QL Today

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

0° QL World

Contents

Advertisers

in alphabetical order

Quanta's QL 2000 meeting will be the biggest event this year for the QL, if you are attending the show, it is a time to put faces to the names we read/speak about. A report on what a Quanta/QL meeting is like is given by Darren Branagh in this issue. A full report from the QL 2000 meeting, no doubt, will be in the next issue and I hope the two day event will be a turning point in deciding the support we give the QL for the future.

Several long running projects are now nearing completion and should be demonstrated at the show these include the much anticipated TCP/IP software to allow an ordinary QL(If there is such a thing) to interact easily with the outside world including email and www browsing. Jon Dent, who is doing the development, recently sent the first ever email from a black box QL to the outside world more details are given in the news section. The colour drivers are being used on the QXL and Q40 and with the new QPC2 soon to be released development of software to exploit them is happening as detailed in this issue. One unexpected new development from Richard Zidlicky is the availability of GCC (GNU C Compiler) for the cross compilation of programs for QDOS. This could make the porting of C programs from other operating systems an easier task.

In this issue Norman Dunbar gives us full details of his new -shock- horror- windows program, no this magazine isn't about to change it's name to Windows Today, it is just a very useful program for those of us stuck when we need to read some Quill documentation but are without any means of using a QL or emulator and have a PC in front of us. What I will say is don't put the icon for the program in a prominent place or you may get some very strange looks.

My thanks go out to all our regular band of authors for this issue and hope you find it informative.

Finally congratulations are due to Dilwyn Jones on his marriage together with his well earned rest from the QL. I would also like to thank him, on behalf of all QL Today readers, for the time he spends on QL Today to make it a reality.

Sinclair C8000
Alternate
Timeline
Investigation
Unit 3

"Hmm, it seems the civilisation died out because they had to keep re-installing their operating systems. They had no Time left for any real work."
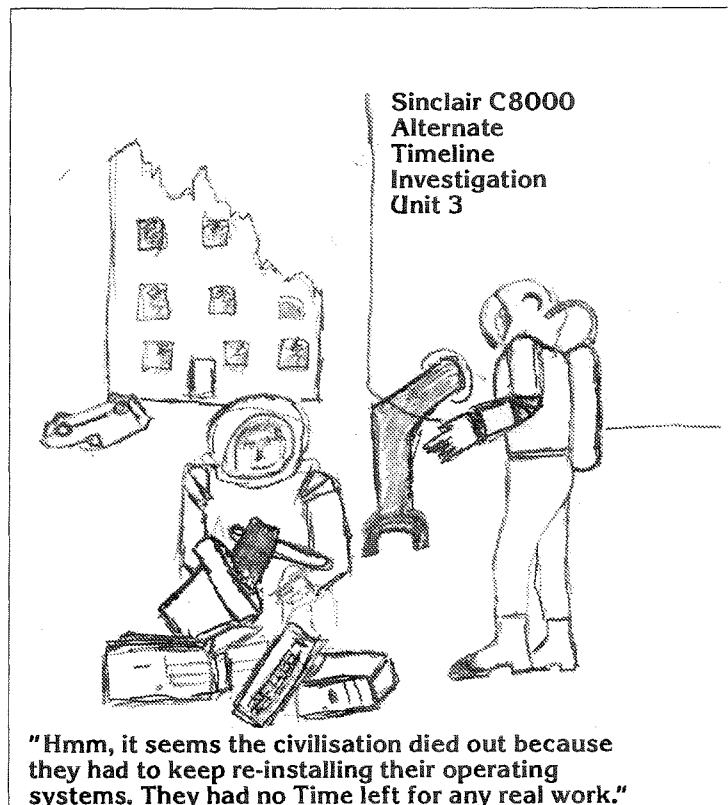
NEWS

First QL TCP/IP Email Sent

Jonathan Dent has sent the very first TCP/IP Email from a QL system Aurora SGC with superHermes:

```
-----Original Message-----
From: jondent4@mythen.ch
[mailto:jondent4@mythen.ch]
Sent: Thursday, August 17, 2000 8:28 PM Subject:
This is a test mail
Regards
Jonathan
```

Jonathan is currently writing the documentation for the implementation and following is a summary of his implementation:
The function soqlStartup() uses a private TRAP#3 driver to let the main TCP/IP job know that a job (user program) is about to use TCP/IP via the soqlLibrary. soqlStartup() passes a pointer to a soqlLink_t object as a parameter with the io.open call, and suspends the job until the main TCP/IP job has completed the initialisation of the object. The main TCP/IP job can restart the user program because soqlStartup() put the job number in the object previously.
After this soqlStartup() does an io.close and all further communication uses the soqlLink_t object. The rest of the soqlLibrary functions use the soqlLink_t object to access code in the main TCP/IP job and can where necessary initiate actions from the main TCP/IP job. An example of this is the local DNS resolution. Any synchronisation which is necessary can be done by the main TCP/IP job suspending and releasing the user program.
The main TCP/IP job also deals (in the present version) with sending data to the Internet. Another job deals with receiving data. This is linked to the main TCP/IP job in a similar way to user programs. Further jobs can be linked in for future enhancements such as PPP and a DNS resolver.

Congratulations go to Jonathan for making this a reality.

PROGS has moved

The new address is
  Mechelbaan 344
  2580 Putte
  Belgium

New telephone number will be +32 15 22 23 26, but not connected yet. He can always be reached by email (PROGS@triathlon98.com) or mobile +32 496 250345

Quanta News

We are delighted to announce the reformation of the East Anglian sub-group of Quanta.
It will meet on the 2nd Sunday of the month in central Norwich. The meetings start at 2pm and end at 6pm, or earlier if we prefer.
At first, while attendance is building up, the meetings will be at a house at 21 Rowington Road Norwich, it is hoped that meetings can move to a local room/hall. Rowington Road is behind the Grove Road shops, which is near Sainsburys on Queens Road. Rowington Road is also near the Bus station and St Stephen's Street.
The first meeting will be Sunday the 10th of September 2000.
Contact: Tarquin Mills 01603-470399

First QL (successor) with Ethernet support

Claus Graf and Richard Zidlicky report that the Q40 using Linux now supports some hardware ethernet cards directly. This allows the Q40 to be part of Intranets and so on. Richard and Claus have tested ethernet based networks with Q40, PC and Apple Macintosh and have been able to browse W3 pages, downloaded software via ftp and use telnet sessions, for example (Q40 as web-, ftp-server). All this is possible with the new Q40 Linux kernel 2.2.6-2. SMSQ/E driver is not available (yet?).
More on the Q40 and Ethernet adapters can be found at http://www.q40.de

RWAP Software - latest News

Our apologies to anyone hoping to see us at the QL 2000 show (Portsmouth) - continued ill-health makes it highly unlikely that we will be able to attend. Please note that overseas customers can now update their software by sending the original disks plus 3 IRCs per program. UK users need to send £1 plus the original disk.
QL Genealogist has now been updated to v3.26. This is to cope with one minor bug which has been found. However, users of this version will find that it is now much easier to Change the parents and spouse details for any given individual. We are looking at improving the DEX program - any suggestions would be appreciated. QL Cosmos has now been updated to v2.03 which overcomes the fact that v2.02 did not work

on all SMSQ/E versions.

One or two people have commented that QL Cash Trader does not work under SMSQ/E. This is possible under v3.7 provided that you have at least SMSQ/E v2.98 - this is due to a disastrous bug in the SMSQ/E machine code vector conversion routines.

We have now released Image D (cost £10) - a program to allow you to create three dimensional views of objects, including hidden line, shading and wireframe techniques. Perspective and magnification can be controlled and multiple objects defined and positioned relative to each other.

Rich Mellor

Web-News and Updates

Norman Dunbar is collecting documentation on the inner working of QDOS and is publishing them on his web site:-

http://www.Dunbar.cwc.net/qdos/qdos.html

Thierry Godfrey wrote:

For those who didn't notice yet, some new and updated software is available from my Web site (http://qdos.cjb.net/english/download.html).

1. New:

- libmcrypt v0.1.0: a strong cypher algorithms library for C68.
- libmhash v0.8.2: hash functions library for C68.

2. Updated

- MicroEMACS v4.00 (08/08/00 release): better tags handling.
- ctags v4.0.3a: this tags file generator now understands S*BASIC and SSB (Structured Super BASIC: hi Tim ! ;-)
- (bug fix): FileInfo II v3.41

My Web site is also undergoing a massive update.

While doing the layout, more news arrived:
Thierry Godfrey has released a new MicroEMACS version (13/09/00) which is available from his web site at http://qdos.cjb.net/download.html The changes are:

- Added the vertical scroll bar.
- There is no more screen flicker when moving around using the page/line up/down loose items.
- Added the missing "arrow down" character into the italic font.
- Added the ‹UNDO› (or ‹PAUSE› for QXL+ SMSQ/E v2.98+) key binding.
- Changed the patch level to 06 into the "about" menu (I forgot to do so on 08/08/00 !).

Dilwyn Jones wrote:

I have now added an update (v3.05k) to Mark Knight's version of Editor 2000 onto my Web site. This fixes a bug which could crash some systems due to the release of Turbo used to compile it, according to Mark. He has also sent me a new version of Heartbeat (v3.01) for inclusion on the same page.

I've also added Norman Dunbar's latest minor update to Winback v2.22 which now includes a note concerning the use of Joinfile on SMSQ/E to restore and join together large files backed up across several floppy floppy disks. Norman's Black Hole device driver is also there, and the QL Net links page includes the link to Norman's new QDOS website.

Meanwhile, Darren Branagh's QL Windows Screen Saver has been added. This is a Windoze screen saver program with a QL feel - basically, he has "stolen" some QL pictures from my website to make a screen saver for Windoze which displays pictures of Super Gold Cards, Qubides, QLs etc (makes my PC look much better!!!). Beware though, it has a picture of Darren in it somewhere. If anyone does download this 1.4MB file, can you let me know if it works, as I had some, err, fun creating the self-extracting zip file and I'm not sure if it works on all PCs or just mine!

Dilwyn Jones
dilwyn.jones@dj.softnet.co.uk
http://www.soft.net.uk/dj/index.html

Just Words! - Style-Check

I have made some additions to the Style-Check data base. The new style_dbf file (9K) can be downloaded from my web page.
http://members.tripod.co.uk/geoffwicks/justwords.htm
Happy checking! Geoff Wicks.
geoffwicks@hotmail.com

Darren Branagh reports on FreeDOS compatible with QPC1

The following came about after the London Show, when Roy Wood actually had to lower himself to buying a copy of MS-DOS 6.22 I was selling for a fiver on my Q-CELT Stand, as he needed it to install QPC for a customer!

I thought that there must be some way of using QPC without Bill Gates benefitting - and there is, its called FreeDOS!

It is a totally free, completely rewritten 100% compatible DOS clone, that has nothing to do with

Microsoft - you can download it at:
www.freedos.org
And I've tested QPC1 on it over the last 3 or 4 days, and although still a beta 5 version, it has worked flawlessly - although disk access is a little slow...
darrenbranagh@hotmail.com

Richard Zidlicky

reports that he has succeded in using gcc (a common unix/linux c compiler) to crosscompile for QDOS. He has diffs to gcc-2.95.1-core to produce a crosscompiler for qdos. It works well enough to compile a working version of c68. It uses c68 libs, as68 and linker.
For more information contact
Richard.Zidlicky@stud.informatik.uni-erlangen.de

TF Services

New from TF Services is Compswitch. It is a 4-way UK trailing socket with a 2m lead. From first appearances it looks like a standard extension socket until you look at the side.
One socket nearest the cable is marked 'Computer' and the other three are marked 'Switched'. A current taking device (computer, light, hifi etc) is plugged into the 'Computer' socket. The remaining 3 sockets then mimic the on/off state of this control device. ie switch on the 'computer' and the remaining devices (sound system, scanner, monitor, modem etc) switch on, together with an LED.
When the 'computer' switches off, then all the other devices switch off.
This can be used with a QL with a switch in its power line, assuming the wall switch is inconvenient.
It can also be used with a QL in a tower case, so that the main case switch acts as a master switch. It can also be used with current IBM ATX computers, which have a soft power-down. In this case switch-off is automatic when the computer powers down. It can also be used with room lights that are connected to the ring main, or 'real' hifi systems, where all components are separate.
The only limitation is that the 'computer' must not take more than 500 watts, and total switched capacity must not exceed 100 watts.
It is not suitable for computers than have a low current consumption - ie laptops.
Cost is £24 inclusive.

New Game

Jerome Grimbert was busy again. He writes:

I now have a Q*Bert game in high colour for my Q40 in 1024x512 (it should also work with GD2 256 colours), but this is not tested).
Available at: **http://grimbert.cjb.net/**
It is about 40K zip (give you a 600 K program)
The text file is an old one made for mode 8.

Domains for sale

Ludovico Camarda has the following domains for sale:
   SinclairQL.com
   SinclairQL.net
   SinclairQL.org
The relative "under construction" pages are already on line. Feel free to make an offer if you are interested.
You can reach Ludovico via email:
lcamarda@esoc.esa.de

Editor: We suggest that, before you make an offer, you better make sure that the Name SinclairQL is free from any copyrights - so that you don't end up in a situation where you paid for the domain and are forced by the copyright holder to return the domain to him. These things have happened several times in the past. It is fairly unlikely that any "Sinclair" or "QL" copyright holder would complain, but the possibility exists.

More MicroEmacs

Just 12 hours before QL Today will be going to the printer, Thierry informed us about even more changes to MicroEmacs. There are too many details to list in the remaining two lines but if youtakee a look at his website, you'll quickly find out what's new.

Helpline

Q. I have a Falkenburg hard disk system on my tower cased QL system. When I try to DELETE a file, it tells me it deleted the file OK, in fact DIR WIN1_ seems to imply it has been deleted, but the next time I switch on the QL, it says the file is still there. What is wrong with the system?
A. These hard disk systems have a WIN_FLUSH command to force file operations to be written out to disk. What is happening is that the file is being deleted from the slave blocks in memory but the changes not written back to the hard disk immediately. If you issue a WIN_FLUSH command before switching off or resetting the QL this will force the deletion to be written to the hard disk itself and fix the problem.

Trader travel experiences

Tony Firshman

Croatian QL Show - epilogue

You may remember the Croatian QL show almost two years ago now. You almost certainly though will not have participated, because it was in a very small room, so we had to turn away most of the 500 visitors («grin»).

On the way Roy Wood, driving my car, had a horrific accident on black ice in Belgium. My insurance company refused to pay out as they said the following clauses actually said no-one except me could drive my car on business:

Entitled to drive: Policyholder and any other persons with policyholders permission

Limitations as to use: Policyholder on personal business (and other non relevant legal jargon).

Does no-one except me hate that 'persons' word - what is wrong with 'people'?

I had asked my broker for *any drivers* as I need to share driving when going to shows on business. Bishopsgate Insurance (now Fortis) said that this actually meant only I was allowed to drive my car on business. My wife Sarah has **exactly** the same clause and she read it that she was insured for other drivers on business.

I suspect any person who uses their own car on company business has a similar clause.

Well Bishopsgate refused to pay. I had to pay the costs of the claim, approaching £2000. I was extremely lucky that we had not ploughed into an army of holiday makers.

We hit the crash barrier at about 65mph, and then got hit broadside by a 40 ton trailer truck going at speed. Such is the strength of my car, that we then did a 2000 mile round trip without further incident. (Well other than Roy destroying a wheel rim in a 1 mph broadside into the kerb).

I repaired the car myself, and saved a great deal of money by stripping it and re-assembling. I also got a complete door/mirror for £90, saving about £400 on the new prices. I used an amazing scrap yard in Aylesbury. I really enjoy mountaineering over piles of rusty heaps - it is amazing what you find. However this 'scrap yard' is a glossy parts shop with bar codes, computers, free baseball caps and coin-in-the-slot coffee! Now **only** 10 year old Volvos just don't happen in scrap yards. The newest I could find elsewhere in my model range was 16 years old - only 1 year after they introduced the model. However you can but try. In a weak voice, I stammer "You don't happen to have a nearside door and electric wing mirror for a 1990 Volvo 740 GLE do you"? "What colour" was the reply, after a millisecond of typing into the terminal. Drat, they don't have my colour. They sold me an identical door from a 12 year old Volvo in perfect condition. It seemed churlish to complain that it was blue and had no heated mirror. Things were looking up.

With the third party cost - a lorry with severe bodywork damage - I had to fork out over £1500, and I still hadn't replaced the wheel rim and the damaged tailgate trim.

I then took Bishopsgate to our insurance ombudsman. Now, as you all know, it is a well known fact of life that the individual stands no chance against the might of the insurance companies. Well that was the case in "Life before Superombudsman". After **only** 18 months of deliberation, they found in my favour plus interest. They said, that they knew what the certificate meant to underwriters, and that it was very common on certificates. However they understood how I did not realise that 'use' meant 'driving'. I contended I was 'using' the car and Roy was 'driving'. Bishopsgate make a great play of how their policy documents have won Plain English Awards. However the only mention of business is on the certificate of insurance - the document drawn up by lawyers.

