lynx Internet browser, FTP, email and other programs.

On Windows XP SP2 and later, you may need to adjust your firewall settings to allow the TCP/IP driver to reach the Internet.

TCP/IP is available by default. You can disable or reenale the QL TCP/IP device driver in the '*QL Configuration*' window.

6 Printer emulation

(available only with the “Expanded QL” registration)

6.1 Virtual printer

QL programs that can output to a printer (for example the Psion suite of business programs) only have drivers that access the printer through a serial port (SER device) and typically only support printer models from the 80's.

PC printers don't connect through the obsolete serial ports anymore, and most current printers don't understand the print commands that QL software would issue.

Q-emuLator provides a solution to print from QL software to any printer you may be able to access from your PC. The printing commands that the QL software sends to the SER port can be redirected to a virtual printer (built-in in the emulator) that interprets these commands and sends the output to your PC printer.

To use this feature, do the following:

- In the QL configuration, set the SER port you want to use to redirect its output to the printer (see Figure 7 - this is enabled by default).
- Your QL software probably has a way to select a type or model of printer. Choose an Epson compatible printer.
- Print from your QL software. A window will pop up to allow you to select the PC printer to use as destination, and optionally change its settings.

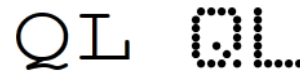


Figure 14 - The word “QL” printed in high quality (left) and ‘dot matrix’ (right) modes

This should result in the QL software being able to print on your PC printer.

6.2 Printer emulation window

Selecting the ‘Printer emulation...’ command from the ‘QL’ menu opens a window (Fig. 15) where you can set two printing preferences:

- The ‘Print quality’ setting switches between high print quality and emulation of the individual dots that would be produced by an early 9-pin dot matrix printer (Fig. 14).
- The ‘Fit page to printable area’ option shrinks the virtual printer page to completely fit inside the printable area of the PC printer. This can result in a smaller print, but at least there should be no pieces missing. If your QL software already provides paper margins, you can try unchecking this option to avoid excessive margins.

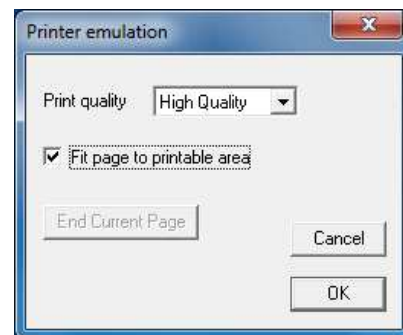


Figure 15 - Printer emulation window

The window also has an ‘End Current Page’ button that is only enabled when the virtual printer buffer contains an incomplete page. Clicking it forces completion of the page and immediately sends it to the printer.

Pending pages are also automatically completed when the serial port is closed or when emulations stops.

7 Advanced topics

7.1 Copy-protected microdrives

The best way to continue to use your copy-protected QL software is to create an image file from the original microdrive (with the “mdump_exe” utility - see paragraph 4.6). The image preserves more information than just copying the files, including most low level details of the microdrive structure that are sometimes used by copy-protection schemes.

If you'd prefer to copy the individual files rather than creating a microdrive image, there is another utility that may work to preserve the copy protection in many cases: The “QL Software\FileUtil” folder contains the “MdvToWin_exe” program that can be used on a QL to copy a file from a copy-protected microdrive to floppy disk and attach to it special microdrive information used by some QL programs to verify the authenticity of the microdrive.

The resulting file should then be copied through the emulator from the floppy disk to a folder in the Windows file system, together with the other files included in the QL software (for the other files, you can just copy them with the COPY or WCOPY commands).

After these steps, Q-emuLator will emulate the original copy-protected microdrive when running this QL program.

7.2 Using Q-emuLator files on the QL

Q-emuLator automatically translates files between QDOS and Q-emuLator for Windows formats when copying files between a Windows directory and QDOS floppy disks or other containers, so the easiest way to move QL software from the PC to the QL is simply to use Q-emuLator to copy it to a QDOS floppy disk.

Should you ever need to translate files on a real QL (rather than from the emulator, where it's automatic) between QDOS and Q-emuLator for Windows formats, you can use the “QLToWin_exe” and “WinToQL_exe” utilities, found in the “Software\FileUtil” folder.

