

QL Today

Volume 6
Issue 5
Jan./February
2002

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

Lots of things that will make you smile!

QPC2 Version 3 is being shipped!

**The Production of Q60 has started!
Welcome D&D!**

QDT is coming on quite well! More News inside!

**More QL Shows planned and confirmed:
England, Eindhoven, USA - details on the back!**

Many Software Updates - details inside!

New Hardware for the QL planned ...

QMenu and Turbo

A good QL start 2002!

Contents

- 3 Editorial
- 4 News
- 9 Uncle Klaus is very old *Geoff Wicks*
- 10 QL Roadmaps *Roy Wood*
- 11 My Boot *John Perry*
- 15 Clocking on *David Denham*
- 20 Gee Graphics! (on the QL?) - Part 26
 H. L. Schaaf
- 21 TurboPTR and QMenu *George Gwilt*
- 28 Which Machine am I running on?
 Dilwyn Jones and Marcel Kilgus
- 31 How Good Is QL-Rhymes V2.0?
 Al Boehm
- 34 Printers and other Sicknesses
 Marcel Kilgus
- 36 Printers and Obsolescence
 Geoff Wicks
- 37 QLDis - part 6 / 1 *Norman Dunbar*
- 45 Hints and Tips
- 45 For Sale
- 46 QDT: Time for User Input *Jim Hunkins*
- 46 Programming the Pointer Environment
 in C - Part 1 *Jerome Grimbert*
- 53 The QL Users Mailing List
 Bruce Nicholls
- 54 Byts of Wood

Advertisers

in alphabetical order

Peter Graf	29, 47
Dilwyn Jones	51
Jochen Merz Software	35, 53
QBox USA	39
QBranch	14, 15
Quanta	41
RWAP	19
W.N. Richardson (EEC)	23
TF Services	43
Geoff Wicks	39

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
Im stillen Winkel 12 Fax +49 203 502012
47169 Duisburg Box1 +49 203 502013
Germany Box2 +49 203 502014
 Mobile +49 1701 222277
 email: JMerz@j-m-s.com
 email: QLToday@j-m-s.com

English office:

Q Branch Tel. +44 1273 386030
20 Locks Hill Mobile +44 7836 745501
Portslade Fax +44 1273 381577
BN41 2LB email: qbranch@qbranch.demon.co.uk
United Kingdom email: QLToday@j-m-s.com

Editor:

Dilwyn Jones Tel. +44 1248 354023
41 Bro Emrys Fax +44 1248 354023
Tal-Y-Bont, Bangor email: dilwyn.jones@dj.softnet.co.uk
Gwynedd email: QLToday@J-M-S.com
United Kingdom LL57 3YT

Co-Editor:

Bruce Nicholls Tel +44 1708 755759
57 Shaftesbury Rd Fax +44 870 0568755
Romford email: qltoday@q-v-d.demon.co.uk
Essex RM1 2QJ email: QLToday@j-m-s.com
United Kingdom

QL Today is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2000 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

We are very pleased to be able to announce that the next generation super QL, the Q60 from the designer of the Q40 Peter Graf has finally gone into production thanks to a small British company, D&D Systems made up of two long standing QLers, Derek Stewart (who runs the Holborn View Bulletin Board System) and Dennis Smith. The QL scene has been blessed with some excellent QL emulators for other platforms in recent years, but the native hardware scene was not quite so fortunate. We at QL Today would like to wish all concerned well with this venture and hope that the Q60 will be as successful as its specification implies it should. Peter Graf has persisted with his dedication to producing high quality QL hardware and deserves everyone's vote of thanks and congratulations. See the news pages for further details.

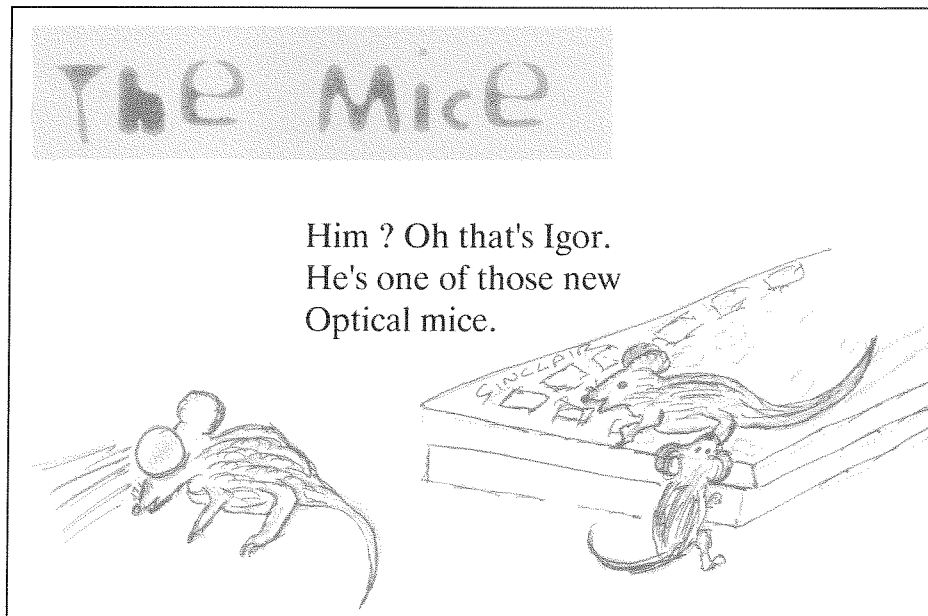
Despite over 15 years of QLing and seeing the dedication of many of us to our favourite operating system, I am still amazed at the ingenuity of some QLers. Take Simon Goodwin's Kodak digital camera software for the QL for example, a neat bit of programming. The RomDisq from TF Services introduced us to flash memory systems. Now, a Greek QLer living in the USA brings us a Compact Flash memory card reader which can apparently be used by Qubide users! Phoebus Dokos located a source of cheap IDE CF card readers which require no software drivers and can be built into a cased QL system rather like a floppy disk drive. See the news pages for

more details. With luck, gadgets like this will provide more reason to bring the Qubide interface back into production.

The new year as expected brought a flurry of Euro-related activity on the QL scene. Andrea Carpi updated the Euro currency converter to take advantage of the GD2 colour drivers, Phoebus Dokos (it's that man again!) contributed Proforma/Prowess fonts which now include the Euro currency symbol, a set of Euro-enabled fonts is now available for QDOS users and of course SMSQ/E users already have the Euro symbol available. Nice to see that despite being a fairly small community, we are still able to keep up to date.

The QL Users Email Mailing List has been buzzing of late with loads of ideas for the future of the QL being discussed by an enthusiastic and loyal bunch of QLers worldwide. QL Forever!

We wish you all a happy new year for 2002 and I for one will be dreaming of two things: A Q60 of course and the long awaited soql internet system from Jonathan Dent.



Cartoon

NEWS

Q60 NEWS from Peter Graf

I have the pleasure to announce that the Q60 series production is now up and running!

The waiting is over, thanks to Dennis Smith and Derek Stewart, who run D&D Systems.

Although the possible user base seems very small, and is already partially supplied by earlier Q40 sales, D&D decided to make the dream of a highend 68060 powered QL come true. When nobody else had the courage to do the Q60 series production, D&D helped. They deserve high respect and a lot of thanks for their decision!

I already had the chance to inspect the first board from their new production and I was very impressed by the fine quality of their work. My own involvement in Q60 production has been reduced to sourcing and preparing parts. Now that the series production is running, I no longer build or sell any prototypes.

Those users who contacted me for a Q60 prototype but were not yet supplied, can now get theirs from series production. For prices and orders please contact:

sales@q40.de

And there is more good news. Instead of just mainboards, D&D will offer complete, cased and tested systems as well!

This is a fine service for those users who do not like to bother with building their own systems, and just like to use a complete and optimized machine.

There will be a choice between

Q60/66: with 68060 at 66 MHz

Q60/80: with 68LC060 at 80 MHz

For a lower cost solution, it is planned to offer an improved Q40 as well. It has similar features to the Q60, but using the 68040 CPU. Details will be published.

All systems are capable of running QDOS Classic, SMSQ/E and 68k Linux. For more information look at <http://www.q40.de>.

[Editor's note: Although he did not include the information in this news item, Peter Graf has published some impressive timings for Q60 on the www.q40.de website. Due to the importance of this project to the QL hardware scene I have reproduced the impressive specification here:

* Q40: 68040 CPU, 40 MHz, Math. Coprocessor, MMU

* Q60/66: 68060 CPU, 66 MHz, Math. Coprocessor, MMU

* Q60/80: 68LC060 CPU, 80 MHz, MMU

* 68060 superscalar architecture, dual execution units

* Up to 160 BogoMIPS performance for QDOS+SMSQ/E (compared to "just" 26.6 for the old Q40 for example!)

* 4 to 128 MB RAM, PS/2 module sockets

* 256 to 1024 kB ROM

* Highspeed 32 bit graphics, with original QL modes

* 65536 colours at 1024 x 512 pixel resolution

* Multisync monitor output

* PC Keyboard interface (DIN)

* 20 kHz Stereo sound

* Battery buffered clock, 2 KB nonvolatile RAM

* Controller for 2 IDE harddisks or CD-ROM

* 2 Serial ports with 115200 Baud, Parallel port, Joystick port (one IO card supplied with mainboard)

* Hardware extension slot supports ISA cards

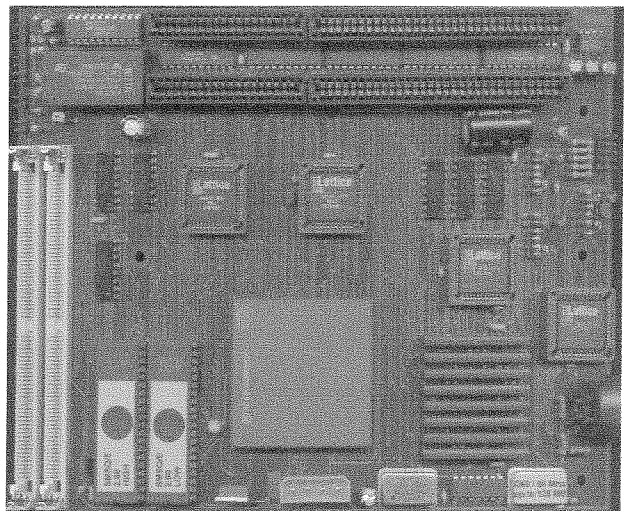
* Fits directly into Minitower or other standard case

* +5V / +12V power supply

* No tinkering, no parts from original QL needed

* Mainboard size 8.2 x 6.3 inch

Now does that sound like every QLer's dream machine or what? - Editor!



D&D Systems,
PO Box 5813,
Ripley,
Derbyshire,
England,
DE5 9ZR
Tel: 01773-740170
Fax: 01773-748399
Email: sales@q40.de

EURO Converter

Andrea Carpi writes:

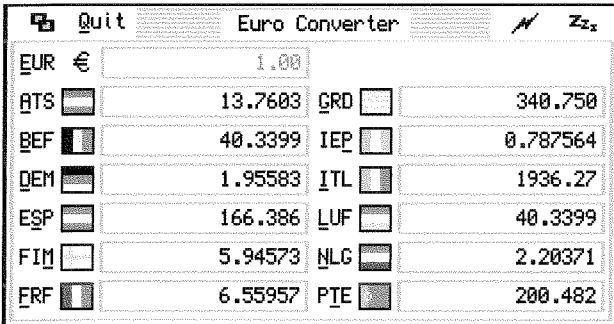
The Euro is here!

And now available from Beginners' Club website, the new version 1.40 of Euro Converter at the address:

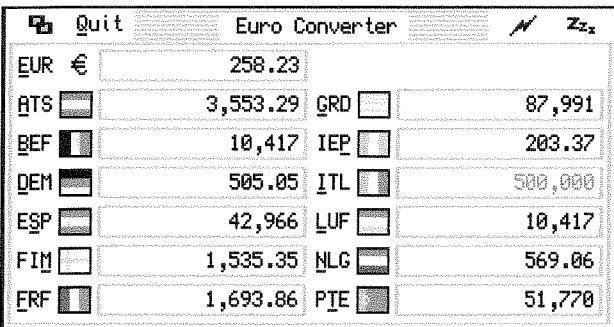
www.beginnersclub.org/e04e.htm

The new version has coloured flags under SMSQ/E 2.98 (or later) with GD2 on QPC2, QXL and Q40/60.

E-mail: info@beginnersclub.org



Currency	Rate	Currency	Rate
EUR €	1.00	GRD	340.750
ATS	13.7603	IEP	0.787564
BEF	40.3399	ITL	1936.27
DEM	1.95583	LUF	40.3399
ESP	166.386	NLG	2.20371
FIN	5.94573	PTE	200.482
FRF	6.55957		



Currency	Rate	Currency	Rate
EUR €	258.23	GRD	87,991
ATS	3,553.29	IEP	203.37
BEF	10,417	ITL	500,000
DEM	505.05	LUF	10,417
ESP	42,966	NLG	569.06
FIN	1,535.35	PTE	51,770
FRF	1,693.86		

News from Tim Swenson

For those that have been waiting, finally, I have finished QHJ #34 and have made it available on my web page (www.geocities.com/svenqhj/).

This issue focuses primarily on TURBO and all of the tools that come with it.

To help in tracking current bugs in TURBO and the TURBO Toolkit, I've created a TURBO Support Page on my web page:

www.geocities.com/svenqhj/

I only have one bug listed, but I'm ready to list more. This page can be used by TURBO users to see if any bug they encounter has been seen or not and if there is a fix on the way.

GWASS V4.15

by George Gwilt

The Gwass assembler program is now at version 4.15. Changes in this version include the ability to input decimal numbers in the range -2^{31} to 2^{32} , which accords with HEX and BIN input. The program may be downloaded from:

<http://www.soft.net.uk/dj/software/other/other.html> and obtained from most PD libraries.

TURBOPTR v4.2

An updated version of the TurboPTR package (which allows Turbo compilation of pointer driven programs) is now available from

<http://www.soft.net.uk/dj/software/other/other.html> and most PD libraries.

Changes to ALTER_TASK v 3.3

(25th December 2001)

1. Made totally free of Toolkit 2.
2. CLOSE is allowed on an already closed channel in SMSQ/E, but not on some other operating systems. ALTER_TASK now only CLOSEs channels which are open.
3. The pointer is repositioned within ALTER_TASK's window after viewing a changed window. (This requires TPTR_EXT v3.2 or later).

Changes to TPTR_BAS v 4.2

(8th December 2001)

1. Removed references to HEX and HEX\$ in `get_x%/get_y%`. Otherwise these were dependent on Toolkit 2.

Changes to TPTR_EXT (Extensions) v 3.2

(25th December 2001)

1. A new function `BASIC_SRT%(dx%)` has been added to return the lower nibble of the first byte of the type word for the keyword with index number `dx%`.
2. `W_MAX` now returns the screen size even within programs with managed windows.
3. The new procedure `SET_PTR wwd,pos` has been added to position the pointer.

BMP2PIC

Bmp2Pic is a new Windows program from Quantum Leap Software (Phoebus Dokos). Before you leap up in protest at the inclusion of news of Windows software in this magazine, let me hasten to add that its purpose is to convert Windows bitmap or .bmp files into QL screen pic files for use with the colour drivers or GD2. It does a quick and simple conversion of Windows graphics files into high colour QL ones, but leaves you with the task of then moving those files onto a QL disk or into a QXL.WIN file depending on what system you use. The program needs the Visual Basic 6 runtimes, a file called MSCOMCTL.OCX to run (most Windows users will already have it, or it's available on many magazine cover CDs for example) – that's not included with the package due to its sheer size!

News from Thierry Godefroy

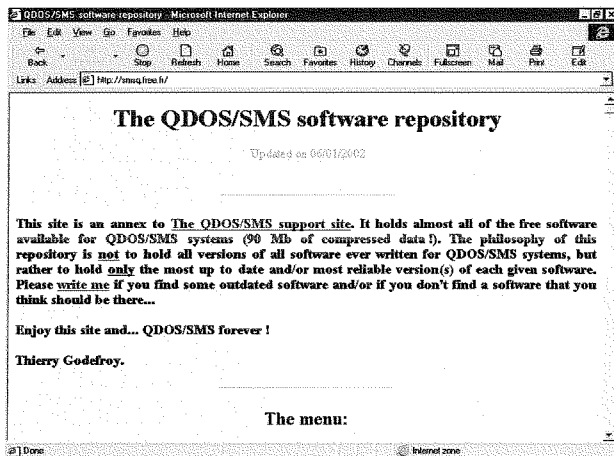
Hi happy QLers !

I've done an (over due) update to my Web site today: <http://qdos.cjb.net/>

and its mirrors, plus I activated the biggest and most up to date download site for QDOS/SMS:

<http://smsq.free.fr/>

Over 90 Mb of compressed software are available there (i.e. all software available on QLCF BBS save the QLCF collection which is only accessible to QLCF members at <http://smsq.free.fr>: 40 more Mb of software !).



USA QL SHOW 2002

Al Boehm

Quanta and NESQLUG are pleased to sponsor the US 2002 QL Show to be held at the Park Inn in Oxon Hill, Maryland on Saturday 1 June 2002. Several European vendors are expected including J-M-S, QBranch and Q-Celt. As usual, Dorothy Boehm will lead a ladies touring group. A Maryland Crab Fest is planned for Sunday.

The Park Inn is a newly renovated motel just off the I-95/I-495 beltway around Washington, D.C. just north of the Potomac River and is 12 km from the White House, U.S. Capital and many free attractions on the Mall. There is a bus service and the Branch Ave. Metro station (www.wmata.com) is five minutes away by car. The special QL room rate is \$59 per night for 1 or 2 people. There is a swimming pool, game room, free parking, free local phone calls, and free continental (coffee, juice, pastries) breakfast. The meeting room is in the Park Inn. For reservations call 301 839-0001 or Fax 301 839-0002 or make reservations on www.parkhtls.com. Indicate QL Show.

NESQLUG will try to arrange airport pickup/return for Reagan Washington National (closest to Park Inn), Dulles, or Baltimore-Washington Airports. Let Al Boehm, tel: 256 859-8051 or email albertboehm@juno.com, know you are planning to come or for more information.

QPC2 Version 3 NEWS

Version 3 of QPC2 is now available, including a new manual. This version of QPC2 includes many changes suggested by users of previous versions.

Screen stretching can now keep the current x:y ratio.

A new power management mode is useful on battery powered laptops for example, because it means QPC only uses as much processing power as is necessary.

Additional configuration items are available.

When running QPC in a window it's now always placed at the center on startup - no more grabbing the title bar to move it to the centre of the screen.

QPC2v3 has revised SER and PAR drivers, they should be much more reliable now. PAR2-PAR4 now available for computers with more than one parallel port. A new "use filter" option is added to the PAR ports. This will allow 3rd party filters to be incorporated into the print process. The check boxes will automatically get enabled when a filter is detected on the system. Option is only available when "printer" is selected for output.

One can now supply a base directory for the DOS device drives. This then acts as the root directory for the drive. Valid entries are normal directory and UNC names, e.g. "C:\WINDOWS\" or "\\MARCELSHARE\". Using the latter variant you can even access drives of other PCs over a network! It's also useful to get around the SMSQ/E filename size limit. There's a new dialog available to adjust the paths. The DOS device now converts characters which are illegal on PC file systems (*:/\ and some more stuff) to some other characters and vice versa. DOS device's make_dir command now correctly returns err_fex if the directory already exists.

Support for wheel mice added. 3 Alt+up/down get stuffed into the keyboard queue for every wheel tick, thus enabling many PE applications to immediately profit from it. Windows 95 implements the wheel differently from all other versions and is not supported. Furthermore, some cheap no-name wheel mice might not generate the necessary messages either.

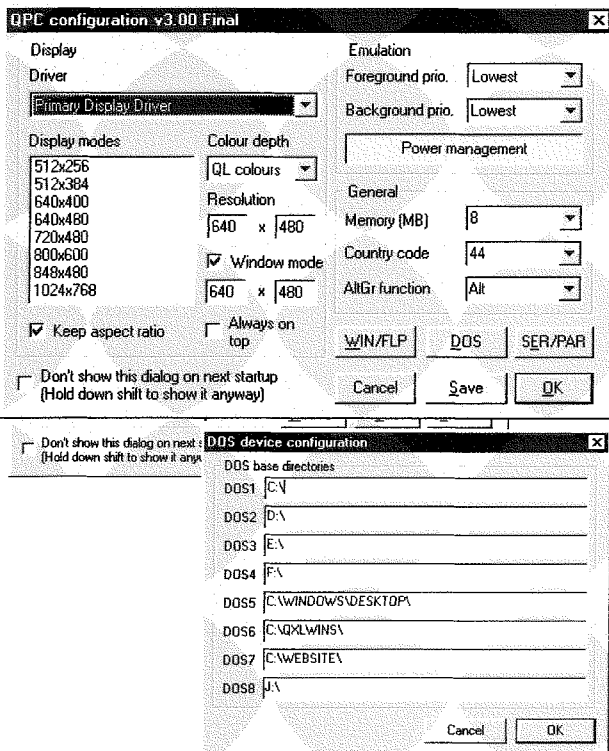
QPC_MINIMIZE/QPC_MAXIMIZE/QPC_RESTORE SBASIC procedures have been added to control the QPC window.

QPC_NETNAME\$ SBASIC function added to get the network name of the current PC. Can be used to distinguish between several PCs in a boot program etc.