My victory will mean that the insurance industry in the UK will have to change their certificates, and will mean many people can make retrospective claims.

It is good to have such a positive result after going all the way to Croatia and back through the worst December weather for 50 years to sell one keyboard membrane, and lose most of my stock in the Belgium ice.

Bishopsgate though got a final stab in the back. They have only paid half the cost of the wheel rim and tailgate tirm (£200) and refuse to compensate me for the £700 I saved them by getting the door, and preparing/re-fitting the car.

Watch for developments in the press......

QL Shows

The few traders left in the QL scene make quite an effort to get to shows, and often organise them.

By the time you read this, Roy

Wood and I will have flown to Itay for the Reggio Emilia show, and Jochen will have driven from Germany.

All the European shows/workshops we attend are free, and we also direct mail people on our database.

It is pretty depressing how few of you though take the trouble to come. We really enjoy the shows, and it is always a good get together. We often eat afterwards at a local restaurant. We have even been known to visit bars occasionally - only to be sociable I hasten to add. The QL more than any other computer I have known, past and present, is a very sociable machine. Most people we meet are very attached to their machines, especially those with the original black job.

However if attendances at the shows drop any more, then they will cease to happen. We know there are a number of people who have never attended a show. If you have never been to one in the past, then you must make a special effort for the QL 2000 bash at Portsmouth in October. It looks like being a very well attended show, and there are even more than 16 people flying in especially from the USA. If they can do it, then there is no excuse for anyone in Europe.

Q40 Colour Etc.
by George Gwilt

Introduction

Over the years there have been various changes in QL hardware and software. Probably the introduction of Toolkit II was the earliest and most widespread advance, since it was effectively a necessary correction and upgrade to the original QL Roms. I would guess that the advent of extended colour to the QL scene will be more pervasive than many of the other changes. It is already in use in Aurora, by means of a version of SMSQ/E. Now it is available in the Q40, again with SMSQ/E and will soon be released for the QXL and QPC. I trust that this exploration of colour will be of interest to some.

What follows is based on my experience with SMSQ/E v2.97 on my Q40.

Colour

The recent advance in SMSQ/E, with which I am concerned here, is the extension in range of colours from the maximum of eight on a basic QL to 65536.

What I want to do here is show how this affects the owner of a Q40.

Actually, if the owner simply switches on his Q40 and does nothing to alter the display, the extended colours do not affect him in the least. Four colours (Mode 4) and eight colours (Mode 8) are available just as before. All programs run as normal. However, as soon as the display is changed all eight colours become available even though the machine appears to be in Mode 4. Indeed you cannot now change mode which remains at "33" instead of "0" for Mode 4 or "8" for Mode 8. To get the look of the old Mode 8 you would have to set the appropriate csize. But why do that? The eight colours are already there, and we don't need bigger pixels to hold the extra colours so the larger csize is unnecessary.

There is one peculiar effect many must have noticed. For example in Perfection the borders look odd. This is because the stipple colour requested by the programmer of Perfection was a mixture of white and blue with a stipple number 2. In an ordinary QL Mode 4, blue is the same as black, so the border is seen as a mixture of white and black. With the extended colours you see a mixture of white and blue instead. This effect is true of several other programs, but on the whole these changes, though possibly slightly upsetting, are benign. However in version A.04 of QD (corrected in v A.07) a defined block became illegible!

To Stipple or not to Stipple

Colour is very useful in the legible presentation of written material to the users of programs. A reader can easily focus upon a particular set of items if they are in a distinctive colour. The more different colours there are, the greater the number of categories that can be highlighted. Thus it has always been a bit frustrating that the legible non stipple colour combinations of paper and ink are so few. For example green and white do not combine well. The best use of that combination is probably for "unavailable" menu items. The original QL offered stippling as a way of increasing the apparent numbers of colours available. But stippled ink is almost bound to fail, and there are perhaps only half a dozen combinations of stippled ink and paper which are acceptable.

This is where the extended colour system can really help. Each colour is solid, so that, provided there is enough contrast between two colours, they can satisfactorily be used as paper and ink. Of course you can use stipples as well for both paper and ink, which gives you more than $5*10^{19}$

different combinations to choose from. The exact numbers available are shown below.

Table Showing Number of Combinations Available

No of colours	Solid pairs	Solid/Stippled pairs
4	6	561
8	28	10,878
256	32,896	13,358,850,040
65,536	214,750,880	57,645,019,790,762,354,688

(It is left as an exercise to find the number of combinations if all 2^{24} colours are available. [They are not available on the Q40!])

The leap from 4 or 8 basic colours to 65536 is really rather daunting, so how should one make practical use of this?

The first thing to look at is how these extended colours can be obtained via SBASIC.

Extra Modes

To replace the Mode instruction there are four new Basic commands

COLOUR_QL
  - Sets the "normal" QL 8 colours

COLOUR_PAL
  - Sets a palette of 256 colours
COLOUR_NATIVE
  - Allows the full range of 65536 colours
COLOUR_24
  - As _NATIVE but with different parameters

The palette consists of 256 colours chosen from the total 65536 available. These can be reset from the default range if wanted. The default range comes with the first 64 colours described in words. For example extra greens available are called Light Green, Dark Green, Lime Green, Apple Green, Pastel Green, Dull Green, Dusky Green, Grass Green and Sea Green. The remainder of the 256 colours in the palette are not named. The implication seems to be that 64 dif-

ferent colours is probably enough choice for the paper, ink and borders which form most of the colour needs of programs.

If COLOUR_QL is set, the colour numbers are as in the QL Manual, 0 = Black, 1 = Blue, 2 = Red and so on. Stipples can be defined either by a single composite number, or by two numbers giving the two colours with stipple 3 assumed, or by three numbers the last being the explicit stipple.

For COLOUR_PAL, on the other hand, a colour is given as a number from 0 to 255. A stipple requires the two colours to be given explicitly, though the stipple number itself need not be given if 3 is intended. Oddly, the first eight default colours, although they are the familiar Black, White, Red, Green, Blue, Magenta, Yellow and Cyan, are given numbers 0 to 7, which do not correspond with the QL numbers, except for Black, Red and Yellow!

COLOUR_NATIVE requires colour numbers from 0 to 65535 (which is $2^{16} - 1$) and this accesses the full range of colours available on the Q40.

COLOUR_24 accesses the same range as does COLOUR_NATIVE, but the colour numbers run from 0 to 16777215 (which is $2^{24} - 1$).

All the COLOUR_ modes except COLOUR_PAL have parameters which directly describe the colour in a simple way. That is, they each indicate how intense the contribution is from the three primary sources, Red, Green and Blue. In the case of COLOUR_QL, the intensity of each is either minimum (ie not present) or maximum. For the extended colours the hardware allows a range of up to 64 intensities. This is achieved by allocating five bits of a word to each of the three primary colours while leaving the least significant bit of the word to be the common least significant bit of a six-bit intensity.

The most significant five bits are assigned to Green, the next five to Red and the next again to Blue as shown below. Since equal intensities of the three colours produces white, the least significant bit is labelled "W0".

This means that in practice the number of shades, except for White, is limited to 32.

| G5 G4 G3 G2 G1 R5 R4 R3 R2 R1 B5 B4 B3 B2 B1 W0 |

The colour parameters when COLOUR_NATIVE is set are the word representations of the bits as described above.

The long word needed to define the colour for COLOUR_24 has a zero most significant byte, followed by bytes for each of Red, Green and Blue. This appears to allow for a range of 256 intensities for each colour, but, as stated above, only five bits are available for each of the three colours, so only the value of the top five bits of each byte is transferred to the appropriate colour. What of the least significant bit? The White bit? Ah, what indeed! There is presumably a logic to it but it seems as if this bit is more or less arbitrarily set. $FFFFF8 and $FDFFF8 set it, $FFF8F8 and $FDF8F8 don't. The moral is - if you want to set the bits precisely, use COLOUR_NATIVE. This will give you full control.

Practical Use

For applications using text, there is no need to access the full range of colours. COLOUR_PAL is probably more than enough to allow pretty borders and paper colours as well as ample variation of ink colours.

However, for graphical work, the extended colours can indeed be useful. For example, the following will enable you to see displayed the range of shades of Magenta.

```
100 COLOUR_NATIVE
110 OPEN#3,SCR_624X420A0X0:PAPER#3,$ABCD
120 BORDER#3,4,$F00,$49F,1:CLS#3
130 FOR x=0 to 31:BLOCK#3,6,300,x*6+200,60,$42*x
```

Screen Background

One other enhancement which is part of the extended colour system is to allow a background. You can set this background, which will show up between any windows on the screen, either to a colour (possibly stippled) or to a screen image previously saved.
The SBASIC commands are

```
BGCOLOUR_QL
```
for the 8 QL colours
```
BGCOLOUR_24
```
to access the extended colours
```
BGIMAGE <scr_image>
```
to set a screen image

These, and all the previous SBASIC commands, are also available as Operating System calls. This means that the extended colours can be used by assembly language programs as well as by SBASIC.

Sprites, Blobs and Patterns

One place where the extended colours might be interesting indeed is in the definition of sprites and patterns. The explanation of how to define sprites using the extended colours is given in the description of the Graphic Device Interface Version 2. Unfortunately this does not enable a programmer to use the system, owing to something faulty having crept in, either in the description itself, or in SMSQ/E v2.97.

"As a result I cannot comment more on this interesting and enticing aspect."

That is what I wrote before I learnt that the software did actually support extended colour sprites. As a result I pried deeper into the SMSQ/E software and found that there was one crucial mistake in the description. In case others may have been equally puzzled, I'll expose this mistake.

Sprites have a header defining the screen mode, size and origin of the sprite. The first word for an ordinary QL sprite is either $0100 for Mode 4, or $0101 for Mode 8. This has been extended to;

$0210 for 256 real colour (GRBGRBGX) where

X=R/B

$021F for 256 palette colour
$0221 for Native colour

In the description issued with the Q40 the first word of the header was wrongly described as having a first byte of '0" or '1', instead of '1' or '2'. For sprites, the definition contains pointers to a 'colour pattern' and a 'pattern mask". For the 256 colour modes, the pattern and mask consist of one byte for each pixel. For the Native colour one word is needed for each pixel.

Odd Quirks

Sometimes I have loaded several programs before I make the extended colours available (eg by DISP_SIZE 0 for the 512x256 screen, or by DISP_SIZE 1024, for the 1024x512 screen). Clearly the channel blocks for all screen channels have to be amended. At the least they will now contain the new address of the screen. This will be $FE800000 and not $20000. One might hope that the colours would have been adjusted too. However, it appears that the contents of the colour masks, which must now contain the NATIVE code for colours of paper etc, are left unchanged. This seems to account for the fact that the ink in channel 0 of a previously opened SBASIC appears yellow, having the code $FF00, instead of green, whose code is, or should be, $F800. It is a slightly less odd fact that if you set

the colour 4 (green) under COLOUR_QL, you will get $F100. If you set colour 3 (the default green) under COLOUR_PAL, you get the correct colour, $F800. If you look very closely you can detect a difference in these two greens.

Two more SBASIC commands are available which can, to some extent help here. They are PALETTE_QL and PALETTE_8. These enable you to set the 8 QL colours and the 256 palette colours. The colour numbers used here as parameters to define the colours wanted are those applicable to COLOUR_24, or "true colours" as they are called. I'm afraid however that if you do use PALETTE_QL to alter green from $F100 to $F800 (which is the default value in the palette) you will be slightly disappointed. This is because the least significant White bit is now set, so that you get $F801 for your pains. This extra 1 appears also in Cyan, Yellow and White. The practical significance of all this is small, so perhaps we can forget it.

Technical Information

One of the programs originally supplied with the Q40 is NET_PEEK. Amongst other things this aims to describe the colours used in any screen or console channel. The only colours recognised were the 8 QL colours, most, if not all, the others being returned as Black. That has been corrected and a new version of NET_PEEK will give in words all the colours named as the first 64 of the palette. Any other colour is shown as the hex NATIVE colour number. In order to reprogram NET_PEEK, I had to investigate the new contents of a SCR/CON channel block. What I found was this.

If the Mode has been altered to 33, that is, if the extended colours are available, the position in the channel block normally used for the colour mask for paper, strip and ink contains instead the colour numbers for the main and contrast colours (both the same if there is no stipple). The stipple number is in the usual place, that is in the most significant two bits of the colour byte for paper, strip, ink and border. If the mode COLOUR_QL is being used the colours are mirrored in the usual colour bytes, but if any other mode is in use, the colour byte is

$00	for no stipple
$20	for stipple 0
$60	for stipple 1
$A0	for stipple 2, and
$E0	for stipple 3.

Anyone following all this carefully might have wondered where the border colours go because there is no "colour mask" for the border in the normal QDOS system. What happens is that one word (but not two!) has been found for the border colour. It is the last word in the extension to the channel block required by the Pointer Environment. It is thus at word $42 from the start of the channel block, or at -2 from the address of A0 if you are using EXTOP to access the channel information. If there is only one colour, its number is stored in that word. If there are two, the number of the first colour is abridged and placed in the most significant byte and the number of the second colour is similarly abridged and placed in the least significant byte.

The abridgement is done by taking the three most significant bits of the red contribution, followed by the three most significant bits of the green contribution and finally the two most significant bits of the blue contribution.

You can see all this for yourself by using NET_PEEK. First, find the channel number by using menu item 5 (Channels) to display the channels. The number under the heading SYS is the internal channel number. By using that as the parameter for menu item 6 (Single Channel) you will see displayed information about that channel. If it is a screen or console channel you will see the colours of paper etc. You will see at the top the address (in hex) of the channel block itself. By using menu item 7 (Display RAM) with the address of the channel you can now see the contents of the channel block.

Etc.

An advance associated with the new screen display is the increase in the size of window available. For some time now, with Aurora and the QXL sizes from 800 by 600 upwards can be set. (So far I have only been able to set sizes 1024x512 and 512x256 on my Q40, but perhaps that will change in future.)

The greater screen sizes have proved popular enough for programs to be altered to take advantage of them. This should be comparatively easy, since all that need be done is to increase the dimensions of any bigger window, if necessary altering also its origin.

A similarly simple, or even simpler, alteration could allow existing programs to take advantage of the extended colours. In the unlikely event of that not being done, it will be left to each user to devise his own colourful programs. It would be a pity if that were the only use of this advance.

Indeed this is part of the general question of how "improvements" can best be made use of.

Using "Improvements"

MINERVA

One such improvement is the tokenisation of integers in Minerva. This can speed up SuperBASIC programs by storing integers either as bytes or as word long numbers internally. As far as I can see this is hidden from the user, but will apply to any SuperBASIC program run. To allow such programs to be compiled by Turbo you are told to turn off this tokenisation. Under the new version of Turbo this will no longer be necessary. This should enable that particular advance to become more generally useful.

However, another change in Minerva, which has also been adopted by SMSQ, is an increase in the number of arithmetic operations available via the vectored routines in the operationg system. Obviously these can be, and presumably are, used by the operating system itself, but cannot be part of an external program written to be usable on all QL type machines, unless this program is machine specific or contains code determining its host machine and corresponding specific code depending on that machine.

As I said earlier, I guess that there will be sufficient interest in the colour capabilities of SMSQ/E for programs using them to appear. I wonder if too we shouldn't take advantage of the more widespread distribution of machines based on a 68020+. It may have been the remark in the Manual for the Super Gold Card

"... *we recommend that your programs do not make use of special 68020 instructions so that they run on all QL systems. This is a particularly important consideration for any program destined for sale or publication.*"

that caused me to do the opposite and to produce my assembler, GWASS, using many, if not all, of these special instructions. Admittedly GWASS was started primarily for my own use and later developed so that I could use the Floating Point Unit on my QXL and not for general distribution. But it is now part of the set of programs attached to the Q40. This shows that there are already in distribution programs depending on higher machines than the old 68000/8. Also available to Q40 users is my program GWDISS, a disassembler and NET_PEEK, already mentioned here. The first two protest on being loaded into a machine which is not 68020+. The last works perfectly well on such a machine, but refuses to disassemble, thus showing that

programs can be written to check what machine they are running on and take appropriate action.

Although it is not quite the same thing I should point out that there are now two versions of Turbo Toolkit, one for ordinary QLs and the other for SMSQ. This shows that machine specific items now exist.

I suggest that the time has come to consider upgrading and writing other programs in a similar way.

68020+

I will end by giving an example of the way that the 68020+ CPUs could be used inside toolkits and programs. This concerns the ubiquitous need to set a number to floating point format. This is required for example when a number up to 2^{32} has to be returned to a machine function added to SuperBASIC. An example of such a function is FLEN. This returns the length of a file as a floating point number. Internally, this number will be a long word. As an example of the enhancement possible in converting a long integer answer to QDOS floating point format I give the following method used in the DIY Toolkit along with the way it could be done using 68020+ instructions.