7.3 Using different QL ROMs

The emulator needs a file with a supported QL ROM image to work. Two ROM images are supplied with Q-emuLator, one is a Sinclair version “JS” ROM, the other is a Minerva 1.97 ROM.

Minerva is compatible with the original Sinclair ROMs, but faster and with more features.

You can use other ROM images with Q-emuLator, you can find them on the Internet (follow the link from the ‘Software’ www page of the Q-emuLator web site), or you can transfer your QL's ROM.

To install a new ROM image that is already on your PC (for example a ROM that you downloaded from Internet), you need to point Q-emuLator to the ROM image: go to the ‘QL Configuration’ window (Choose ‘QL Configuration...’ from the ‘QL’ menu), click the uppermost ‘Select...’ button and find the ROM image file in the file selection window that appears. Click ‘OK’ to close the window. Done!

7.4 Creating a ROM image on a QL

To create a ROM image on your QL, use the `"SBYTES FLP1_QL_ROM,0,49152"` command.

Note: This command doesn't work on Gold Card and Super Gold Card systems, as you get a modified ROM image instead of the original one. Therefore, if you have a (Super) Gold Card, you must disconnect it before using the SBYTES SuperBASIC command and copy the ROM to a microdrive, then from there to a floppy after reconnecting the card.

On a Super Gold Card system (but not on a Gold Card), you may also be able to get the unmodified ROM image with the command `"SBYTES FLP1_QL_ROM,hex('400000'),49152"`, without having to disconnect the card.

Once you have the ROM file on a QDOS floppy disk you can read it from Q-emuLator on your PC. Follow these steps:

1. Launch Q-emuLator.
2. Insert the QDOS floppy disk with the ROM file image in the PC drive (A: or B:).
3. Click on the gray rectangle representing the left microdrive slot (below the window with the QL picture); choose *'QDOS Floppy Disk'* from the pop-up menu that appears, and choose either *'A:'* or *'B:'* from the sub-menu, depending on where you inserted the floppy disk. Q-emuLator looks at the disk and displays its name on the microdrive slot.
4. Choose the *'Install QL ROM from disk...'* command from the *'File'* menu. This instructs Q-emuLator to copy the ROM image to the *"QL ROMs"* directory and use it for emulation.
(Warning: this command requires Q-emuLator to reside in a writeable directory and not on CD-ROM or other read-only media).

7.5 Using the Toolkit II ROM

Q-emuLator includes a Toolkit II ROM containing many SuperBASIC extensions that are commonly used by QL software.

To use the Toolkit II ROM, make sure that *'Back ROM'* is *'Active'* in the *'QL Configuration dialog'*, and that the name of the *'Back ROM'* is *'TK2_ROM'*.

7.6 Using other 16KB QL expansion ROMs

A 16KB expansion ROM can also be optionally used by the emulator.

If a 16KB expansion ROM is installed on your QL, you can get an image of it with the `"SBYTES FLP1_MY_ROM,49152,16384"` command. To tell Q-emuLator to use the ROM image, copy it to your hard disk, go to the *'QL Configuration'* window (Fig. 5), press the second *'Select...'* button (the one on the *'Back ROM'* line) and select the ROM image on your hard disk.

7.7 Running SMSQ/e for the Gold Card

(available only with the "Expanded QL" registration)

If you have a Gold Card or Super Gold Card QL expansion and you are running a recent copy of the SMSQ/e operating system on it, you can use your copy of SMSQ/e with Q-emuLator, too. Just configure the emulated QL to start with any Sinclair or Minerva ROM and plenty of RAM, and load and execute the SMSQ_GOLD or GoldCard_bin file as you usually do on your (Super) Gold Card system. (E.g. `LRESPR win1_SMSQ_GOLD`, assuming you have the Toolkit II installed in the emulator and you the SMSQ_GOLD file is in a PC directory associated with win1_).

Q-emuLator now emulates some of the Gold Card's hardware. Not all of it, but just enough to support SMSQ/e. The emulator also detects SMSQ/e being loaded and automatically adds its own device drivers to SMSQ/e, allowing for example access to the Windows file system. A mouse driver is also installed when you load SMSQ/e, allowing you to use the PC mouse right away.

