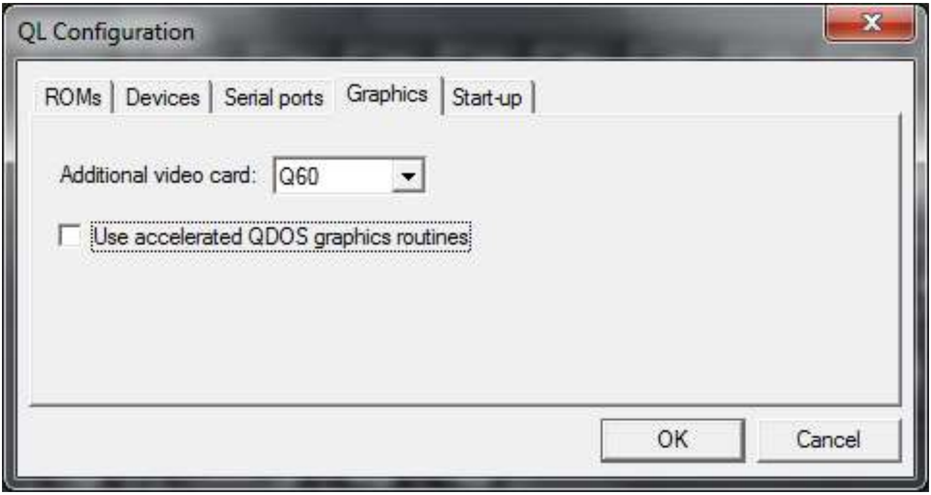
QemuLator = Q60? Dilwyn Jones

Our Editor saw an interesting little post on Daniele Terdina's Q-emuLator blog http://qemulator.blogspot.co.uk/, where the author had taken advantage of the new terms of the SMSQ/E licence and released a version of SMSQ/E tailored to allow Q-emuLator to support a 16-bit graphics mode, similar to a Q60.

I have in the past used the Aurora Gold Card version of SMSQ/E to run a 256 colour graphics mode compatible with the Aurora video modes, which was great. Users of Q-emuLator may have noticed that in its Configuration menu's "Graphics" tab it supports a couple of additional video card emulation modes. One is the Aurora mode I mentioned. The other is Q60 mode – see the screen dump below:

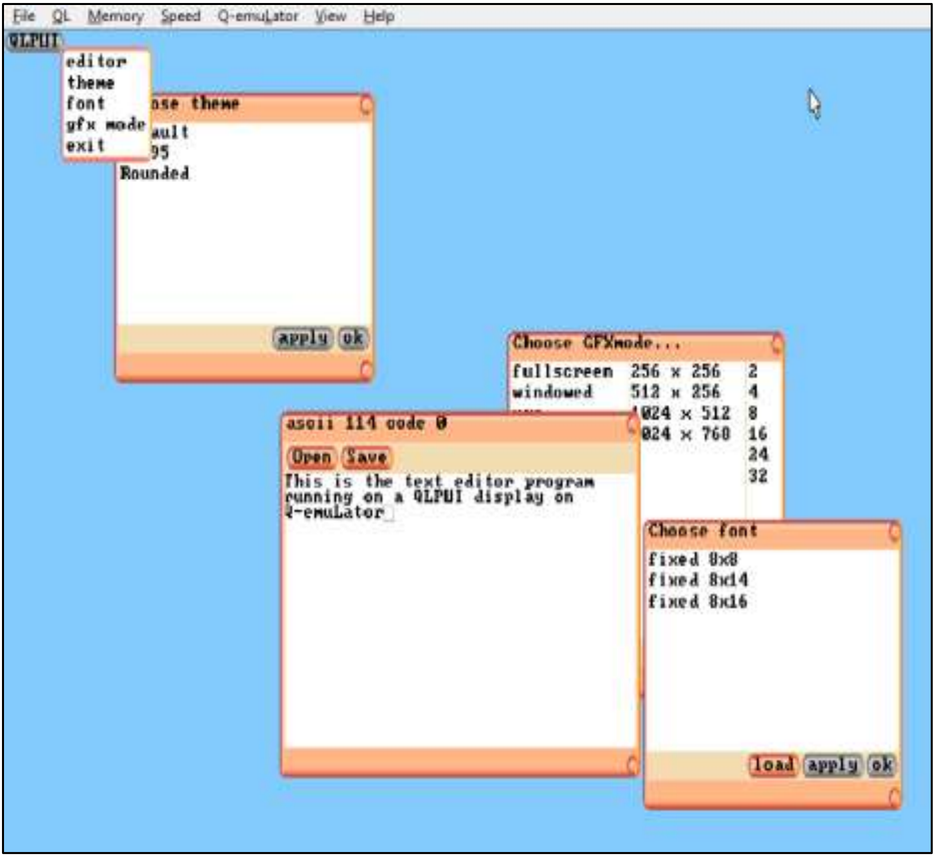


Additional video card: Q60

□ Use accelerated QDOS graphics routines

Selecting the Q60 video card emulation in the Configuration menu

Originally this had been produced to allow Peter Graf to develop a new modern prototype GUI (graphical user interface) for QDOS, called QLPUI. There is a demonstration version of this available as a QLPAK file on Daniele's website's software page at http://www.terdina.net/ql/software.html and a sample screen dump is shown below.



This is the text editor program running on a QLPUI display on Q-emuLator

QLPUI sample screen from the demo QLPAK.

Daniele Terdina had made a version of SMSQ/E able to use the 16-bit colour modes at resolutions up to 1024x768 back in 2010. But back then it wasn’t possible to publish the binaries because of the licensing issues. More recently, the terms of the SMSQ/E licence changed to make it easier to release self-assembled versions of SMSQ/E to be released, which allowed Daniele to release this experimental version of SMSQ/E which allows you to use the 16-bit Q60 colour mode (which is not quite the same as the equivalent QPC2 and QXL 16-bit colour modes) at resolutions up to 1024x768. The Q40 computer’s resolution of 1024x512 seems to be possible as well, which is really useful for testing programs to see if they work on a Q60 mode 33 display.

To explain the differences between the two types of 16-bit colour modes:

The QPC2 and QXL 16-bit colour modes are referred to as Mode 32 displays. Mode 32 is the one generally used on Intel-style processors such as Windows PCs – QXL and QPC2 are both designed for Windows PCs.

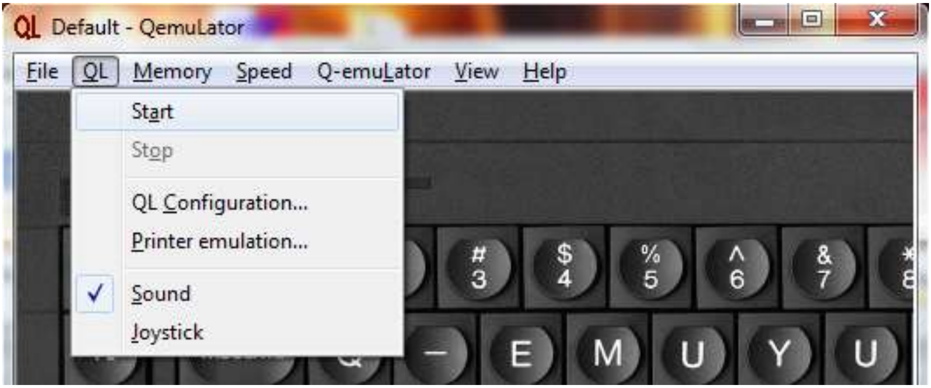
The Q40 and Q60 16-bit colour modes are referred to as Mode 33 displays. Mode 33 is generally used on Motorola processors – the 68000 family of micro-processors, as used in Q40 and Q60 computers. Q-emuLator chooses to emulate the Mode 33 display.

Although described as something which should be regarded as an "experimental" release of SMSQ/E I got it to work pretty easily on my dreaded PC (there's a first for everything!).