QPC_EXEC filename [parameters] is a new SBASIC procedure added to start external Win-

dows programs or files, e.g. "QPC_EXEC 'notepad','c:\text.txt'" or "QPC_EXEC 'c:\text.txt'" to start the default viewer for txt-files.

QPC can now emulate the original 512x256-QL screen at \$20000 in all resolutions and colour depths. This feature can be controlled using the QPC_QLSCREMU command. -1 as parameters enables the automatic mode (emulation depends on the last MODE call), 0 disables the emulation (this is the default), 4 and 8 force the emulation to the specified colour mode. In QL colour modes (4/8) the emulation always uses the same mode, only the 16bit mode can emulate both independantly. With this feature some older applications work quite seamlessly again. Shadows added to PE windows in 16bit mode. This version of QPC2 needs at least SMSQ/E V2a99 (which is supplied with QPC2)



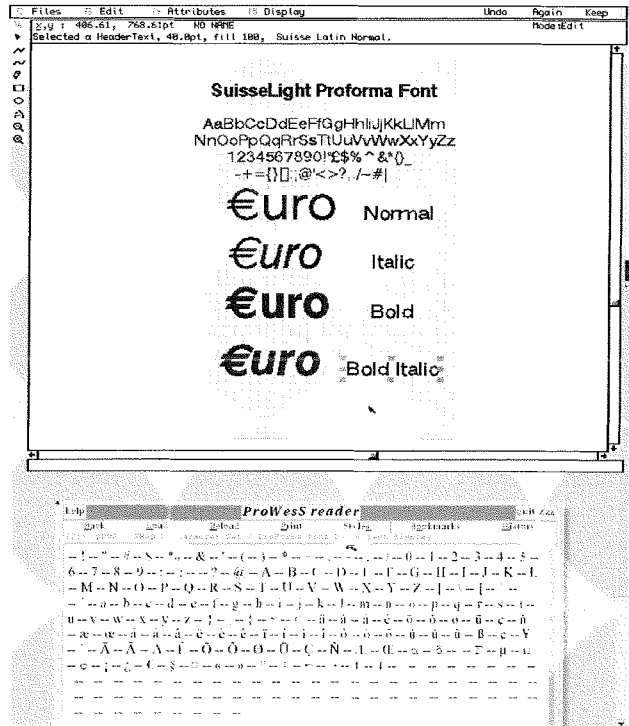
EURO Font

Phoebus Dokos of Quantum Leap Software has released a set of Proforma fonts containing the new Euro currency symbol. The first pack of 4 fonts is available from the usual sources of freeware software, including

<http://www.soft.net.uk/dj/software/other/other.html> and the USA mirror site of Phoebus Dokos. The pack includes normal, italic, bold and italic bold versions of the SuisseLatin font, with further fonts to follow. The Euro symbol is accessed with the CTRL SHIFT u keypress, or with the \paragraph\ entity name in text.

The second pack of 4 fonts is also available at the time of writing, this one being the Guardian

font, also in normal, italic, bold and italic bold flavours.



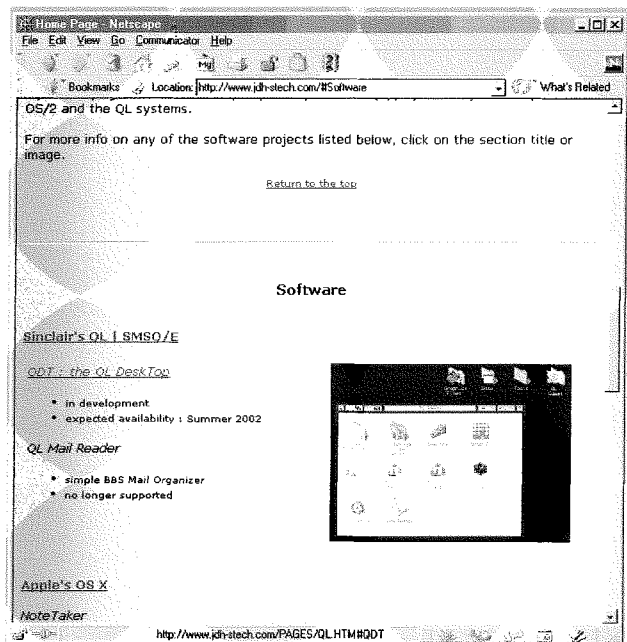
QDT News

from Jim Hunkins

I would like to announce that www.jdh-stech.com (J. D. Hunkins Software Technology) is now on line. Details of the QDT (QL Desktop) project are fully updated. Enjoy.

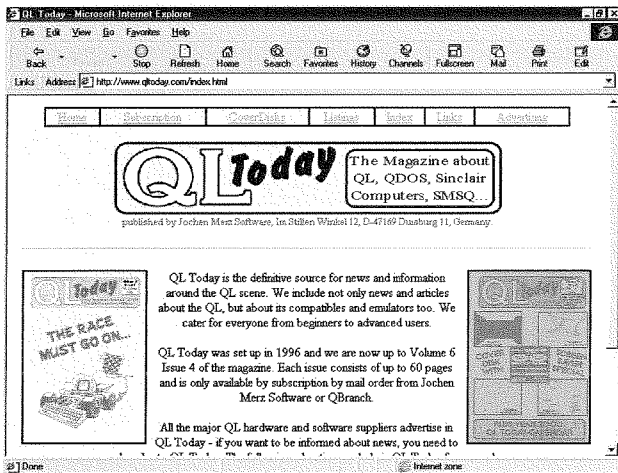
<http://www.jdh-stech.com>

For those of you with QL websites, feel free to add a link to the main page. I would prefer that the link goes to the main page instead of directly to the QL software page as internal locations may change around in the future while the primary page will stay the same.



QL Today's New Website

New at www.qltoday.com is the latest QL Today website. It includes sections on subscription details, cover disks to download (e.g. if you have mislaid your copies), listings, volume index (up to volume 4 so far), links to other QL-related websites and a section on advertising. The website was prepared by Bruce Nicholls.



Compact Flash Adaptors from Phoebus R. Dokos

I secured a batch of 50 IDE compact flash adaptors for \$15.99 each (+ s/h). (about 17 Euros). These are not hot swapable (the hot swapable ones will go for about double the price) but they can accept huge CF cards and even IBM Microdrives (the irony of that!). They work right out of the box with ANY QL with IDE i/f (Qubide/Qubide II, Q40/60) and if I receive enough mail I could order them and pass them at no extra cost to anyone that wants one. I got three myself (well, two, since one goes for QL development purposes) and they are great and extremely fast. If anyone's interested let me know. My email address is webmaster@redoak.net

Or contact me by snail-mail at:

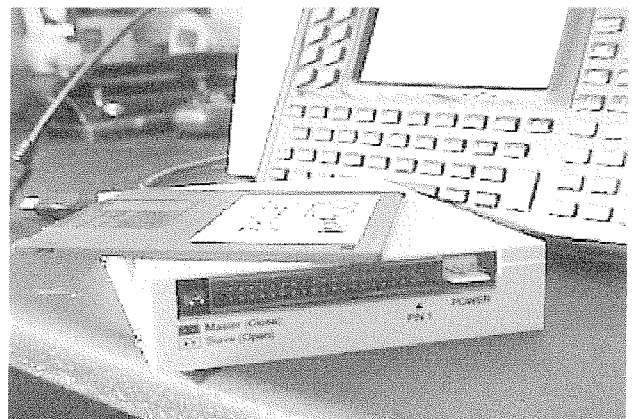
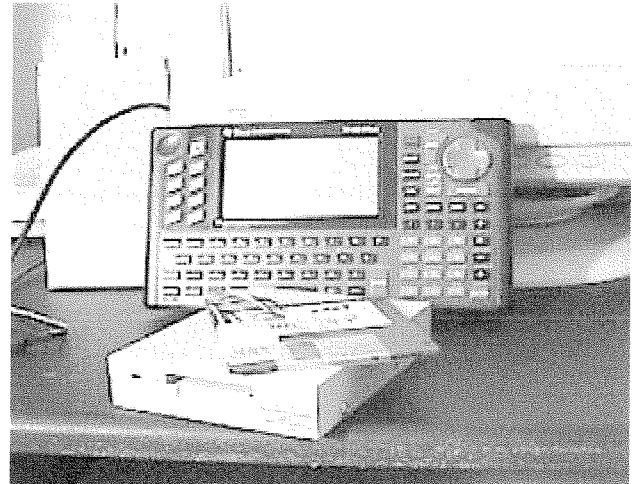
**Phoebus Dokos, 941 Lilac Street Apt.#1
Indiana, PA 15701-3340 USA
Telephone +1(724) 464 0199**

Here are the complete specifications:

Read/Write Led. Master/Slave selectable via jumper. Include screws and power cable adapters (from regular sized power outlet to 3.5" floppy type sized ones) Fastest way to Upload and Download files to or from a CompactFlash card or IBM MicroDrive (up to 40 times faster than USB). Think of the possibilities! Transparent to any operating system, does not require any drivers - it will simply show up as another drive on your computer. Easy way to install disk drive with solid

state storage in existing systems. You simply attach an IDE cable to it. Mounting holes and cable connectors are the same specs as a 3 1/2 Floppy drive. Fits neatly into any 3 1/2 inch floppy drive bay. These are not hot-swapable, you must restart computer each time you need to switch cards. Fully assembled and ready to go... NOT A KIT!

If we can get 100 people that price may drop to 12\$ (About 14 Euros)



EURO QDOS

The My Freeware page on my website now includes a QDOS font with a Euro currency symbol at CHR\$(181) like SMSQ/E. On British keyboards, for example, the Euro symbol is accessed with the CTRL SHIFT u keypress. The font is freeware (please copy at will for all QLers!). A short text file is included to describe how to use it.

EUROFONT.zip may be downloaded from:
<http://www.soft.net.uk/dj/software/freeware/freeware.html>

and in time should be available from all QL PD libraries.

Dilwyn Jones

Uncle Klaus is very old

Geoff Wicks

Many years ago when I was living in the Netherlands, my work colleague responsible for the money was looking glum. The combination number of the safe had just been changed and this was governed by strict legal regulations. She had been told the new code just once and from then on had to remember it. She was strictly forbidden from writing it down. Realising that the Dutch word for safe, "kluis" and the name "Klaus" looked and sounded very similar, I suggested she could not be legally prevented from writing the birthday of her uncle Klaus in her diary. She grinned wickedly and then said solemnly, "Uncle Klaus is very old. I'm surprised he's still alive". Uncle Klaus' birthday was soon transformed into part of his telephone number.

Most of us do not have to put up with this sort of bureaucratic nonsense, but we are increasing being given numbers we have to remember. My Dutch bank card has two PIN codes, one for getting money out of cash machines and large purchases and the other for small purchases. With online banking it is even worse. To access my account I have to type in a 6 digit account number, then a 6 digit PIN code given by my bank and finally a 6 digit PIN code I chose myself. With every transaction I have to enter another 6 digit code that is one of 100 I have been given by the bank.

Usually I have little difficulty in remembering numbers, but if I have not used one of my bank cards for a couple of months, I have to think long and hard to remember the PIN code.

Al Boehm emailed me about this problem. He has to cope with numerous phone and voicemail numbers, bank PIN codes, bicycle lock, driving licence number, automobile tag number etc.,etc. He suggested it should be possible to write a simple computer program to help us remember these numbers by converting them into words.

Telephones have recently regained letters alongside the number and he suggested we could use these letters to generate words. Most bank PIN codes have 4 digits and, in most cases, each number on a telephone has 3 letters associated with it. Thus there would be $3 \times 3 \times 3 \times 3 = 81$ letter combinations or 162 if we allowed reversed words. It would be a simple piece of programming to generate each of these combinations and check against a QTYP dictionary or ASCII word list to see if any were valid words.

Al also pointed out a complication. Not all numbers on a telephone have letters attached to them, as can be seen from the practice in the UK:

1: ----
2: ABC
3: DEF
4: GHI
5: JKL
6: MNO
7: PQRS
8: TUV
9: WXYZ
0: ----

We could use the letter I for the number 1 and the letter O for the number 0, but it would still not solve the problem of a PIN

code like 5795, which would have no vowels.

There are, however, other ways of remembering PIN codes. When they were first introduced the Dutch Postbank produced a booklet giving various suggestions, but I have thrown away my copy and can no longer remember them.

This then is the question. Do you have a system or do you know of a system for remembering PIN codes? Is it a system that could easily be made into a simple program? Al suggests there would be interest in a QL program for converting PIN codes into easily remembered mnemonics. All suggestions and ideas welcomed.

Jochen replies: I suffer from the same problem: too many PINs, and it is very difficult to remember even half of them if you don't frequently use them. The solution to is pretty obvious, I would think, and I wonder why no one has come up with this solution (at least Siemens, Nokia, Eriksson and Motorola have not, as far as I know). We're not talking about Grandma with just one PIN, we're talking about people who have lots of PINs for devices. With a quota of over 50% people in Germany owning a mobile phone, how can it be that all these phones do not have a password area which can be opened with one master password? You can buy these devices (cheque card format), which let you store 10 PINs and lock them with a master password, but why carry extra stuff around if you carry a mobile phone anyway? They have put calendars, alarms, games etc. into the phones, and keyboard, memory and display exist too! So why not add this feature? It would be so easy and useful!

QL Roadmaps

Roy Wood

1985

Bought my first QL from a friend who ran a rehearsal studio in Holloway. The concept at first was to transfer stuff from the PSION II using the Transform software but, as soon as I got it running I got hooked. Subscriber to QL World.

1985

Protests from my then wife and two young children about my hogging the TV set to run the QL caused me to hunt out a small black and white monitor to run it on. Much better picture but the screen, at only 10", was hard to read.

1986

A friend bought a lot of computer stuff at an auction. He wanted the printer and monitor for his own system (Amiga I think but not sure). I got a CST disk interface and 5.25" twin disk drive as well as a second QL. Suddenly found I did not have time to make the tea while Quill was loading.

1987

Bought ICE the 'Icon Driven Front End' and mouse system and ArtIce the drawing Program. I had been struggling with QL Paint and using it to draw stage plans but I found that drawing with a mouse was so much easier. Big boost to my productivity. I also bought Icicle but my unexpanded QL did not have enough memory and I did not have enough money to buy an expansion. Back in the box.

1989

Got a 14" Colour monitor at a local junk shop. What a revelation!!

1991

Moved to Hamburg, Germany and began working for a German sound company. Finally had enough money for a Gold Card and twin HD Drives. WOW. So fast! My enthusiasm ramped up a notch. My wife worked at the local theatre and had longish hours so I began to improve my SuperBASIC and finally realised I had enough memory to use Icicle. Icicle used a simple superBASIC program to poke new commands into the PSION suite so that some of its functions could be called by using a mouse. This was another revelation and I have been a big supporter of mouse based systems ever since.

1992

Bought the first bits of software from Jochen Merz. Found that the twin drives I had bought were out of alignment (probably damaged in post on the way to Hamburg). The good thing about this was I had several conversations with Jochen trying to work out what was wrong. (The drives would read and write perfectly to disks formatted on them but would not read disks formatted on other machines.) We have been friends even since. Got my first tower cased system with a Falkenburg Hard Disk interface and 20Mb drive - so much space, I'll never fill it.

1993

Filled Hard disk and bought a 30Mb. Jochen introduced me to the Pointer Environment and I bought a lot of new programs. QPAC 2 took a while to master but soon I had a screen full of buttons!

1993

Bought Text 87 and LINEdesign. Found out what a word processor should do and have never used Quill since. LINEdesign took a while to get to grips with at first but I offered to help to correct some of the english in the manual and that task taught me how to use the system.

1994

Bought a QXL to go in the 286 Laptop that Bob Dyl got for me. Another revelation! A portable QL at last and I was able to take it on tour with me. In fact it nearly got me fired from one tour because I spent so much time fiddling around with it. Also bought my first Super Gold Card. More speed - More Memory - Whoopee!

1995

I moved back to the UK with my wife and daughter. Started Q Branch. Got one of the first superHermes interfaces from T.F. Tony opened my case in his usual delicate way with a cold chisel and inserted the interface with a lump hammer but the system survived and I had an independent mouse at last. SMSQ/E arrives. A vast improvement on my system.

1996

Bought a better (486) laptop. Much better screen though still not in colour. Just in time for the arrival of QPC. I also began using a modem on the bulletin boards. This was an invaluable resource for sending and downloading program updates. Mostly made possible by superHermes and Jonathan

Hudson's QTPI. IQLR Dies but QL Today rises From the ashes.

1997 1997 Got onto the internet for the first time. Email the first big advantage because Q Branch Customers could contact me easier. Not strictly QL though. The Big QL event was the Aurora. Slightly marred by the three hours of head scratching and one hour of phone calls to Ron Dunnet to find out why it did not work. Turned out to be the monitor which could not handle the signal. Also found that I could not use bigger screens because the Falkerburg HDD interface overwrote some screen lines. Had to buy a Qubide and IDE HDD. After that it was an 'all systems go. Super system'!

1998 RomDisq allowed a stable storage system which would not be overwritten when I made a mistake copying files at a show.

1999 Arrival of Mplane allows the construction of a MinisQL. An Aurora system in a small box. Ideal for show. Q40 finally ready for sale. QPC 2 arrives. I buy a better laptop and get the Q Branch website up and running.

2000 The Q 40 colour Drivers arrive fully. Colours are also readied for the QXL and QPC 2.

2001 Thierry Godefroy's beta test version of the CD drivers for the Qubide and Q40.

These are my own personal landmarks in 16 years of using the QL. How about you ? what are yours?

My Boot

John Perry

I have been meaning for a while now to pass on my wisdom concerning BOOT programs for the QL. I have never really set pen to paper on any QL subject so please forgive me if this is not advanced enough for readers of QL Today, as my computing knowledge does not extend beyond superbasic and pointer environment (PE). On the other hand, not everyone is an expert and this short article might help those wanting to dip their toes in the waters of the pointer environment.

A boot program is pretty essential for using a QL with pointer environment, as the pointer environment isn't built into the operating system, so it always has to be loaded from disk. The boot program fires up automatically when you start a QL - I tend to leave the boot disk in or on the drive overnight so it's ready to go when I switch on. Despite war-

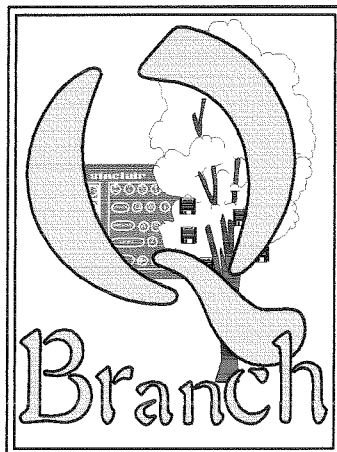
nings, I've never really had a problem with a disk being in the drive when I switched on, I've only ever had problems if the drive light was on when I switched off (presumably that meant the head was in contact with the disk and cause damage either when it was switched off or switched back on).

```
100 REMark BOOT fairly simple
110 TK2_EXT : REMark ENSURE TOOLKIT 2 IS ACTIVE
120 LRESPR FLP1_PTR_GEN : REMark POINTER INTERFACE
130 LRESPR FLP1_WMAN : REMark WINDOW MANAGER
140 LRESPR FLP1_HOT_REXT : REMark HOTKEY SYSTEM
150 LRESPR FLP1_MENU_REXT : REMark QMENU EXTENSIONS
160 LRESPR FLP1_QPAC2 : REMark QPAC2 MAIN FILE
170 LRESPR FLP1_TURBO_TK_CODE : REMark TURBO TOOLKIT
175 LRESPR FLP1_QLOADREF_BIN : REMark QLOAD and QREF
180 ERT HOT_WAKE('x','EXEC')
190 ERT HOT_LOAD('Q','FLP1_QUILL',P) : ERT HOT_PICK('q','quill')
200 ERT HOT_LOAD('A','FLP1_ABACUS') : ERT HOT_PICK('a','abacus')
210 ERT HOT_LOAD('R','FLP1_ARCHIVE') : ERT HOT_PICK('r','archive')
220 ERT HOT_LOAD('E','FLP1_EASEL') : ERT HOT_PICK('e','easel')
230 ERT HOT_CMD('c','charge')
240 ERT HOT_CMD('?', "ed qfind(')"&CHR$(192)&CHR$(192))
250 ERT HOT_CMD('/', "ed qfind"&CHR$(10))
260 HOT_GO
```

On this system, I mainly use pointer environment, QPAC2, Menu Extension and Turbo Toolkit and Compiler. Other programs are loaded ad-hoc

I use more than one boot program in fact, depending on which machine I'm using. For my version JS QL, which has a Trump Card expander and twin floppy drives, I use a fairly simple boot program, which is the one I'll describe here in case it helps newcomers get started.

from floppy disk as and when required. I rarely use hotkeys as I can't remember which keys I programmed (age is the excuse).



Text 87

£ 79.00

Typset 94 £ 29.00
 Fountext 94 £ 39.00
 2488 drivers £ 29.00
 Epson ESC/P2 drivers
 £ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handling the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

PROGRAMMING

QD 98	£ 45.00
QD + QBasic	£ 59.00
QD + Qliberator + QBasic	£ 100.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyprtr pt 1 & 2 (together)	£ 30.00
Easyprtr pt 3 (C library)	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

Happy Q Year

There is still a little 2nd User hardware here at Q Branch Towers so give us a call if you are looking to upgrade. This coming year should prove interesting with a promise of a new IDE hard disk controller and even an ethernet package. These are, at present, just in the design stage but the enthusiasm for this and the GoldFire project is there. We also have seen the results of a new Text 87 Laserjet driver. This driver is very close to release and offers some of the best results I have seen from Text 87 with added support for background images. Watch these pages for a release date.