Both methods to start with:
D1.L contains the number to be set in FP format

```
MOVE.W    D1,D4       This sets a zero exponent
MOVE.L    D1,D5       This might set a zero mantissa . .
BEQ       DONE        . . yes it did ->
MOVE.W    #$81F,D4    OFFSET
ADD.L     D1,D1       Have we finished? . .
BVS       DONE        . . yes. D5 contains the mantissa ->
SUBQ.W    #1,D4       Adjust the exponent for a doubling of D1
MOVE.L    D1,D5       Get ready to find the answer
```

At this point the methods diverge

DIY Toolkit

```
          MOVEQ     #16,D0      Set the maximum shift
;
; The loop
;
L1        MOVE.L    D5,D1       D1 is the working register
          ASL.L     D0,D1       Can it be shifted D0 bits up? . .
          BVS       L2          . . no
          SUB.W     D0,D4       The shift worked - adjust exponent
          MOVE.L    D1,D5       Reset the mantissa
L2        ASR.W     #1,D0       Try 8, 4, 2, 1 shifts
          BNE       L1          If D0 > 0, more to do
DONE
```

The answer is now in D4.W (the exponent) and D5.L (the mantissa)

68020+ method

```
          BPL       J1
          NEG.L     D1              Make D1 positive
J1        BFFFO     D1{1:31},D1     Set D1 to the . .
          SUBQ.W    #1,D1           . . shift needed
          LSL.L     D1,D5           Shift the mantissa . .
          SUB.W     D1,D4           . . and adjust the exponent
```

As for the DIY Toolkit, the answer is in D4.W/D5.L

NOTE: BFFFO finds the 1st non-zero bit in D1 from bit 30 to bit 0

The DIY method requires five journeys round the loop L1 to the end, whereas the 68020+ method requires no loop so is shorter and about 4 to 5 times faster.
Given that this is a very common set of code which is executed many many times in normal programs using machine code functions, it looks like a good way to speed up many programs running on on Super Gold Cards, QXLs, Q40s and any other system which uses a 68020+.

Before I finish, I would like to report that I tried my NET_PEEK with QPC. The results were indeed interesting. First of all, the registers displayed for a job are just what you would expect on a QL. Also the content of the system variables at $28000 was correct. However, when I tried to disassemble code, the program fell over. I think that was due to two things.
1. I suspect that only 68000/8 instructions were emulated, and

2. The instruction
```
MOVEM.L   A7,-(A7)
CMPA.L    (A7),A7
```
set the ZERO flag, thus signalling that the machine was a 68020+.
These instructions do not set the ZERO flag on a 68000/8.
Clearly these are inconsistent and I think that either the QPC should emulate the full 68020+ set of instructions, or, a second best, it should emulate a 68000/8 correctly and set off the ZERO flag.

Conclusion
I hope that this excursion into colour is useful. I hope also that we will see programs appearing that make use not only of the colour capabilities but also of the 68020+ range, maybe even of the Q40's FPU.

Who knows!

QPC2 or QPC2?
What can the future of the QL be called
Peta Jäger

This question may seem silly at first but at present it is the only answer to the question - What can the future of the QL community be called?
The Q40 is a beautiful thing and I admire the enthusiasm which people find for the development. I have furthermore great respect appreciating Tony Tebbys work in the fulfilment of a real QL successor. I am however really disappointed about the lack of far-sightedness of Tony Tebby with regard to a real QL platform in software or hardware. It is a slap in the face of those people who supported the QL all these years, who bought new hardware (QXL) and software, who believed in the promise of what was still to come (colour driver, disk access in the background etc.) and who see that the colour drivers are developed for the Q40 first and second to QXL and QPC. Even though the

programming is slightly easier for Q40, it isn't correct to treat those QXL users who have been waiting for years for the completion of the drivers. After all many people have bought the QXL for a lot of money and many have then payed for SMSQE once again. In addition, the colour drivers were already demonstrated for the QXL at a French meeting years ago, admittedly not with great success (they ran only on Tony's computer). That QPC (2) then completely unnoticed at last gets the drivers, that is not nice for Marcel Kilgus (the programmer) nor for the QPC users.
In my opinion QPC is the only true alternative to the QL and in the following paragraphs I give my reasons why.
Microsoft's operating systems unfortunately have spread everywhere on this planet (presumably they spread everywhere else). We couldn't do anything

about that. Several things are important for an operating system to survive, if we do not want it to disappear like OS/2, Atari and Amiga os or other operating systems such as our QDOS:
1. software downward compatibility
2. broad availability (should run everywhere, notebook computer, PDAs etc.)
3. it should run parallel to other operating systems.
4. all available and future available hardware should be supported.
5. it shouldn't be expensive.
6. speed which makes work pleasant.
7. modern and simple to use software.
All these things which I find important are only fulfilled by QPC(2).

1 - Q40 as the only available QL alternative (not emulated) offers little more compatibility than QPC2, it's actually even less at the moment. This is at least what I have heard because I don't have any Q40. The compatibility has more to do with the operating system

than with the platform. All programs, which run under QDOS also run under QPC (partly with tricks). I have forced even old digital Precision programs to run under QPC. But as I have said before the compatibility has more to do with the operating system than the emulator or the hardware platform. The printer port under QPC is working correctly, the serial interfaces work upto 115k baud without recognisable problems (tested among others in the mailbox environment). Disc access works with DOS and QDOS diskettes, though not with ED diskettes which are formatted with 3.2 MB as it is possible for Goldcard systems. Till now, disk access over the QXL-WIN file has never had problems. The DOS DEVICE is still in the proof-testing, it will be available soon and might be a small revolution since it will finally be possible to transfer data between DOS and the QXL_WIN file directly. Unfortunately the network in the classic form does not work but QPC can use the DOS/Windows network. If two DOS computers are connected with standard PC network cards, then QPC can access the second PC and it has the fastest QL network available which is only dependent on the performance of the installed PC hardware!

2 - The availability of Q40 is, as with every proprietary hardware, naturally very bad. Even for such unusual operating systems like e.g. Apples MacOS, there are PC emulators to use in notebook computers etc... QPC2 runs practically on every hardware configuration with processors from AMD, Intel and also Cyrix with at least 166 MHz. It runs on notebook computers, desktop systems and within emulators. One can "take

along" QPC, everywhere to.

3 - Q40 only runs parallel to Windows if you have a second PC right next to it. Not incredibly exciting and a very expensive variant! QPC runs under MSDOS, Windows 3.1, Windows 9x/NT/2000 at the moment only in the full window mode, but the frame mode will be available shortly.

4 - This is nearly impossible due to the Q40 design. Only parallel and serial devices can be attached right now. Using QPC there is not much more to be connected right now, but this will change very soon, as I will explain now. In the foreseeable future an absolute hardware availability will be possible for QPC (2), since all of today's PCs are equipped with USB ports, for which you can buy practically any peripheral. If QPC supports the USB port in the future then no one has to write drivers for the actual hardware anymore because the SMSQE driver for the USB port suffices to operate all current hardware.

5 - Well, they are not cheap but if you consider the few installed SMSQ systems available then the price for QPC or Q40 is ok. Though I would recommend QPC if you already have a pc at home.

6 - It is said the Q40 is very fast and this is clearly correct because a 68040 is very fast. But QPC2 is also fast. At the moment practically no PCs are offered under 400 MHz clock frequency and that is why QPC is almost as fast as a standard QXL (68040 inside). PCs now available have clock frequencies of 1000MHz and more, using these QPC(2) will then outperform the Q40. Even an available Q60 in the future can't be much faster, it will be also much more expensive than

QPC especially as QPC can use current hardware (UDMA fixed disks, AGP graphics cards etc.). Marcel Kilgus is working on several projects which will increase the speed of QPC soon. The new emulation core may well bring a 20% or more performance increase. The transfer of the SMSQ/E floating point-routines to the floating point core of the PC CPU will bring a speed increase of about 400%. All these things will be put in concrete terms very soon.

7 - The greatest flaw in the QL scene is the absence of software. There is practically nothing. There are no tools for programming, especially since many programmers have gone to the PC side. OK you'd say, we have QSPREAD, we have QD, Linedesign, SUQCESS, even some of Digital Precision progs will get used once in a while, but to be honest people, Qspread is a program only a few people use, even by the people who bought it (me for example). QD is despite all improvements only an editor, no word processing program is in sight even though that is a program which we are missing in the QL scene. There is a group of people who will now suggest: we have Paragraph, text87 and Quill, but to be honest, all these programs do not even have all the possibilities which Wordpad offers in Windows 9x! Moreover you get this little program for free. Is there a good graphics program (Linedesign is a good program but since there is virtually no support and lack of speed, I don't use it much these days) which supports actual printers? Only Jonathan Hudson has brought the QL up to date in the data transmission area. Yet we have put ourselves into the offside even more, if we do not

get internet connection to the rest of the world soon. The programs for e-mail and browsing are already available but instead of programming something easy like internet connection, the programmers waste there strength in customisation for new hardware, which will not be up-to-date in a couple of years (months). The QXL has shown us that we can not keep up with the speed the hardware develops nowadays. QPC does the right thing in saying: "Hey, why hardware, the PC is efficient enough. Let us concentrate on software and develop programs, which compete with PC programs". Who like me has to work with PC hardware and software everyday professionally and can sing a song of how badly programs are written and how unreliably PC-software is (QPC excepted). QPC could close a gap

there. With SuperBASIC, C or machine code we could develop programs in a shorter time which can do everything one really needs and are also stable. If QPC runs in a window, one could develop programs which run in parallel on the windows desktop or Apple desktop in the PC emulation. The possibilities are then unlimited and our well-loved operating system could perhaps establish itself again and not die out. The Q40 however is a niche product with lower future prospects. E.g. The problem is, the QXL fans know all to well, is if the developers have no more desire to develop the Q40 any further (e.g. because the sales figures are declining) then it is over for all of them if they need spare parts which they can't buy anymore. A PCI graphics card that is needed for the milan is hard to get nowadays

(AGP is now the name). You won't find ISA slots in actual motherboards because of PC99! Watch out QXL fans: get a spare board with an ISA slot as a precaution ... P2/2 EDO-RAMS also have to be obtained (for Q40) more difficult, too. For the tinkerer and QL-Hardcore fan the Q40 is the only one.
For everybody else it can only be QPC!
For all you sceptics out there: QPC runs for me and, many others I know, every day without problems. The current version 1.54 makes it possible to exchange ASCII data between Windows and the QL scrap!
Additional information about QPC, Q40 and everything around the Sinclair QL can be foundon my redesigned home page: http://www.sinclairql.de soon.

Jochen's Comments

Peta's article forces me to reply. Others should reply too, but the lack of time (we're 2 weeks behind schedule) does not allow other replies for this issue. Maybe we find something from, for example, Tony and Peter Graf in the next issue. I don't want to comment on all points; with some, I agree, but with many, Peta forgets some facts (or does not know them). Beforehand, I would like to say: QPC is great, without any doubts, and I also sell it, as most of you know. I also sell SMSQ/E, so I think I can put some things right.
QPC's SMSQ/E is adapted by Marcel and Tony does not have the sources. This logically implies that Tony first programs the changes/improvements on the operating systems for ATARIs, QXL, Q40 or whatever, and then, afterwards, Marcel

adapts it for QPC. Since Tony sometimes modifies many source files, Marcel often has to spend quite some time with them. That's given order which cannot be changed for obvious reasons.
Regarding the implementation of the colour-drivers and "slap in the face": a bit, strong, isn't it? If I were in Tony's position and I had had the choice between a stable machine with genuine Motorola hardware, which we know, works (Q40) or an unstable slow software-protocol, then it would be easy for me to choose - and so it probably was for Tony. The QXL-PC hardware interface (designed by Miracle Systems) is by far not as good as it could (or should) be, and why waste a lot of debug time for trouble shooting without knowing if it is due to the unreliable inter-

face or the actual driver which needs to be debugged? It is more than understandable that the drivers were implemented on the Q40 first. Have a guess why Tony used the ATARI Mega STE for programming and not the so much faster QXL? For the same reason - a solid, reliable system is required for development, and why waste time for ruling out problem factors which do not exist on other systems?
Moreover, it has taken longer for the QXL, because it was quite clear that the old QXL-PC-software interface would be far too slow for the additional amount of data required for the 65536 colour resolution. So the interface had to be redesigned as well to make sure it would not be dead slow. Peta also forgets that QL revenues do not earn a full living - for none of us, unfortunately - it is probably as much hobby for

us dealers than it is for the customers. If we put programming-time in right relation to the revenues nothing at all would happen. Therefore I think accusations are fairly inappropriate.

Many people bought SMSQ/E for the QXL. They got far more than the SMSQ shipped with the QXL offered them, and most of them got it bundled for a third of the full price. This has nothing to do with the speed of additional implementation, so why complain!?

Yes, and I must correct another illusion, unfortunately: a USB-driver should allow the connection and access of all possible peripherals!? Ooops, no, it would be nice, but each USB-device requires its own drivers. There may admittedly be a common protocol, however a printer, a mouse, a CD-ROM, a modem, a CD writer or a TV-card cannot be driven by the same driver.

Anyway, let's see what the future will bring.

I think Peta provided us with a lot of provocative discussion-material, and I hope that we will have some replies to this in the next issue.

I don't think either that there is only black/white, Q40/QXL - many users still use their QLs, SuperGoldCards, ATARIs, etc, and so do I.

Regarding the provocative statements about the software: I prefer to use a stable program with fewer features - it does not need to look very fancy as long as it does the job I want it to do. I don't need 96% of the features Word offers me. I also don't need the information window that Word or Windows or whatever has just crashed and what I typed was in vain. I use Qspread more than any other program since I upgraded it last year - you can have a look at my harddisk to see how many _tab files I've created over the last two years. Many people speak to me at the show, and the fact that they tell me about their improvement suggestions shows, that they use the software. I think many users do not believe that others still use their QDOS compatible system and actually work with it - they seem to have forgotten that writing an application for their own use is still very easy compared to other systems. I have seen what problems can be caused by "today's" development systems like Delphi, where nobody knows what components do internally. Simple things can take 5 times as long as with BASIC!

So what do you think? Don't disappoint me and write to us!

Gee Graphics! (on the QL?) - part 18

Herb Schaaf

Matrix operations on the QL

Matrices are such a massive mathematical topic that I hardly know where to start. I'll just share a few PROCedures and a FuNction that seem to do the job of getting the "right answers".

There are indications(1) that the Chinese were using matrix concepts thousands of years ago, but the name "matrix" was first publicized(2) in 1858 after having been used in correspondence between James Joseph Sylvester and Arthur Cayley. Together they developed much of today's matrix theory and are described as the "Invariant Twins"(3). Sylvester was a wild, impetuous, poetic extrovert who invented lots of math terms. Cayley was a quiet, methodical, reserved mountain-climbing introvert who wrote lots of math works. Both were Englishmen who attended Cambridge. Both showed an aptitude for math. Cayley was senior wrangler in the math tripos in 1842, and Sylvester took 2nd place in 1837. Both took up the law because of religion, Cayley in 1849 at the age 28, and Sylvester in 1850 at the age 36. In the early 1850's they became friends at the Court of Lincoln's Inn and bounced their thoughts off each other. Together they made a great pair of math enthusiasts. I expect there are many of you who know much more about math history and matrices, and your comments, insights, corrections, suggestions, etc. are invited.

Most of you probably know that a matrix can be considered to be a rectanglar array of "elements" neatly arranged in rows and columns. The "elements" are usually numbers. So far all the matrices I've run into are 1 dimensional (row OR column, and referred to as "vectors") or 2 dimensional (rows AND columns). Do higher dimensional matrices exist in higher mathematics? Should they? Could they? Are there triangular matrices?

According to MacDuffee(2) matrices are much more than arrays. He states "A matrix is a number of total matric algebra". Matrices can be scaled, added, subtracted, multiplied(4), and raised to powers. In a matrix if the number of rows equals the number of columns then it is a "square" matrix and has a "determinant", and if the determinant is not zero then the matrix has an "inverse" that can act as a reciprocal. It would probably be best if you refer to textbooks(5) rather than have me try to explain. When indexing matrices some choose to count from zero, and others count from one. QL S*basic can handle either. I have chosen to write my FuNction and PROCedures with an option base of 1. I like to think we might find another clever way to make use of the zeroth rows and columns later on.

PROCedure MAT_SCALE(input_mat, scale_factor, output_mat)
changes every element of an array by the same ratio.

PROCedure MAT_ADD (mat1, mat2, mat_sum)
adds two matrices to form a sum, both must be the same size, that is to say they both must have the same number of rows and both have an equal number of columns.

PROCedure MAT_SUB (mat1, mat2, mat_diff)
subtracts matrix2 from matrix1, both must be the same size.

PROCedure MAT_MUL (mat1, mat2, mat_product)
multiplies two matrices to form a product matrix, the number of columns of the first matrix must be equal to the number of rows of the second matrix, and it can make a difference as to which matrix comes first(4). The product matrix will have as many rows as the first matrix, and as many columns as the second matrix.

FuNction MAT_DET (mat_name)
returns a single number, the 'determinant', from a 'square' matrix. If the determinant is zero, the matrix is 'singular'.