The recommended version of SMSQ/E for Q-emuLator can be downloaded as a zipped file from http://www.terdina.net/ql/software.html . There is another version on the same page which allows Q-emuLator to emulate an Aurora board with a 256 colour display.

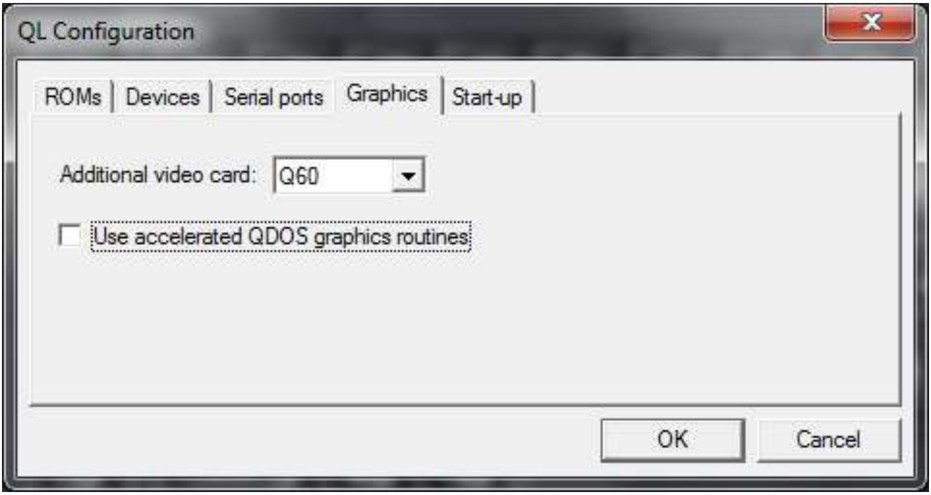
Simply save and transfer the zipped SMSQ/E to one of your QemuLator WIN drives and unzip the SMSQ/E. This should leave you with a file called SMSQ_QEM

In the QL menu of Q-emuLator, select the QL Configuration option to start the Configuration menu:



'QL' menu

Then select the Graphics tab within the QL Configuration menu:



Additional video card: Q60

□ Use accelerated QDOS graphics routines

Configuration menu, Graphics tab

In this tab there is a drop-down list adjacent to "Additional video card:". This will probably be set to None at first. Change it to Q60, then click on OK. Ignore the "Use accelerated QDOS graphics routine" option for now. Start the emulator by clicking on the picture of the QL.

You can now LRESPR the SMSQ_QEM file from BASIC once in Q-emuLator. Assuming it's in drive 1 in Q-emuLator:

```
LRESPR win1_SMSQ_QEM
```

Normally, this will start in a 512x256 four colour (Mode 0) display. From here you can issue a DISP_COLOUR command to make it change colour mode and screen resolution.

```
DISP_COLOUR colour_mode,x_size,y_size
```

Or just

```
DISP_COLOUR colour_mode
```

The colour mode numbers are usually:

0 for standard QL colour mode
1 for 16 colour modes (which don’t exist on any QL system I know of)
2 for 256 colour or 8-bit modes like an Aurora
3 for 16-bit colour mode (includes QPC2, QXL and Q40/Q60)
4 for 24-bit colour mode (no such QL system yet that I know of)

For our purposes now, we will use 3 for the Q40 and Q60 style modes.

The x-size and y-size parameters refer to the screen resolution, the number of pixels across and down the screen respectively. These values can be up to 1024 across (x-size) and up to 768 down (y-size). Bear in mind that these high colour modes need a lot more memory than the standard QL colour modes, so may put a limit on the maximum amount of RAM available to you in the emulator – the QemuLator manual states that you can “only” have up to 4MB or 8MB of RAM because of the address space taken up by the larger Aurora and Q60 displays. QL software is pretty compact on the whole and so this should never really be an issue in 99% of cases.

If we just want to change the colour depth (number of colours) we can just use:

```
DISP_COLOUR 3
```

If we want to change the display resolution at the same time, we can use the longer version:

```
DISP_COLOUR 3,1024,768
```

If your PC doesn’t support that size of screen display, try lower values such as 1024,512 or even smaller until you find values that work on your PC.