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00

(Includes Pointer and non-pointer driven versions)

(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)

Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

QL Rhymes £ 10.00

Rhyming Dictionary

Spelling Crib : PD program £ 1.50 +SAE
 or Free if you buy all three programs

New Program !

UTILITIES

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread v3.00	£ 48.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

We are currently out of stock of the SuperBasic Reference Manual

Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.

compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !

£ 40.00

+ postage

Q Branch

Feeling out on a limb?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

Tel +44 (0) 1273-386030 fax +44 (0) 1273-430501
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor

P Demo version £ 1.50 + postage
Full Registered version £ 18.00
Version 2.03 available now !

New Fax Number

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II	£ 100.00
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 50.00 *
Recycled Aurora	£ 75.00 *
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

14" and 15" monitors for the Aurora - Call.

Qubide upgrades to version 2.01 £ 8.00

SMSQ/E

Gold Card / Atari / QXL Version

£ 69.00

Various Atari versions : call for details

QPC 2 v3 full colour version!

£ 84.00 (£ 68.00 SMSQ/E Owners)

Special offer !

Get Cueshell for only £15.00 with any copy of SMSQ/E or QPC 2 upgrade !

Upgrades from QPC 1 £ 46.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Upgrades from QPC 2 v2 £ 14.00
If Bought/upgraded after 15/10/01 £ 5.00

Q Branch Programs

The Knight Safe 3	£ 35.00	Q - Route v1.08C	£ 25.00
upgrades from previous versions	£ 5.00	Route finding programme	
Q - Count	£ 25.00	The Fractal Collection	£ 35.00
Pointer driven home accounting			

Post and Packing is included with Software Prices



We can accept payment by Visa, Mastercard and Switch. We also accept cheques drawn on a UK Bank in Sterling.

You can also pay directly into our bank in Euros.

Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)



Line 110 simply activates Toolkit 2. TK2_EXT is the command to activate Toolkit 2 on systems where the default is 'Toolkit 2 not activated', as on the Trump Card.

Line 120 installs the Pointer Interface. This is the part of the pointer environment which gives you access to the on-screen pointer with mouse and keyboard. My system has a second hand QIMI mouse with 2 buttons (I think it is actually an Atari mouse). The QIMI should have a battery backup for the QL clock according to the person who sold it to me, but I have never been able to get it to work, possibly a faulty battery perhaps? PTR_GEN also saves and restores window contents with its extended console driver - you can see this in action when using CTRL C to gho from program to program if you have more than one program running.

Line 130 installs the Window Manager. I have never really understood enough about this, but I gather it's essential for all those pretty pointer environment program looks - the standard headings, borders, etc.

Line 140 gives me hotkeys. I can set up keys which when pressed with the ALT key will load programs from disk, even if I'm running another program at the time.

Line 150 installs the Jochen Merz Menu Extensions. Apparently it's possible to write your own programs using extensions in this file. I only install it to enable me to run some programs which need these extensions.

Line 160 installs the main QPAC2 file. QPAC2 is a term referring to a big collection of menus and utilities in the one file called QPAC2.

Line 170 installs the Turbo Toolkit. I use the Turbo compi-

ler a lot, so this is quite useful for me. In order to use the compiler, I have to have this little toolkit installed and it provides me with a lot of useful little extensions to basic.

Line 175 loads the Liberation Software QLOAD and QSAVE (fast load and save for basic) extensions, along with the QREF programming aids. QREF is an extension which lists names of variables used, procedure names etc. Extremely useful if you write a lot of basic programs like me.

Line 180 is quite important for me. QPAC2 includes a collection of menus for file handling, job control, channel listings etc and most of these can be accessed directly from something called the EXEC menu. Although the name implies its a menu to execute programs with, in fact what it does is to bring up the QPAC2 menus and a few other 'things' (pun intended: QPAC2 is quite hot on "things" even though I don't really understand the principle) So if I use ALT x a menu called EXEC comes up on screen. If I want to run the FILESmenu to copy some files, I move the pointer down to the FILES entry (or press F) and up comes the files menu. If I have a lot of programs running and I want to zap one of them, I do the same but call up the RJOB menu instead. If I want to jump to a particular program without having to CTRL C round the lot of them, I call up the PICK menu. Couldn't be simpler.

It would be quite easy to set up a similar hotkey to start the files menu directly if you like. ERT HOT_WAKE('f','files') would set up hotkey f to start the Files menu. Or if you are in the habit of running many programs at the same time (why else would some people have 2MB or 4MB memories I suppose!) you could use a similar

definition to set up a hotkey to call the PICK menu to help you choose one of the programs running on your QL, a kind of task switching facility. Incidentally, the HOT_WAKE extension is one that calls up a program or menu which has already been installed in memory with RESPR or LRESPR or the equivalent HOT_RES commands. In this case, the FILES, EXEC and PICK menus have been installed as part of the QPAC2 file.

The next set of hotkeys have been defined to load programs from disk for me. When I press SHIFT ALT R, it calls up Archive for me. I've used an upper case definition here - you can make separate entries for upper and lower case keys. I tend to use the convention that upper case is for loading programs, lower case for picking programs. I pick more than I load, and so as I'm lazy I made the most used one lower case to avoid having to wear out the shift key!

There is a command called HOT_RES which appears to do the same thing, but has an important difference. HOT_LOAD executes a program directly from disk. When you quit from the program, you have to start it from disk again. HOT_RES puts a copy in memory and executes it from there each time. This means it never leaves memory until you switch off or reset the QL. For programs you use a lot (I tend to use Quill more than any other QL program I must admit despite the efforts of some to wean me off it!) HOT_RES can be very useful if you have plenty of free memory and wish to avoid having to repeatedly change disks to load that program.

I treat Quill as a special case when it comes to HOT_LOAD. It has the peculiar habit of

grabbing almost all free memory when it starts. Even if you only want to create a very short 20KB document it might grab over half a megabyte on my Trump Card. QPAC2 has the facility to tame this, by reserving memory before it can get its hands on it. Adding a 'P' option to the HOT_LOAD command will take some memory to one side reducing the amount Quill is permitted to grab.

```
ERT HOT_LOAD ('Q',
'FLP1_QUILL',P memory_value)
is a version which will hand Quill a fixed, known amount of memory, e.g. it could be allocated 128KB. The value is specified in kilobytes. I tend to use the value-less option, which asks for how much memory each time. That way, if I'm only going to write a short letter I can offer it the default of 32KB. If I'm writing a lot of text like this article I can give it more memory, 256KB for example. Giving it slightly higher values than it really needs also helps prevent it running short of memory and slowing things down by writing a DEF_TMP file to disk.
```

Lines 190 to 220 are commands concerned with loading or picking the Psion programs. Line 230 is rather different. I

use the Turbo compiler and that needs the command CHARGE from basic to begin compilation. Very often I'm writing notes in Quill while programming, so I switch between Quill and basic. But it's useful to be able to compile just with an ALT c keypress whatever I'm doing. ERT HOT_CMD sets up a hotkey which first picks basic then enters a basic command. Obviously, it only works if the basic cursor is flashing in channel 0's window and basic can take input.

Lines 240 and 250 are for use with the QREF extensions, more specifically the QFIND function which searches for the line number containing the name given. For example, PRINT QFIND('x') tells you the first line number at which the variable 'x' is used. When used in conjunction with ED or EDIT it greatly speeds up finding and editing lines of basic. So EDIT QFIND('x') for example would bring that line up for immediate editing.

? and / keys are on the same keys, so since QFIND by itself (no name to search for) jumps to the next occurrence of the last search, it makes sense that EDIT QFIND can be used repeatedly to bring up lines referring to the given name. So

I have arranged that SHIFT ALT / or ALT ? brings up the equivalent of ED QFIND("") with the cursor cleverly positioned two spaces from the end ready for you to type in the name required and press ENTER. This is achieved with:

```
240 ERT HOT_CMD('?',ed
qfind("") &CHR$(192)
&CHR$(192))
```

The pair of CHR\$(192)'s move the cursor back from the end of the line to the position needed, for convenience.

```
250 ERT HOT_CMD('/',ed
qfind' &CHR$(10))
```

Note how CHR\$(10) is used at the end of this command to simulate pressing ENTER. It shows how even non-printable characters can usefully be used with hotkeys.

Line 260 issues a HOT_GO command. This 'wakes up' the hotkey system. If your hotkeys fail to work, this is probably why.

Conversely, if you need to LRESPR some more extension files and attempting to LRESPR them gives you 'in use' or similar errors, it may be because the hotkey system is active. Stop it temporarily with the HOT_STOP command. LRESPR should then work, and afterwards you can restart it with HOT_GO.

Clocking on

David Denham

Of all the QL subjects discussed in QL Today, time and clocks rarely seem to be covered (apart from the millennium bug which seemed to pass us by with rather less trouble than on SOME computers!), so I'll remedy that subject myself.

SETTING AND ADJUSTING THE CLOCK

First of all, we'll check if our clocks are correct and on time.

```
PRINT DATE$
```

That should print something like

```
2001 Sep 10 12:45:52
```

We'll assume that was wrong and this is how to correct it:

```
SDATE 2001,10,21,12,30,0
```

The SDATE command takes 6 parameters in the order of year, month, date, hours, minutes, seconds. Just enter them all as numbers.

Want to know what day it is? PRINT DAY\$ will tell you. After setting the date with the command above, enter PRINT DAY\$ should tell you it was a Sunday.

Programmers may find it advantageous to have the time returned as a floating point number. In fact, the value returned by PRINT DATE is the number of seconds elapsed since the first possible clock date. When did time start as the QL knows it (or rather, the part inside the QL which deals with time!)? Try PRINT DATE. It may surprise you to know how many seconds have elapsed since then. OK, so the huge number of seconds is meaningless. If you supply DATE\$ with a number of seconds, it will tell you the date calculated by adding that number of seconds to the first possible date. Try PRINT DATE\$(0). How many QLers had been born by then? Sadly, it won't tell you that..

DATE can be useful for timing purposes, but since it only counts in whole seconds it is not very accurate as a short period stopwatch.

```
100 LET start_time = DATE
110 INPUT "Press ENTER ";z$
120 PRINT DATE-start_time;" seconds elapsed."
```

DATE\$ with one numeric parameter effectively converts the date from numeric to string form. So PRINT DATE\$ and PRINT DATE\$(DATE) are pretty similar (the only difference being that PRINT DATE\$(DATE) needs fractionally more time to do the same thing.

ADATE is a command to adjust the QL clock by a given number of seconds. So if your QL clock is 30 seconds slow, you could move it forward with ADATE 30. Similarly, if it is fast, you can move it back with ADATE -30.

Would you like to know what day you were born? Or how many seconds between given dates? Here are some handy little tricks using the QL clock commands and functions to impress your friends:

Take my son's date of birth - 3rd January 1963. Store the present date in a variable to allow you to restore it later, set the clock value to his date of birth then get the QL to calculate the day:

```
100 LET current_date = DATE
110 SDATE 1963,1,3,0,0,0
120 PRINT DAY$
130 SDATE 1961,1,1,0,0 : REMark zero the clock
140 ADATE current_date : REMark move forward to
    today
150 PRINT DATE$ : REMark check it got it right!
```

Note that on some systems, this will only work for dates after 1960. I don't fully understand why,

but DATE=0 means 1st of January 1961 and I guess something odd is happening when the clock seconds count goes below 0 as SDATE 1961,1,1,0,0,0 (zeroing the clock) wraps round to the year 2097 on some versions of the operating system.

[Refer back to Mark Knight's articles on the QL clock systems for a possible explanation of this, due to the use of signed/unsigned date values. After 1961, on QPC2 anyway, the date seems to wrap around to 2097 for 1960, 2096 for 1959, PRINT DATE(-1) gives 2097 Feb 06 06:28:15 - editor]

To work out the number of seconds between two dates (and from that you can work out days, months, etc since there 86400 seconds per day for example) you can use something like this:

```
100 CLS : current_date = DATE
110 SDATE 1963,1,3,0,0,0 : time1 = DATE
120 SDATE 1963,12,31,23,59,59 : time2=DATE
130 SDATE 1961,1,1,0,0,0 : ADATE current_date
135 PRINT time2-time1;' seconds'
140 PRINT DATE$
```

To find out when 1000 days from now will be, try:

```
PRINT DATE$(DATE+(1000*60*60*24))
```

That's based on adding the number of seconds per hour times hours per day times 1000 days - this can be useful where legalities give you so many hundred days to do something for example.

Clocks

This is all well and good, but how do we actually get the QL to print the time for us? The simplest way is to put PRINT DATE\$ into a loop like this, which will print the current date in the top left of screen window #1:

```
100 REPEAT loop:AT 0,0:PRINT DATE$
```

To time how long you've been working you could use a routine like:

```
100 start_time=DATE
110 REPEAT loop:AT 0,0:PRINT DATE-start_time
```

Since the QL allows us to multi-task programs, a clock is a good example of a program which genuinely needs to run at the same time as another program. Provided the other program

doesn't cover the whole screen to obscure the clock, the time can be continuously displayed alongside your main program. And if you are running SBASIC as opposed to SuperBASIC, you could probably write a little time display routine in another SBASIC to leave your main SBASIC free to run BASIC programs, enter commands etc, without having to compile your clock program or write it in machine code, C etc.

Toolkit 2 fortunately provides us with a ready made clock routine. To see it in action, make sure Toolkit 2 is active on your system and enter the command CLOCK. It starts a 2 line digital clock display in the top right of the #0 window when in Monitor mode. It's a simple little job which has positioned its window there; bury it (CTRL C to BASIC if using pointer environment for example) and it disappears until you CTRL C back to the clock again.

To get rid of the clock, enter the command RJOB 'clock'. The word clock has to be in quotes or you'll get a bad parameter error! Please note that the clock will not operate correctly in mode 8 since the characters will be too large (CSIZE 2,0) to allow the date string to fit into this size of window.

This command features a lot of optional little extra features. For example, if you open a SuperBASIC channel window at a position where you prefer to have the clock, you can tell the clock to run in that window:

```
OPEN #3,SCR_120x10a0x0
CLOCK #3
```

To get rid of this clock, simply close the channel, in this case CLOSE #3. The clock can run in 8 colour mode if you take care over the window size to allow for the character width. Don't issue another CLOCK command without removing the first one or you'll get 2 clocks running, possibly on top of each other. The QL might be multi-tasking but there's no need to take it to extremes!

The clock window needs to be 120 pixels wide and 10 pixels high minimum. If you prefer, a 2 line clock can be set up to run in a window 60 pixels wide by 20 pixels high.

For those lucky enough to have high resolution displays on their system, you can position the clock well out of the way of the normal BASIC windows by, for example, placing the clock

window bottom left of an SVGA 800x600 screen:

```
OPEN #3,SCR_120x10a0x590:CLOCK#3
```

This will keep a clock display going while you are editing a BASIC program for example, but the clock display disappears if you CTRL C to another program.

Toolkit 2 provides some control over the format of the clock if you provide a second parameter after the channel number to specify what to print. This string can include text, names of days and months and of course any of the 6 date numbers (year, month, day, hour, minute, seconds).

If the string contains a dollar symbol, the clock prints the first three letters of the day name if the dollar is followed by d or D, or the first 3 letters of the month name if followed by a m or M:

```
CLOCK #1,"Day is $d, Month is $m"
```

Including a percentage symbol lets you specify the time digits to be printed. The letters can be upper or lower case.

%y includes the last 2 digits of the year number

%d includes the day of the month number

%h includes the hour

%m includes the minutes

%s includes the seconds

The default is '\$d %d \$m %h:%m:%s' which prints time such as Sun 14 Oct 12:30:05

If you just want the hours and minutes for a tiny clock display, use:

```
CLOCK #1,'%h:%m'
```

Which will simply print 14:30 or whatever in a tiny window which should be 36 pixels wide and 10 high. Handy for creating small clocks about the size of a QPAC2 button for example. Now there's a thought, using some of the available toolkits, it might be possible to write a program which runs in the button frame to show the time constantly!

Since the format of DATE\$'s output is fixed, we could extract the time information from it with simple string slices and use those to create an analogue clock if we prefer that to digital.

The listing is rather long since it contains routines for smooth animation of the clock hands. It draws a clock face by putting 12 points at 360/12 degrees apart in the circle (stretching my

schoolboy maths a bit to remember how to calculate points in a circle!), then uses the same sort of code to draw the hands of the clock. Hands are erased when they move by using OVER -1 to exclusive OR the original out. Line 460 even puts in a ticking noise every second. You could even modify this listing to chime every hour by checking if 'minute' and 'second' are both zero and putting in a different beep or even a short tune of beeps! Incidentally, it should be possible to simplify this listing by removing the checks on prev_hour, prev_minute and prev_secod and just erasing and redrawing the hands each pass around the loop, but this introduces some unpleasant screen flicker.

The Listing

```

100 INK 7:PAPER 0:CLS : OVER -1 : CIRCLE #1,50,50,10
110 FOR hour = 1 TO 12
120   POINT #1,50+(8*SIN(RAD(hour*30))),50+(8*COS(RAD(hour*30)))
130 END FOR hour
140 hour = 0 : minute = 0 : second = 0
150 prev_hour = hour : prev_minute = 0 : prev_second = 0
160 first_run = 1
170 REPEAT analogue_clock
180   REMark draw hours hand
190   t$ = DATE$
200   :
210   hour = t$(13 TO 14)
220   IF prev_hour <> hour THEN
230     REMark erase previous hour hand
240     IF first_run = 0 : LINE 50,50 TO 50+(4*SIN(RAD(prev_hour*30))),
       50+(4*COS(RAD(prev_hour*30)))
250     prev_hour = hour
260     LINE 50,50 TO 50+(4*SIN(RAD(hour*30))),50+(4*COS(RAD(hour*30)))
270   END IF
280   :
290   REMark draw minutes hand
300   minute = t$(16 TO 17)
310   IF prev_minute <> minute THEN
320     REMark erase previous minute hand
330     IF first_run = 0 : LINE 50,50 TO 50+(7*SIN(RAD(prev_minute*6))),
       50+(7*COS(RAD(prev_minute*6)))
340     prev_minute = minute
350     LINE 50,50 TO 50+(7*SIN(RAD(minute*6))),50+(7*COS(RAD(minute*6)))
360   END IF
370   :
380   second = t$(19 TO 20)
390   IF prev_second <> second THEN
400     REMark erase previous second hand
410     INK #1,4
420     IF first_run = 0 : LINE 50,50 TO 50+(6*SIN(RAD(prev_second*6))),
       50+(7*COS(RAD(prev_second*6)))
430     prev_second = second
440     LINE 50,50 TO 50+(6*SIN(RAD(second*6))),50+(7*COS(RAD(second*6)))
450     INK #1,7
460     BEEP 50,10 : REMark tick
470   END IF
480   first_run = 0
490 END REPEAT analogue_clock

```

RWAP SOFTWARE

QL Cash Trader v3.7 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT returns, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of the current tax year rules.

Q-Help v1.05 £10

Q-Index v1.04 £5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.

The PD toolkits referred to are available for £2.

Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and find out the commands which relate. Launch Q-Help for further info on the chosen command.

Sidewriter v1.08 £10

Produce landscape printouts of Easel/Qspread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix to laser printers. A most useful utility by Dilwyn Jones - you know it must be easy to use.

ProForma ESC P2 Drivers v1.03 £8

New improved colour and monochrome printer drivers, providing up to 720dpi for all programs written for use with ProWesS, such as LineDesign and Paragraph. Works on all Epson inkjet printers which support binary mode compression (740, 850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.26 £20

Genealogy For Windows £50

Store your family tree for posterity. Add individuals with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family. QL version now supports FileInfo II and QMenu as well as allowing you to link both male and female trees. Sample tree of the Royal family since 1066 included. PC version is event driven - enter the details as they appear in documents and it generates the tree from these. QL data and GEDCOM can be transferred to the PC version. Upgrade to latest PC version (v5.21) for £8 Both programs easy to use and complete with a step by step tutorial.

** QL USERS upgrade to PC version for £25 ONLY **

D-Day MKII v3.04 £10

Grey Wolf v1.8 £8

War In The East MKII v1.24

(Upgrade Only) £5

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while. Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

Image D v1.03 £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

SBASIC SuperBASIC Reference Manual £40

Updates £6 each, £10 for 2 (Current Version - Rei 3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date with regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note: Price for the book does not include post & packing).

QL Cosmos v2.04 £5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars and planets for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

Q-Route v2.00 £25

Upgrade from v1.xx £5

The latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

Flashback SE v2.03 (Upgrade only) £5

The ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately only available as an upgrade from the original version (original still available from Sector Software).

Return To Eden v3.08 £10

Nemesis MKII v2.03 £8

The Prawn v2.01 £8

Horrorday v3.1 £8

West v2.00 £5

The Lost Kingdom of Zkul v2.01 £5

A wealth of QL adventures - mainly text only. Save the Galaxy from the ambitions of the evil dictator Nemesis.

Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horrorday. Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.

Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.

Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.

Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.

All six adventures are available together for only £25.

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for? All 6 programs only £28.

Open Golf v5.20 £8

QuizMaster II v2.07 £5

Stone Raider II v2.00 £5

Hoverzone v1.2 £5

Deathstrike v1.5 £5

Flightdeck v1.05 £10

The latest maps for Q-Route. Maps of various areas of Britain have been created by cutting them out of Big Britain Map - they will use less memory and can contain more detail. Areas covered: Scotland, NE England, NW England, S&W Yorkshire, Wales & Derbyshire, London area and South England. Latest version of Q-Route is recommended.