PROCedure MAT_INV(input_mat, output_mat)
finds the inverse of a 'non-singular' square matrix, similar to finding the reciprocal of a number. The inverse matrix can then be used in matrix multiplication to accomplish 'matrix division'.
In MAT_DET and MAT_INV a lot of looping, swapping, and other calculating can be involved, and it is possible to have wicked rounding errors accumulate. Inputs with 'reasonable' numbers will probably work out OK, but 'extreme' input values might result in wierd results. Is there or could there be a LRESPR'able "double" or "quadruple" precision S*Basic math package for the QL?
I've tried to make the MAT_ PROCedures and FuNction general. QL S*Basic can call arrays and parts of arrays by name, but mass assignment of arrays is not allowed.

PROCedure MAT_COPY(from_mat, to_mat)
is used to make mass assignment of values from the called input

array(s) to "working" arrays as well as to transfer the results back to a named output array. In all cases we must have created properly DIM'd named arrays before we can call them. Inside the PROCedures 'Work_ Arrays' are created, so as to preserve the called named arrays if desired.

PROCedure MAT_SHOW (mat_name)
prints out the matrix in rows and columns.

PROCedure MAT_COMPARE(mat1, mat2)
compares two matrices for differences, prints them out and reports on the total number of differences found. I use this to check textbook answers with QL calculated results.
MAT_SHOW and MAT_COMPARE are convenient when debugging and tracing programs.

These PROCedures and FuNctions seem to work OK, but are not especially elegant nor optimized. Since we may even find a use for some of these matrix operations in the future, I've started the listing "MatFuNPROCs_bas" with a high line number so that we can merge it in more easily. Apologies to those who abhor long listings.

Projective geometry can go on beyond Euclidean geometry into Non-Euclidean geometries. There we find concepts such as homogeneous coordinates, and learn that parallel lines do meet in 'ideal' points. Far out! Matrices are used in computer graphics to do accomplish rotations, translations, scalings, perspectives, and projections. Hope we can learn how.

Notes--
(1) Struik, Dirk J. "A Concise History of Mathematics" 1976 Dover Publications. Inc. New York
ISBN 0-486-60255-9
(2) MacDuffie, C.C. "What is a Matrix?" June-July 1943 American Mathematical Monthly, Vol. 50 No.6. pp. 360-365
(3) Bell, Eric Temple "Men of Mathematics" 1937 Simon & Schuster, New York
(4) Matrices can almost be treated like numbers, but there are exceptions. If Matrix A is different from Matrix B then "Matrix A times Matrix B" is NOT equal to "Matrix B times Matrix A". Sir William Rowan Hamilton became so excited when he discovered this exception on October 16, 1843 that he promptly carved it in stone on a Dublin bridge. He used the terms 'Linear and Vector functions' instead of 'matrix' when he published his Dublin "Lectures on Quaternions" in 1853.
(5) Dover has several less expensive paperback classical math reprints. Here are a few: "Matrices and Transformations" by Anthony J. Pettofrezzo, "Matrices and Linear Algebra" by Hans Schneider and George Phillip Barker, "Fundamental Concepts of Geometry" by Bruce E. Meserve.

Listing "MATFuNPROCs_bas"

```
20000 REMark MATFuNPROCs_bas
20010 REMark to go with GG#18
20020 REMark HL Schaaf August 17, 2000
20030 :
20040 DEFine FuNction MAT_DET (mat_name)
20050    LOCal i, swap_row
20060    IF DIMN(mat_name,1)<>DIMN(mat_name,2) THEN
20070      PRINT #0;"Error - not a square matrix " : STOP
20080    END IF
20090    DIM Work_D(DIMN(mat_name,1),DIMN(mat_name,2))
20100    MAT_COPY mat_name, Work_D
20110    determinant = 1
20120    FOR pivot = 1 TO n
20130      biggest = ABS(Work_D(pivot,pivot))
20140      swap_row = 0
20150      FOR row = pivot TO n
20160        Abs_element = ABS(Work_D(row,pivot))
20170        IF Abs_element > biggest THEN
20180          swap_row = row
20190          biggest = Abs_element
20200        END IF
20210      END FOR row
20220      IF (swap_row) THEN
20230        FOR i = 1 TO n
20240          SWAP Work_D(pivot,i) , Work_D(swap_row,i)
20250        END FOR i
20260        determinant = -determinant * Work_D(pivot,pivot)
20270      ELSE
20280        determinant = determinant * Work_D(pivot,pivot)
20290      END IF
20300      IF (ABS(determinant)<1E-10) THEN
20310        PRINT#0;"Determinant == 0 ":EXIT pivot
20320      END IF
```

```
20330      pivot_inv = 1/Work_D(pivot,pivot)
20340      FOR col = pivot+1 TO n
20350        Work_D(pivot,col) = Work_D(pivot,col)*pivot_inv
20360      END FOR col
20370      FOR row = pivot+1 TO n
20380        FOR col = pivot+1 TO n
20390          reduce = Work_D(pivot,col)*Work_D(row,pivot)
20400          Work_D(row,col) = Work_D(row,col) - reduce
20410        END FOR col
20420      END FOR row
20430    END FOR pivot
20440    RETurn determinant
20450 END DEFine MAT_DET
20460 :
20470 DEFine PROCedure MAT_INV(input_mat, output_mat)
20480    LOCal n,i,j
20490    n = DIMN(input_mat,1)
20500    IF n <> DIMN(input_mat,2) : PRINT "not a square array " : STOP
20510    IF n <> DIMN(output_mat,1) : PRINT "rows mismatched " : STOP
20520    IF n <> DIMN(output_mat,2) : PRINT "columns mismatched" : STOP
20530    DIM pivot_flag(n)
20540    DIM swap_row(n)
20550    DIM swap_col(n)
20560    singular_flag = 0
20570    DIM Work_I(n,n)
20580    MAT_COPY input_mat, Work_I
20590    FOR i = 1 TO n
20600      biggest = 0
20610      FOR row = 1 TO n
20620        IF (NOT pivot_flag(row)) THEN
20630          FOR col = 1 TO n
20640            IF (NOT pivot_flag(col)) THEN
20650              Abs_element = ABS(Work_I(row,col))
20660              IF (Abs_element >= biggest) THEN
20670                biggest = Abs_element
20680                irow = row
20690                icol = col
20700              END IF
20710            END IF
20720          END FOR col
20730        END IF
20740      END FOR row
20750      pivot_flag(icol)=1
20760      IF (irow <> icol) THEN
20770        FOR col = 1 TO n
20780          SWAP Work_I(irow,col),Work_I(icol, col)
20790        END FOR col
20800      END IF
20810      swap_row(i)=irow
20820      swap_col(i)=icol
20830      IF (NOT(Work_I(icol, icol))) THEN
20840        PRINT#0;" Bad news ! singular matrix ! no inverse"\
20850        singular_flag = 1
20860        EXIT i
20870      END IF
20880      pivot_inverse = 1/Work_I(icol,icol)
20890      Work_I(icol,icol)=1
20900      FOR col = 1 TO n
20910        Work_I(icol,col)=Work_I(icol,col)*pivot_inverse
20920      END FOR col
20930      FOR row = 1 TO n
20940        IF (row<>icol) THEN
20950          temp = Work_I(row,icol)
20960          Work_I(row,icol) = 0
20970          FOR col = 1 TO n
20980            Work_I(row,col) = Work_I(row,col) - Work_I(icol,col)*temp
20990            IF ABS(Work_I(row,col))<1E-8 : Work_I(row,col) = 0
21000          END FOR col
21010        END IF
21020      END FOR row
21030    END FOR i
```

```
21040    IF NOT(singular_flag) THEN
21050       FOR swp = n TO 1 STEP -1
21060          IF (swap_row(swp)<>swap_col(swp)) THEN
21070             FOR row = 1 TO n
21080                SWAP Work_I(row,swap_row(swp)), Work_I(row,swap_col(swp))
21090             END FOR row
21100          END IF
21110       END FOR swp
21120       MAT_COPY Work_I,output_mat
21130    END IF
21140 END DEFine MAT_INV
21150 :
21160 DEFine PROCedure MAT_SHOW (mat_name)
21170    LOCal i
21180    FOR i = 1 TO DIMN(mat_name,1)
21190       PRINT mat_name(i,1 TO DIMN(mat_name,2)),\
21200    END FOR i
21210 END DEFine MAT_SHOW
21220 :
21230 DEFine PROCedure SWAP(n1,n2)
21240    n1 = n1 + n2 : n2 = n1 - n2 : n1 = n1 - n2
21250 END DEFine SWAP
21260 :
21270 DEFine PROCedure MAT_COMPARE (mat1,mat2)
21280    LOCal i, j, rows1, cols1, rows2, cols2
21290    rows1 = DIMN(mat1) :cols1 = DIMN(mat1,2)
21300    rows2 = DIMN(mat2) :cols2 = DIMN(mat2,2)
21310    IF ((rows1==rows2)AND(cols1==cols2)) THEN
21320       diff = 0 : diff_flag = 0
21330       FOR i = 1 TO rows1
21340          FOR j = 1 TO cols1
21350             diff = mat1(i,j) - mat2(i,j)
21360             IF (ABS(diff)>1E-6) THEN
21370                PRINT "at row ";i;" col ";j;" diff = ";diff,
21380                PRINT mat1(i,j),mat2(i,j)
21390                diff_flag = diff_flag+1
21400             END IF
21410          END FOR j
21420       END FOR i
21430       PRINT diff_flag;" differences"
21440    ELSE
21450       PRINT "matrices are of different sizes"
21460    END IF
21470 END DEFine MAT_COMPARE
21480 :
21490 DEFine PROCedure MAT_COPY (from_mat, to_mat)
21500    LOCal  i, j, rows1, cols1, rows2, cols2
21510    rows1=DIMN(from_mat,1) :cols1=DIMN(from_mat,2)
21520    rows2=DIMN(to_mat,1) :cols2=DIMN(to_mat,2)
21530    IF ((rows1<>rows2) OR (cols1<>cols2)) THEN
21540       PRINT #0;"Mismatch in MAT_COPY !":STOP
21550    END IF
21560    FOR i = 1 TO rows1
21570       FOR j = 1 TO cols1
21580          to_mat(i,j) = from_mat(i,j)
21590       END FOR j
21600    END FOR i
21610 END DEFine MAT_COPY
21620 :
21630 DEFine PROCedure MAT_MUL (mat1, mat2, mat_product)
21640    LOCal i, j, jj, k, rows1, cols1, rows2, cols2
21650    rows1 = DIMN(mat1,1) : cols1 = DIMN(mat1,2)
21660    rows2 = DIMN(mat2,1) : cols2 = DIMN(mat2,2)
21670 REMark what about a [1 x N] or [N x 1] matrix ?
21680 REMark ie a row or column vector ?
21690 REMark transpose into a QL DIM array(1,N)
21700    IF (cols1 AND rows2 AND (cols1<>rows2)) THEN
21710       PRINT#0;"mismatch in MAT_MUL ! ": STOP
21720    END IF
21730    not_cols1 = 0
21740 REMark suppose cols1 = 0  (actually = 1)
```

```
21750    IF NOT(cols1) : cols1 = rows1 : rows1 = 1 : not_cols1 = 1
21760    DIM Work_1(rows1,cols1)
21770    FOR i = 1 TO rows1
21780      FOR j = 1 TO cols1
21790        IF (not_cols1) THEN
21800          Work_1(i,j) =  mat1(j)
21810        ELSE
21820          Work_1(i,j) =  mat1(i,j)
21830        END IF
21840      END FOR j
21850    END FOR i
21860 REMark for cols2 = 0 then DIM as column transpose matrix ?
21870 REMark when QL shows DIMN(array,2) = 0; it is actually 1 !?
21880    IF (cols2) THEN
21890      DIM Work_2(rows2,cols2)
21900      MAT_COPY mat2, Work_2
21910      DIM Work_3(rows1,cols2)
21920      FOR i = 1 TO rows1
21930        FOR jj = 1 TO cols2
21940          FOR k = 1 TO cols1
21950            Work_3(i,jj) = Work_3(i,jj) + Work_1(i,k)* Work_2(k,jj)
21960          END FOR k
21970        END FOR jj
21980      END FOR i
21990      FOR i = 1 TO rows1
22000        FOR jj = 1 TO cols2
22010          IF not_cols1 THEN
22020            mat_product(jj) = Work_3(i,jj)
22030          ELSE
22040            mat_product(i,jj) = Work_3(i,jj)
22050          END IF
22060        END FOR jj
22070      END FOR i
22080    ELSE
22090      DIM Work_2(1,rows2)
22100      FOR i = 1 TO rows2
22110        Work_2(1,i) = mat2(i)
22120      END FOR i
22130      DIM Work_3(1,rows1)
22140      FOR i = 1 TO rows1
22150        FOR k = 1 TO rows2
22160          Work_3(1,i) = Work_3(1,i) + Work_1(i,k)* Work_2(1,k)
22170        END FOR k
22180      END FOR i
22190      FOR i = 1 TO rows2
22200        mat_product(i) = Work_3(1,i)
22210      END FOR i
22220    END IF
22230 END DEFine MAT_MUL
22240 :
22250 DEFine PROCedure MAT_SCALE(input_mat, scale_factor, output_mat)
22260    LOCal i, j
22270    DIM Work_F(DIMN(input_mat,1),DIMN(input_mat,2))
22280    FOR i = 1 TO DIMN(input_mat,1)
22290      FOR j = 1 TO DIMN(input_mat,2)
22300        Work_F(i,j)=input_mat(i,j)*scale_factor
22310      END FOR j
22320    END FOR i
22330    MAT_COPY Work_F,output_mat
22340 END DEFine MAT_SCALE
22350 :
22360 DEFine PROCedure MAT_ADD (mat1, mat2, mat_sum)
22370    LOCal i, j, rows1, cols1, rows2, cols2
22380    rows1 = DIMN(mat1) :cols1 = DIMN(mat1,2)
22390    rows2 = DIMN(mat2) :cols2 = DIMN(mat2,2)
22400    IF ((rows1==rows2)AND(cols1==cols2)) THEN
22410    DIM Work_A(rows1,cols2)
22420      FOR i = 1 TO rows1
22430        FOR j = 1 TO cols1
22440          Work_A(i,j) = mat1(i,j) + mat2(i,j)
```

```
22450      END FOR j
22460      END FOR i
22470      MAT_COPY Work_A, mat_sum
22480    ELSE
22490      PRINT "matrices are of different sizes"
22500    END IF
22510 END DEFine MAT_ADD
22520 :
22530 DEFine PROCedure MAT_SUB (mat1, mat2, mat_diff)
22540    LOCal i, j, rows1, cols1, rows2, cols2
22550    rows1 = DIMN(mat1) :cols1 = DIMN(mat1,2)
22560    rows2 = DIMN(mat2) :cols2 = DIMN(mat2,2)
22570    IF ((rows1==rows2)AND(cols1==cols2)) THEN
22580    DIM Work_S(rows1,cols1)
22590      FOR i = 1 TO rows1
22600        FOR j = 1 TO cols1
22610          Work_S(i,j) = mat1(i,j) - mat2(i,j)
22620        END FOR j
22630      END FOR i
22640      MAT_COPY Work_S, mat_diff
22650    ELSE
22660      PRINT "matrices are of different sizes"
22670    END IF
22680 END DEFine MAT_SUB
22690 :
22700 REMark end of listing MATFuNPROCs_bas
```

You and Your Software - just Good Friends?

Part 10 - Commercial and Non-Commercial

Geoff Wicks

Correction

Astute QL readers have pointed out mistakes in my last two articles. Last issue there was a correction from Wolfgang Uhlig about Easyptr and now Ian Pizer writes that it is possible to get an alphabetical directory list in Text87. This is done by pressing TAB when the unsorted list appears.

Many thanks to Wolfgang and Ian for making my QL work and that of others easier.

Geoff Wicks.

Part 10 start here

When I first started writing this series of articles about 18 months ago I suggested there were skilled programmers, who were unskilled in program presentation, and skilled presenters who were unskilled in programming.

This statement had been inspired by a program I had been evaluating. It did some clever things with words, but I was never able to discover quite what. The programmer, who is no longer a QL user, had been unfortunate because initially his program refused to run on my system. This was not his fault. The problem was a minor extension I had temporarily installed while testing another third party program.

I recently looked at his program again to discover what I did not like about it. After loading the program it started by asking two questions. The first warned about memory, ram disk use and extensions that should be present. The second asked whether or not I was using QPC. Were these questions really relevant? A plus point was that the program correctly

adjusted to the screen resolution, but then the opening screen was almost blank. Just a request to enter the parameter file plus a few instructions. My immediate reaction was, "What is the parameter file?" I pressed D for the defaults. The next screen was a list of 17 parameters, which could obviously be amended, and the statement "action?". Again I was uncertain, but there was a small amount of help, so I pressed I for information.

At this stage I started to despair. The first information screen told me nothing about how to use the program. It was a long statement of its history and the system requirements. The next screen listed the files on the disk and what the program did, but not how to use it. After reading four screens I was still no wiser. I then discovered another set of information screens, all 6 of them. The author had written some clever routines for displaying text on screens of different resolutions and was showing these off, but they

were irrelevant to the working of the program. I was soon wanting to scream out loud, "Just tell me what to do".