If you have the version of SMSQ/e for the Gold Card that includes the "GD2" color drivers, you can use the Aurora display from SMSQ/e.

Note: Compatibility with SMSQ/e for the Gold Card is a new and still experimental feature. In particular, some problems may be expected with the file system and I/O in general, as the Q-emuLator's device drivers are not 100% compatible with the ones in SMSQ/e. Some preliminary testing has been done using SMSQ/e for the Gold Card versions from 2.97 to 3.13, and Q-emuLator is not expected to run with older versions of SMSQ/e. Please report any problems you find running SMSQ/E on Q-emuLator.

7.8 Hardware add-on API

(available only with the "Expanded QL" registration)

For programmers and developers of QL hardware, Q-emuLator for Windows exposes a programming interface that allows emulation of simple QL hardware add-ons. A programmer can create a DLL that emulates a QL hardware interface and place it in a directory named 'Extensions' for Q-emuLator to use it.

Appendix I

Format of .QCF files

A configuration file is a text file used to store the emulated QL's configuration and some other Q-emuLator's settings.

Inside configuration files, each line of text describes a different preference or setting, identified by a keyword. The format of each line is:

Keyword = Value

You can, for instance, set the QL's ram amount to 640K with the following line in a configuration file:

Ram = 640K

Configuration files have the ".QCF" suffix. You can have multiple configuration files and you can store them anywhere you like.

The following table (Table 4) describes the keywords you can find and use in configuration files. For each keyword the table shows the possible values, their meaning, and the default value. The last column shows how you can alter the setting from Q-emuLator's user interface.

Note: Some of these keywords may be ignored by unregistered copies of Q-emuLator.

Keyword	Value	Meaning	Default	Q-emuLator
RAM	<i>N</i> <i>N K</i> <i>N M</i>	QL ram amount: <i>N</i> bytes <i>N K</i> Kilobytes <i>N M</i> Megabytes	128K	RAM menu
MainRom	<i>name/path</i>	Name/Path of main (48KB or greater, at address \$0) ROM Image	Minerva	QL Config. Window
BackRom	<i>name/path</i>	Name/Path of back (16KB or greater at address \$C000) ROM Image	TK2	QL Config. Window
BackRomActive	Yes/No or On/Off	Enable back 16KB ROM?	Yes	QL Config. Window
ExpRom	<i>name/path</i>	Name/Path of external (16K at address \$C0000) ROM Image	(none)	QL Config. Window
ExpRomActive	Yes/No or On/Off	Enable external 16KB ROM?	No	QL Config. Window
UseFloppyName	Yes/No or On/Off	Access MDV slots via floppy disk alias?	Yes	QL Config. Window
FloppyName	<i>name</i>	Floppy disk alias/device name	FLP	QL Config. Window
UseHardDiskName	Yes/No or On/Off	Access MDV slots via hard disk alias?	Yes	QL Config. Window
HardDiskName	<i>name</i>	Hard Disk alias/device name	WIN	QL Config. Window
HasRamDisk	Yes/No or On/Off	Enable built-in Ram Disk?	Yes	QL Config. Window
RamDiskName	<i>name</i>	Name of Ram Disk Device	RAM	QL Config. Window
HasParPort	Yes/No or On/Off	Enable Parallel Port Device?	Yes	QL Config. Window
ParPortName	<i>name</i>	Name of Parallel Port Device	PAR	QL Config. Window
TCP	Yes/No or On/Off	Enable TCP/IP drivers?	Yes	QL Config. Window