Britain-map v1.11 £2

Big Britain Map (needs 2MB) v2.03 £5

Various Britain Area Maps (ask for details) £2 ea.

Ireland Map v1.00 £5

Belgium Map v1.01 £2

Catalonia Map v1.03 £2



RWAP Software, 7 Common Road,
Kinsley, Pontefract, West Yorkshire
WF9 5JR

TEL: 01977 614299



Cheques in £sterling
payable to 'R.Mellor'

* Also known as Trading Accounts

Gee Graphics! (on the QL?)

- Part 26

H. L. Schaaf

```
100 REMark code51_bas
110 REMark HL Schaaf Dec 23, 2001
120 REMark to go with GG#26
130 REMark O(n^4) Delaunay triangulation algorithm
140 :
150 REMark QL SBASIC adaptation of Code 5.1 on page 187 in
160 REMark "Computational Geometry in C", 2nd edition,
170 REMark by Joseph O'Rourke, 1998, Cambridge Univ. Press
180 REMark ISBN 0-521-64010-5, and 0-521-64976-5 (pbk.)
190 :
200 MODE 4 : WTV : CSIZE 0,0
210 wh = 202 : ww = 448 : REMark window height & width
220 wx = 32 : wy = 12 : REMark offset of upper left
230 bw = 1 : REMark border width
240 graspix = 476/645
250 IF VER$ = 'JSU' : graspix = 344/549
260 ws = wh-1 : REMark window vertical scale
270 REMark now find the graphic width
280 gw = (ww-4*bw)/(wh-2*bw) * graspix * (ws+1)
290 WINDOW ww,wh,wx,wy : BORDER bw ,4
300 SCALE ws,-gw/2,-ws/2 : PAPER 0:INK 7: CLS
310 :
320 N = 3 : RANDOMISE 42
330 REPEAT Delaunay
340 DIM P(N,3)
350 FOR i = 1 TO N
360 P(i,1) = (RND * (gw*.9)) - (gw*.9)/2
370 P(i,2) = (RND * (ws*.9)) - (ws*.9)/2
380 P(i,3) = P(i,1)^2 + P(i,2)^2
390 END FOR i
400 show_points : start = DATE
410 FOR i = 1 TO N
420 FOR j = i+1 TO N
430 FOR k = i+1 TO N
440 IF (j<>k) THEN
450 xn = (P(j,2)-P(i,2)) * (P(k,3)-P(i,3))
460 xn = xn - (P(k,2)-P(i,2)) * (P(j,3)-P(i,3))
470 yn = (P(k,1)-P(i,1)) * (P(j,3)-P(i,3))
480 yn = yn - (P(j,1)-P(i,1)) * (P(k,3)-P(i,3))
490 zn = (P(j,1)-P(i,1)) * (P(k,2)-P(i,2))
500 zn = zn - (P(k,1)-P(i,1)) * (P(j,2)-P(i,2))
510 flag = (zn < 0)
520 IF flag THEN
530 FOR m = 1 TO N
540 IF (m<>i) AND (m<>j) AND (m<>k) THEN
550 test = (P(m,1)-P(i,1))*xn
560 test = test + (P(m,2)-P(i,2))*yn
570 test = test + (P(m,3)-P(i,3))*zn
580 flag = flag AND (test <= 0)
590 END IF
600 END FOR m
610 END IF
620 IF flag THEN
630 show_triangle i,j,k
640 show_points
```

"Connecting the dots"....

That's what my wife explained to a friend who asked what I was doing on my QL. I've been diverted again, this time to ideas such as Dirichlet tessellations involving Voronoi diagrams and Delaunay triangles on the 2D plane.

Rene Descartes was probably using the diagrams in the early 1600's. Peter Gustav Lejeune Dirichlet described the concepts in a 1850 publication so his name was connected. In 1908 the Ukrainian mathematician Georges Voronoi published seminal papers that caused his name to be connected with the ideas. Theissen used the ideas in geographical ways in 1911, so his name is also connected. Delaunay dedicated his 1934 paper to Voronoi, and thenceforth the name Delaunay was connected with the triangles.

One way to think of it is to have a number of sites such as Post Offices located on a map of a country. Each Post Office site is a Delaunay point. Anyone in that country has a nearest Post Office site. Each site thus has a service region surrounding it. The shape of the region is known as the Voronoi polygon. The boundaries between regions are Voronoi edges. Where three Voronoi edges meet is a Voronoi point.

If I understand it (?) each Voronoi point is the center of a circle through 3 Post Office sites, and that circle is the circumscribing circle of a triangle having those 3 sites as vertices and known as a Delaunay triangle. Furthermore all the sites (Delaunay points) can be connected by a unique network of Delaunay triangles with no triangle edges crossing. Furthermore the Delaunay triangulations includes the convex hull of all the Delaunay points, and also includes the

Gabriel graph, which includes the Minimal Spanning Tree. I see that a whole series of Gee Graphics articles could be used to cover each of these topics. For the moment let's just see what a Delaunay triangulation looks like. I've adapted an algorithm from O'Rourke's book* that he describes as brutally slow, but with amazingly concise code.

If you run the listing 'code51_bas' you will get an idea of what a Delaunay triangulation of points looks like. It starts off with 3 points chosen at random. The next cycle will be 4 random points, then 5, then 6, etc. Sixteen points takes about 24 seconds on my QL with Gold Card and SMSQ/E. How long does it take on your QL system?

Next time I hope to have figured out a way using the QL to show the Voronoi diagram that goes with a Delaunay triangulation.

Dilwyn Jones has kindly offered to put the Gee Graphic listings on his website for download, so I shall try to collect them and have a brief 'readme.txt' file that explains them, then offer them to Dilwyn.

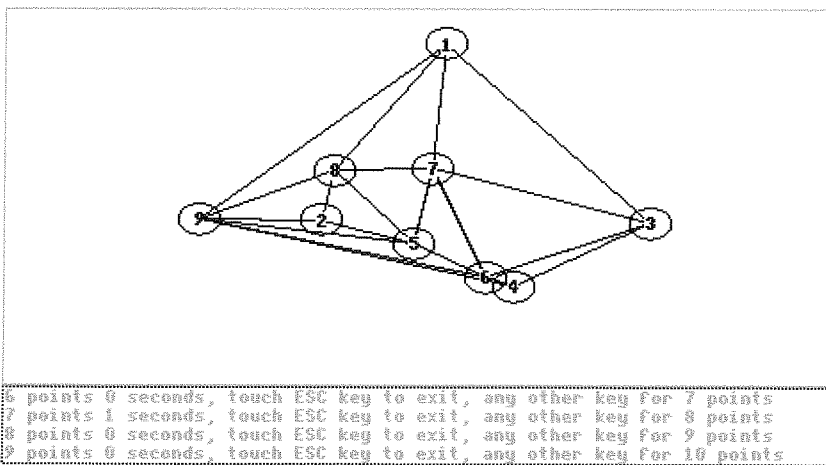
* see listing 'code51_bas' for details about book.

The code will soon be available on the QL Today Website.

```

650     END IF
660     END IF
670     END FOR k
680     END FOR j
690     END FOR i
700     PRINT #0;N!'points'!DATE-start!'seconds';
710     PRINT #0;"; touch ESC key to exit";
720     PRINT #0;"; any other key for ";(N + 1);' points'
730     IF INKEY$(-1) = CHR$(27) : EXIT Delaunay
740     N = N + 1 : CLS
750     END REPEAT Delaunay
760 :
770     DEFine PROCEDURE show_triangle(a,b,c)
780     LOCAL i
790     POINT P(a,1),P(a,2)
800     FOR i = b, c, a
810     LINE TO P(i,1),P(i,2)
820     END FOR i
830     END DEFine show_triangle
840 :
850     DEFine PROCEDURE show_points
860     LOCAL i, i$
870     FOR i = 1 TO N
880     CIRCLE P(i,1),P(i,2),ws/24
890     CURSOR P(i,1),P(i,2),-3*(1+(i DIV 10)), -4
900     PRINT i
910     END FOR i
920     END DEFine show_points
930 :
940     REMark end of listing 'code51_bas'

```



TurboPTR and QMenu

George Gwilt

A recent update to TurboPTR did two things. First it rendered it free of TK2. This was done because it seems that some emulations of QDOS do not automatically include TK2 so the new version of TurboPTR should be suitable in these cases.

The second thing the update did was to include in the Examples, which are designed to show how TurboPTR can be used, EX5a_BAS and its compiled version EX5a_TASK. This new example

shows how QMenu can be incorporated if it is resident.

At this point, in case any reader is unfamiliar with QMenu, I should give some explanation. I was goaded into examining QMenu by the suggestion from John Sadler that I should include in TurboPTR an easy method of dealing with files. More specifically a method is needed of looking through directories to identify files. Anyone who uses QD will be familiar with the elaborate window that appears if a request to load or save a file is made. It must have struck those who also use MenuConfig that the same system for identifying files appears in that program too. But it

was when I noticed that MicroEMACS also uses the same system that the penny dropped. LRESPRng MENU_REXT, which is necessary for both QD and MenuConfig, produces something that is used also by MicroEMACS (v4.00 anyway). Surely then TurboPTR can get in on the act! Well, MocreEMACS, written in C, accesses the routines it wants in a quasi assembler way. So how can that help TurboPTR, written in S*BASIC? Investigation showed that MENU_REXT does two main things. First it sets up two Extension Things, "Menus" and "Scrap Extensions", which contain many assembler subroutines which are available for use. Second, it loads several keywords, which are available for use by an S*BASIC programmer, and so also to anyone using TurboPTR. My investigation would have been easier if I had asked Jochen Merz for information before rather than after discovering the assembler subroutines and the keywords. But I think you learn more by searching on your own and trying out things for yourself than by simply reading a manual.

About QMenu

Some of the keywords added by MENU_REXT concerning files and which are available to programmers are:

Keyword	fn/proc	Used in EX5a of TurboPTR (*)
CHAR_SELECT\$	fn	
DIR_SELECT\$	fn	
EXT_SELECT\$	fn	
FILE_ERROR	fn	
FILE_SELECT\$	fn	*
GET_DEFAULT\$	fn	
ITEM_SELECT	fn	*
LIST_SELECT	fn	
READ_STRING\$	fn	*
REPORT_ERROR	proc	
SET_DEFAULT	proc	
VIEW_FILE	proc	

Anyone who has MENU_REXT loaded can try these out straight away by PRINTing the functions and simply typing the procedures. Several of these will produce a window from which you may be able to escape easily, or not. Others might respond "faulty parameter". In general though you will see first that "FILE_SELECT\$" produces the familiar window allowing you to search for a file. You should also notice that a number of other functions produce windows which may look familiar too. Since these keywords are intended to function with the Pointer Environment, it is necessary for their proper operation to call them from a managed window. Anyone who has TurboPTR

should have LRESPRd TPTR_EXT in which case typing BOUTL will cause the window to be managed. Now the keywords that behaved oddly before should be better behaved. But it would be advisable to have a look at the brief description in Table 1 below before trying again.

To give the flavour of the possibilities, however, I'll describe FILE_SELECT\$ here.

The syntax is:

```
FILE_SELECT$ ([title], [filename],
[directory], [ext], [lines], [x_pos], [y_pos],
[col1], [col2])
```

title	appears as the heading for the window
filename	is the suggested Filename
directory	is the directory to be used
ext	is the extension to be used
lines	governs the vertical size: 0 = maximum 1 = just a request >1 = number of lines - 1
x_pos	for most keywords the horizontal window position but seems to have no effect for FILE_SELECT\$
y_pos	the vertical window position
col1	main window colour
col2	list colour

Colours are numbers from 0 to 3.

	Ink	Paper	Border	Title paper
Number 0	Black	White	Green	Green/White
1	White	Black	Red	Red/Black
2	Black	White	Red	Red/White
3	White	Black	Green	Green/Black

If col1 is given as -1, the colour set to item 64 is used. (See Get_Default\$)

If col2 is given as -1, or is omitted, the colour used for col1 is used for col2 as well.

All the parameters within "[]" are optional and final commas are not needed.

As an example

```
PRINT FILE_SELECT$(,,"win1_misc")
```

will set FILE_SELECT\$ to display the files in the directory "win1_misc_". Note that the final underscore, "_", need not be given in the parameter.

Technically, FILE_SELECT\$ is a function returning a string. Indeed, pressing F3 and typing a sting of characters will cause that string to be returned. However, FILE_SELECT\$ can do much more than that. Before actually returning a filename, you can search through hard disks, floppies, ram1 to ram8

CAMBRIDGE

Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

MOBILE: 07808 576118

E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL, PC AND MAC.

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) *



WE DO REPAIRS, PART EXCHANGES, AND BUY Z88's & PARTS

NEW! CAMBRIDGE Z88 WITH OZ4+512k Internal Ram £150

ALSO AVAILABLE, 512k RAMPACK £40

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	£120	\$180
CAMBRIDGE Z88 COMPUTER	£99	\$130
★ RECONDITIONED Z88	£60	\$90
32k RAMPACK	£18	\$25
128k RAMPACK	£38	\$50
32k EPROM PACK	£18	\$25
128k EPROM PACK	£24	\$33
256k EPROM PACK	£55	\$75
EPROM ERASER	£32	\$43
PARALLEL PRINTER LEAD	£39	\$50
SERIAL PRINTER LEAD	£12	\$16
* MAINS ADAPTOR (230vac; 6v,500ma)	£10	
TOPPER (PROTECTIVE COVER)	£12	\$16
CARRYING CASE (PLASTIC)	£10	\$12

**1/2 PRICE
STARTER PACK
1 - Z88 ★ PLUS
TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!**

FILE TRANSFER TO OTHER COMPUTERS

SPECIAL Z88-QL SERIAL LEAD	£10	\$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	\$3
PCLINK KIT (For PCs) }	£25	\$35
Z88 TO MAC KIT }	£25	\$35
Z88 TO BBC KIT }	£25	\$35

ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

etc. You can even look at files (binary or text) by setting "view" and left clicking on the filename. You can display all files in the given directory and all its subdirectories by setting "tree". This is of course most useful if the extension is set so that only a subset of files is displayed. This provides a method of searching through a directory for all files ending with, say, _ASM.

The complexity of QMenu is such that I would not want to produce a parallel system to use in QMenu's absence. Hence EX5a an TurboPTR shows a simple request for a filename if QMenu can't be used, just as MicroEMACS does.

Is QMenu There or Not?

Since QMenu is a commercial product, only those users who have bought it can use it. Thus any non commercial programs written using TurboPTR should emulate MicroEMACS and contain a test of QMenu's presence and use an alternative "files" procedure in its absence. I'll now show how this can be done in a program intended to be compiled by Turbo.

The first task is to test whether QMenu is loaded or not. This can be done by using Turbo Toolkit's BASIC_INDEX%, which gives a negative answer if the keyword is not there and the position in the Name List otherwise. Thus:

```
1000 num = BASIC_INDEX%("FILE_SELECT$")
1010 IF num < 0
1020 a$ = Get_Fname$: REMark Function to use if
      QMenu absent
```

will show the presence or absence of FILE_SELECT\$.

The program would continue:

```
1030 ELSE
1040 a$ = FILE_SELECT$("Input Filename",
      "flp1_test", "flp1")
1050 END IF
```

If run, a program containing this snippet should work whether or not QMenu is loaded. However, if the program is compiled by Turbo, it will not work if QMenu is not loaded at run-time, thus rendering null and void the object of the exercise, which was to provide a program which operates in QMenu's absence.

What went wrong? When a program is compiled, Turbo keeps a note of all keywords appearing in the program and stores these. When the program runs, a check is made that these stored keywords are in fact loaded. Thus running the

compiled program in the absence of QMenu results in the error message

FILE_SELECT\$ not loaded.

A method of solving this problem is based on Turbo TK Code's function TURBO_V, which returns an absolute address in a compiled program's Vector Table.

This Vector Table contains one long word for every variable name used in a program. For names which are keywords, the entry is the absolute address of the keyword's code. Turbo arranges that when the keyword is used in a program, a branch will be made to its address. Thus the entry in the Vector Table for DATE\$ will be its address. If we were to alter the entry in the Vector Table for DATE\$ to the address of another machine code routine, then whenever DATE\$ was used in the program the other routine would be used instead.

This simple, but effective, method, allows us to use any routine even though its name does not explicitly appear in the program as a function or procedure.

To make this work in our program we need to know the address of FILE_SELECT\$. This is recorded in the second long word of its entry in the Name Table. For each name in the Name List there is a corresponding entry in the Name Table. We know the entry number for FILE_SELECT\$. It is "num", found in line 1000 of our program. Since the first entry has number 0, the second 1 and so on, the number of FILE_SELECT\$'s entry is num + 1. To find the machine code's address all we do is to add

$(\text{num} * 8) + 4$

to the address of the start of the Name Table. This address could be found by using Turbo Toolkit's BASIC_POINTER function. Thus

```
BASIC_POINTER(24)
```

gives the address we want. It is the address stored in the long word at byte 24 of the Basic variables.

There is one slight snag, however. At any time, between any two instructions, Basic may move. The absolute address of the Name Table will alter if a move happens. So, if a move takes place after using BASIC_POINTER to find the Name Table's address and before using that address to discover the machine code address we want, we are in deep trouble and heading for a crash. A

partial solution to this is to check that the address of the Name Table is the same before and after the operation. If it isn't, we simply repeat the operation. This ensures that the only way we could get a false address for the machine code is if BASIC moves after the first use of BASIC_POINTER and moves back to its old position before the the second use of BASIC_POINTER but after determining the machine code address. The chance of this is pretty small and can probably be ignored. However it is not foolproof, so a new keyword has been devised which will be added either to TPTR_EXT or to Turbo Toolkit. The keyword is BASIC_ADR.

BASIC_ADR(num)

returns the machine code address of the keyword whose index number is num. If num is not a valid index into the name table, pointing outside its valid range, or if the name is not a machine code function or machine code procedure, BASIC_ADR returns zero.

Thus the program becomes:

```
1000 ads = BASIC_ADR(BASIC_INDEX%("FILE_SELECT$"))
1010 IF ads = 0
1020 a$ = Get_Fname$
1030 ELSE
1040 POKE_L TURBO_V(DATE$),ads
1050 a$ = DATE$("Input Filename","flp1_test", "flp1")
1060 END IF
```

We have to be careful about which keyword we substitute for FILE_SELECT\$. It must be a keyword of the same general type. The four types are: procedures and the three types of functions returning integers, floating point numbers and strings. Also, certain keywords are specially dealt with by Turbo. Some of these will have a zero address in the Vector Table. Others will have a well defined parameter list which is checked by Turbo. This eliminates several candidates for the substitution but many remain. To be on the safe side it is suggested that the choice be restricted to the following four suitable keywords.

For procedures	-	ADATE
integer functions	-	BASIC_B%
floating point functions	-	DATE
string functions	-	DATE\$

If a keyword used as a substitute is required to be used in its own right in the program, its address from the Vector Table must be restored before its use. Otherwise a program could make the substitution once only at the start.

Odd Happenings Explained Away

This section could be useful to those who have tried out what has been described and who have found that some frustratingly peculiar things have happened. For me at any rate this tends to be the normal case when I try a new system for the first time.

Only two things are mentioned here. All other odd happenings have probably sprung from misuse of the system or careless(?) reading of its description.

1. The program compiles but stops with the message that there is an invalid parameter at line 1050.

This could be because in line 1040 DATE\$ has been misspelt. For example if DATE\$ were typed in as DSTE\$, S*BASIC would set DSTE\$ as a name in the Name List so that "ads", the address of FILE_SELECT\$, would be POKEd into the space allocated in Turbo's Vector Table for the spurious DSTE\$ instead of into the desired location, the space for DATE\$. The program would thus treat DATE\$ as DATE\$ and not as FILE_SELECT\$, with the resultant "invalid parameter" error.

Solutions: Type the name correctly. Alternatively type in the parameter for TURBO_V in quotes. By doing this, misprints will be picked up at compile time, since the presence of quotes will prevent the misprinted item being put into the Name List. This in turn will prevent Turbo allocating space in the Vector Table for the item. That is unless the misprint is in fact a variable used in the program! Barring such mischances, Parser_task will signal an error at the line where TURBO_V appears, because it needs an entry in the Vector Table to have been allocated to process that function.

2. The program stops at line 1050 with "out of range" error.

This could be because the title given to FILE_SELECT\$ will not fit into the space available. This is a very unlikely error. Also the minimum size of window which will accommodate FILE_SELECT\$ is 444 pixels by 98 so smaller windows will cause trouble.

Solution: Obvious! Well all right – restrict the title to a reasonable length and see that the program's main window is no smaller than the minimum.

Table 1 - Brief Description of QMenu Extensions

QMenu is a comprehensive system with many options. This list of some of the facilities of some of the extension keywords should give an idea of what is available and enable TurboPTR programmer's to use these in their own programs.

In all the functions and procedures the parameters:

```
title
x_pos
y_pos
col1
col2
lines
```

have the same meaning when they appear. See FILE_SELECT\$ above.