I pressed ENTER to proceed, and was told the dictionary could not be found. When I finally got the program working, it was almost an anticlimax, leaving me with a feeling of "Was that it?". When I left the program, I discovered it had left all its temporary files on my hard disk.

Perhaps surprisingly, I still considered this program to be a potential commercial product. Here was clearly a skilled programmer who enjoyed working with words, and who probably had the potential to produce a range of recreational word programs. They would have been of minority interest and perhaps a little difficult to market, but would have still been viable, although probably not profitable. What the program lacked was the "fun" element and ease of use. It needed more visually attractive screens and a simpler way of inputting data and parameters to make it addictive.

I discussed the program with another trader, who had also been evaluating it. When I said the program needed some changes, the other trader replied, "No chance. I suggested a different front end and he just exploded". With these words a potential commercial product bit the dust.

About a year later I was at an Eindhoven show, and was asked to take a look at a program. It had an instant appeal. It looked different from the average QL program, and the author had obviously given much attention to its design. When I discussed the program with him, it was clear he had a good working knowledge of

the Easyptr suite and could exploit it to full practical advantage. As he demonstrated the program it did most things he claimed for it. Even without hands-on experience, I felt this program had commercial potential.

The main snag for me was that the program did not fit into the Just Words! range because it was not a word based program. However other traders were also taking a close look at it, and it is now on sale. When I add that it has become a SUQCESS, you will recognise the program and the author.

A third program I evaluated was a Scrabble game. The author had put a lot of effort into it, and I was favourably impressed. It was bug ridden, making it too easy to cheat, but I was reasonably certain the author had the skills to correct this. Although I would have loved to have added the program to the Just Words! range, I turned it down without any hesitation. This had nothing to do with the quality of the program. There was just no way I could solve the obvious copyright problems.

These three examples illustrate how a dealer must think when evaluating a potential new product. It is not simply a matter of whether a program works or not, or does what is claimed. There are also marketing, commercial and legal aspects. When a dealer sells your program or other product, he is also laying his own reputation on the line, and taking calculated financial risks. In the QL world these risks can be very real. The financial loss to Just Words! caused by the collapse of IQLR was greater than 13% of that year's income, and over 5 times the gross profits for that year.

It is therefore reasonable that a dealer places demands on a software author to make his program more commercially attractive. When he does this, it is not necessarily a criticism of the program or the programmer. It is just that the dealer has the better knowledge of what QL users are wanting, and how they will react to the product. If a software author is unwilling to make changes to his program before commercial release, then how will he react if the first users discover a bug or request new and improved features? A trader has to be able to trust his software authors to produce the final product QL users are wanting.

If you have written a program or routine you think may be of wider interest, do not be afraid to discuss it with other users and traders, but do not expect an easy ride. If it does have commercial potential you will not be on your own. The trader will be there to offer help in rewriting the program or will know the people who can help in rewriting. Even if you are not a skilled programmer, but have a viable idea, we may be able to make a commercial product out of it. The important point, no matter how good or how bad your programming is, is a willingness to be flexible, to learn and to make changes.

The QL community is full of creative people. Their talents should be shared.

This series is now complete. It has been well read and attracted a lot of interest. Many thanks to all for the many comments and feedback.

More on SERNET

Tim Fuller

The four articles on SERNET in QL Today Volume 5 Issue 1 were of great interest to me as I have acquired a Q40 and now have the task of copying many megabytes of files from my Aurora hard disk to that of the Q40.

SERNET seemed just right for the job, so I made up a lead with 25-pin connectors on each end according to Figure 1 in the article by Dilwyn Jones. To test the lead I tried copying files between ser1 on the Q40 and ser3 of the superHermes on the Aurora, and found it would copy only to the Q40. The reason must be, I think, that in the first place superHermes thinks it is connected to a real modem so does not expect to receive any data from the line until a carrier has been detected by the modem, and in the second place superHermes does not contain a pull-up resistor to assert the DCD signal when the pin is not connected. Presumably the Q40's serial interface does have such a pull-up, and a QL ser1 or ser2 would not need DCD as these interfaces have only one handshake signal in each direction. My theory is supported by the fact that file copying was possible in both directions when I had connected DCD to DSR at each end of the lead, though I should have been able to get away with the connection only

at the superHermes end. However I wanted the lead to be generally useful. So I conclude that a true null modem lead should have DCD connected to DSR at each end. The pins will be 8 and 6 on a DB25, but 1 and 6 on a DB9.

The reason I tested my lead on ser3 was force of habit, as I know from experience that ser3 works faster and more reliably than ser1 and ser2 on a QL. Also I hoped I could deceive SERNET into thinking ser3 was ser2 by using the redirection facility provided by the ipcextuk_bin resident extensions that come with super-Hermes, despite the warnings in the articles. However this did not work, so I have been using the Qpac2 Files menu to back up my Aurora to the Q40 using ser2 at 9600 Baud. It does seem slow, but I can see progress when the Q40 hard disk activity light flashes regularly. Sometimes though the flashing stops, so I assume that then SERNET is repeatedly trying to send the same packet without success.

SERNET would be much more

useful to superHermes users if it could be made to work with ser3. The speed would be four times greater, perhaps more because of the increased reliability. It appears that SERNET does not work with ser3 because it opens separately the receive and transmit sides of the serial port it is using (see pages 35-37 of the SMSQ(E) manual). If this is essential to SERNET and cannot be altered there would still remain the possibility of having superHermes allow separate opening of stx3 and srx3. I wonder if there is anyone willing and able to do the work?

Jochen replies: Tim is absolutely right: independant receive and transmit channels are essential for SERNET. The reasonable solution would be a driver for superHermes which also supports independant rx and tx channels.

But you can double the speed by cheating: SER2 sends happily at 19200 baud, and SER3 receives at 19200 baud. If you make up a special cable with three connectors, two at the QL side which connect to SER2 for transmission and SER3 for reception, and configure SERNET to use SER2 and SER3, then you can not only double the speed but also use SERNET on non-SMSQE machines (ooops, I'm cutting sales here!)

Programming ProWesS in SBasic - Part 3

Wolfgang Lenerz

As promised last time, we will now look a bit closer at the PW function. We have already seen this in the example program (line 210), but I haven't really explained what it does and why it is necessary, so here goes:

The PW function

As you will have noticed in the description concerning the *PWcreate* keyword, you have to give this keyword a type followed often by several tags. This is also true for other keywords, notably *PWchange* and *PWquery* which we will see a bit further on. Now, types and tags are actually numerical values, i.e. the tag "TYPE_OUTLINE" is equal to the value $4F55544C$ and the tag "POSITION_BELOW" is equal to the value 11000004. In other words, they are numeric constants.

You can, of course, use these direct values in the functions and keywords instead of the pretty names. So you could write

```
object=PWcreate(0,$4F5554C,....)
```

thus creating an outline type. This however, is not very legible, and then you would have to know the value of each tag. In the documentation, only the names are given, not the numerical values. Also, the code would not be very readable...

Moreover, since there are over two hundred tags and a few types, this would mean that learning all the values by heart would be a bit difficult. This problem doesn't exist for 'C' or assembler programmers who can define 'include files' which contain all of these values and define them as "variables", since these variables later no longer exist, but this is not really feasible in basic.

One could, of course, simply have a procedure initialising all the variables, something like:

```
DEFine PROCedure init_tags
    TYPE_OUTLINE       = $44544C
    POSITION_BELOW     = $11000004
    (... and so on for the 220 remaining tags)
END DEFine init_tags
```

and then re-use that procedure in every program using Prowess.
Since there are more than 220 definitions like that, each Sbasic program in Prowess would already be at least 220+ lines long (or, if you number in increments of ten, lines 10 to 2210+ would already be taken...) without you having done anything! This would lead to an inflation in program size, and every SBasic program in Prowess would have to include this, making the inflation even larger still!

This is why the PW keyword exists. It is a function which takes as parameter the name of a type/tag and returns the value for it.
Thus, PW is a function, used as follows:

```
result=PW(tag$)
```

tag$ is the tag or type to get. It would be best if this were within quotes. It is possible to leave it without quotes since the keyword should be intelligent enough to figure this out. However, leaving the quotes out might give rise to problems when compiling your program later on, so it is preferred to give the name of the tag between quotes.
The names for the tags (and types) can be found in the part of the Prowess Sbasic Interface

manual containing a detailed description of them. If you have a look at the manual, you will notice that they all start with 'PW', and have the format PW("*name*") - e.g. PW('TYPE_OUTLINE'). The 'C' manual uses the entire name, including the 'PW_'. In the **PW** function, you only type the *name* part.

It doesn't matter whether you write the actual name in upper or lower case. However, since in the manual they are all described in upper case, this has beenused throughout this series (as well as in the SBasic manual) and in all of the examples.
As an example, in the manual, you might find the name PW('OUTLINE_QUIT_ACTION') for a certain tag. In SBasic, you would thus write:

```
object=PWcreate(...., PW('OUTLINE_QUIT_ACTION')
...).
```

There, that wan't too difficult, was it?

And now, for something completely different.

Using strings in Prowess
Whilst using Prowess in SBasic is generally not very difficult, curiously enough, one must pay a little bit more attention to what one does with strings, if they are used as parameters for the PWxxxx keywords. (This short explanation does not concern the **PW** function explained above nor the other keywords which don't start with 'PW', where strings are handled normally). Some of the tags or types passed to the keywords use string parameters: for example, the text in the title of an outline object. Unless you adhere to the rules explained below, there can be a slight (ahem) problem. But first, a little history lesson is in order:

A little history
The potential problem here stems from the question: how long is a piece of string? This happens to be not only a bad pun, but also the true reason things get so involved here. It was originally foreseen that Prowess would be programmed in 'C' (and, possibly, Assembler). To a large extent, Prowess itself is programmed in 'C'. The 'C' language uses a certain way to determine how long a string is: it starts at the start of the string (it knows where that is), and then strings along (groan) until it finds a \0, i.e. a CHR$(0). It then knows where the end of the string is, and so it can determine how long the string itself is.

As you may know, the QL normally uses another way of handling strings: the characters of the

string are preceded by two bytes, which contain the length of the string. So, there is no CHR$(0) at the end, and the same string "Isn't computing great" looks, in memory, as follows:

in normal QL mode:
```
(length word) Isn't computing great
```
and in 'C':
```
Isn't computing great0
```

Use CHR$

Of course, these two ways of doing things are incompatible... This means that you, the programmer, must make sure that your strings always end with a CHR$(0) as is required by Prowess: if you pass a string variable (such as a$), make sure that a$ contains a CHR$(0) at the end: a$=a$&CHR$(0). Likewise for arrays: if you pass (part of) an array, this must end with CHR$(0):

```
DIM B$(20,40)
 ...
B$(5)=B$(5)&CHR$(0)
```

Strings are not copied

The strings you pass to Prowess are not always copied to a safe place, since Prowess in many cases thinks (just like most 'C' programs) that if you pass it a string, then you should damn well keep the string until it has finished. This means that you cannot pass a direct string (e.g. something between quotes), or the result of a string function, to any of the PWxxxx keywords, you must always pass them a string variable (with the CHR$(0) at the end).
Moreover, this string variable should not be a LOCal variable, since these are thrown away when you quit the function/procedure where they were declared (unless you are sure that the object won't need the string any more when you quit the function/procedure, e.g. if you remove the object before leaving the function/procedure.)
Fortunately enough, there are one or two exceptions: Some of the tags allow a string parameter with 'COPY', which means that Prowess **does** copy the string to a safe place. In these cases you can pass a direct string. These tags generally contain the word 'COPY' at the end, so they are easy to recognise. This is also mentioned in the manual where the types and Tags are detailed.

The rules for passing strings

In short, here are the rules for passing strings:

IF the tag with which you use the string, is a COPY tag, you can use any kind of string:
- string variables, LOCal or global
- (slices of) string arrays, LOCal or global
- direct strings
- (results of) string functions

IF the tag is not a copy tag, use only:
- global string variables
- (slices of) global string arrays

but do NOT use a direct string or a string function. You MAY use LOCal variables if you are sure that you won't leave the function/procedure where they are declared before removing the object.

In all cases, append a CHR$(0) at the end.

Attention, string functions are not only those you make yourself in Basic and which return a string, but also such machine code functions as "&": Never pass as parameter:
```
a$&CHR$(0)
```
but always write
```
a$=a$&CHR$(0)
```
and then pass a$.

Finally, one small bonus: whenever you can pass a direct string (or string function), there is no need to append a CHR$(0) at the end, this will be done automatically.

Array handling

In the 'C' interface, some tags allow you to pass string arrays (for example PW('MENU_ADD_ARRAY'). It has been attempted to keep this functionality, so that you can also pass SBasic arrays to the corresponding SBasic keyword. This is no problem, if you keep the following in mind:

-› All array elements must end with a CHR$(0). The reason for this is explained above.
-› Only pass a two dimensional string array (e.g. DIM a$(10,20)).
-› When you pass an array after a tag requiring such an array, the array itself is generally preceded by two numbers which contain:
  - The number of the elements in the array. This is quite straightforward and should correspond to the first dimension of the array (i.e. DIMN(a$,1)+1, which would be 11 in the above example). The "+1" is due to the fact that arrays start at element 0, but DIMN returns the number of elements as used in DIM.

- The maximum length of each element of the array +2 (or 3). The maximum length is of course the second dimension of the array (i.e. DIMN(a$,2), which is 20 in the above example). You must always add either 2 or 3 to this number - 2 if the length of the array is an even number, 3 if it is an uneven number. This takes care of the two bytes in front of each element of the array and which, as is usual in the QL, contains the actual length of this element of the array, and of the possible padding byte if the length of the elements is uneven. I would recommend you always use arrays of an even second dimension, so you can forget about whether you should add 2 or 3 and always add 2 automatically...

-> Don't forget that arrays start at element 0! Example:

```
DIM a$(2,20)
a$(0)="Element one"&CHR$(0)
a$(1)="Element two"&CHR$(0)
a$(2)="Element three"&CHR$(0)
(...)
xobj=PWcreate(...PW('MENU_ADD_ARRAY'),3,
DIMN(a$,2)+2,a$,...)
```

or a more generic example:

```
DIM a$(x,y)
a$(0)="Element one"&CHR$(0)
a$(1)="Element two"&CHR$(0)
a$(2)="Element three"&CHR$(0)
(...)
a$(x)="Element x"&CHR$(0)
(...)
length=DIMN(a$,2)+2
length=length + (length && 1)
xobj=PWcreate(...PW('MENU_ADD_ARRAY'),
x+1,length,a$,...)
```

The length=length + (length && 1) part of this program makes sure that 1 is added to length if length is uneven: (x && 1) returns 1 if x is uneven, and 0 if it is even.

Some additional string handling keywords

There are also some additional keywords concerning string handling, the purpose of which is to make some part of the string handling a bit easier:

MKSTRING$

Sometimes, some of the query or change tags do not return a string, but a pointer to a string. This is an address where the string lies in the (completely useless) 'C' format. The MKSTRING$ function looks at the string located at that address, and returns it to SBasic as a real string. The Syntax of this function is as follows:

`String$=MKSTRING$(address)`

where address is the address returned by Prowess.
Example:

```
l_item_t=PWquery(my_loose_item_object,
PW('LOOSE_TEXT',))
item_text$=MKSTRING$(l_item_t)
```

This queries the item (a loose item) and gets a pointer to the text of the item from it, and then makes a string out of this, which is returned in item_text$.

MKLEN

Some of the Prowess calls will set a string passed to them to a new value. Of course, Prowess only sets this in the 'C' format - i.e. it forgets to set the new length of the string in the precedding length word (actually, Prowess doesn't even know that there is a length word before the string). MKLEN does that - it makes sure that a string passed to it will be the correct length - it stops just before the first CHR$(0) in that string. Use this only when the string was modified by Prowess, i.e. when it is a (call and) *return* parameter to Prowess. The length of the string that will be set does not count the CHR$(0) at the end - in other words, this will no longer be part of the SBasic string.
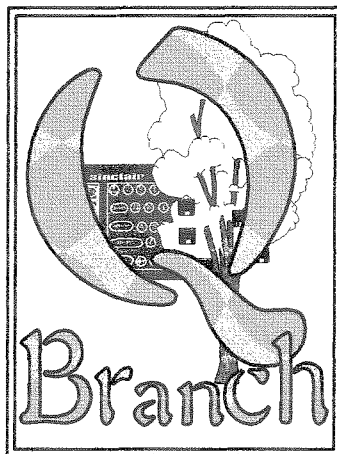
The Syntax of this keyword is:
`MKLEN string$`

MKLENO

This does the same as *MKLEN*, but keeps the CHR$(0) at the end.

Both of these keywords should be used ONLY when you are sure that there is, indeed, a CHR$(0) at the end. If not, and especially in a compiled program, it could happen that the variable stops where another starts - without a CHR$(0) to separate them! The new length would then count until the first CHR$(0), i.e. at the end of the next variable (perhaps!).

In the next instalment of what is no longer really a mini-series, we'll try to refine our example program a little bit.