Keyword	Value	Meaning	Default	Q-emuLator
Subdirs	Yes/No or On/Off	Enable Level2 file drivers?	On	QL Config. Window
FastStartup	Yes/No or On/Off	Skip initial QL memory test?	No	QL Config. Window
AutoStartSession	Yes/No or On/Off	Launch emulation session when Q-emuLator starts?	No	QL Config. Window
FirstKey	None F1 F2 F3 F4	No special action on F1/F2 prompt Simulate a F1 keypress at startup Simulate a F2 keypress at startup Simulate a F3 keypress at startup Simulate a F4 keypress at startup	None	QL Config. Window
Sound	Yes/No or On/Off	Sound Emulation State	On	Sound command in the QL menu
Joystick1	Default or None	CTL1 Joystick Emulation State	None	Joystick command in the QL menu
SER1	None Printer COM1 COM2 COM3 COM4	Disable QL serial port 1 (Ser1_) Send output to the virtual printer Assign Ser1_ to PC COM1: Assign Ser1_ to PC COM2: Assign Ser1_ to PC COM3: Assign Ser1_ to PC COM4:	Printer	QL Config. Window
SER2	as above	As above but replace Ser1_ with Ser2_	Printer	As above
SER3	as above	As above but replace Ser1_ with Ser3_	None	As above
SER4	as above	As above but replace Ser1_ with Ser4_	None	As above
SLOT1	Empty DISK_A DISK_B QXL:full_path MDV:full_path FLP:full_path ZIP:full_path PAK:full_path path	The slot (device) is empty QDOS Floppy in Drive A: QDOS Floppy in Drive B: QXL.WIN file at full_path Microdrive image Floppy disk image ZIP archive QLPAK package Windows Filesystem directory identified by path. Path should be an absolute path (ie. c:\ql\)	Empty	Popup menu accessible by clicking on each microdrive slot
SLOT2	as above	As above	Empty	As above
SLOT3	as above	As above	Empty	As above
SLOT4	as above	As above	Empty	As above
SLOT5	as above	As above	Empty	As above
SLOT6	as above	As above	Empty	As above
SLOT7	as above	As above	Empty	As above
SLOT8	as above	As above	Empty	As above
Speed	QL GoldCard Full	Original QL Speed QL with Gold Card Speed Full Emulation Speed	Full	Speed Menu
DisplayMode	Window 640x480 1024x768	Q-emuLator Window Fast Screen Mode Full Screen Mode	Window	QL Config. Window
VideoCard	Aurora Q60	Aurora video card Q60 video emulation	<None>	QL Config. Window
AcceleratedGraphics	Yes/No or On/Off	Enable QDOS Graphics acceleration?	Yes	QL Config. Window
WindowHeight	Height	Height of Q-emuLator's window in pixels	376	Mouse or maximise icon

Table 4 - .QCF file options

- Notes:
- Keywords are not case sensitive and can contain spaces. For example, you can write 'FloppyName', but also 'FLOPPYNAME' or 'floppy name'.
 - Paths can be absolute or relative. In the latter case the path is relative to the working directory or to Q-emuLator's directory

As an example, the following is the content of a configuration file that specifies the default values:

```
Ram=640K
MainRom=QL_ROMs\QL_ROM_JS
BackRom=QL_ROMs\TK2_rom
BackRomActive=Yes
ExpRom=
ExpRomActive=No
UseFloppyName=Yes
FloppyName=Flp
UseHardDiskName=Yes
HardDiskName=Win
HasRamDisk=Yes
RamDiskName=Ram
HasParPort=Yes
ParPortName=Par
TCP=On
Subdirs=On
Speed=Full
FastStartup=Yes
AutoStartSession=No
FirstKey=None
DisplayMode=Window
AcceleratedGraphics=Yes
Sound=On
SampledSound=On
Joystick1=None
SER1=Printer
SER2=Printer
SER3=None
SER4=None
Slot1=Empty
Slot2=Empty
Slot3=Empty
Slot4=Empty
Slot5=Empty
Slot6=Empty
Slot7=Empty
Slot8=Empty
WindowHeight=376
```

You are encouraged to create a personalised default configuration file that Q-emuLator will load each time it starts: customize your settings through the Q-emuLator user interface (most settings are in the QL Configuration window), choose the *Save Configuration* command from the 'File' menu, and save the file in the Q-emuLator's directory, naming it '**default.qcf**'.

Appendix II

Format of QL files stored in the PC file system

QL files have a special piece of information associated with them, called the 'QDOS file header'. The header stores such information as the file name and whether the file is an executable program.

Q-emuLator for Windows stores part of the header at the beginning of files. The header is present only when it is useful, ie. only if it contains non-default information.

The header has the following format:

OFFSET	LENGTH(bytes)	CONTENT
0	18	"!QDOS File Header"
18	1	0 (reserved)
19	1	total length_of_header, in 16 bit words
20	length_of_header*2-20	QDOS INFO

The first 18 bytes are there to detect whether the header is present (ID string).