CHAR_SELECT\$([title],[range],,[font],[x_pos],[y_pos],[col1],[col2])

Returns a character from the range displayed.

range =	0	all characters
	1	non printable
	2	digits
	4	lower case characters
	8	upper case characters
	16	remaining printable characters
	64	cursor control characters

A combination of ranges may be given as a parameter
Thus 12 -> characters of both cases

font = the address of a font

DIR_SELECT\$([title],[lines],[x_pos],[y_pos],[col1],[col2],[dir])

Returns a directory

dir = directory first used

EXT_SELECT\$([x_pos],[y_pos],[col1])

Returns an extension

FILE_ERROR(err_num,[opt],[x_pos],[y_pos],[col1])

Returns	-1	err_num not applicable
	0	ESC or Abort
	1	Retry
	2	Overwrite
	3	Edit

err_num is the error number (eg -6)

FILE_SELECT\$ - see above

GET_DEFAULT\$(item)

Returns the default for item.

item	0 to 7	default extensions
	32 to 39	default directories
	64 to 66	colour combinations

The colour combinations are those indicated in the description of FILE_SELECT\$ above.

Item number 64 is for the main window
65 is for the list window
66 is for the button

ITEM_SELECT([title],[prompt],ch1\$,[ch2\$],[ch3\$],[x_pos],[y_pos],[col1])

Returns 0 if ESC is pressed
1 if ch1\$ is pressed
2 if ch2\$ is pressed
3 if ch3\$ is pressed

ch1\$ to ch3\$ are the choices available

LIST_SELECT([title],lst\$,,flgs,[lines],[cols],[x_pos],[y_pos],[col1],[col2])

Returns the number of the item selected (0, 1, 2 etc) or
-1 for ESC and -2 for OK

lst\$ is a string array containing the items
flgs = 1 for underscore the 1st character of each item
4 for set an underscored 0, 1 etc in front of each item
16 centres the items (may be used with 1 or 4)

Notes:

1. The empty item between "lst\$" and "flg\$" adds further complexity and is not described here.
2. Since LIST_SELECT requires an array to be passed as a parameter, it cannot be compiled by Turbo.

READ_STRING\$([title],[prompt],[default],[lg],[x_pos],[y_pos],[col1])

Returns the edited "default"

default is the string presented for editing
lg is the maximum length of string

REPORT_ERROR err_num,[x_pos],[y_pos],[col1]

This procedure reports the error "err_num" and waits until an appropriate key (eg ESC or ENTER or mouse click) before disappearing.

SET_DEFAULT item,def\$

This procedure sets the default for "item" to "def\$".
See GET_DEFAULT\$ for item values.

VIEW_FILE filename,,,[col1],[col2]

This procedure displays the file "filename".
The missing parameters are not described here.

Which Machine am I running on?

by Dilwyn Jones and Marcel Kilgus

An interesting problem which faces QL programmers now is testing to see which of the multitude of QL-Compatibles the program is actually running on. To facilitate this, a largely undocumented system variable has been added and SBASIC for example has inbuilt BASIC keywords to test and return values associated with this. In general, software would of course run on any machine type, but it is possible that assembler programmers may wish to identify the machine or processor type to build time-critical loops, or to take advantage of facilities available on more recent machines by introducing two routines, one to take advantage of the new facilities to speed up or otherwise improve the program, the other a more general set of code to work on all machines.

The new system variable `sys_ptyp` at offset `$A1` (decimal 161) contains a byte holding values which identify the type of processor and if any floating point unit is available.

The top 4 bits (high nibble, or bits 4-7) identify the processor type:

HEX	DECIMAL	PROCESSOR TYPE
0x	0+	68000 or 68008
1x	16+	68010
2x	32+	68020
3x	48+	68030
4x	64+	68040
6x	96+	68060

The bottom 4 bits (lower nibble, or bits 0-3) contains information about any floating point unit available. It takes the form:

HEX	DECIMAL	FPU DETAILS
0	0	No FPU available
1	1	Internal MMU
2	2	68851 MMU
4	4	internal FPU
8	8	68881 or 68882 FPU

The new system variable at `$A7` (decimal 167) contains a byte holding information about the machine type. Bit 0 indicates if a Hermes or Blitter chip is installed (depending on machine), bits 1 to 4 indicate the machine or emulator type, while bits 5 to 7 indicate the display type as shown in Marcel's list below.

<code>sys_mtyp equ</code>	<code>\$00a7</code>	byte	Machine type / display type
	HEX	DECIMAL	
<code>sys.mtyp equ</code>	<code>\$1e</code>	30	machine ID in these bits
; bit 0			
<code>sys.blit equ</code>	<code>+1</code>		Blitter fitted
<code>sys.herm equ</code>	<code>+1</code>		Hermes fitted
;bits 1 to 4...			
<code>sys.mst equ</code>	<code>\$00</code>	0	ordinary ST
<code>sys.mstr equ</code>	<code>\$02</code>	2	Mega ST or ST with RTC
<code>sys.msta equ</code>	<code>\$04</code>	4	Stacy
<code>sys.mste equ</code>	<code>\$06</code>	6	ordinary STE
<code>sys.mmste equ</code>	<code>\$08</code>	8	Mega STE
<code>sys.mgold equ</code>	<code>\$0a</code>	10	Gold Card
<code>sys.msgld equ</code>	<code>\$0c</code>	12	Super Gold Card
<code>sys.mfal equ</code>	<code>\$10</code>	16	Falcon
<code>sys.mq40 equ</code>	<code>\$11</code>	17	Q40
<code>sys.mtt equ</code>	<code>\$18</code>	24	TT 030
<code>sys.mqxl equ</code>	<code>\$1c</code>	28	QXL
<code>sys.mqpc equ</code>	<code>\$1e</code>	30	QPC
;bits 5 to 7...			
<code>sys.mfut equ</code>	<code>%00000000</code>	(dec. +0)	Standard QL or Futura type of display
<code>sys.mmon equ</code>	<code>%00100000</code>	(dec. +32)	Monochrome monitor
<code>sys.mext equ</code>	<code>%01000000</code>	(dec. +64)	Atari Extended 4 Emulator
<code>sys.mvme equ</code>	<code>%10000000</code>	(dec. +128)	Atari QVME emulator
<code>sys.mqlc equ</code>	<code>%11000000</code>	(dec. +192)	QL mode LCD
<code>sys.mvga equ</code>	<code>%11000000</code>	(dec. +192)	VGA
<code>sys.maur equ</code>	<code>%10100000</code>	(dec. +160)	Aurora

Are you waiting for...

- ... an all new QL?***
- ... superscalar 68060 power at up to 80 MHz?***
- ... integration instead of tinkering with many add-ons?***
- ... more than 65000 colours at high resolutions?***
- ... speed and new features without losing QL compatibility?***
- ... sampled stereo sound for QDOS and SMSQ/E?***
- ... improved peripherals?***
- ... full control over your system by open hardware docs?***
- ... booting your favourite OS from ROM in a few seconds?***
- ... a real 68060 QL instead of emulators?***
- ... QDOS Classic and SMSQ/E with up to 160 BogoMIPS?***
- ... even 68k Linux loading from QDOS or SMSQ/E?***
- ... the fastest QL ever?***

***No more waiting.
The new Q60 has arrived!***



Take the power back in your hands.

***Available from D & D Systems, P.O. Box 5813, Ripley, Derbyshire, England DE5 9ZR
Tel: +44 (0)1773-740170, FAX: +44 (0)1773-748399, Email: sales@q40.de***

***More information about the Q40i and Q60 at the official Website: <http://www.q40.de>, Email: info@q40.de
The Q40i is also available. It offers similar features like the Q60 at lower cost, using a 40 MHz 68040 CPU.***

To locate these system variables from assembler, ensure you call sms.info (trap #0, d0=\$00, system variables address returned in a0) then add the offsets mentioned to the base address and analyse the values at those locations. From BASIC, use any of the commonly available toolkit extensions to locate the system variables, for example, on Minerva or SMSQE you can use the VER\$(-2) function to return the base address of the system variables.

Examples

This little BASIC program will list the information it can extract, somewhat along the lines of the equivalent SBASIC extensions MACHINE, PROCESSOR and DISP_TYPE in SMSQ/E.

```
100 sv = 163840 : REMark fixed location on original QL
110 t$ = VER$ : REMark check version of BASIC
120 IF t$ = 'JSL1' OR t$ = 'HBA' THEN
130   REMark Minerva or SBASIC, supports ver$(-2)
140   sv = VER$(-2)
150 END IF
160 processor_type = PEEK(sv+161)
170 REMark separate processor and FPU type
180 fpu_type = processor_type && 15
190 processor_type = processor_type DIV 16
200 machine_type = PEEK(sv+167)
210 REMark mask out machine_type into the machine and display components
220 display_type = machine_type DIV 32
230 machine_type = machine_type && 31
240 PRINT "Processor Type:";
250 SElect ON processor_type
260   =0 : PRINT "68000 or 68008"
270   =1 : PRINT "68010 (possibly QPC)"
280   =2 : PRINT "68020"
290   =3 : PRINT "68030"
300   =4 : PRINT "68040"
400   =5 : PRINT "Someone must have made a 68050 unit!"
410   =6 : PRINT "68060"
420   =REMAINDER:PRINT "Unknown"
430 END SElect
440 IF fpu_type > 0 THEN PRINT "FPU available"
450 PRINT "Display type:";
460 SElect ON display_type
470   =0 : PRINT "QL style display"
480   =1 : PRINT "Monochrome display"
490   =2 : PRINT "Atari Extended Mode 4"
500   =4 : PRINT "Atari QVME display"
510   =5 : PRINT "Aurora display"
520   =6 : PRINT "QL Mode LCD"
530   =7 : PRINT "VGA displays"
540   =REMAINDER : PRINT "Unknown"
550 END SElect
560 PRINT "Machine Type:";
570 SElect ON machine_type
580   =0,1 : PRINT "Unknown, probably a standard QL"
580   =2,3 : PRINT "Mega ST or ST with RTC"
590   =4,5 : PRINT "Stacy"
600   =6,7 : PRINT "Atari STE"
610   =8,9 : PRINT "Atari Mega STE"
620   =10,11: PRINT "Gold Card"
630   =12,13: PRINT "Super Gold Card"
640   =16,17: PRINT "Falcon"
650   =17,18: PRINT "Q40"
660   =24,25: PRINT "Atari TT 030"
670   =28,29: PRINT "QXL"
680   =30,31: PRINT "QPC"
690   =REMAINDER : PRINT "Unknown"
700 END SElect
710 IF machine_type && 1 : PRINT "Hermes or Blitter installed"
```

Or if you prefer a set of functions to return the hardware details, try these:

```
1000 DEFine FuNction S_V
1010   LOCal t$
1020   t$ = VER$
1030   IF t$ = 'JSL1' OR t$ = 'HBA' THEN
1040     RETurn VER$(-2)
1050   ELSE
1060     RETurn 163840 : REMark QDOS QL fixed location
1070   END IF
1080 END DEFine S_V
1090 :
1100 DEFine Function MACHINE_TYPE
1110   LOCal mt
1120   mt = PEEK(S_V+167)
1130   RETurn (mt && 31)
1140 END DEFine MACHINE_TYPE
1150 :
1160 DEFine FuNction DISPLAY_TYPE
1170   LOCal mt
1180   mt = PEEK(S_V+167) && (128+64+32)
1190   RETurn (mt DIV 32)
1200 END DEFine DISPLAY_TYPE
1210 :
1220 DEFine FuNction PROCESSOR_TYPE
1230   LOCal pt
1240   pt = PEEK(S_V+161)
1250   RETurn (pt DIV 16)
1260 END DEFine PROCESSOR_TYPE
1270 :
1280 DEFine FuNction DISPLAY_TYPE
1290   LOCal pt
1300   pt = PEEK(S_V+161)
1310   RETurn (pt && 15)
1320 END DEFine DISPLAY_TYPE
1330 :
1340 DEFine FuNction HERMES_OR_BLITTER
1350   LOCal mt
1360   mt = PEEK(S_V+167)
1370   RETurn (mt && 1)
1380 END DEFine HERMES_OR_BLITTER
```

How Good Is QL-Rhymes V2.0?

by Al Boehm

Occasionally I write songs. I even sold one or two which by a loose interpretation makes me a professional song writer. Now us professional song crafters are always looking for good tools so I jumped at the chance to review QL-Rhymes. QL-Rhymes is a rhyming dictionary set up to help those who are searching for words that rhyme. Right up front I will say that QL-Rhymes does this task admirably, and I highly recommend it if you have any need at all for rhyming.

Capability

The most important criterion of any tool is its ability to do the job. A rhyming dictionary must be able to come up with a lot of rhymes. Since

these rhymes are mainly used to assist in the creation of poems, the dictionary must be easy to use. It must assist and not slow down the creative thinking of the poet.

For many years, I have used Kevin M. Mitchel's (1996): *Essential Songwriter's Rhyming Dictionary*, Alfred Publishing Co. Los Angeles, CA. It is a good rhyming dictionary but suffers the defect of all paper dictionaries - you have to spend time looking up the word. While I am thinking about the alphabetizing, I loose my train of thought.

With QL-Rhymes I just type in a word, and it quickly and without any thinking on my part displays a whole set of potential rhymes. It is effective. But does the dictionary have enough rhymes? To check this I compared QL-Rhymes against Mitchel's Dictionary by selecting a few random words and seeing how many rhymes, including the original word, showed up.

The result is pretty impressive:

	Random words				
	hole	room	silk	time	thunder
QL-Rhymes	43	52	4	43	8
Mitchel	28	14	4	14	5

Aside from words that have a phonetic rhyme, QL-Rhymes also can list words that have the same spelling ending. This feature greatly expands the number of words you may select from. Of course, some words end the same but do not sound the same at all. In short, QL-Rhymes is a top-notch rhyming dictionary.

Program Aspects

QL-Rhymes requires the Pointer Environment and Toolkit 2. Although the program is only 52k in size, at least 768K of ram (Trump Card or better) is needed for the data base. While the Pointer Environment is required, it is not supplied with the QL-Rhymes disk. It is Q-Liberated and has the Q-Liberator extensions embedded in the program, that is, a separate load of extensions is not required. Also two of Simon Goodwin's DIY routines are built into the program as well as some of Albin Hessler's EASYPTR routines.

The program window is fixed to fit within the standard QL size - 512 by 256 pixels - which is fine for the intended purpose. It can be moved to other locations on the screen. The program can be put to sleep as a button. Up to 24 words are viewable at one time.

Rollovers are used to select options. A rollover is a type of icon also called a sprite that changes appearance when the pointer rolls over it. At this time, if the mouse is clicked or ENTER pressed, whatever the icon represents is done. QL-Rhymes uses SPACE as well as ENTER which is OK, but not consistent with some of the other Pointer Environment programs that do different things for SPACE and ENTER. Rollover is HTML jargon. I think they are called loose menu items in EASYPTR terminology. Whatever they are called, they include text prompts which make it obvious what they do. For example, there are ones for quitting the program, putting it asleep as a button, moving the window, and Help. The most important are the Phonetic and the End Letters menu items to select which manner is to be used in finding rhymes. For words with more than 24 rhymes, a page down rollover prints the next screen of words. A page up rollover allows previous pages to be viewed.

There is a smart menu feature that fades icons that are not applicable at a given stage. For example, the page down rollover has no meaning

and is faded if only one page of rhymes is found. There is a Main Menu rollover that goes back to start over. As in most Pointer Environment programs, the menu items can also be initiated by a key stroke of the letter underlined in the menu prompt. Also the pointer can be moved by the arrow keys as well as the mouse. This extra capability puts the Pointer system ahead of the other mouse/click systems.

As many of you know, I have trouble with my eyes. I count this a blessing in disguise since it has made me think more about visual displays. I had no trouble reading the QL-Rhymes menu items. Except for Configure and Word List explained below, I had no trouble in figuring out what they do. If you will allow me to be picky, some of the title blocks at the top looked liked menu items and I tried to click on them.

Configure Confusion

It's a good thing to be able to tailor a program to your system and your desires in such things as paper and ink color, etc. Four ways to configure a compiled program are:

1. Use a CONFIG block embedded in the program. The disadvantage is a separate CONFIG or MENU CONFIG program is needed.
2. Use a configuration file that stores the desired attributes and is read in by the program. The disadvantage is that a separate file is needed.
3. Add a parameter string to the EX command when the program is started. The disadvantage is that if option switches are used, these strings can be quite arcane and hard to remember.
4. Change parameters on the fly temporarily in the program. The disadvantage here is that these changes are in effect only as long as the program is running unless some other provision such as a configuration file is used to save them.
5. SET parameters in the boot program that EXECutes the compiled program. The disadvantage is that some small knowledge of SuperBasic is required.

QL-Rhymes uses 1, 2, 3, and 4 of the methods. This variety took me a little while to figure out. A two stage set of configure information is used:

1. Location (device and if needed directory) of definition (_def) file. This location must be known when the program starts. Thus, the program must be CONFIGed or started with an attached parameter string. It comes CONFIGed with FLP1_. The only thing the CONFIG program configs is the location of this file.

2. The definition file gives the location of the rhymes word list file plus several other parameters such as ink color and printer baud rate. These parameters can be changed on the fly in the program and once changed can be written to the `_def` file. Also I was able to change the `_def` file directly using an editor program.

This two stage configuration had me baffled for a while particularly since the on-the-fly changes are accessed through a menu item labeled Configure but have nothing to do with the CONFIG program. In addition, the location of the rhyming word list has its own menu item on the main page but can only be saved if the Configure menu item is invoked. When I used Menuconfig, there seemed to be some type of glitch - a lot of gibberish was printed in the little window where you are to enter the location of the `_def` file. To be fair, if you follow the instructions in the manual precisely, the program will end up configured OK.

A Model Manual

The QL-Rhymes Manual is the very model of a modern major manual! It starts off with a concise statement of what QL-Rhymes does. Good. Next it gives system requirements and acknowledgements. It has a short Quick Start section then goes into details of the various menu items. Some theory on rhymes and the format of the rhyming data base is given. Plus there is an index. I did have to look under OUTPUT FILE instead of FILE to find out where directions were for sending rhymes to a file. But since I did find it quickly, I have to give the index a passing grade. Since the Manual is very good, the only thing I can question is the need for a paper manual in the first place. I received the Manual in the mail about two weeks after I received program and accompanying files zipped over the internet. Except for the configuring, I was able to use the program usefully minutes after I received it. This was due to a well thought out overall design and a six paragraph help file. If a paragraph or two was added to the Help and the rhyme theory and data format placed in a doc file, the manual, for me, would be superfluous.

For every program I wrote in the 1980's, I was strongly told to put out a Users Manual, Theory Manual, Programmers Manual, and an Executive Overview Manual. It is my practice, to read the instructions carefully beforehand. But two things have shaken my faith in comprehensive manuals. When I moved into a new house five years ago, I had a plethora of manuals to read: the alarm

system, the microwave oven, the ice maker, the heater-air condition system, etc., etc. Although I really did try, I never did come to grips with the VCR instructions. What finally sent me over the edge was a two page manual on how to use a mattress.

The second thing is: We have a 17 year old, high school, German exchange student staying with us. He is very much into computers and indeed put together a 486 PC using spare parts I had lying around. He doesn't read instructions! This scared me until I noticed that his success rate and time to get something running right was better than mine. Are paper manuals going the way of the dinosaur?

The answer I think can be found in the QL-Rhymes program. To use it, I really did not need a manual because it was well designed, prompts were informative, and output readily understandable. The manual really did not help me much with figuring out the configure. I just played with it until I understood what was going on. Perhaps better prompts or a different design would have made configuring simple. On an economic level, printing and postage for the manual has to be a large proportion of production costs. Yes, I know there are people who really like the feel of a paper manual in their hands. But the trend is clear.

Desired added capabilities

The main idea of a rhyming dictionary is to nudge the creative process. Sometime I just browse through the paper dictionary to see where the biggest groups of rhymes lie. So a browsing and a search for large sets of rhymes would be helpful. For those of us with poor vision, a large font option would be appreciated. The menu items need not be larger since one tends to memorize what they say, but it would be nice if the words themselves were optionally larger. The words are printed on the screen in two columns. It looks like there is room for three columns. The number of columns could be an option.

Summary

The QL-Rhymes program is an effective aid for getting rhymes. It is easy to use. It does require the Pointer Environment (not provided) and Toolkit 2. Follow the manual exactly when configuring. It is probably better to use the parameter string option rather than CONFIG. QL-Rhymes was written by Geoff Wicks and is available from Just Words for 15 EUROs. Thanks are given to John Terry who inspired and tested the program.

Printers and other Sicknesses

Marcel Kilgus

Sooner or later most QDOS/SMS-ers are confronted with the problem: how do I get my printer working? This was not entirely simple in the past, but it has now become a major problem. The trend in printers is firmly in the direction of "Idiot proof". Formerly printers had a degree of intelligence needed to transfer the ASCII codes in their different fonts to paper, but today's price wars in the printer market have made it a habit to reduce the art of printing to the intelligence of a crisp bread by leaving the individual printing task completely to PC software. This new art of peripheral production is summarised by the term "GDI", which means Graphics Device Interface and designates the graphics subsystem of Windows. The consequence is that printers work with Windows, but not, for example, QPC. I am often surprised by mails from customers who assume their emulator has printer problems. The reality is that QPC, just as any other QDOS/SMS Platform, is only able to access the printer port, and it is the software that has to speak the right language.

The ideal solution for the QPC user would be for the emulator to have a built in ESC P/2 to GDI interface, because all QL software can generate ESC P/2 and when that is converted to GDI it would be possible to use every conceivable Windows printer.

This has two drawbacks:

- 1) I would have to do the entire work (and I know little about printers and their languages and
- 2) the solution is strictly for the QPC only.

Drawback one is the knockout punch as I have no time to do it, but drawback 2 is also important.

What are the alternatives?