Big news this month is that we have dropped the prices on many of the items that we have to buy in from Europe. The continuing weakness of the Euro on the exchange markets has allowed us to make large cuts in the prices of software from Jochen Merz, Progs etc. We are now selling QPC 2 for only £82.00 to non SMSQ/E owners and £67.00 pounds to those who already have a copy of SMSQ/E for a different platform. We will only be able to keep prices this low as long as the Euro remains weak so buy now and take advantage the situation.

Q Branch

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

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

Please Note our new address and phone numbers

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
Demo version £ 1.50 + postage
Full Registered version £ 18.00

¶ Version 2.03 available now !

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.

Few only

Super Gold Cards	£	125.00
QXL	£	Call
Recycled superHermes	£ 80.00 *
Recycled Gold Card	£ 60.00 *
Recycled Aurora	£ 75.00 *
Aurora	£ 90.00
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.

SMSQ/E

Gold Card / Atari / QXL Version

£ 69.00

Various Atari versions : call for details

The Colour Drivers are running on the Q Branch Q 40 !
Coming to a computer near you in 2000!

QPC 2 is here !

£ 82.00 (£ 67.00 SMSQ/E Owners)
Upgrades from QPC 1 £ 27.00
(return master disk)

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E

Q Branch Programs

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

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

Only £ 35.00

There are so many extras available for the Q 40 that it is hard to list them here. If you are interested in one of these boards than contact us and we will supply the details. the basic prices are shown here.	Q 40 board with 16Mb RAM & I/O card £ 330.00
	+ SMSQ/E £ 30.00
	Extra 16 Mb RAM £ 30.00
	Tower Case £ 45.00
	All prices do not include shipping.

Q40/Linux Journal

Tim Swenson

About 10 years ago, I was editing the Captial Area Timex Sinclair User Group Newsletter. I found that there were a number of articles on programming that I wanted to write, but I felt they really did not fit the readership of a User Group letter. So, I decided to create a newsletter just for such articles, and the QL Hacker's Journal was born.

Now with Linux available on the Q40, the discussion of Q40/Linux related issues really don't fall within the scope of a QL publication like QL Today. Given that a good idea can still be a good idea 10 years later, I've decided to create the Q40/Linux Journal.

The aim of the Journal is to cover topics that apply directly to running Linux on the Q40. General questions about how to create user accounts, how to change file permissions, etc. are best left to the whole horde of Linux books available. The Journal will focus on topics such as; using the sound on the Q40 with Linux, support for ethernet cards, features of the newly ported kernels, what packages have been tested and work on the Q40, and so on.

The audience of the Journal is from the experienced Unix user (and know that there is still a lot to learn) and the relative newcomer to Unix that is trying Linux because it is available on the Q40. My first goal is to document the things that I want to know and add other areas that readers would like to see covered.

I've been thinking about this Journal for a number of months and have thought about how best to approach it. I had thought that a single, updatable, document would be the best way to document using Linux on the Q40. The downside to this idea is that it would be kind of difficult for users to keep track of which version of the document they have, and what recent changes went into the document. With a Journal format, users would know what is new as it would all be in a single document.

The best medium for distributing the Journal is via the ql-developers mailing list. As much as it is listed as the ql-developers list, it is really the Q40/Linux mailing list. I don't know if all Q40/Linux users are on the list, but I do believe the majority are. I plan to post each issue to the mailing list and then place it on my web page. With past issues being on a web page, anyone coming late into the Q40/Linux world can quickly get up-to-date with

what's going on, and hopefully use the Journal to answer questions before posting them to the mailing list.

The ql-developers mailing list has been a good source of information, and with the list archive, the discussions can be documented for posterity. But, it can take some time to plow through the archive to find the right answer to a question. A lot of the discussions going on are the group, collectively figuring out the answer to a problem, sometimes taking a wrong turn. The advantage of the Q40/Linux Journal is that it will be documenting the answer after it has been found. Being a good procrastinator, I do not have an issue ready, but by making such a public statement pretty much puts me on the hook. I do have some ideas for articles and do know who to "volunteer" to write on areas that they know.

Like the QHJ, the Q40/Linux Journal will probably come out on an irregular basis, depending on what new work has been done and what new things users have tried. Like the QHJ, I know that feedback will be limited, but I do hope that the work involved with benefit the QL community.

Now I need to go forth and get the first issue done. I hope to have the first issue ready for the QL 2000 show.

To get on the list, send an email to **Bruce@qltoday.com**

Why always me...? or How bad luck turns into something positive

Jochen Merz

In the past few weeks I have had very bad luck in terms of hardware problems: first, the fast serial port SER4 of my main ATARI TTs died (permanent shortcut in the serial cable). Rather tragically for me, since this is the fastest port and I used it to connect my two TTs via SERNET. This

ensures that I have always the same data on both computers as a backup - which is a pretty safe method. Other serials ports are occupied by the dot matrix printer, modem etc. and can only handle a maximum of 19200 or 38400 Baud. The mailbox TT's serial ports are also used up completely by the modems etc.

If that was not enough: my main TT died one day after the Eindhoven meeting. A strange smell, and a few moments later it was dead. OK - power-supply swapped - nothing, no sign of life. Panic!

There is a reason why I call this TT my "main TT", I use it for virtually everything (calculations, invoices, programming etc.) - except for internet access. And just as bad: I also still use it as a native ATARI, in order to print QL Today on it and do the layouts.

But I had still the Q40. Because it was still lacking 4 serial ports it led a shadowy existence and I actually took it only to QL meetings. Well, a very good replacement for the QL/SMSQ part of the TT. The TT's harddisk had survived, but it is not possible to connect this SCSI harddisk (or my Syquest removeable, SCSI too) to the Q40. But the solution was relatively simple: connect this harddisk to the mailbox TT and transfer via SERNET to Q40.

The main problem was to having only two serial connectors on the Q40. One was already used by the mouse, so only one remains. So I have to use it for the printer, sometimes for SERNET, sometimes for a modem to send faxes. Not the best solution, but what could I do...? I was glad to have a working system.

Very positive: The speed of the Q40 is impressive! I am using the Q40 now for some time and am very positively surprised. All my programs run - much faster! Unfortunately I did not have much time due to whole bad situation and the time lost thereby did not allow me to experiment with the colours (But George Gwilt has dealt with the colours in detail in his article).

The solution for the lost ATARI side of the TT was not so simple, unfortunately. There is Calamus for the PC, but this

meant I had to get all my data from the ATARI to the PC. Fortunately, I still had another SCSI removeable drive, which works both with the PC and with the ATARI. Only one single 32MB-Partition was allowed to exist, otherwise the PC does not like the ATARI format - and the PC does not support more than one partition on one medium, but better this way than nothing at all. After some hours all my data was on the PC. Windows prints correctly if you print to a printer connected to the parallel port. But if the same printer is connected via serial (the parallel port goes to the Q40 and I do not want to swap cables here too, like I have to do on the serial port of the Q40 as mentioned before), then some different data is sent and the printout is never correct. Solution: new printer drivers from the Internet. Not quite, even more faults in the printout. No solution - swap cables! I won't continue with the problems I have with the ATARI-PC solution, it probably won't interest you.

On with Q40. I had to get used to it slightly. The screen resolution is similarly to the TT and easily readable on a 17"-monitor. You have to have at least a 15", 14" is too small. But even 17"-monitors do not cost a fortune nowadays. The TT can only handle 4 colours (QL mode 4), the Q40, however, has 65536 colours, the stored displays occupies much much more memory than before. I could easily start 30 display-filling QSpreads to the 20MB of my TT without running out of memory. Q40 already complains about starting the 8th QSpread. Each job grabs about 1MB only for memory to store the display background. For the mailbox functionality, for which my Q40 was actually planned, 16MB is enough; but I would recommend to every-

one to equip Q40 with 32MB. After few days only plugging the cables all the time was getting on my nerves and I decided I had to get a new serial card. At first, when I got Q40, I had two cards (hard disk, floppy, 2 Ser, 1 Par) in the computer, but there were interrupt conflicts (the well-known PC problem - and PC cards are used in the Q40) and so I had to remove a card. It took some searching, but I got a freely configurable ISA card (no Plug&Play), not inexpensive, but available. The card has 4 serial and 3 parallel interfaces and must therefore be put in together with a card which holds the HD interface, floppy controller etc., otherwise I would have no floppy and hard disk. Soon I had the mouse working on all four serial ports - I was happy!

All three PARs also worked at the first attempt. Wow!

Too much luck for me, probably, the Q40 did not want to speak to my other TT via SERNET. I had gone through all the configurational possibilities but it wasn't until I used a serial interface tester at SER4 and figured out that SER4 of my other TT was totally dead - all lines with the exception of Carrier Detect were gone. No idea what has killed it. Whether it was simply an unfortunate coincidence or a result of this short-cut - no idea! Especially since serial ports are desgined for "hot plugging" and should survive wrong connections and short-cuts. But the computers have their age and were also very reliable for many, many years.

I thought: use a slower port... better than nothing at all. But, as soon as one sends data faster than 300 Baud, half or more of the date gets lost on the receiving Q40 side. Again in a deadlock!? Despair, many telephone calls with Tony

Tebby, trial, and now finally the serial ports work. There must be only one interrupt per port, otherwise hardware problems arise. Tony's software can handle all devices sharing one interrupt, but as soon as more than one device is connected to the same interrupt, they fight for the interrupt level. The new card is, fortunately, very freely configurable, so that I can take e.g. 7 and 11 for serial port 3 and 4. It works! Hurray!! Still better: my main TT arrived back from repair. I copied everything back, and the Q40 can take over the mailbox TT and do the job I bought it for.

At present I copy the mailbox data over a slower port, the Q40 should soon replace the mailbox TT and you should benefit from faster BBs's. More (hopefully) in the next issue.

3 days later: the Q40 has replaced the BBS TT!

Letter-Box
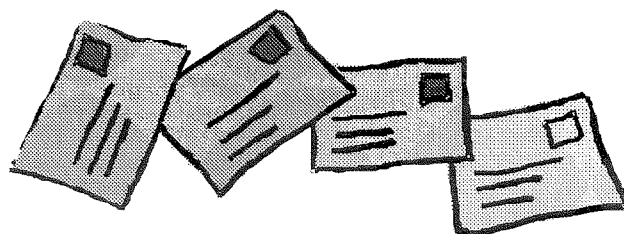
Jerome Grimbert writes to tell his experience of the new colour drivers for the QXL:

last evening, I made some additional experiments with disp_colour:

contrary to what I read, it is possible on the QXL to switch between the GD2 driver and the old one (and vice-versa):

I started in 1024x768 x 16 bits. I never was able to get 1024x768 x mode 4/8; but

    disp_colour 0,800,600

does work and puts the display in 800x600 x mode 4/8

    disp_colour 3,800,600

does work also, giving the 800x600 x 16 bits. Also 640x480 did work fine.

But I never succeed to get 512x256 working, even when starting with SMSQE configurated to 512x256x16 bits (display shows garbage).

But the good news is that it means that text87 will not require a reboot to get the right screen mode...

My web pages for the QL are now at

    http://www.crosswinds.net/~grimbert/ql

which includes my High Colour Sprite Editor.

.W. Wright writes a response to the keyboard interface help request in QL Today:

In reply to John Rish "Desperately seeking..." query in QL Today Volume 5 Issue 2.

It is a Schön PS/2 keyboard interface. I obtained one from TK Computerware in 1989.

The kit comprised the interface, a modified QL top case, a PC type keyboard. The top case had the key area cut away and resulting hole being covered with a piece of fibre board fixed with small self tappers at the corners. The 6 pin DIN plug was fitted to the left hand bottom corner of the fibre board. I later found that the keyboard had been fitted with a small additional circuit board containing 2 transistors and a small number of resistors etc between the connecting cable and the circuit board.

This kit was fitted to my QL and I found that I had a very stable system which accepted further hardware updates without a flicker. After 9 years of faultless use I started to get occasional missed keystrokes and additional characters.

I met Tony Firshman at a computer fair and discussed these problems with him. He suggested cleaning the keyboard contacts. This I did, curing the missed keystrokes, but not the additional characters. Over the following months further faults developed in my QL, mainly faulty ram chips (old age?) which resulted in the destruction of my hard disk. I again contacted Tony through TF Services asking if my computer could be repaired. He agreed to try but after 1 or 2 attempts the keyboard still gave trouble. At this point I decided to eliminate the problem by asking him to fit a superHermes. This was done and the various pieces of redudant hardware were returned with my repaired QL.

I have compared the photographs with my Schön interface and it matches exactly in all respects.

From memory, I recall that an earlier version of this interface using the existing keyboard gave trouble with key bounce, etc and was given bad reviews. A later review of the PS/2 version said that it had cured all the earlier version's faults (hence my purchase). The bad reviews may have affected sales as it took TK Computerware about 7 months to obtain my kit.

I hope the above ramblings will answer John Rish's query and prove useful.

Ralf Reköndt also told us - with surprise - that it is a Schön interface.

Peter Marsh writes:

The keyboard interface on page 7 was sold under the name of "SCHON" back in 1987 WITH KEYBOARD for £99.95 .

About Printing Accented Characters

Ian Pizer

In order to see accented characters on your QDOS screen you need to have a list of the combination of keys to get what you want. Below is my list. (consult Jan Jones Handbook page 184 or the original QL USERS GUIDE, or just try the combinations shown below).

ASCII	HEX	CHAR	KEYS
130	82	å	CS2
141	8D	à	C-
140	8C	á	C,
142	8E	â	C.
128	80	ä	C esc
144	90	è	C0
131	83	é	C#
145	91	ê	C1
143	8F	ë	C/
148	94	ì	C4
147	93	í	C3
149	95	î	C5
146	92	ï	C2
151	97	ò	C7
150	96	ó	C6
152	98	ô	C8
132	84	ö	CS4
154	9A	ù	CS;
153	99	ú	C9
155	9B	û	C;
135	87	ü	C'
136	88	ç	CS9
137	89	ñ	CS0
163	A3	É	CSC
172	AC	α	CSL
177	B1	π	CSQ
176	B0	μ	CSP

C=CTRL, S=SHIFT

But these combinations may not suit your printer so you need to load a TRAnslate program which translates the codes necessary for your printer.

I use a BASIC program written by Jan Bredenbeek of Holland which will do this for you provided you know which codes need to be translated. This program was once available on a Bulletin Board but you can find it on disk UG02 of the QUANTA Library. (in extremis ask me for a copy: **ianpizer@compuserve.com**).

This program allows you to use 1, 2, or 3 codes in the translation. For example: e backspace ^ would give you e with a circumflex. (101 8 94). I use this translate method when I print from QD but if TRAnslate is operative when I use Text87 I must print to PARD to avoid the translation which is not needed (thanks to Roy Wood).
Once you have run the BASIC program it makes a TRA program for you whose size is 263 bytes. To make the translation operative you do

```
a=alchp(263)
lbytes win1_TRA,a
TRA a
```

You cannot do it with LRESPR. To make inoperative do TRA 0. I have recently heard of a program called Clavier (French for Keyboard) which may be even more flexible than TRA. It can be downloaded from Jochen Merz's BB.
In QUANTA March 2000, Page 34, Dr Teply has managed to have about 30 accented characters for the Czech language and some with an inverted circumflex (the character above 6 on keyboard that I call circumflex but it probably has a proper name!), which the method above cannot do.

Foreign character sets are an issue which matter to quite a lot of QL Today readers. Thanks to Ian for giving us an example which works on QDOS and SMSQ/E.
We also get requests about other language options for SMSQ/E. As most of you know, English, French and German keyboard tables, translation tables and character sets are built into SMSQ/E. It is possible to choose between the languages by selecting them with LANG_USE, KBD_TABLE or TRA, followed by the country code (GB, F D) or the phone prefix for that country (44, 33, 49). This will select the error messages, keyboard layouts and printer translations of the given country.
It is possible to extend this scheme by adding new modules to SMSQ/E. You can, therefore, add a module to SMSQ/E which allows you to select Denish keyboard by typing KBD_TABLE DK.
However, before this is possible, somebody needs to prepare the right keyboard tables and error messages. This should be done by somebody who has a minor knowledge of machine code (it is mainly changing values in a table) and who knows the keyboard table very well. We can't do it - I have no concept of other languages than English or German, and some keyboards have fancy features like non-spacing characters (for accents).
If you would like to prepare a table and the messages, please contact us! (This invitation goes to everybody who uses a keyboard different to English, French and German).

Programming in Assembler - Part 9
Norman Dunbar

You can relax this time - this part is fairly short.

Maths Stack

The maths stack is where all internal mathematical calculations of floating point variables are done. It is also used to allow parameters passed to machine code procedures and functions to be 'collected' from the user and passed to the registers etc for use by the procedure or function.

The maths stack is simply an area of memory which can be used for all these fancy calculations, parameter handling etc. There is nothing (much) special about it and it is ALWAYS addressed internally using register A1 (relative to A6 - but you knew that didn't you?)

One of the first things I learned when writing extensions to SuperBasic was that on entry to a function or procedure, the A1 register is set to a value corresponding to the top of the maths stack. This is a MYTH and is not correct.