The headers Q-emuLator supports can be 30 bytes or 44 bytes long (the value of the corresponding byte at offset 19 is either 15 or 22). In the first case, there are 10 bytes with the values present in bytes 4 to 13 of the 64 bytes QDOS header. In the second case the same piece of information is followed by 14 bytes containing a microdrive sector header, useful for emulating microdrive protection schemes. Additional header information (file length, name, dates) is obtained directly from the file through the host file system.

Some QL programs to translate between QDOS and Q-emuLator for Windows file formats are included in the Q-emuLator package.

The translation is transparently performed when you move files between QDOS floppy disks and Windows directories through Q-emuLator.

There is no need for you to use the provided conversion utilities, except when you want to copy files from a copy-protected microdrive to a floppy disk.

Note: While the OS X version of Q-emuLator uses exactly the same file format, the old version for Mac OS uses a different scheme: the QDOS information is stored in the file's resource fork. (On Mac OS all files have both a data fork - corresponding to a Windows sequential file - and a resource fork, containing structured information called 'resources'). The latest version of Q-emuLator for Classic Mac OS can read files created by Q-emuLator for Windows, but not vice versa.

Appendix III

Included ROMs and copyrights

Please note that although copyright holders allow usage of the included ROMs, they retain the copyright: These ROMs are not in the public domain! (With the possible exception of Minerva versions 1.89 and earlier.)

QL_ROM_JS is a Sinclair ROM version "JS". It's used by default unless a different ROM is installed in the '*QL Configuration*' window. The "JS" was the last version of QDOS released by Sinclair for the UK market.

Outside of North America, Sinclair ROMs are © 1983-1986 Amstrad Plc. Amstrad has kindly given permission to redistribute Sinclair ROMs (free of charge) and to include them with emulators, provided that the ROM's copyright notice is not altered.

In North America, Sinclair QL ROMs are © 1983-1986 Frank Davis and Paul Holmgren. Frank and Paul have kindly given permission to include Sinclair QL ROMs with Q-emuLator, on condition that users may not distribute these ROMs without first asking and obtaining explicit permission from them.

If you reside in North America, you are NOT allowed to distribute the Sinclair ROMs included in the Q-emuLator package without explicit permission from Frank Davis and Paul Holmgren.

MINERVA_197 is a Minerva ROM version 1.97.

Minerva ROMs are public domain. Tony Firshman kindly made version 1.89 and earlier public domain in 1998, and in 2003 Laurence Reeves (who wrote Minerva) made all versions public and released the latest sources under the GNU license. (Download Minerva sources from <http://bergbland.info>).

TK2_ROM is a Toolkit II ROM version 2.10.

Toolkit II is © 1985 by Tony Tebby. Tony Tebby generously changed its distribution from commercial to freeware in January 2005 (when he also changed to freeware the HOT_REXT, PTR_GEN and WMAN Pointer Environment extensions).

Appendix IV

QL ROMs supported by Q-emuLator

The following table (Table 5) shows the current compatibility status of Q-emuLator with available QL-compatible ROM images.

ROM Name	QDOS Version	SuperBasic Version	Author	Compatible	Size	Notes
AH	1.02	AH	Sinclair	●	48KB	
TB	1.03	TB	Sinclair	●	48KB	Reduced keyboard functionality
JM	1.03	JM	Sinclair	●	48KB	
JS	1.10	JS	Sinclair	●	48KB	
JS-4M	1.10	JS	?	●	48KB	Better to use the regular JS instead
JSU	1U10	JSU	Sinclair	●	48KB	
MG	1.13	MG	Sinclair	●	64KB	
MGUK	1Σ13	MGUK	John Alexander	●	48KB	
MF	1.14	MF	?	●	48KB	RAM must be set to 1920KB or less
MGx	1x14	MGx	Sinclair	●	48KB	MGE, MGG, MGR, MGI, etc.
Minerva	1.6x to 1.98	JSL1	Q-View	●	48KB	
M89-T6	1.89	JSL1	H.P. Reckenwalt	●	80KB	
Tyche	2.05	TY05	?	●	64KB	RAM must be set to 1920KB or less
68K /OS	2.x	n/a	GST	●	32KB	File system not emulated