There is an additional command called DISP_SIZE. This will let you change the display resolution without changing the colour mode.

```
DISP_SIZE 1024,512
```

Sets the emulator to 1024 pixels wide, 512 high.

If you want to check which colour mode the emulator is, try this function:

```
PRINT DISP_TYPE
```

Or

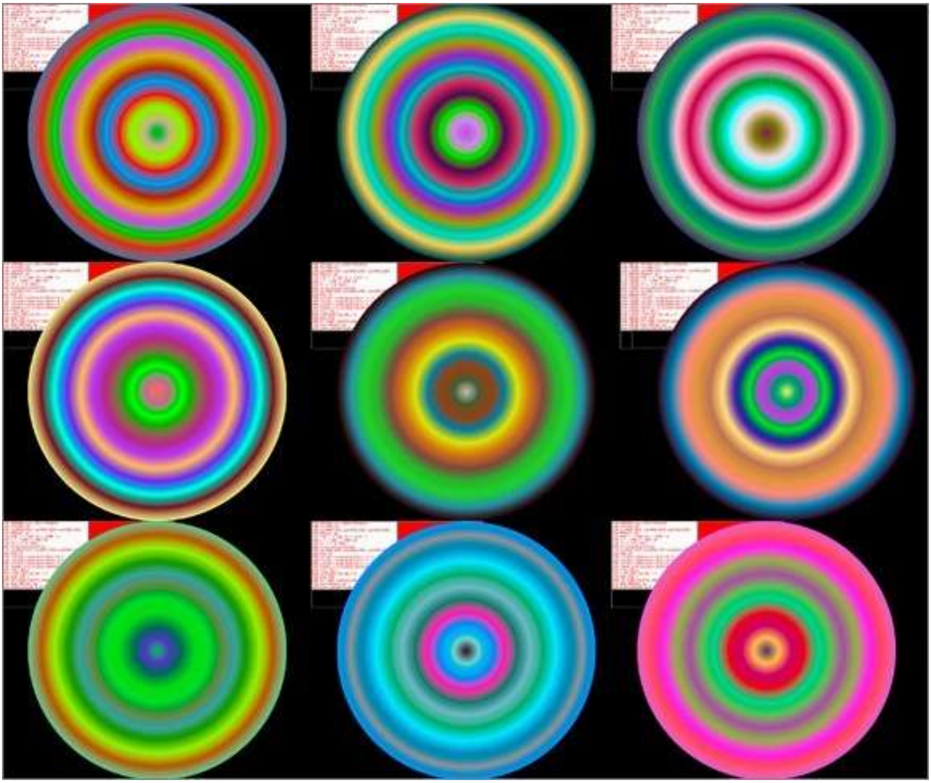
```
LET mde = DISP_TYPE : PRINT mde
```

This will return the screen mode number. Possible values are:

0 = 4 colour mode
8 = 8 colour mode
16 = 256 colour mode (Aurora graphics)
33 = Q40/Q60 16-bit colour mode
32 = QPC2 or QXL 16-bit colour mode, which as far as I know you can’t get on Q-emuLator.

And that’s really all there is to it. Most of your QL software (at least the ones which don’t do anything specific to a particular colour mode – games and graphical programs sometimes write direct to fixed memory locations and may run into problems on enhanced displays) will continue to work as normal and when writing your own programs you’ll have the benefit of all the extra facilities available to you from SMSQ/E and SBASIC.

The following picture is “stolen” from Daniele’s blog and shows the output of the “circles” demonstration program. Not particularly effective in the black and white printed magazine of course, but you may need dark glasses if reading this in the PDF magazine!