The value in register A1 can be anything on entry to a machine code function or procedure. I have done a lot of investigating (thanks to QMON2) and come up with the following rules:

If you want a suitable value in A1 for the top of the maths stack, then either fetch some parameters, or load it from BV_RIP

This means that if a function wants to return a value - which functions usually do - and the function has no parameters then you must load A1 from BV_RIP(A6) before calling the BV_CHRIX vector to reserve space. As I found out to my cost, not setting A1 is a good way to trash the system!

If your function does have parameters, then AFTER they have been fetched, A1 is set ok, up until that time, it is not and has the following possible values:

A1 Negative
If A1 is a negative number, then your function has been called as part of an expression such as:

```
PRINT 10 * MY_FUNCTION(p1, p2, p3 ....)
```

The number in A1 is the number of bytes that have already been used on the maths stack for the '10' in this case. This will be -6 as the 10 will be stored as a floating point number.

A1 Zero
If the number in A1 is zero, then your function has been called thus:

```
PRINT MY_FUNCTION(p1, p2, p3 ....)          or
PRINT MY_FUNCTION(p1, p2, p3 ....) + 10
```

and no bytes have been used on the maths stack yet.

A1 Positive
If A1 is greater than zero then this implies that there are that many bytes available on the maths stack and calling bv_chrix to allocate stack space will not move the maths stack around in memory.

NOTE: I have NEVER seen this documented and it has been discovered by me during long debugging sessions. Now that SMSQ is here, the above information may no longer be valid. The ONLY thing to remember is that on entry to a procedure or function, A1 DOES NOT hold a suitable value for the top of the maths stack as stated in various documents.

So that is the real situation and not as specified in the documentation. I took ages to debugg one simple function I wrote, which had no parameters and required some space on the maths stack for its result. Take a look at the code in the colour functions (green, red etc) we wrote back at the start of this article and you will see the following code:

```
return_d7    move.l  bv_rip(a6),a1   ; Because we had no parameters passed
             moveq   #2,d1           ; Size of stack space required
             move.w  bv_chrix,a2     ; Routine to allocate maths stack space
             jsr     (a2)            ; Go get some space NO ERRORS OCCUR !
```

As you can now see, we load A1 from BV_RIP because none of the functions had any parameters passed. Had that one line of code been missed out, your QL would have crashed. Try it if you like!

Values on the maths stack must be stored at even addresses. For integers, long integers and floating point values, this is not a problem. Strings, on the other hand, must be set up correctly with the word defining the size on an even address and the bytes of the string following. Odd length strings should have an extra padding byte to keep the A1 maths stack pointer even.

If you read back to the section above entitled 'Keeping things even' then you will see how to do this. If you are returning a string from a function, you will need to reserve space for the string, its word count and a possible spare byte for padding. Refer to the explanation above and you will see why the following code 'just works':

```
ret_string   move.w  (a0),d1         ; Assume the string to return is at (A0)
             addq.w  #3,d1           ; +2 for size word + 1 for padding if reqd
             bclr    #0,d1           ; Make even - removes padding if not reqd
             move.w  bv_chrix,a2     ; Routine to allocate maths stack space
             jsr     (a2)            ; Go get some space NO ERRORS OCCUR !
```

Of course, I am assuming that A1 holds a suitable value. The code above will request an even amount of space for a string result. First we fetch the length into D1 - this is the number of characters in the string only.

We then add 3 to D1. This is 2 for the word count and one for a possible padding byte. By clearing bit zero of D1 we force the number to be even and can then carry on with the request for space etc. Easy stuff this!

Retruning Values from Functions

When returning values on the maths stack you must be very careful. When a function exits there must be a value on the top of the maths stack the pointer to this value needs to be stored in BV_RIP(A6) and D4 has to have a values in it which defines the returned parameter type.

```
+--------------------------------+
| D3 | Return parameter type     |
|--------------------------------|
|  1 | String                    |
|  2 | Floating point            |
|  3 | Word integer              |
+--------------------------------+
```

Notice anything missing? Although we are allowed to fetch long integers as parameters, we are not allowed to return them. This is a problem and the usual fix is to convert a long integer to a floating point and return that instead. This will be covered in another thrilling episode!

Channel Tables

In our procedure PSI_CLS, we use a channel number in SuperBasic. In assembler, this is no use to us as all internal operations that require a channel (CLS, PAPER etc) require a channel id which is a 32 bit long number which bears no resemblance (or only coincidentally) to a SuperBasic channel number.

In QDOS there is a channel table - for the entire system, and there is the SuperBasic channel table which is used to convert channel numbers into channel ids which is what we require. SuperBasic keeps us away from nasty things like internal representations - assembler does not.

The routine we used above, channel_id, is all that is required to convert a channel number to a channel id. It looks at the SuperBasic channel table and for each channel that has been opened (even if it is now closed) there will be an entry in the channel table. Each entry is $28 bytes long (40 bytes) and has the following format:

```
+------------------------------------------------------------------------------+
| Offset | Size    | Purpose                                                   |
|--------------------------------------------------------------------------------|
| $00    | Long    | QDOS internal channel id                                  |
| $04    | 6 bytes | Graphics cursor X position (Floating Point format)        |
| $0A    | 6 bytes | Graphics cursor Y position (Floating Point format)        |
| $10    | 6 bytes | Turtle angle (Floating Point format)                      |
| $16    | Byte    | Pen status (0 = up or 1 = down)                           |
| $20    | Word    | Character position on line for PRINT and INPUT etc        |
| $22    | Word    | Width of the channel. Set by WIDTH command in SuperBasic  |
|        |         | but defaults to 80 when OPEN is called.                   |
| $24    | Long    | Spare - currently unused                                  |
+------------------------------------------------------------------------------+
```

When a channel is opened in SuperBasic, an entry is created (or reused) in this table. At startup channels #0, #1, and #2 are pre-created and that is all. If you now open #4, a new entry will be created for it. If you open channel #10, then blank entries are created for all the 'in-between' channels (5 to 9) and entry 10 is then created and initialised on top.

A channel that has never been opened can therefore still have an entry in this table - channels 5 to 9 in the above example. All of these use memory so it is advisable to start with 3 and work upwards opening channels as you go, rather than opeing #100 or something similar which needlessly wastes 40 bytes of memory for each unused channel.

A channel that is closed, or has never been opened, has a QDOS channel id which is negative.

In the Basic variables area in QDOS (to be covered in a later issue - and by the way, I refer to the variables that hold information about SuperBasic, and not variables you create in SuperBasic !) BV_CHBAS holds the offset from A6 to the first entry in the table (ie channel #0) and BV_CHP holds the offset from A6 to the first byte AFTER the last entry in the channel table. Don't forget that these are offsets and that everything in SuperBasic is relative to A6 - simply because by doing this the base address for the job (SuperBasic is just another job in the machine) is held in A6. If everything else is stored as an offset then moving the job in memory is simple as only the A6 register has to be updated.

Take a look at the code for channel_id again and note how we are using addresses that are relative to A6. Make sure that you understand because all fiddling in the bowels of SuperBasic requires that you understand relative addressing.
Most of the time you will only be interested in the conversion from SuperBasic channel number to QDOS channel id.

Homework

As an exercise, why not add a new procedure called PSI to the code for PSI_CLS. This new procedure will carry out all the same work as PSI_CLS but it will not do the CLS part of it. This will be useful when you want to set the colours for a window but not clear it. I will NOT be giving the answers out next time, but here are a few hints:

- update the definition table with details of the new procedure.
- in the proc's code, set D6.B to zero for PSI and 1 for PSI_CLS. Do this as the first instruction in both procedures.
- In the PSI procedure, simply set D6 and jump to the code in PSI_CLS.
- Just before doing the actual CLS part of PSI_CLS, check the value in D6.B and if zero, don't do the CLS simply BRA.S to error_exit instead.

All in all, I think this can be done in about 10 extra lines of code, maybe less, not counting the extra lines in the definition block.

BEWARE: Adding even a few lines of code can sometimes cause any 'short' branches to go out of limit and this will cause errors in the assembly. If this happens, simply find the ones in error and remove the '.s' from the 'bsr' or 'bra' instructions.

Next Issue

That's it for this issue. In the next article we will be exploring the depths of QLTdis again with a pseudo-code run through of exactly how we are going to decode all those instruction families - no typing you'll be glad to hear. See you then.

The EPSON Saga ...

Jochen Merz

Readers of QL Today will know that I always used to recommend EPSON printers - they always were very flexible, and compatible with existing QL programs. Unfortunately, the range of printers which supported the required ESC/P2 is permanently shrinking. The only available model is the Stylus Color 900 - a high end printer. All lower models come with the very cut-down version of this printer operating system, called ESC/P Raster. It seems that EPSON saves a few pence by using a smaller EPROM and, saving programming work by providing just a "dumb dump" device without "intelligence".

Pity for us, but maybe there's a way out of this. All one needs to know is: what is ESC/P Raster? Very few instructions only, which I will list in the following article. I have not tested them myself, I hope, somebody is going to try it, and to provide the QL scene with a method of device driver emulation to convert "ordinary" output into raster graphics which is understood by all ESC/P Raster printers - as a result all QLers would benefit from the fact that virtually every EPSON colour printer could be used again for printing.

Please note that only the following commands should be used - they must not be mixed with any ESC/P2 commands.

The whole raster graphics print is based on the command ESC . (dot), hex $1B $2E. This is supported by all ESC/P2 printers too, so you can try it even if you do not have a ESC/P Raster "only" printer. The definition of the ESC . command can be found in any EPSON documentation (ESC/P Reference Manual, for example). It is quite complex, as it allows raw data as well as compressed raster (RLE, TIFF and Delta Row). To make sure it works on all printers, use uncompressed data. If you write a driver, allow the specification of the printer model so that the driver can decide, depending on the printer, whether to use compression or not. To give you an idea about the complexity: some more recent cheaper printers do not support compression (e.g. the Color 400), whereas older printers (Color II, quite old) did. Some do not support higher raster bands with compression methods, some don't if Microweave is on. Better use uncompressed, even if it takes longer. RLE compression, however, seems to be OK on all printers. If you try it on dot matrix printers, avoid 720dpi, best start with 180dpi and then try 360dpi.

Before your print raster graphics, you must "**Enter raster graphics mode**" (all hex values):

1B 28 47 01 00 01

Inside this mode, you cannot print text. You should not mix text and graphics printing on tjthe same page. Only a very limited range of printer codes is available, which are:
LF, FF, CR, ESC EM, ESC @, ESC ., ESC (i, ESC (c, ESC (C, ESC (V, ESC (v, ESC @, ESC $, ESC r, ESC U, ESC +, ESC (U

To exit this graphics mode, send

1B 64

Now, as you've entered the raster graphics mode, back to printing with ESC . :
The "**Print Raster Graphic**" command is (all hex)

1B 2E c v h m n_L n_H d_1 d_2 ... d_n

where
c is the print mode:
  0 = uncompressed raster
  1 = RLE compressed raster
  2 = TIFF compressed raster
  3 = Delta Row compressed raster
v is the print density in the vertical direction, v/3600 dpi. Possible values are 5, 10, 20
h is the print density in the horizontal direction, v/3600 dpi. Possible values are 5, 10, 20.
m is the raster band height in dots.
  Possible values are 1, 8, 24.
  The values depend highly on the printer model, resolution used and other settings (e.g. micro-weave, compression).
n_1+256*n_h is the number of dots in the horizontal direction.
d_1, d_2 ... d_n is the raster data, row by row, left to right.

Another (not quite as important) command is "**Select Dot Size**". Again, a feature not supported by every printer - and even the same settings are different in their result (at least the wording is different). The command is (all hex):

1B 28 65 02 00 00 n2

Where
n2 defines the dot size:
  0 = default dot size
  1 = small (smaller, micro) dot, depending on model

2 = standard (normal) dot size, depending on model
3 = double dot size (supported only by more recent models).

For the Stylus 600, 0 is the same, for the rest add 1. 1 itself is not implemented (seems to me like a bug, or maybe EPSON wants to be difficult).
This command is very printer model dependent, so better not use it unless you know the printer. It has to be issued before printing raster graphics, prior to paper feeding and should not be changed in the middle of the page.

EPSON recommends the following order of commands:

1. Initialisation of the printer:
  Initialize printer   ESC @
  Select graphics mode   ESC (G
  Set unit   ESC (U
  Set page length   ESC (C
  Set page format   ESC (c

2. Printing method control:
  Select uni-direction mode   ESC U
  Select MicroWeave print mode ESC (I
  Select monochrome/color   ESC (K
  Select dot size   ESC (e
  Select print speed   ESC (s

3. Then for every raster line
  Set vertical position   ESC (V or
                                ESC (v
  Then for every color per raster line
  Select color   ESC r
  Set horizontal print position ESC \ or
                                ESC $ or
                                ESC (\
  Print raster data   ESC .

4. New page   FF

5. Terminate printing: ESC @

The command to select MicroWeave is also printer-model dependent, and so is the command to select monochrome/color printing. The latter can be used to select higher performance monochrome mode, but is limited to more recent printers only.
I have no idea what these commands do on printers which do not support them - whether they are just ignored are whether the printer will print garbage. Any feedback would be helpful.
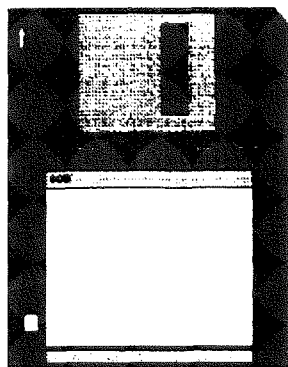
The London QL Show - A Review

By Darren D. Branagh

Saturday the 2nd September was the date for the latest QL Show, and the first in the Greater London area for quite a few years. It was held at a new venue - a hall not too far from Ken Brickwoods house, who helped organise the event, along with Malcolm Cadman and various other members of the London QL Group.

The entranceto the London Show Venue

I for one was looking forward to it, as I always remember how popular and busy the London shows of old were. The first QL Show I ever attended was at the old venue near Ladbroke Grove, some 6 or 7 years ago, and I remember how busy it was.

Sadly, this QL show was to be a lot quieter. They were several reasons put forward for this - the main ones being the fact it was held on a Saturday when most QL shows are on a Sunday, the fact that England were playing cricket at the Oval (just down the road from the venue), and the fact that maybe many people were

saving up for the forthcoming QL2000 event - all of which didn't help. Having said that, although numbers were scarce, a lot of interesting things happened.

The first stand on entering the main entrance of the hall was the QUANTA desk, ably manned by Bill Newell and Basil Lee. They were selling their range of Sweatshirts and other hardware and software items and taking subscriptions, and the QUANTA Library was available to Members. I discussed the forthcoming QUANTA CD-ROM Disk, which I am producing with a few people and interest was very encouraging - the disk will contain the entire QUANTA QL Library of programs all in a single QXL.WIN file for use with QPC, Q-Emulator, and the QXL, or any system that can read them - over 100Mb of files in total. This should be available from the QUANTA desk at the QL2000 show for a modest fee of about £5 each.

Next to QUANTA was Roy Wood of Q-Branch, selling his vast range of pointer driven programs, and also clearing a lot of cables and small hardware pieces as Roy has recently closed his Shop - the Bank Volt, near where he lived, to concen-

trate on other matters. Lots of bargains could be had, and as ever Roy had a crowd around him for much of the day. I at least got a laugh when Roy was reduced to buying a copy of MS-DOS from me in order to do an install of QPC for a customer!! Roy is currently testing the beta drivers of the 16-Bit colour version of QPC2, and said that is was coming along well and hoped to release them very soon indeed. This coupled with the fact that the first ever QL emails have already been sent by Jon Dent is very encouraging, and we will all probably be using our QL's for email and web access by year end at the latest.

Star of the Show award though goes to Tony Firshman on the TF Services stand, who actually premiered a brand new product at the London Show - called a compswitch. This is an ingenious little device that closely resembles a 4-way trailing socket that many of us are familiar with - indeed you will find one in virtually all QL'ers homes. However, the compswitch has some very clever additional features.

Compswitch is designed to switch on/off all computer peripherals automatically with a computer, when it is plugged into one of the sockets marked "computer". It will mimic exactly the "power-out" connector you
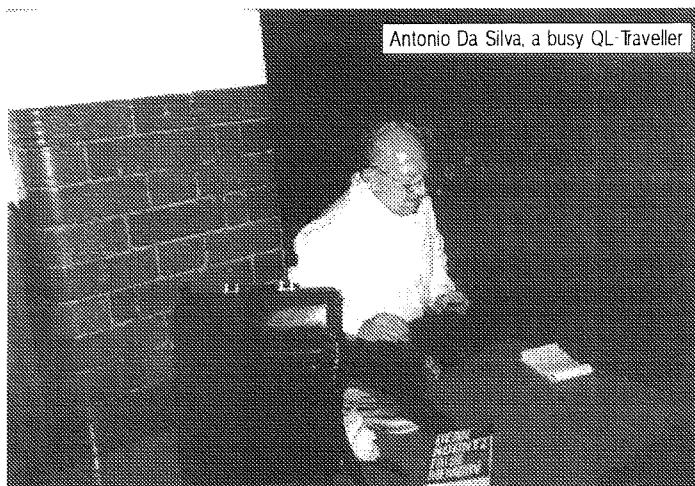
Antonio Da Silva, a busy QL-Traveller

may remember on IBM AT computers. This connector was on early IBM ATX systems, but is now generally not available. The Compswitch has one socket marked æcomputeræ, nearest the fuse, and you simply plug a computer into this socket - such as your PC, and any other peripherals or devices into the other three sockets which are all marked 'switched'. Now you can switch off/on your Scanner, Speakers, Modem, Monitor, even your Hi-Fi and Printer via the computer switch! As Supplied, the Compswitch has four standard UK 3 pin sockets, but Tony told me adaptors can be used in these if needed for other use and the maximum power on the 'computer' socket is 500 Watts, with at least 10 Watts beings required to switch on the 'Switched' sockets. A very clever device indeed, and one that I am sure will sell well, as indeed it did - I know one Ql'er who went home with several of them!!