It is often worthwhile to take a look at the Linux world, which, as the main "Windows Competitor", often has similar problems. Many will have heard of "Postscript". Postscript is a page description language mostly used in laser printers. Some will be familiar with PDF data files and these are related to Postscript, which is not surprising because both standards come from the same producer - Adobe. Under Linux Postscript is now the favourite printer language, just as ESC P/2 has become for QDOS/SMS. At the moment few home printers can use Postscript, because large memory and a good processor are necessary. The solution is Ghostscript, which understands Postscript and comes with many drivers for different printers. It renders the page (similar to GDI protocol) using PC software and sends it via special drivers to the printer.

"Good", people say, "but does that get us any further?", to which my answer is a definite "yes". Ghostscript is not just for Unix/Linux. There is a Windows ported version, and surprise, surprise, also for QDOS/SMS thanks to Jonathan Hudson. To cut a long story short it is possible with a little effort, a small program called Redmon and the Windows version of Ghostscript to link with the QPC's PAR interface without having to change a single line of QPC code. The nice bit is that GS for Windows comes

with a GDI driver, which can address all Windows printers. I have tested it and now have a Postscript capable QPC. If you send Postscript code to the parallel interface it prints correctly on any Windows printer. Given that there is now a QDOS/SMS version of Ghostscript, this is also a possibility for modern QL compatibles. I have not explored that further, but it is something for people who are interested.

Fine, I now have a parallel port with a uniform language for every printer. But the problem is not completely solved, because I know of no QDOS/SMS program that produces Postscript code. Well, I do know one. It is called MPS, was originally developed for Linux and which I compiled for a small experiment for QDOS/SMS. This program converts ASCII data into Postscript and can be used, for example, as a filter for QD. The result is I now have a QPC on which QD will work with all printers! It is a small, but not bad beginning.

If people are interested in going this way (and I can honestly see no other way) then all applications will have to be provided with a Postscript output. In the long term direct Postscript implementation would be the best (above all for ProWesS since from a single implementation all its applications would have Postscript), but in the short term an ESC P/2 to Postscript convertor in a device driver would be a solution. In that way all programs, including older applications would have access to the newest printers. And the nicest thing for me would be that almost anyone could do the work, as it would be completely independent of QPC. If someone feels confident of doing this and wants to make a contribu-

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
<http://www.j-m-s.com/smsq/index.htm>

QPC2 Version 3

Marcel did it again: major improvements to QPC! A long list of new features, which we won't list here, so please ...

see "News" in this issue!

To avoid confusion: If you already own QPC2 (Version 1), then the upgrade is NOT free! The naming is a bit confusing (we should have called it QPC3, but now it is too late, isn't it?), so here is the complete price list:

Upgrade to QPC2 Version 3

From QPC1 (or just QPC - the first QPC!)	£45	EUR 71,53
From QPC2 Version 1 (or just QPC2)	£29	EUR 45,97
From QPC2 Version 2	£13	EUR 20,40
(if bought/upgraded after 15. Oct. 2001)	£4	EUR 6,08

You will get a new manual (the QPC part)! Please send in your master disk for an upgrade. Please note: QPC1 will not be supported on this disk anymore! If you want to continue to use QPC1, please make yourself a backup copy of the master disk (you should do this anyway).

QPC2 Version 3 Full version £83 EUR 127,31

TERMS OF PAYMENT

Postage and package [Germany] EUR 4,86 (if total value of goods is up to DM 50,- then only EUR 3,32).

[Europe] £4,50 / EUR 7,41 (if total value of goods is under £15 / EUR 25,56 then only £3 / EUR 4,86).

[Overseas] between US\$7.50 (1 item) and US\$17.50 (maximum).

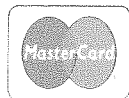
Cheques in EURO and International Money Order in EURO (both have to be drawn on German banks).

We accept Internation Reply Coupons as payment (can be bought at any post office): 1 IRC = EUR 1.02 redeeming value.

Exact prices are given in EUR ... the price in £ and/or US\$ are approx. conversion based on the exchange rate at the time of creation of this ad. Prices may be adjusted in case the exchange rate falls considerably out of this range in both directions!

We can charge your credit cards in £'s, US\$, EURO or in DM - please state the currency you prefer.

Please do not send any UK bank cheques in £ - our banks have increased the fee for handling them by 600% (no joke!) so we cannot accept them anymore, unless you add £6 for clearing the cheque. E&OE.



tion to the QDOS/SMS world, then it could be worthwhile to take a look.

Now, I think I could at least ask for a short overview (this article was written under time pressure). I would like to have a reaction, preferably in the next issue. If you have questions they can be asked through the usual channels. My little MPS

port is also available for your own experiments.

Addendum

Since the print of the original German article many months have passed by and one thing actually changed: the mentioned ESC P/2 emulator for QPC is currently under development by a 3rd party person.

This, however, is a pure QPC only solution. And though I'm very grateful for this development, the "proper" (and on all other platforms "only") way of doing it is still the one mentioned above. The conclusion remains, if you have further questions feel free to contact me. This could really be a very important step in the QDOS/SMS development.

Printers and Obsolescence

Geoff Wicks

My desktop is slowly disappearing. In the back right hand corner is the 486 QXL PC on which this is being written. Working anticlockwise there is a bubble jet printer, a few pens and pencils, a laser printer, a scanner, an overflowing in-tray, my laptop from which I shall send this article to QL Today, the PC keyboard and mouse. Somewhere in the middle is a mass of disks, the electronic equivalent of my overflowing in-tray.

Ideally to make more room I would like to get rid of the bubble jet. It has served me well for over 7 years, but I do not like ink jets. Their cartridges are expensive, they run out of ink too quickly and the ink is soluble. The laser printer would do me fine, but, alas, I cannot (yet) use it on the QL. All QL laser printer drivers date from 3 - 4 years ago and the world has since moved on.

Anyone who has recently tried to buy a printer for their QL will know how difficult it is to find a compatible one. There is no systematic advice on QL compatible printers other than the rule of thumb to look at the top range of Epson. There are dan-

gers in this policy, because no one knows how long Epson printers will remain QL compatible. People with experience of Epson printers are discovering that ESC P/2 is still being developed, and it is becoming increasingly difficult to discover the precise changes.

It was therefore with some interest that I heard that Marcel Kilgus had written an article for the German QL Today on using Postscript and Ghostscript to print from QPC on practically any Windows printer. Marcel kindly dug out the article from the depths of his hard disk and sent it to me. I felt the article deserved a wider publication and Marcel has agreed to an English translation appearing in this issue of QL Today. It even has an up to date addendum containing an interesting snippet of news!

Marcel's article started me thinking about the concept of obsolescence as it affects the QL. He suggests two ways of solving the printing problem, one of which could be used only by QPC. The other should be adaptable for the Q40/Q60, and this is important now we have the news that the Q60 is

likely to go into production. Both solutions leave original black box hardware users in the cold.

Now the controversial bit. In the QL world we have made great efforts to ensure downward compatibility, and most of our software has remained useable on all hardware. We have never planned QL obsolescence, although obsolescence has occurred. Who now uses a 128K black box with microdrives? We assume that all QL users have disk drives, Toolkit 2, memory expansion and often a fast processor. What we can no longer guarantee is a compatible printer for the original hardware users. Has the time now come for planned obsolescence?

There are two subjects of for discussion, and these should be kept separate, because I would not want my deliberately provocative suggestion to detract from Marcel's constructive view of solving QL printer problems. Is his vision the way we should be going, and how difficult would it be for software authors to adapt their programs for his preferred Postscript option?

There is scope for a healthy debate on both topics in future issues of QL Today.

QLTdis – part 6 / 1

Norman Dunbar

Last time I started with the first 16 instruction families and gave you the code for disassembling them. At the end of that article I mentioned that all the sub-routines were not included, so you wouldn't be able to actually run the code – this month, I will not be giving you the type 17 through type 31 decoding routines, because I am having a few problems with a couple of them and may need to modify my own code yet again (!) before I turn it loose on my readers.

So, this month, I'm going to give you the code for the various sub-routines which you will need for the last chunk of code and for the next one – when I eventually get things sorted out.

So, without any further ado, here come those sub-routines. These get added at the end of the file DISS_ASM and when we get the remainder of the type decoding routines tested, we can insert those between the last lot of code and this installment. That way, your file will match up to mine and the 32 bit offsets from the jump table to the code routine won't overflow.

First up, we have the condition code decoder. This is called whenever we have to decode the condition codes in a Scc, Bcc or DBcc instruction. Most of the condition codes are two characters long, but the first two are only one, so we test for those two first, and treat them as exceptions. Anything other than 'T' or 'F' is read from the table as two characters.

```
*-----
* Condition codes routine
*-----
cond_code  cmpi.b  #0,d0          ; Special case, zero ?
           bne.s   cc_1         ; No, try one
           moveq   #'T',d4      ; Yes, add 'T' to buffer
           bsr     str_add_b
           rts                ; Done

cc_1       cmpi.b  #1,d0          ; Special case, one ?
           bne.s   cc_rest      ; No, do the rest
           moveq   #'F',d4      ; Add 'F' to buffer
           bsr     str_add_b
           rts                ; Done

cc_rest    movem.l d0/a3,-(a7)   ; Save working registers
           lea     cc_table,a3   ; Table of condition codes
           lsl.w   #1,d0         ; Multiply D0 by 2 to get offset
           move.w  0(a3,d0.w),d4 ; Get word from offset into table
           bsr     str_add_w      ; Add it to the buffer
           movem.l (a7)+,d0/a3   ; Restore working registers
           rts                ; Done

cc_table   dc.w    "T "        0 = T => Special case – never read from here
           dc.w    "F "        1 = F => Special case – never read from here
           dc.w    "HI"       2 = HI
           dc.w    "LS"       3 = LS
           dc.w    "CC"       4 = CC
           dc.w    "CS"       5 = CS
           dc.w    "NE"       6 = NE
           dc.w    "EQ"       7 = EQ
           dc.w    "VC"       8 = VC
           dc.w    "VS"       9 = VS
           dc.w    "PL"      10 = PL
           dc.w    "MI"      11 = MI
           dc.w    "GE"      12 = GE
           dc.w    "LT"      13 = LT
           dc.w    "GT"      14 = GT
           dc.w    "LE"      15 = LE
```

This next routine adds the instruction size to the buffer, so if we are currently decoding something like 'MOVE.W D0,D3' this routine adds the 'W' part of the instruction to the output buffer. As a useful side effect, it sets D5W to 1, 2 or 4 which corresponds to 'B', 'W' or 'L' for later use in decoding the effective address.

In case you wonder why I am 'fudging' the output buffer if there is a dot in it already, it stops the output looking like 'MOVE.W D0,D3' and saves me having to remove a whole lot of dots from the op-code strings table. OK, I admit to a bit of bad design in this case – but for the purposes of the magazine, it was easier to fudge this one routine rather than have to reprint a lot of the op-code table again.

```

*-----
* Size decoding routine.
*-----
size_decode move.w  d0,d5      ; Copy op-code
                bsr     size_d0  ; Extract size part of op-code
                exg     d0,d5    ; Swap op-code & size over

*-----
* At this point D5 = 1 for byte, 2 for word and 4 for long. We need to check for
* 1 and adjust (temporarily) to zero to get the correct text from the table.
*-----
                move.w  d5,d4    ; Copy the size specifier
                cmpi.w  #1,d4    ; Byte sized instruction ?
                bne.s   sd_dot   ; No
                clr.w   d4       ; Yes - adjust back to zero

*-----
* If the final character in the buffer is a dot, remove it first !
*-----
sd_dot         cmpi.b  #'.',-1(a5) ; Do we have a dot ?
                bne.s   sd_text  ; No, skip
                subq.l  #1,a5    ; 'Remove' the dot
                subq.w  #1,d6    ; Adjust the counter too

sd_text        lea     s_table,a3 ; Table of sizes
                move.w  0(a3,d4.w),d4 ; Fetch the correct size specifier
                bsr     str_add_w ; Add to the buffer
                bsr     space    ; Followed by a space
                rts           ; Done

s_table        dc.w   '.B'
                dc.w   '.W'
                dc.w   '.L'

```

These two routines convert the source or destination register bits in the op-code into a digit in D4 ready to be added to the buffer.

```

*-----
* Source Register routine.
*-----
src_reg        move.w  d7,d4      ; D4 is where we want to be
                andi.w  #$07,d4  ; Bits 0 - 2 = source register
                bsr     reg_no   ; Add to buffer
                rts

*-----
* Destination Register routine.
*-----
dest_reg       move.w  d7,d4      ; D4 is where we want to be
                andi.w  #$0e00,d4 ; Bits 9 - 11 = dest register
                lsr.w   #8,d4    ; Shift down
                lsr.w   #1,d4    ; 8 is the maximum remember ?
                bsr     reg_no   ; Add to buffer
                rts

```

When we decode one of the many bit operation instructions, the next routine extracts the data from the op-code and adds the correct instruction text to the buffer. This is called from BTST, BCHG, BSET and BCLR decoding routines.



Who knows what a new year will bring? At Just Words! we are positively buzzing with creativity and have plans for new commercial and freeware products.

One of our ideas is so esoteric that, as far as we know, it has never been done before. We don't even know if it's possible, but we'll have a try.

Back down on earth we have good news for our European customers. You can now pay us in Euros through our Dutch Postbank Account. Now let's hope Bolkie is successful in pushing down international bank charges!

QL-THESAURUS
STYLE-CHECK

£10 or €15 each

QL-2-PC TRANSFER
QL-RHYMES



Geoff Wicks, 28 Ravensdale, Basildon, Essex, SS16 5HU, UK.

Tel: +44 (0)1268 281 826 Email: geoffwicks@hotmail.com

Web: <http://members.tripod.co.uk/geoffwicks/justwords.htm>

Just Words! - Software for Writers and Word Lovers.

QBOX-USA BBS



Operating since 1993 on a Sinclair QL from Utica, Michigan, USA

Supporting ALL Sinclair and Timex users

Message and File Areas for QL, Z88, Spectrum, TS2068, ZX81, TS1000

Modem speeds 300 bps to 33.6k bps supported

24 hour operation - call anytime

810-254-9878

```

*-----
* Bit operation routine.
*-----
bit_op      andi.w  #$c0,d0      ; Mask out all but bits 6 & 7
            lsr.w   #4,d0       ; Bit 7 -> bit 3
            lea    bits_table,a3 ; Table of bit op instructions
            move.l  0(a3,d0.w),d4 ; Collect a long from the table
            bsr    str_add_1     ; Add to the buffer
            rts                    ; Done

bits_table  dc.l    'TST '      Note trailing space !
            dc.l    'CHG '
            dc.l    'CLR '
            dc.l    'SET '

```

Next we have a simple routine which extracts the size bits from the op-code and returns D0W as 1, 2 or 4 to signify 'B', 'W' or 'L' accordingly.

```

*-----
* Get size bits in D0
*-----
size_d0     andi.w  #$c0,d0      ; Keep only bits 6 & 7
            lsr.w   #5,d0       ; Move right 5 bits
            tst.w   d0          ; D0 = 0, 2 or 4
            bne.s  size_exit    ; Not zero
            moveq   #1,d0       ; Size is 1 for .B

size_exit   rts                    ; Done

```

These next two routines are called from the MOVEM decoding process to extract a list of address or data registers from a bitmap in D4.B. They look so sweet and innocent, don't they? Wait till you see 'big brother' the innocently named 'reg_list' routine - that is a nightmare and I'm sure it will be a candidate for reworking when I get around to testing the MOVEM decoding.

```

*-----
* ADDRESS Register list routine.
*-----
addr_reg    tst.b   d4          ; Nothing to do if D4 is zero
            beq.s  dr_exit     ; Thought so !
            movem.l d0-d4,-(a7) ; Save the workers
            move.b  d4,d0       ; Reg_list expects the mask in D0.B
            moveq   #'A',d2     ; Doing address registers
            bsr.s  reg_list     ; Call register list routine
            movem.l (a7)+,d0-d4 ; Restore the workers

ar_exit     rts

```

```

*-----
* DATA Register list routine.
*-----
data_reg    tst.b   d4          ; Nothing to do if D4 is zero
            beq.s  dr_exit     ; Thought so !
            movem.l d0-d4,-(a7) ; Save the workers
            move.b  d4,d0       ; Reg_list expects the mask in D0.B
            moveq   #'D',d2     ; Doing data registers
            bsr.s  reg_list     ; Call register list routine
            movem.l (a7)+,d0-d4 ; Restore the workers

dr_exit     rts

```

Here comes trouble! This routine is not my own work. Well, I did translate it into assembler from the original SuperBasic by Andy Pennell. I suspect that it will work, but only serious testing will determine this. Be prepared for a reworking if I find any problems. What this does is takes the register list bit map in D0.B and converts it into something like 'D0,D1-D3,D5,D7' in the output buffer.

QUANTA



Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

Now in our Nineteenth Year

Further details from the Membership Secretary

Bill Newell, 213, Manor Road

Benfleet, Essex, SS7 4JD

Tel. +44(0)1268 754407

or

Visit the Quanta Web Site

<http://www.quanta.uni.cc>

email: quanta_membership@uk2.net

```

*-----
* Register list routine. Due to the technical difficulty of this routine, I am
* commenting it slightly more than the rest of the routines.
*-----
reg_list moveq #8,d1 ; Bit number counter initialised to 1 + highest bit
        lsl.w #8,d2 ; Move 'A' or 'D' to high byte. D2.W = 'A0' or 'D0'
*-----
* The start of the main loop. Test the current bit of the bitmap and if set
* carry on processing, otherwise jump to the end of the loop and see if we are
* finished. If not finished, we will end up back here.
*-----
rl_530 subq.b #1,d1 ; Adjust for the correct 'next' bit number
        btst d1,d0 ; Test that bit
        beq.s rl_650 ; Bit not set so skip to test for end of main loop
*-----
* The current bit in the bitmap is set so we have a register to process.
* Subtracting the bit number from ascii '7' gives the correct register number.
* The register type (in D2) is OR'd into the register number to get the correct
* register name. This is then added to the buffer.
*-----
        moveq #'7',d4 ; Assume R7 register
        sub.b d1,d4 ; Subtract the bit to get the correct register number
        or.w d2,d4 ; Mask in the register type 'A' or 'D'
        bsr str_add_w ; Add register to buffer
*-----
* If we have not just processed bit 0 then we still have more to do and must
* skip over the next bit of code which deals with the end of the string.
*-----
        tst.b d1 ; Reached bit zero ?
        bne.s rl_532 ; Not yet
*-----
* We have processed bit zero so add a slash and skip to the bit that removes it
* WHY IS THIS REQUIRED ? Andy Pennell, are you there ?
*-----
rl_531 bsr slash ; Add a spare slash
        bne.s rl_650 ; Then remove it again !!!
*-----
* Having found a set bit, we must look for others. If the next bit down is CLEAR
* then we need to append a slash to the buffer. If we have reached bit zero then
* we terminate the string with '-R7' as we obviously have a list of registers.
*-----
rl_532 subq.b #1,d1 ; Down to the next bit
        btst d1,d0 ; Test it
        bne.s rl_531 ; Next bit is clear - add a slash
*-----
* Next bit is set also but are we at bit zero now ? If not, keep looking for
* more bits.
*-----
        tst.b d1 ; Done yet ?
        bne.s rl_570 ; No - keep looking for the next CLEAR bit
*-----
* We have reached bit zero so our list is completed by adding the end of the
* list to it.
*-----
        move.l #$20037,D4 ; Sets D4 to 'x-x7' where 'x' = binary zero
        or.w d2,d4 ; Add in the 'D' or 'A' - D4 = '-D7' or '-A7'
        bsr str_add_3 ; Add '-D7' to buffer
        bra.s rl_done ; And finished - get out of this routine
*-----
* Look at the next bit down. If set continue scanning
*-----
rl_570 subq.b #1,d1 ; Next bit
        btst d1,d0 ; Test it
        bne.s rl_572 ; Set so continue scanning

```

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost..... **£24**

*****NEW*****

superHermes

A major hardware upgrade for the QL
All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software) **£90** (£92/£93)
IBM AT UK layout Keyboard..... **£11** (£13/£15)
Serial mouse **£8** (£8.50/£9)
Capslock/scrollock LED **£1** (£1.50/£1.50)
Keyboard or mouse lead **£3** (£3.50/£3.50)
High speed serial (ser3) lead **£4** (£4.50/£4.50)

Hermes available for **£25** (£26/£27) Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead) **£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send **£3** (+£5 for manual if reqd).