I did run into a small problem with the SMSQ/E. It only seemed to work in full-screen mode where the emulator takes over the entire PC display. If I then switched back to windowed mode with either F12 or ALT TAB to use something else in Windows, I lost control of the emulator, which resorted to showing the original QL display before switching modes (presumably the various screen modes are held in separate memories?) and sometimes flashing violently and losing part of the drive displays, forcing me to minimise the display or switch back to full screen mode with F12 or SHIFT F12, mildly annoying. I reported this to Daniele in case it was a bug, but more likely a “feature” of my PC. As I only started using this version of SMSQ/E on the day of the deadline for articles in this issue, it didn’t allow time for a reply so I may need to follow this up in the next issue.

This is a brave new departure for Q-emuLator which so far has largely been promoted and sold on the strength of its QL and QDOS compatibility and ability to run software (including older software) with a high degree of compatibility. It has long been possible to buy the Aurora SMSQ/E to give Q-emuLator a 256 colour Aurora compatible display and now it can be Q60 display compatible too. Although I haven’t yet gone very far in testing this, it will probably be very useful just for testing my software for Q60 display compatibility. Don’t think of it as just being a little something extra for testing software – this has taken Q-emuLator to a whole new level by offering the choice of using QDOS or SMSQ/E and access to SBASIC, choices of display sizes and colour depths, together with a choice of speeds (the emulator can be set to QL speed, Gold Card speed or as fast as your computer can go!) and RAM sizes.

Q-emuLator is available for both Windows and Mac OS X. I have only tested this SMSQ/E on a Windows copy, but there is nothing I can see which says it cannot work or ought not to be used on the Mac version of the emulator. Perhaps someone who uses Q-emuLator on a Mac OS X system can try it and let me know.

STOP PRESS

Just as this issue was going to print, a reply was received from Daniele Terdina concerning the ALT-TAB problem I experienced with the Windows version of Q-emuLator. This should not be an issue at all on the Mac OSX version of Q-emuLator.

In the reply he states that this is currently a known issue, that the High Colour modes are intended to be full screen only on a Windows system. Should you need to use ALT-TAB to switch to another Windows program, you may find it results in the emulator display being in an unknown state and the need to press F12 once or twice to recover.

When you are in full screen mode, it is better to use F12 than ALT-TAB. After the F12 you'll see the normal QL display area, not the video display area used by SMSQ/E, but F12 again should allow you to go back again.

Apparently, use of ALT-TAB control passes back to Windows without the emulator knowing and without exiting full screen mode. This is something which may get improved in future versions, but stick to using F12 for now to work around this.

QemuLator update released

Daniele Terdina has released version 3.1.3 of QemuLator for Windows, which is available to download from his website at: http://www.terdina.net/ql/winql.html

Changes in this release include:

- Exit and re-enter full screen mode on Alt-Tab.
- Return cartridge name in "get medium info" trap for microdrive images (*.mdv).
- Fixed crash when QCF file was passed on the command line.

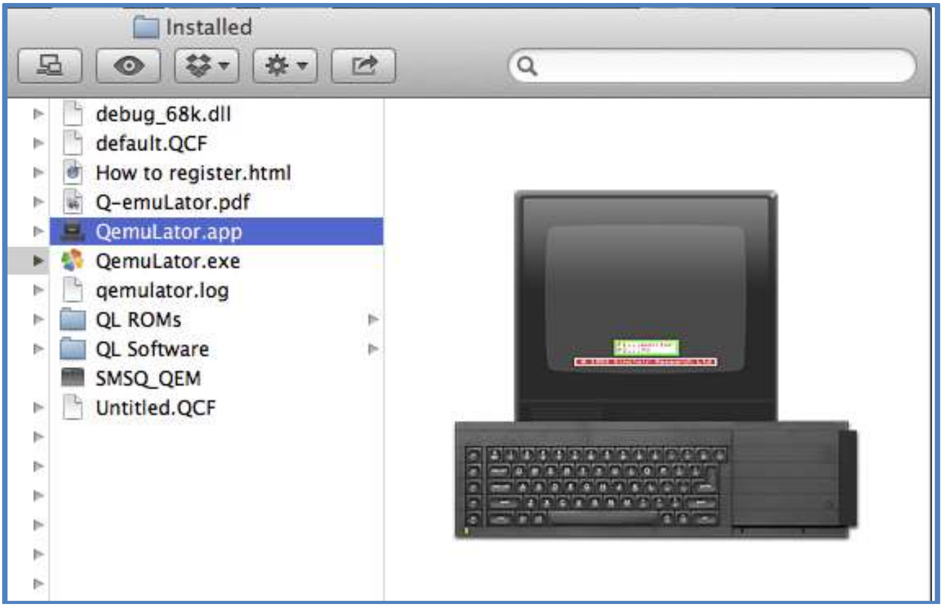
QemuLator is Shareware, but the code you get when you register will also be valid for unlocking (automatically, no re-registration needed) future bug fixes and minor improvements (any future QemuLator for Windows releases with version number 3.x).