Table 5 - QL ROMS supported

Appendix V

Unzip code license

Q-emuLator contains altered parts of the Info-ZIP code. Those parts of the code have the following copyright and license:

*"This is version 2005-Feb-10 of the Info-ZIP copyright and license.
The definitive version of this document should be available at
<ftp://ftp.info-zip.org/pub/infozip/license.html> indefinitely.*

Copyright (c) 1990-2005 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals: Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ed Gordon, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Steven M. Schweda, Christian Spieler, Cosmin Truta, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software [the original Info-ZIP code] for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

- 1. [Does not apply] Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.*
- 2. [Does not apply] Redistributions in binary form (compiled executables) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled.*
- 3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).*
- 4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases."*

Appendix VI

Debug_68k Users' manual

Debug_68k.dll is a component that can be used by emulators to add a debugger for an emulated Motorola 68000 CPU.

1. Installation

Put the debug_68k.dll in the directory containing the emulator executable file. This is automatically done by Q-emuLator's installer during a 'Full' installation.

2. The debug_68k window

When you invoke the debugger (press F11 in Q-emuLator when emulation is running), or when execution hits a breakpoint, the debug_68k window pops up (Figure 16).

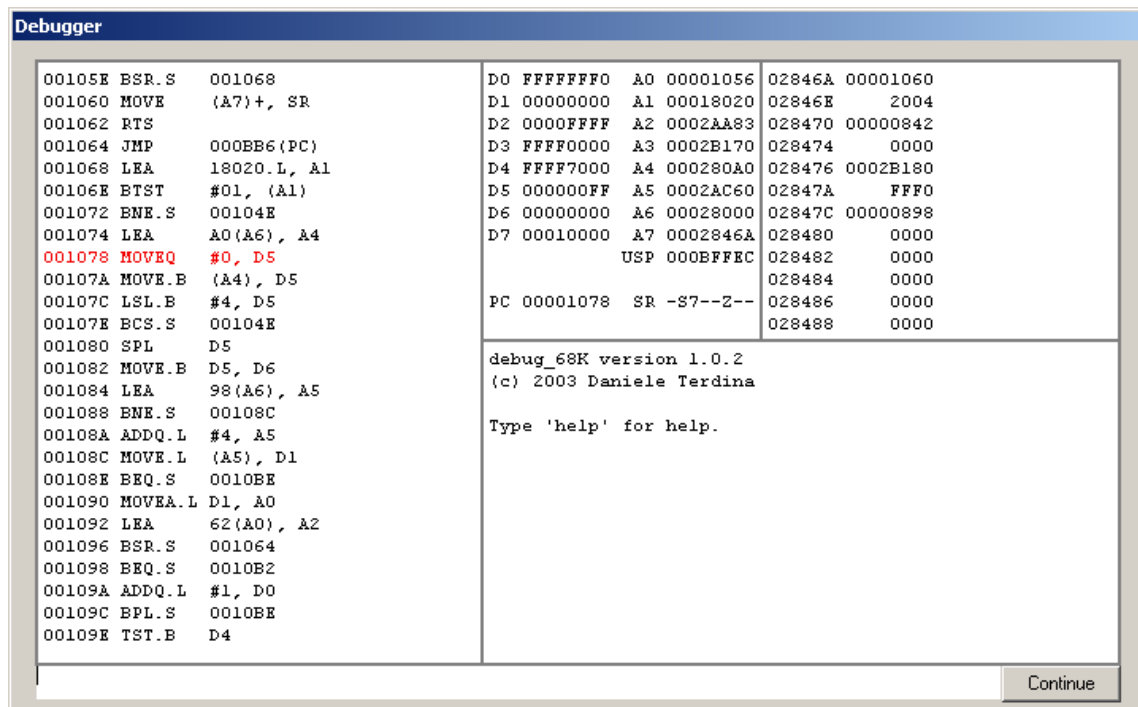


Figure 16 - Debug_68K Window

At the bottom of the window there is a text input box (it looks like an horizontal white bar), with a "Continue" button to its right. Pressing the button closes the window and resumes execution.

The upper part of the window is divided in four parts. From left to right they are the disassembly area, the register display and the stack display, and the output area which is under the register and stack displays.

When you digit a debugger command on your keyboard, it appears in the input area, and it gets executed when you press the **Enter** key. When the command gets executed, the command and possibly its results are displayed in the output area.