Send disk plus SAE or two IRCs

MKL...**£40** (£41/£43) MKII...**£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£40/£41)
4mbytes RomDisq..... **£65** (£66/£67)
8 mbytes RomDisq..... **£98** (£99/£100)
Aurora adaptor..... **£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost **£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors) **£40** (£43/£44)
4 amp total (for motors etc) **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..... **£25** (£27/£28)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting..... **£30** (£31/£32)

Temp probe (-40°C to +125°C)..... **£10** (£10.50/£11)

Connector for four temp probes..... **£10** (£10.50/£11)

Data sheets..... **£2** (£2.50/£3)

Control software & manual (for all I/F) **£2** (£2.50/£3)

QL SPARES

Keyboard membrane **no longer on sale**

1377 PAL **£3** (£3.50/£4)

Circuit diagrams **£3** (£3.50/£4)

68008 cpu or 8049 IPC..... **£8** (£8.50/£9)

8301/8302 or JM ROM or serial lead..... **£10** (£10.50/£11)

Power supply (sea mail overseas)..... **£12** (£17/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address, postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details

10 May 01

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: 01442-828254

Fax/BBS: 01442-828255

tony@firshman.demon.co.uk

http://www.firshman.demon.co.uk

```

*-----
* Aha, a clear bit has been found after a (number of) set bit(s) so we need to
* close our list of contiguous registers. This time the bit number in D1 is the
* bit number of the CLEAR register so we subtract from '6' and not '7' to get
* the correct register number for the last set register in the list.
*-----
    move.l    $$2d0036,d4          ; Sets D4 to 'x-x6' where 'x' = binary zero
    sub.b    d1,d4                ; Register number in ascii D4 = 'x-xn'
    or.w     d2,d4                ; Add the register type D4 = 'x-Dn' or 'x-An'
    lsl.l    #8,d4                ; D4 = '-Dn ' or '-An '
    move.b   #'/',d4              ; Assume more registers. D4 = '-Dn/' or '-An/'
    bra.s    rl_650                ; Check if we are done yet.

*-----
* Still haven't found a clear bit - but are we now at bit zero ?
*-----
rl_572  tst.b    d1                ; Done yet ?
        bne.s   rl_570            ; No, scan again

*-----
* We are now done. Bit zero has been reached. We must close the list of
* registers.
*-----
    move.l    $$2d0037,D4         ; Sets D4 to 'x-x7' where 'x' = binary zero
    or.w     d2,d4                ; Add in the 'D' or 'A'
    lsl.l    #8,d4                ; D4 = '-Dn ' or '-An '
    move.b   #'/',d4              ; D4 = '-Dn/' or '-An/'

*-----
* This is the end of the main loop. At this point - which we arrive at from
* numerous places - we could have just processed bit zero and if so, we want to
* tidy up and get out of here. On the other hand, if not then we want to carry on.
*
* If we are not yet processing bit zero, then go back and process the next bit.
*-----
rl_650  tst.b    d1                ; Are we at bit zero ?
        bne.s   rl_530            ; Do next bit - we are not yet finished

*-----
* Our output buffer has a '/' too many and is one character too long so
* adjust the buffer pointer (A5) and the buffer length (D6) before exiting
* The output buffer now contains the full register list.
*-----
    subq.w   #1,d6                ; Adjust size of buffered string
    subq.l   #1,a5                ; And output buffer

*-----
* All done, bye bye
*-----
rl_done rts                        ; Get out of here quick !

```

A simple one now, and one which was the subject of a challenge in the QL news group some time back. How to reverse the bits in a register word. This is needed because the bit maps used for the MOVEM instruction are the 'wrong' way round for one form of the instruction, and rather than code two nightmare routines like the above, I decided to swap the bits over and use a common decoder.

```

*-----
* Reverse bits in D2.W routine.
*-----
swap_d2  moveq   #15,d4           ; Using d4 as the counter
rev_one  asr.w   #1,d2            ; bit 0 -> X flag
        roxl.w  #1,d3            ; Move the bits of D3 left by one bit AND
*                                     ; then copy the X flag -> bit 0 of D3
        dbra   d4,rev_one        ; Do the other (15) bits of d2
        move.w d3,d2            ; D2 is now reversed
        rts

```

Next issue, we continue with the decoding of the effective addresses.

Hints and Tips

Q. Is it possible to generate Euro currency characters in Line Design?

A.1. With suitable fonts, yes. Phoebus Dokos has produced some fonts which contain the Euro symbol at CHR\$(181) – the same location as SMSQ/E – and these can be displayed by using

```
PFshow gstate, whatever
draweuro
PFshow gstate, whatever
```

Maybe someone finds this useful. Wolfgang Lenerz (www.wlenerz.com)

```
DEFine PROCedure draw_euro
rem print approx. Euro sign
rem this presumes that 2 global variables exist:
rem   - gstate
rem   - charsize : CURRENT character size
LOCAL box(3),xs,ys,wid
wid =charsize DIV 12
IF NOT wid : wid=1           : rem make line width
PFShow Gstate, 'C'         : rem show a 'C'
get_box Gstate,box         : rem get max Font sizes
ys=box(1)/5                 : rem magic!
PFWidth Gstate,'C',xs      : rem get xsize of C char
xs=xs-1
PFPathMethod Gstate,1
PFLineWidth Gstate,wid
IF charsize<10:wid=.5
PFMoveR Gstate,-xs-1,-ys-.5 : rem now draw the two lines
PFLineR Gstate,xs,0
PFMoveR Gstate,-xs,-wid*2
PFLineR Gstate,xs,0
PFMoveR Gstate,1,ys+.5+wid*2
PFPathDraw Gstate          : rem and show them
END DEFine draw_euro
:
DEFine PROCedure get_box (Gstate,box_size)
rem gets the size of a fontbox into the box_size array
rem passed as param
LOCAL xo,yo,xs,ys
PFFontBbox Gstate, xs,ys,xo,yo
box_size(0)=xs
box_size(1)=ys
box_size(2)=xo
box_size(3)=yo
END DEFine get_box
```

the CTRL SHIFT u key combinations on UK keyboards. The \paragraph\ entity name when entering text seems to produce this character, though I am not sure if this was intentional or not.

A.2. (from a reply by Wolfgang Lenerz on the QL Users Email Mailing List):

Whilst waiting for the Prowess Euro fonts, I have devised a small basic proc (using Proforma extensions) that draw a euro. I use it as follows:

FOR SALE:

QL with a few microdrive cartridges, etc. Offers invited from a good home (seems a shame to chuck it away!).

Enquiries to Andy Cawley (Doncaster area, England) on 0786-482005

QDT: Time for User Input

Jim Hunkins

The QL DeskTop program (QDT) has the major goal of being an enhancement to the user experience. With this in mind, it is necessary to get an insight as to how different users use their QL and as to their program preferences.

With the summer release date rapidly approaching, now is the time that I need everyone's input. A series of requests follow. I would appreciate it if everyone could take a few minutes and respond to the following. An email address is supplied with the questions.

I am also pleased to mention that a couple of people have contacted me and offered their assistance with programs and in other ways. This brought up the subject of policy for contributors. To avoid boring everyone here, I have added to my QDT website www.jdh-stech.com a policy statement which outlines rules for programs that are used and/or linked to with QDT along with the policy for those who wish to contribute programming, translation, graphic design, and other assistance.

While I am talking about the websight, don't forget to keep an eye on it. I try to update the progress chart every few weeks and will be putting up more screen dumps as they develop. It also has email links for giving me feedback as the project continues.

Here are the questions that I need your assistance with. Please send your responses and a copy of your boot file to:

development@jdh-stech.com

It would also be helpful if you would include the word SURVEY in your subject. And thanks for the help!

1. QDT uses different programs for defaults to handle different files. The intent is to have a set of free programs and a set of cost programs. In the case where the cost program is found it

would be the default; else the free program would be the default. Please list both your favorite free and cost program for the following:

- a) text editor
- b) graphics displayer
- c) HTML (web browser)

2. QDT will automatically try to set up your initial desktop if requested. In order to do this, along with supporting as many programs as possible with the Tabbed Notebook Configuration capability, I need to know what programs you normally use (along with what category you would place them in). For example, you might use LineDesign and would probably call it a graphics editor. Please list all the programs that you use and their category.

3. QDT will eventually supply a printer object capability.

- a) which printer(s) do you use
- b) do you use/have loaded ProWesS
- c) do you ever print in color?

4. QDT will need to do some minor modifications to your boot file. QDT will also scan the boot file to help it set itself up properly for your normal environment. Please answer the following questions and send a copy of your boot file(s).

- a) do you have more than one boot file; if so, how do you select which one?
- b) do you call other basic setup type programs/files from your boot file?
- c) is your boot file always present on your disk
- d) is your boot file always loaded?

That is the end of the questions for now. Please don't forget to send a copy of your boot file(s). Also, please let me know if it would be okay to contact you with questions by email as to your system usage (I will not use your email address to send you anything else, unless you specifically ask me to). And thanks for making QDT happen!

Programming the Pointer Environment in C - Part 1

Jerome Grimbert

QLToday ran a series of articles about programming Prowess from SBasic, but what about programming the Pointer Environment in another Language. Nobody has written about

this yet so I present here an article on Programming the PE in C. Before starting, you will need the C68 compiler installed on our favourite system. To be able to use the example you need the Xmenu library available from

(<http://ql.grimbert.cjb.net/> and <http://www.imagnet.fr/~godefroy/>).

Once C68 installed, it is pretty easy to do the compilation.

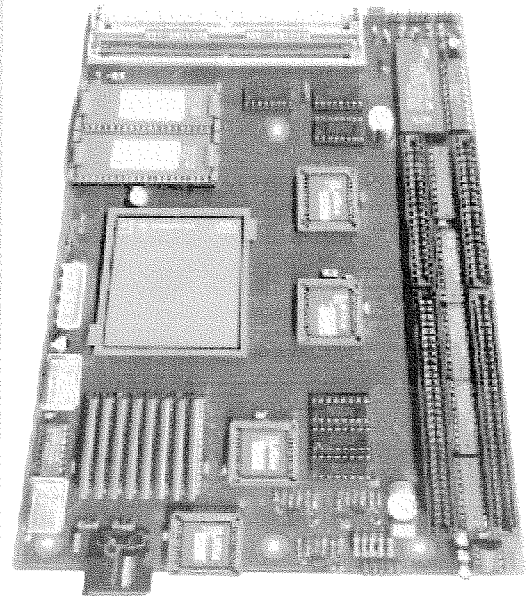
Q60

Q40

The powerful Sinclair QL successor

Have you ever dreamt of QLs that are more than 100 times faster, have many thousands of colours, high resolution graphics, stereo sound, fast harddisks and modern peripheral capabilities? New complete computers that run your QL software and open the door to unreached 68060 speed and advanced applications? Here they are!

- x Q40i : 68040 CPU, 40 MHz, Math. Coprocessor, MMU
- x Q60/66: 68060 CPU, 66 MHz, Math. Coprocessor, MMU
- x Q60/80: 68LC060 CPU, 80 MHz, MMU
- x 68060 superscalar architecture, dual execution units
- x Up to 160 BogoMIPS performance for QDOS+SMSQ/E
- x 4 to 128 MB RAM, PS/2 module sockets
- x 256 to 1024 kB ROM
- x Highspeed 32 bit graphics, with original QL modes
- x 65536 colours at 1024 x 512 pixel resolution
- x Multisync monitor output
- x PC Keyboard interface (DIN)
- x 20 kHz Stereo sound
- x Battery buffered clock, 2 KB nonvolatile RAM
- x Controller for 2 IDE harddisks or CD-ROM
- x 2 Serial ports with 115200 Baud, Parallel port, Joystick port (on IO card supplied with mainboard)
- x Hardware extension slot supports ISA cards
- x Fits directly into Minitower or other standard case
- x +5V / +12V power supply
- x No tinkering, no parts from original QL needed
- x Mainboard size 8.2 x 6.3 inch



The Q40i and Q60 give you the choice of three different operating systems:

- **SMSQ/E** - The QDOS compatible, but faster and more modern operating system. The first choice for most Q40 and Q60 users. Allows you to run the majority of QL programs, has a much improved BASIC. SMSQ/E supports a lot of the advanced Q40 and Q60 hardware capabilities, including full **highcolor graphic drivers** and sampled sound !
- **QDOS Classic** - A multitasking operating system with good compatibility for older QL programs. Includes harddisk and sound support. It was the first operating system for the Q40. Full 68060 support.
- **68k Linux** - The free Unix-like multiuser / multisesion operating system. The Q40 and Q60 are the first and only QL style computers that run Linux. Linux on the Q40 and Q60 offers you an endless amount of application software, including X Window System, full Internet access, graphical Webbrowser, CD-ROM writing, MP3 player. There is a special **Q60 Linux distribution** with graphical installation tool on CD-ROM.



The Q40i and Q60 are offered as mainboards or cased systems by:

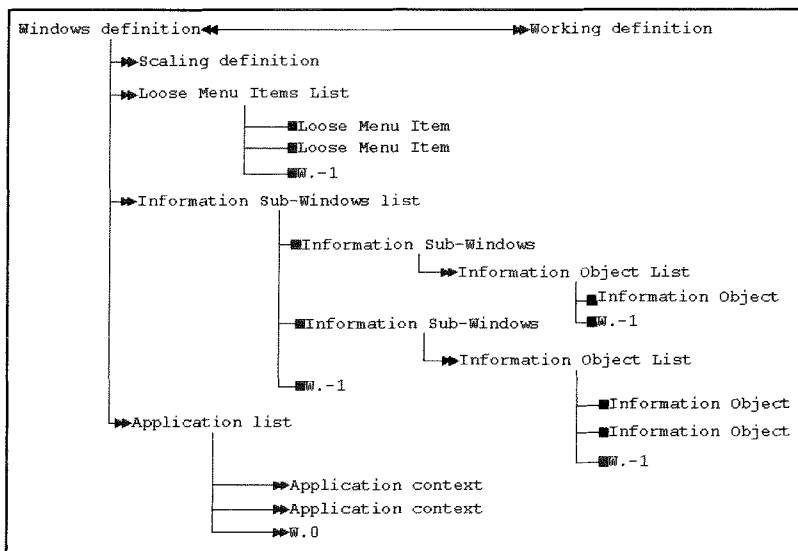
D & D Systems, P.O. Box 5813, Ripley, Derbyshire, England DE5 9ZR
Tel: +44 (0)1773-740170, FAX: +44 (0)1773-748399, Email: sales@q40.de

More information about the Q40i and Q60 at the official Website: <http://www.q40.de>, Email: info@q40.de

Just to describe my installation, I have put all the program files in win1_bin_, the include files in win1_bin_include_ and the library files in

win1_bin_lib_.

In the figure, I tried to represent the various objects needed by the PE.



Some explanation of the symbols used on the figure: The double triangle is for a pointer. The square is for a part of the structure. For instance, the windows definition has a pointer to a "loose menu items list". This list is a contiguous memory region in which we find a repetition of "Loose Menu Item". Every list is ended with a special word value (-1 or 0) which is right after the latest element.

Do not be scared by the apparent complexity of the figure, we are now going to explore it slowly. It will also give us the time to detail each part.

So, first, let's just see what's the minimal information needed and what it does. The minimal is a simple window, and it is done with only two structures: the windows definition and the working definition.

The "working definition" is filled by the PE when the window is displayed, so all we have to usually do is simply have enough memory for it and to set a pointer back to the "window definition".

The "window definition" is just really that: it's a structure which defines the size of the windows, the colours used and keeps pointers to all the objects of that window.

The "scaling definition" defines at least the minimal window's size, the maximal window's size as well as the allowed stepping. The minimal window's size is the smallest size the window can be shrunked, and of course the maximal size is the largest it can be. If you do not want any resize, setting the minimal and maximal size both to the initial size of the window is the simplest thing to do.

The stepping is the amount added or subtracted (according to the resize direction) on each axis, unless the value reaches either the minimal or maximal, in which case the limiting value is used.

Let's see that on a very simple program:

```
#include <stdio.h>
#include <stdlib.h>
#include <qdos.h>
#include <qptr.h>

/* for new C68, hide startup problems */
struct WINDOWDEF _condetails =
    { (char) 0, (char) 0, (char)0, (char)4 ,
      (short)2, (short)1, (short)0, (short)0};

char _conname[] = "con_2x1a0x0";
/* mask startup problems, for old one */
char *_endmsg = NULL;
/* and stop when I say */

char _PROG_NAME[] = "PE in C tutorial 1";
```

That's just the beginning for a small file. First we include some useful things (that's pretty standard for C), and then we perform some settings so the our program will look ok once started (We hide the default windows, and we set the program name). That's also nearly standard for PE program in C.

```
static QD_TEXTI(quit, "QUIT");
static QD_TEXTI(title, "PE in C test 1");
```

We want some label, so we need to declare them. For this small first program, we only want a title and a button to quit.

```
static long ACTION_QUIT(struct WM_wwork *wwk);
struct WM_action action_quit =
    { JSR, wm_actli, ACTION_QUIT};

static long ACTION_QUIT(struct WM_wwork *wwk)
{
    exit(0);
}
```


Here comes the tricky part of PE in C: writing the wrapper for the various actions. Because the PE expects machine code routines, it's a little strange to fill a data structure with what will be executable code, but it works fine. There is a full

explanation in the C68 docs about that if you want more details.

So, so far, we have a correct setting, two text strings and an action/callback.

```
struct WM_wstat * init_status(struct WM_wwork *wwp)
{
    struct WM_wstat *result;
    /* Default struct has 40 loose item, that's enough */
    result=(struct WM_wstat *)malloc(sizeof(struct WM_wstat));
    result->wwork = wwp;
    wwp->wstat = result;
    result->wdef = NULL;
    result->chid = 0;
    result->swnr = result->xpos = result->ypos = 0;
    result->kstck = result->kprs = 0;
    result->evnt = 0;
    result->xsiz = result->ysiz = result->xorg = result->yorg = 0;
    result->ptpsx = result->ptpsy = 0;
    result->wmode = FORM_QL4; /* THIS is important */
    result->spar0 = 0;
    result->spar1 = 0;
    result->ciact = NULL;
    result->citem = -1;
    result->cibrw = result->cipap = result->cispr =
    result->cihxs = result->cihys = result->cihxo =
    result->cihyo = 0;
    result->litem[0] = 0;
}

```

This is just a function which creates and fills the Working definition. It will be called later, but it is a good thing to handle it in a modular way. On the same way, having a function which

creates and fills the Windows definition is also a good approach, because it is the only thing to update when changing the design of the window! So, let's have one too.

```
struct WM_wwork * init_window()
{
    struct WM_wwork * result;
    struct WM_litm *loose_list;
    struct WM_infw *infw_list;
    struct WM_info *info_list;

    info_list=(struct WM_info *)malloc(sizeof(struct WM_info)*2);

```

We need TWO information objects, one is the title and the other is just for the end of the list. We assume that there is no memory problem. There is now a lot of fields to fill.

```
    info_list[0].xsize=14*6;
    info_list[0].ysize=10;
    info_list[0].xorg=0;
    info_list[0].yorg=0;
    info_list[0].type=TYP_TEXT;
    info_list[0].spar=0;
    info_list[0].attr.t.ink=0; /* Should be from a config block */
    info_list[0].attr.t.cwid=0;
    info_list[0].attr.t.chgt=0;
    info_list[0].pobj=&title;
    info_list[1].xsize=-1; /* end of list */

    infw_list=(struct WM_infw *)malloc(sizeof(struct WM_infw)*2);

```

We also need TWO Informations Sub-Windows. The first one will contains the title, and the second is yet another end of list.

```
    infw_list[0].xsize=14*6;
    infw_list[0].ysize=10;

```

```

infw_list[0].xorg=2;
infw_list[0].yorg=3;
infw_list[0].flag=0;
infw_list[0].borw=0;
infw_list[0].borc=0;
infw_list[0].papr=0324; /* Should be from a config block */
infw_list[0].pobl=info_list;
infw_list[1].xsize=-1;

loose_list=(struct WM_litm *) malloc(sizeof(struct WM_litm)*2);

```

Also, we need TWO Loose Menu Items. As usual, the last one is just the end of the list.

```

loose_list[0].xsize=4*6;
loose_list[0].ysize=10;
loose_list[0].xorg=15*6;
loose_list[0].yorg=3;
loose_list[0].xjst=0;
loose_list[0].yjst=0;
loose_list[0].type=TYP_TEXT;
loose_list[0].skey=K_CANCEL;
loose_list[0].pobj=&quit;
loose_list[0].pact=&action_quit;
loose_list[0].item=0;
loose_list[1].xsize=-1; /* end of list */

result = (struct WM_wwork *) malloc(sizeof(struct WM_wwork));

```

Now that we have all the small parts we would need, it's time to organise them for the PE in the Windows definition.

```

result->wstat=NULL; /* filled later */
result->chid =0;
result->pprec=NULL;
result->psave=0;
result->spar1=0;
result->spar2=0;
result->spar3=0;
result->pullid=0;

```

That was just some fields to fill. Next is something usually with more customisation: The sprite to use for that window and the original size (as well as position from the mouse pointer).

```

result->splst=NULL;
result->xsize=20*6;
result->>ysize=16;
result->xorg=20; /* initial position of mouse */
result->yorg=8;

```

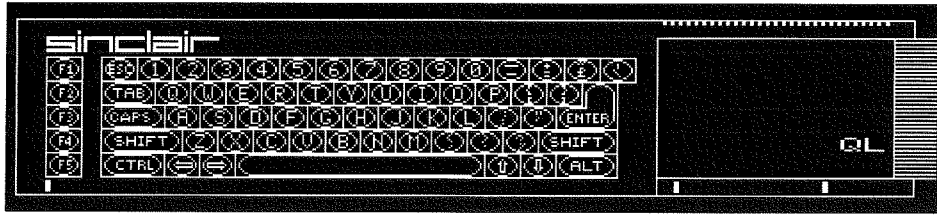
And then, still more fields to fill. The following are related to the aspect of the windows (colour, border, shadow and such things).

```

result->flag=1;
result->borw=1;
result->borc=0; /* Should be from a config block */
result->papr=7; /* Should be from a config block */
result->sprite=NULL; /* default pointer */
result->curw=1;
result->curc=2; /* Should be from a config block */
result->uback=4; /* Should be from a config block */
result->uink=2; /* Should be from a config block */
result->ublob=NULL;
result->upatt=NULL;
result->aback=7; /* Should be from a config block */
result->aink=0; /* Should be from a config block */
result->ablob=NULL;
result->apatt=NULL;
result->sback=0; /* Should be from a config block */
result->sink=7; /* Should be from a config block */

```

DILWYN JONES



QL P.D. SOFTWARE LIBRARY SERVICE

A brand new PD software library service from QL author Dilwyn Jones. Hundreds of freeware, shareware and PD programs available on floppy disk at a price of just £1.00 per disk (or just 75 pence if you supply the floppy disk). Programs are normally supplied on HD disks unless you specifically request DD disks.