TESTING SMSQ/E ON THE MAC Lee Privett

Well the luxury of having just a large screen size resolution was enough to spur me in to testing the MAC version of QemuLator with

SMSQ/e. I had already followed instructions, both from the Daniele's website and Dilwyn's, LRESPR the code.

The first thing was to type EXTRAS to check the plethora of commands and extensions SMSQ/e gives. The next job was to write a suitable generic boot program to make use of the larger screen size, with the resultant code below with hopefully plenty of explanatory REM statements.

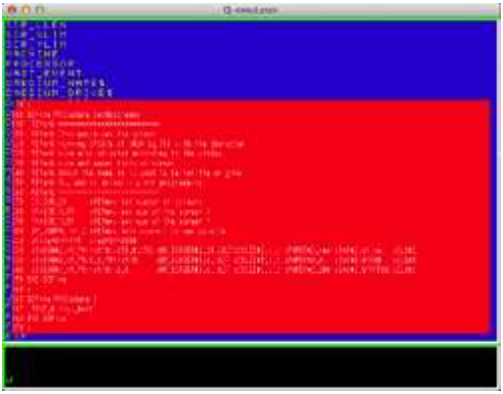


debug_68k.dll
default.QCF
How to register.html
Q-emuLator.pdf
QemuLator.app
QemuLator.exe
qemulator.log
QL ROMs
QL Software
SMSQ_QEM
Untitled.QCF

Screen shot of QemuLator folder on the iMAC

```
100 REMark =========================================
110 REMark Menu_Boot Version 2 for SMSQ/E version 3+
120 REMark Lee Privett 2013 February for Q-emuLator
130 REMark =========================================
150 :
160 SetUpScreens
170 :
180 DEFine PROCedure SetUpScreens
190 REMark ==========================
200 REMark This maximises the screen
210 REMark running SMSQ/e at 1024 by 768
215 REMark with the character
220 REMark size also adjusted according to the
window
230 REMark size and makes fonts on screen
240 REMark about the same as it used
241 REMark to be for the original
250 REMark QL, and is primarily for programming
260 REMark ==========================

270 COLOUR_24 :REMark Set number of colours
280 XM=SCR_XLIM :REMark maximum of the screen X
290 YM=SCR_YLIM :REMark maximum of the screen Y
300 SP_JOBPAL -1,2 :REMark Sets colour palette
310 White=$FFFFFF :red=$FF0000
320 WINDOW#2,XM,YM-(YM/8)-150,0,150

321 WM_BORDER#2,20,1027:CSIZE#2,1,1

322 PAPER#2,red :INK#2,White :CLS#2

330 WINDOW#0,XM,YM/8,0,YM-(YM/8)

331 WM_BORDER#0,6,1027 :CSIZE#0,1,1
332 PAPER#0,0 :INK#0,$FF00 :CLS#0
340 WINDOW#1,XM,YM-(YM/8),0,0
345 WM_BORDER#1,6,1027 :CSIZE#1,1,1
347 PAPER#1,200 :INK#1,$FFFF00 :CLS#1
350 END DEFine
360 :
365 DEFine PROCedure S
367 SAVE_O flp1_boot
```

368 END DEFine



```
SCR_LLEN
SCR_XLIM
SCR_YLIM
MACHINE
PROCESSOR
LAST_EVENT
[illegible]
[illegible]
[illegible]
```