All numeric values displayed in the debug_68k window are always hexadecimal.

Addresses are displayed with six hexadecimal digits, as this covers the 16MB address space of the 68000 processor.

3. The disassembly window

This window shows the disassembly for the 68000 code corresponding to a small area of memory. There's one machine code instruction per line (or a `DC .w` followed by a 2-byte value for codes that don't represent a valid Motorola 68000 instruction), preceded by its starting address in memory.

A line with red text in the disassembly area shows the machine code instruction that will be executed next.

After execution of one or more 68000 instructions, the range of disassembled addresses will change to show the code at the next `PC` (Program Counter), and some code before and after the next `PC`.

You can scroll the code with the up arrow, down arrow, page up and page down keys.

To display the code around an address `<addr>`, you can use the command `"u <addr>`". (`"U"` stands for "Unassemble").

If you hover the mouse over some of the operands that are specified through addressing modes, a pop-up will show the address value. The address value is computed according to the current register values, which may be different than the value it will have when the instruction will get executed. The exception is the `PC` register: when used in addressing modes, it takes the value of the instruction's address instead of the current value of the `PC` register.

4. The register display

Red text in the register display area highlights values that have changed after execution of the last instruction or group of instructions.

The bottom line in the register window may either be blank or display emulator-specific status. (For example, in Q-emuLator I use it to show whether special system code is executing, like system calls or interrupt handlers.)

To change a register value, type in `"<reg>=<value>"` (e.g. `"D0=0"`). `<value>` can be any valid expression (type `"expr"` for expression help).

5. The stack display

The upper right part of the debugger window shows the content of the stack, starting with the current stack pointer address on top. The content is shown in words, or longs when the values look like valid addresses.

6. Debugger commands

Type `"help"` for a list of available commands.

Note that there is currently no "clear breakpoint" command. However, you can use `"bp <bpNumber> 0"` to clear a breakpoint.

In addition to the debugger commander, when running in Q-emuLator there is an extra SBK command available that takes the name of a SuperBASIC function or procedure as a parameter and returns the address in memory of the code that implements the command. The SBK command only works for built-in functions or functions installed as SuperBASIC extensions.

7. Stepping through code

Use the `F1` key to step to the next instruction. `F1` steps into function calls.

Use `F2` to step to the next instruction, but step over function calls as if they were a single instruction.

`F3` works like `F2`, with the additional rule that if the instruction is a conditional branch, then it executes instructions until the next PC is reached (i.e. 1 instruction if the branch is not taken, or any number if it is taken). It is similar to setting a breakpoint to the next instruction and then restart emulation. This is especially useful for the branch instruction that ends a loop, when you want to go to the first instruction after the loop is completely executed.

`Shift F3` is complementary to `F3`: For conditional branch instructions, it runs until the branch target address is reached.

To run unconditionally (or until execution hits the next breakpoint), use `F4`, or type 'g', or press the 'Continue' button.

8. QL specific commands

The `"jobs"` command lists the QL jobs currently executing.

The `"sbk <keyword>"` command finds the starting address in memory of the implementation of a QL built-in function or procedure. The command also works for installed SuperBASIC extensions.

9. Using debug_68k.dll in emulators

It's easy to add debugging capabilities to an emulator of a M68000 based system by using `debug_68k.dll`. If you are an emulator author, write to qemuLator@kagi.com to receive instructions, the header file with the interface, and sample glue source code. Use is free provided credit for it is given in the emulator's documentation, and the `debug_68k.dll` and this document are distributed in their original form with the emulator. Emulators can extend the debugger to provide emulator-specific commands.

Appendix VII

Contact information

Q-emuLator support site

You can find news and information about Q-emuLator (both Mac and Windows versions) at the following World Wide Web address:

<http://www.terdina.net/ql/q-emulator.html>

Q-emuLator blog

Official Q-emuLator blog:

<http://qemulator.blogspot.com>

Subscribe to the blog's RSS feed to automatically receive notifications for Q-emuLator updates.

E-mail address for feedback

Please send feedback about Q-emuLator (support requests, bug reports, questions, comments, suggestions, etc.) to the e-mail address present on the Q-emuLator web support site (it's at the bottom of most pages).