CATALOGUE

To obtain a free catalogue of the entire software library:

1. Send me a formatted HD or DD disk and return postage, or
2. Download the catalogue from my website (see address below), or
3. By email – just send me an email to ask for the catalogue as a text file.

CD-ROMs

(Also available from Q-Celt Computing in Ireland)

QL EMULATORS CD **£5.00** - All of the available QL emulators!

QL PD-CDR **£5.00** - Gérard Plavec's software collection!

LINE DESIGN CLIPART CD **£10.00** - Huge collection of Line Design clipart!

FAMOUS FACES CD **£10.00** – Line Design pictures of the famous!

QL RELIGION CD **£10.00** – A collection of bibles, religious texts, clipart etc

DJ PD LIBRARY CD **£10.00** – The entire PD library on a CD! Start your own!

The CD-ROMs are supplied in QXL.WIN format on an ISO-9660 CD-R for use with QXL, QPC, QemuLator v2, uQLx, Q40, Q60 (systems able to read QXL.WIN media)

POSTAGE UK/Europe: add £1.00, Rest Of World: add £2.00.

PAYMENT In UK Pounds Sterling currency. Cheques payable to DILWYN JONES.

Dilwyn Jones, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT, U.K.

Email: dilwyn.jones@dj.softnet.co.uk

Website: <http://www.soft.net.uk/dj/index.html>

Note: Although this service is run by the editor of QL Today magazine, it is a completely separate venture which has no connection with the publisher of this magazine. Software is supplied free of charge, apart from copying and media charges etc to cover costs, and without any warranty other than to the replacement of defective media (e.g. "bad or changed medium" errors etc.). In other words, usual PD library terms. E & OE.

```

result->sblob=NULL;
result->spatt=NULL;

result->help=NULL;

```

And now, we are simply linking our small objects into that big windows definition. There is one Sub window, which uses one information object, one loose item and no application window.

```

result->ninfo=1;
result->ninob=1;
result->pinfo=infw_list;

result->nlitm=1;
result->plitm=loose_list;

result->nappl=0;
result->pappl=NULL;
}

```

It's time to have a main which really does something.

```

main()
{
    struct WM_wwork *wwp;
    struct WM_wstat *wsp;

    /* text may become a problem if started in mode 8
    ** but the high_color has no problem to work like mode 4
    ** So, to simplify our setting, if it's mode 8, then
    ** this program will want to run in mode 4... assuming
    ** that all hardware who support mode 8 also have mode 4
    **
    */
    short mode, type;
    mode = -1;
    type = -1;
    mt_dmode(&mode, &type);
    if (mode == 8) { mode = 4; mt_dmode(&mode, &type); }
}

```

Now the problem with any text is resolved, we can call the two functions in order to fill the structures.

```

/* Create the window structure */
wwp = init_window();
wsp = init_status(wwp);

```

We are nearly ready, all that is missing is a window channel and checking that the PE is here. (Of course it is!).

```

wwp->chid = fgetchid(stdout);
if (!(wm_findv(wwp->chid)))
{
    exit(-1); /* there is no PE here, so stop */
}

```

Ok, it's time to really make something on the screen. We want a primary window, right under the mouse.

```

/* show the window, at the mouse pointer */
wm_prpos(wwp, -1, -1);
wm_wdraw(wwp);

```

And now, the best part: an infinite loop which will read the event and dispatch them. Actually, we care only for the 'QUIT' loose item, and that is taken care directly by the PE. So There is really not a lot to do.

```

/* Now, let's get the events */
while (!wm_rptr(wwp))
{
    exit(0);
}

```

That's all. It's pretty long to start, but then adding new objects is rather easy.

PE in C test 1 QUIT

The QL Users Mailing List

Bruce Nicholls

Dotted throughout QLToday, past and present, are references to The QL Users Mailing list but what exactly is this magical list?

The list was started in 1997 by Bruce Nicholls who once ran Quo Vadis Design selling QL Software and also published a magazine called QReview. The list was setup for any discussion

related to QL/SMS be it news, help, queries, for sale etc. The list now has over 150 active members including most of the QL Traders. A mailing list is a method by which people who have the same interest can exchange information with each other via computers. In the past this was done via bulletin boards but now the communication utilises the most common electronic communication known as email. Email is used in preference to other methods as it accessible by most people who either have personal internet access or work email. You can even use the QL

JOCHEN MERZ SOFTWARE

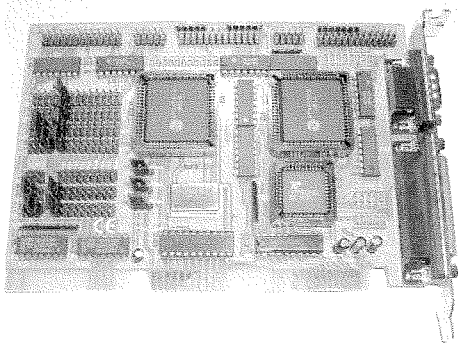
Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
<http://www.j-m-s.com/smsq/index.htm>

Very Special Offer - rare item!

ISA-Card with 4 serial ports (FIFO) and 3 parallel ports.

Ideal for Q40 and Q60 (they act as SER1 ... SER4 and PAR1 ... PAR3) - no plug and play - all IO addresses and IRQs can be jumpered (large jumper fields on the left of the board). Including all cables!

As only one item is available and QL Today reaches some readers fairly late, I decided to sell it to the highest bid. Minimum bid: EUR 30,-. Bids by post and Email to smsq@j-m-s.com. Payment ONLY by credit card (customers with a bank in Germany can pay by account debit). Bidding will end March 15th 2002. The highest offer will get the card. Postage as described on ad on page 35 applies.



to send and receive emails! The mailing list works by people subscribing to the list and then sending an email to the mailing list address which then sends the message out to all the other people who have subscribed (joined) the list. People can then respond to the email message either directly to the person who sent the email or to the mailing list thus promoting discussions in real time. A lot of people on the list are able to monitor the list throughout the day and you may get an answer to a question/problem within a few minutes.

How to subscribe

To subscribe to the list you need to send an email to the address:

majordomo@nvg.ntnu.no

with the following in the body of your email:

subscribe ql-users

Your request then needs to be approved by the list owner before you can start sending emails. This process was added as the list was being abused by people sending spam emails to the list. In the joining welcome message you will get all these instructions together with the address you need to send emails to.

How to Unsubscribe

To unsubscribe from the list you need to send an email to the address:

majordomo@nvg.ntnu.no

with the following in the body of your email:

unsubscribe ql-users

If you need to unsubscribe an oldaddress from a new email account you need to send an email with the following in the body of your email:

unsubscribe ql-users oldaddress@???.com

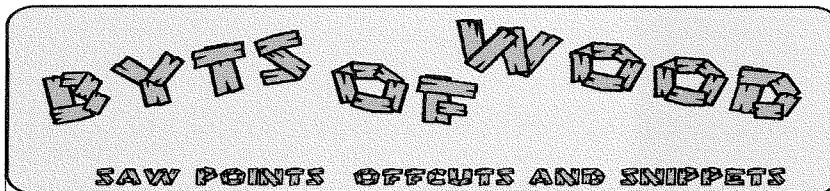
If you need help with any of the above send an email to

owner-ql-users@nvg.ntnu.no

and the list owner will try to help.

QL-Developers List

There is also a list called ql-developers which was going to be just for commercial developers but the needs changed and this list is now for the discussion of Q40/Q60 Linux plus Hardware. To join this list just substitute QL-Users in the above examples with QL-Developers.



It is customary, at this time of the year, to look back on what the last year has brought us and to look forward to the coming year, either in anticipation or in apprehension. Leaving aside the political events, upon which this is not the place to comment, the year for QLers has been quietly uneventful.

Those who work hard as a matter of course have been just as productive as ever but there has been precious little activity from other sectors of the community. We seem to be relying on a diminishing group of individuals to provide all of

our software and hardware and that cannot be a very healthy situation.

Queue here for QPC2 version 3

Marcel has been working very hard as usual and QPC 2 has now reached version 3 with many improvements over the previous version. Most of these will be documented elsewhere in this magazine so I will not go too deeply into the changes. I am writing this using it and, since a large part of my working life is devoted to wrestling with the various pro-

blems that Windoze throws in your face, I can only say that I wish that other Windoze programs were as stable. There is, as usual, the vexed problem of printing but I will get into that a bit further down the line. I will return to the subject of QL printers in the next column.

Planting a Ceed

Thierry Godefroy, in spite of a long spell under water in the Indian Ocean towards the end of last year, put in his usual good work with the release of the first versions of the CD drivers. Unusually, these were incredibly cross platform devices which would enable use by people with either Q40/60s or QL/Aurora users with a Qubide attached. In their current form they need 'QXL

Tools' to access the drives but these are very much the early seeds from which many directories may flower.

Duncan Neithercut was pretty swift off the mark with QCDEZE. This is a pointer driven front end for the CD drivers which will be a very welcome piece of code. I have downloaded this file but QL Today deadlines have not given me any time to play around with it so I will have to postpone my report on that until the next missive. As I have been writing this there have been a few emails from users who have had success in using them.

The same goes for Duncan's latest version of CSB (Clip Scrap Board). Both of these files are available from Thierry Godefroys website and BBS should you feel moved to try them out. I hope to bring a better report in the next column. If Duncan's past record is anything to go by then they should be well worth a look.

The Rhyme of the (not so) Ancient Lexicographer

Geoff Wicks has also been taking his doggerel for a walk. He released the wrything dictionary - or should that be.... what was the HOTkey for that spelling crib program of his? Seriously though this is another neat piece of work from the Just Words house which is both useful and fun. Now what rhymes with QL? Oh to hell with it. Well done Geoff.

The Flame Still Glows

Nasta has kept the embers of the GoldFire going throughout the year with numerous postings onto the QL Users list. The project may seem like vapourware to many of you out

there but it is still taking up a lot of Nasta's time and, given the support of the users and the enthusiasm of those of us who would very much like to see it appear, it may achieve corporeal form soon. As I have mentioned before the software for it will be a different matter but we'll burn that bridge when we come to it.

A Spoonful of Medicine

The TCP/IP / QL internet camp has been very silent for a while and I hope this does not mean that the trail has gone cold. This remains one of the 'Holy Grail' products for many QL users who have not succumbed to the lure of the PC. Progress was, at first, dizzyingly fast and there have been emails sent out and received by users of fairly standard QL systems. Although I have a pretty powerful PC system I would still like to see the QL making more of the online world if only to take part in the debates which rage along the QL - User's mail list. Let us know where you are Jon (apart from Switzerland that is).

SMSQ/E

Progress in SMSQ/E has been slow in the last year but progress in this department has often stalled to be revived again later. Tony Tebby's time is always short and there is precious little financial remuneration to be found in writing QL code. There are still some problems I would like to see solved. The latest version (2.99) still seems to have problems with DD drives on my Aurora system and vexed question of the caches on the Q40 / Q60 platforms is yet to be addressed. It has, however, made some big leaps since it's inception and there are more things to be seen, I am sure.

A New Hard Man on the Block

Those of you who read the QL Users list on the internet will have noticed the flurry of mail that came about when a new name appeared. Dave Parks has been working with Nasta on new designs for QL Hardware. He has been suggesting a combination of the new version of Qubide that Nasta and Tony were working on a while ago with an Ethernet card. This would be very welcome because he wants to make it compatible with the Q40 / Q 60 system. An interesting trick if it can be pulled off. It seems from his writing on the users list that he has the determination and the knowledge to do these things and is providing welcome support to Nasta who has had to further his visions for QL expansion alone for some time.

I agree with those who say that the QL bus is the bottleneck for both the QL and the Aurora but for many people a complete new system is beyond their reach and a partial upgrade could improve the performance of their ageing systems whilst freeing some of the other expansion devices for the second user market.

The Modern Way

One thing that has pre-occupied me over the last year, given the declining numbers of QL users, has been the concept of user desire. Given that most QL systems are not, by any stretch of the imagination, modern systems in their functionality and capabilities what can we do to improve matters? Windoze may be a hated system for many of my readers but one thing it has done is to introduce standardised hooks into system functions so that

the programmer only has to know how to pass the information to the routine he wishes to use and how to interpret the output from it. This has freed the program writer from endlessly having to write file manipulation, printer and screen control procedures and allowed him to work on the core of the program itself. The QL programmer has long had some of these features available to him but some have refused to use them. There may be a number of reasons for this including a dislike of the way that the functions are implemented, the fact that some of them are not free to the user, and a basic dislike of the Pointer Environment in general but, by not creating and accepting standards for programming we have created a system which is far more hotch potch than it really needs to be.

What's on the Menu?

Jochen Merz's Menu Extensions for example - I know I have harked on about this before - provide the best, fastest and easiest way to access files for loading and saving. OK there is a small charge if you want include the extensions with the program distribution itself but you do not have to include them if you don't want to. If you want to add a hook to the files part of the Menu Extensions you only need to add two or three lines to your program and you can allow the user to decide if he has them or not and configure the program accordingly.

This one simple example would help many people who have to laboriously type in file names for saving and loading. There are many other areas in

which the system could be standardised like this and the user would see a fairly harmonious set of menus and commands to do the basic tasks.

Time for QDT?

As I mentioned above one great hope for me was the demonstration by Jim Hunkins of his QDT concept. For once there seemed to be some hope that we could integrate the programs into a system with a proper desktop feel to it. There have been attempts at this before.

The ICE system was any early desktop shell system giving both pointer access and a suite of utilities such as calculators, calendars etc. The user was able to configure the desktop to add his own programs and to produce an individual system. This fell by the wayside through lack of support both from programmers and users. If it had succeeded and we were all using it would Jim be writing 'Iced QDT'?

Cueshell also set out to do a similar thing although it lacked the ability to put programs on it's desktop and was seen by many as just another file manipulation tool it does have a lot of other features as many of you who tried it on the demo disk that came with this magazine will have found out.

QDT is, however, a whole new approach and one which could pull together many of the disparate aspects of the QDOS/SMSQ scene. Jim's article in the last QLT was a taste of what it looks like and what it will do and certainly whetted my appetite for the real thing. Maybe this year's US show will see a functioning demo - eh Jim?

Old Software for New Users

A New user appeared on the QL Users list recently. He had managed to obtain a QL and a Super Gold Card and was trying to get the system into shape. As always it is hard to get someone started on a system when you are so familiar with it. We may all have a lot of knowledge to impart but it is the order in which it is passed on and the bits we leave out because we think they should be obvious that can sometimes leave a neophyte dazed. One thing that struck me was that, when he asked what software he should use for word processing, spreadsheets, databases etc. he was advised to try Xchange. Now I have nothing against exchange as such and, in its day, it was a remarkable piece of software. My main thought was that we should not introduce new users to the 21st Century world by getting them to try 80's software. If we are to keep people and get them to really use the system then we should stress the modern aspects of it. By all means give them Xchange but get them to try the demo programs like QSpread, DATA-design, Cueshell and others so they can see what is possible. We may lack the orchestras of bells and whistles that the PC has but we can still blow our own trumpet to a good tune.

A Broken Rainbow

A long running saga came to an end last month. Way back last year we supplied a second user Aurora to a customer in Germany. This came from a complete working system so I was fairly confident it was a fully functional board. A few weeks later the customer contacted me to say that it would not work.

In a flurry of emails we tried all the things we could think of to get it working. We investigated the Gold Card in case it had not been modified correctly and all of the other possibilities. The board, however, stubbornly refused to display anything but half a startup screen. IT did not help that the customer had little english and my poor grasp of German was not up to a technical discussion - I barely even manage to converse with my father in law.

Anyhow we got the board back with all of the bits. I plugged it all up and he was right - it did not work. I then plugged in another Aurora I had using the same peripherals. It worked! OK so the board was duff. I then changed all of the plug-in chips. No change. OK. For some reason I decided to plug in a Super Gold Card. That worked - now I was completely baffled. Plug the Gold Card back in - it does not work. Three hours later and I had tried everything. I modified another Gold Card and that did not work either. I was lost - time to call in a expert. I dropped the whole kaboodle in to Keith Mitchell.

Keith was puzzled at first but finally found the answer. One of the diodes on the board had been fitted the wrong way round. This gave no problems with a Super Gold Card but was a complete disaster when used with a Gold Card. A quick flip of the soldering iron and all was well. We returned the board and got a big thank you from the customer.

The Republic of Fasgro- lia

Many of you may recall the acronymbatics that i have collated here in the past. As always these thing fascinate me

with their ability to be re-interpreted. I unwittingly used one of these in a conversation with Steve Hall once and he said, 'That's a TLA' 'A What?', I asked. 'Three Letter Acronym', he said.

The Guardian Newspaper in the UK had a couple of interesting and amusing definitions in its IT section. I pass these on to you with any representation of they're having been born of my fevered brain.

Flyspeck 3

- Text that is too small to read.

Janet and John 20

- Text that is too large to read comfortably

and, of course;

FasGroLIA

- Fast Growing Language of Initialisms and Acronyms.

We are all citizens of the republic of Fasgrolia whether we want to be or not. Just examine your own conversations.

Honourable Mentions in Despatches

This issue the award must go to all of you. We have made it through another year and the QL is still going and still, for many of us, an interesting machine. There will always be slack times and busy times. These days the hardware people seem to be very active and that should lead to an improvement in the software section. Lets hope so. We may have lost a few people this year, some have gone on to the PC or LINUX and some, sadly, have died. I will miss Tim Fuller who was always interesting when he called.

Most of all though the Most Honourable Mention must go to the people who have ma-

naged to produce this magazine for so long on a no profit basis. We will soon enter our seventh year of publication and, during those years Dilwyn and Jochen have worked tirelessly to get this magazine out on time and with such a good blend of articles. Bruce Nicholls has also played a pivotal role in keeping us on track so, if you see any of them at a QL show, shake their hand and tell them how much you appreciate the work. Every little helps.

And Finally - as they say on the News.....

Censusless

Those of you who have an interest in geneology could not have missed the shambles that followed the UK governments posting of the 1907 census results on a UK website. For those of you in foreign climes the outcome was that the site was completely overwhelmed by visitors and had to be closed down.

Now it may be slightly off topic but was there not another census around this time of the year some time ago? This census was also heavily oversubscribed and people had to be turned away. No Internet in those days so the people came in person of course. Should we be looking for evidence of a cybernetic second coming? Reports of a woman giving birth while trying to log on to the census site? Or on a stranded train from Virgin Express?

If you get an email from three wise men or a bunch of shepherds doing a wildcard search for some sheep all using the * key don't say I didn't warn you.

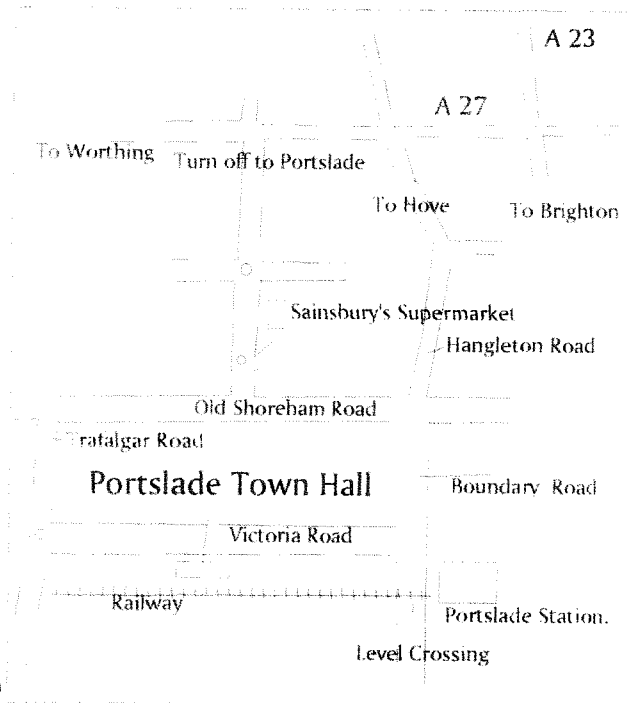
Dot.com all you faithful.....



The QL Show Agenda



Hove Workshop - (UK)



3rd March 2002 at Portslade Town Hall on the south coast (same venue as last time).

10am to 5pm. Good parking and easy access from Portslade station. This is the seventh year we have run this show and after the successful change of location last year we use my local Town Hall again. This is a much larger venue with better parking and easier access. I have also arranged a bevy of local ladies to do some catering so the food and drink situation should also be better.

We expect all of the usual QL culprits to be there. Hotel details will be available in the flyer which will be sent out in February. If you have not returned the postcard from the QL 2000 flyer please contact Tony Firshman at T.F. Services to make sure you are on the mailing list.

See you all there. Roy Wood - Q Branch.

QL Meeting - (NL) Eindhoven

Saturday, 23rd of March, 10:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

Quanta AGM and Workshop - (UK)

Manchester. A 2-day event on Saturday 13th and 14th April 2002, Venue: 3rd Davyhulme Scout Headquarters, Conway Road, off Lostock Road.
Public from 2pm Sat and from 10am Sun. AGM 2.00pm Sunday
For more details see previous issue of QL Today.

North American US Show 2002

Saturday, 1st of June

This time, the event takes place near Washington.
For details, please check page 6 of this issue.