I was next to Tony, and although failed to match the uniqueness of the Compswitch, had quite a lot of items for Sale. The newest revisions of the Dilwyns Jones QL Collection were available on Disk and CD-ROM, as was the new LineDesign Clipart CD-ROM containing over 5,000 pieces of Clipart. I had a demo beta version of the new Spectrum Emulator CD for the QL, with thousands of ZX Spectrum games and utilities on it, including the ZeXcel Emulator by Ergon. Dilwyn couldn't make the Show as he was preparing for his wedding a fortnight later, but was however beavering away on several new CD's for launch at the QL2000 Show, including the Religion Texts CD, which will contain over 150Mb of religious text files from all

the various religions of the world including everything from the King James Bible, to the Holy Koran, to the teachings of Buddha and Confucius, and even a Bible Dictionary and Some Christian Jokes used by Vicars and Priests etc. in their Sermons. Dilwyn also struck on an idea to do a CD containing the full QL Club International Newsletters, from beginning to end. I also had a lot of novelty mice (!) which sold well throughout the day, and a small range of QL hardware and second user bits and bobs.
At the top of the room, just in front of the stage area was Jochen Merz of JMS. He as ever had an ever growing queue in front of his stand with a lot of people getting upgrades to his vast range of software. Jochen was demoing the latest version of QPC at his stand, and although not quite ready for public release at the time, it was showing a lot of promise. QPC can now be used in a Window under Windows (yes, confusing isn't it?) and can therefore be moved around the screen on the Windows desktop, without the need to take over the entire screen unless you want it to. The colour drivers are due to be implemented soon too, and will probably be available at the QL2000 Show or by the time you read this. Jochen was also selling the new "The Wall" game by Wolfgang Lenerz, as reviewed last issue and continued

nued to amaze me by selling blank recordable CD-R disks for less than 5.00 for a box of ten - they cost three times this much at my local discount computer shop!! His DD and HD disks were similarly very cheap indeed.
Of the traders, last but not least was Geoff Wicks of Just Words! Who was next to Jochen at the front of the house. As ever he was selling his range of programs for the word sleuth and Word processor user alike, and Geoff spent much of his time bent down in front of his Laptop giving demos of his products.
Sadly missing from the traders were Rich Mellor of RWAP Software, who is still very ill and will unfortunately miss QL2000 also (Roy Wood is bringing some of his wares to QL2000 on Rich's behalf, and is looking for someone to maybe help out in the collection of stuff from Rich, if anyone can help, contact Roy) Bill Richardson of EEC was also not at the show as he too is also recovering from an Angioplasty operation a while ago - hopefully he will be back on top by QL2000.
Ken Brickwood, Malcolm Cadman, and other members of the London Group manned the Kitchens for the day, and provided a range of Sandwiches, Rolls, Cakes and Biscuits all at very
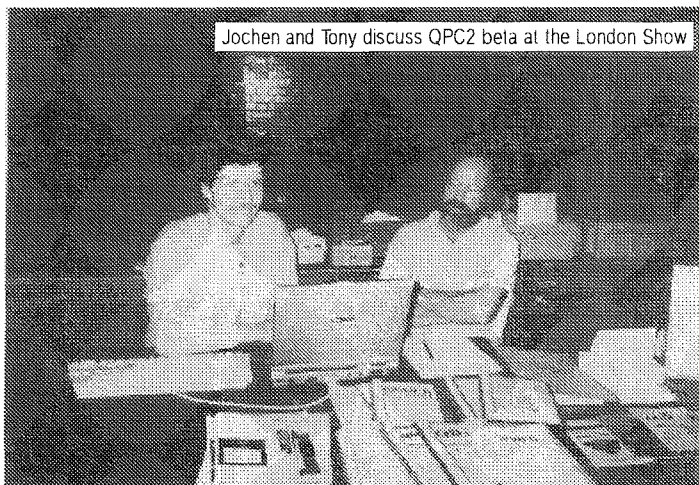
Jochen and Tony discuss QPC2 beta at the London Show

cheap prices, including the usual tea and coffee (Including some very good Organic sauces - thanks Malcolm!)

They also stretched themselves to looking after the bring and buy stall, containing items for sale from the collections of the London members, including items that once belonged to the Late Cyril Phillips, who sadly passed away a short while ago, and had been donated by his wife I think. Ken

Brickwood brought along a lot of items from his own house too, and sold a lot bits and bobs - he had at least a half dozen Digital Diaries for sale - is this man the most organised on earth?

Towards the mid afternoon, a former QL user called David O'Connor brought in several large boxes of QL related gear in an attempt to sell it all off as he hadn't used it at all in years. QL'ers gathered like vultures from every corner of the hall in a matter of seconds at the prospect of a potential bargain. There was a wealth of older QL software and Hardware there, including loads of Microdrive based stuff and a lot of people left the area with large smiles on their faces, as most items were yours for a couple of quid. These included a new young QL user, called Tarquin Mills, who bought a complete QL system from David very cheaply. Nice to know the QL can still attract new faces at this stage of the game - very encouraging.

Even I went home happy, as I managed to find a QJUMP QEPIII Eprom Programmer interface among Davids stuff for only a tenner - it even still had the original VISA payment slip inside the box for 139.95... :-))

As I was staying with Malcolm Cadman in his flat in Deptford, I stayed back and helped tidy up after the Show, and we then went around the corner to Ken Brickwood's house for a cuppa. While the Tea brewed, I began playing with Ken's Macintosh Powerbook, and under the 'Virtual PC' program I managed to get QPC running on a Mac!! Basically what we had in front of us was an Apple Mac emulating a PC which in turn was emulating a QL - and it worked. It even ran at a usable speed (a little slow, but usable nonetheless) and we were able to view the QUANTA Libguide disk on it easily.

An interesting way to end a QL Show, and it just goes to show the platforms on which a QL/QDOS/SMSQ environment is now available. With the London Show over, we all agreed to hold another next year, as it had been much too long since the last one. Thanks to the London Group for organising it, particularly to Ken and Malcolm. Then our thoughts turned towards QL2000 - hoping it will be everything we all hoped it will be - stay tuned for the answers....

New E-Mail Addresses

Thanks to Bruce, the domain qltoday.com can be used for QL Today. He also introduced various, very useful E-Mail addresses - very easily recogniseable! You can reach individual members of the QL Today team with the following E-Mail addresses:

bruce.nicholls@qltoday.com
dilwyn.jones@qltoday.com
jochen.merz@qltoday.com
roy.wood@qltoday.com

If you want to contact the editors (in their function of being one of the editors), please use

editor@qltoday.com

which will reach both Dilwyn and Bruce.

The Editor Gets Hitched...

By *Darren D. Branagh*

Saturday the 16th of September was a big day for the Editor of QL Today, one Dilwyn Lloyd Jones. He decided to tie the knot with his girlfriend, Christine.

Yours Truly was best man, and I even manged to remember the rings The entire day (thankfully) went without a hitch, and Dilwyn now has a new first love in his life (instead of his QL). The venue was the Registry office in Bangor, near where Dilwyn lives, followed by photos in the beautiful grounds of Bangor Cathederal. The reception was held at the lovely Eryl Mor Hotel in Bangor, which had a beautiful sea view overlooking the pier and puffin Island in the distance. A really delicious buffet lunch followed conprising seafood, samosas, drumsticks and ribs and steaks followed by some of the most wonderful desserts I've ever tasted (I tried most of them!)

All in all a great day, and one I'm sure Dilwyn and Chris will remember - especially the speeches!!

Dil and Chris in the grounds of Bangor Cathedral

Dilwyn and his new wife, Chris, in a gazebo on Bangor pier

Dil and Chris with the wedding car - don't worry, there is a QL in the back seat :-)
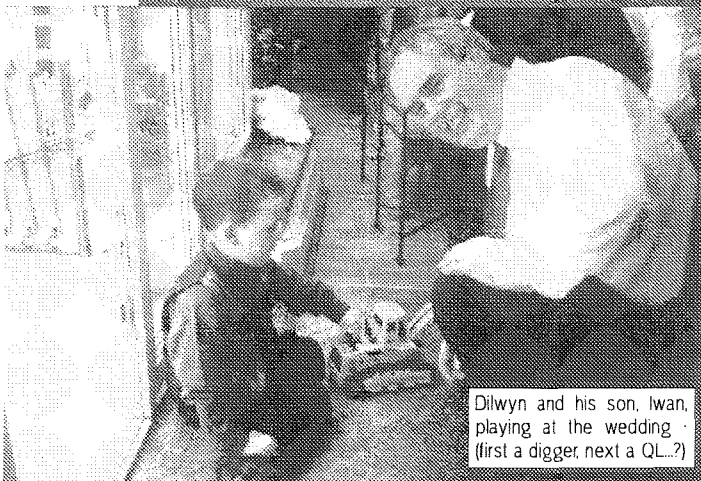
Dilwyn and his son, Iwan, playing at the wedding - (first a digger, next a QL...?)

Dilwyn and Chris outside their hotel room after the wedding

Inside GoldFire - Part 2
"Nasta"

1.2.8 Comparison between GoldFire and original QL bus specification

Several lines have changed names and in some cases functions. Great care has been taken to keep the modifications completely compatible with every known existing peripheral that works with a Gold Card or Super Gold Card. In fact, even some peripherals that do not, may work with the GoldFire. In some cases lines have changed function. This type of change was used for lines that were either unused, or have outlived their usefulness with the advent of previous well accepted peripherals. With all the changes, it is expected that only some very unusual and possibly homebrew add-ons will have problems.

Here is a comprehensive signal name translation table:

QL name	GF name	C	Comment
D0..D7	Dx,Ax,BSx	Y	Compatible for 8-bit access, new features
A0..A19	Dx,Ax	Y	Compatible for 8-bit access, new features
ASL	-WTRQ	?	Not used by existing QL peripherals
DSL	-NTRQ	Y	New features
RDWL	-WR	Y	New features
DTACKL	-TACK	Y	Compatible for 8-bit access, new features
BGL	-WDATA	N	Remove 68008 CPU when GF w. QL motherboard is used
BRL	-ATRQ	N	Remove 68008 CPU when GF w. QL motherboard is used
RESETCPUL	-RESET	Y	
CLKCPU	CLK	Y	
R,G,B	reserved	?	Not used by existing QL peripherals
CSYNCL	-RFSH	C	Compatible for (Super) Gold Card use
VSYNCH	POLL	C	Compatible as poll interrupt source
VPAL	-VPA	C	Depends on motherboard, see below
E	A23,D23	C	Redefined as address line, 6800 chips not supported
FC2	A22,D22	C	Depends on CPU replacement used, see below
FC1	A21,D21	C	Depends on CPU replacement used, see below
FC0	A20,D20	C	Normally not used on QLs
ROMOEH	ROMOE	Y	Depends on motherboard, should be phased out
DBGL	reserved	?	Not used on QL existing peripherals
DSMCL	DDEC8	Y	
IPL1L	IPL1	C	IPC cannot cause int. level 5 and 7 on GoldFire
IPL0L	-HINT	N	Subject to interrupt routing on GoldFire
BERRL	-NINT	N	Not used on QL existing peripherals
EXTINTL	-EINT	Y	
SP0..SP3	GROUND	Y	Were originally connected to GND on QL motherboard
VIN	+5V	C	Compatible for modified 5V systems
VP12	+12V	Y	
VM12	-12V	Y	
GND	GROUND	Y	

NOTES:
C Compatibility:
Y - yes, may also have added features
N - no (usually lines that were not used by existing peripherals)
C - conditionally compatible, see comment and additional notes
? - Unknown, peripherals should not be using these lines anyway

Certain lines that have 'compatibility' status, may not be compatible with their original QL bus definition when a GoldFire is used, but will still retain compatibility with peripherals, and back-planes. This means that they will in general be fully backwards compatible but that peripherals designed specifically for use with the GoldFire may not work with older boards carrying a CPU, such as (Super) Gold Card.

Lines that have a comment that reads 'depends on motherboard' are not used by the GoldFire, but should an older card with a CPU be installed instead of the GoldFire, the motherboard or the backplane might be required to produce these signals correctly for the older card to function. This will not be a problem as long as the 'motherboard' is an Aurora or an original QL motherboard, and either a Qplane or an Mplane backplane is used.

Lines that have a comment that reads 'depends on CPU replacement used' are redefined by GoldFire in a backwards compatible manner and there should be no problems with existing peripherals and motherboards. Should mother-boards be modified to take advantage of the GoldFire signal definition, replacing the GoldFire by a Gold Card for such modifications will render the system inoperable (no damage will occur), because the Gold Card relies on the lines be-having according to the original QL bus signal definition.

1.3 Bus protocols

The GoldFire uses two main bus protocols, the narrow transfer protocol, implementing an 8-bit non-multiplexed QL-like bus, and the wide transfer protocol, implementing a 32-bit multiplexed protocol which uses the same electrical signals, but will not be seen by standard 8-bit peripherals. This enables both types of peripherals to be mixed and matched without resorting to new backplanes and bus connector definitions.
For ease of interfacing common 8-bit ROM or ROM-like memory to carry driver and OS expansion code, a special variation of the narrow transfer protocol is also provided, that will be transparent to existing 8-bit peripherals, avoiding addressing range restrictions that would otherwise apply.

1.3.1 Narrow (8-bit) bus protocols

The GoldFire uses two types of narrow transfers, both based on the same protocol, but with some important differences:

1) The QL compatible narrow transfer protocol. This enables the GoldFire to access QL type hardware in a completely compatible manner. It also offers an extension to the original QL specification by being able to control the speed of the bus. The protocol uses all the usual QL signals in their usual manner:
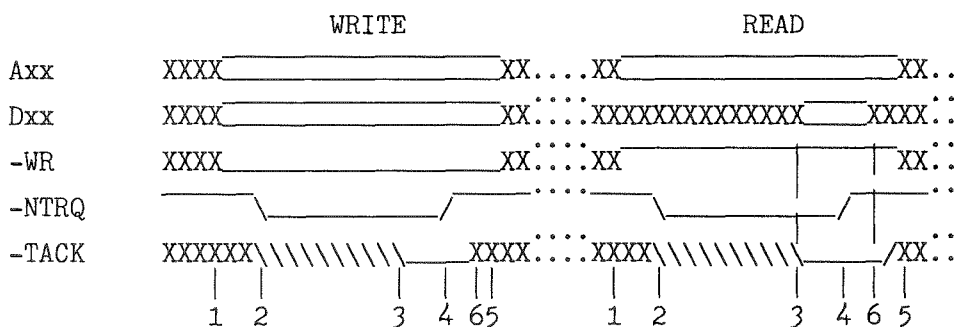
Diagram points:
1..2   Address setup time, data setup time on write cycles
2..3   -TACK dead time during which recognition of a low level is suppressed
3..4   Input setup time, (Data on read, -TACK on any cycle)
4..5   Address hold time, data hold time on write cycles
4..6   Input signal hold time (Data on read, -TACK on any cycle)

Timing data:
t(1,2) = min. 2 CPU clock cycles
t(3,4) = min. 6ns (signal MUST be stable by point 4) (final spec)
t(4,5) = min. 1 CPU clock cycle (final spec)
t(4,6) = min. 6ns
t(2,3) = default 11 CPU clock cycles, configurable from 0 to 15 CPU clock cycles

1 CPU clock cycle is 1/40000000 seconds = 25ns

The QL compatible access cycle has 5 phases:

Phase 1: The GoldFire bus controller outputs the address (and data for write cycles) on the bus, and sets the -WR line according to the direction of transfer, read or write.
Phase 2: The GoldFire bus controller sets the -NTRQ low to indicate that a narrow cycle is requested. The -NTRQ signal is compatible with the QL bus signal DSL.
Phase 3: A response is required from the peripheral in form of pulling the -TACK signal low. This signals that the peripheral has either taken the data supplied on a write or has supplied data on a read. A special dead time feature is built into this phase of the protocol, which prevents recognition of the -TACK signal being low for a programmable length of time. This is done to slow down the bus to speeds that can be safely handled by QL peripherals, in effect, it provides compatibility. The user can choose what the dead time will be (and hence limit the maximum speed of the bus) by using an appropriate address for the access, see '2. GoldFire address map specification'. The -TACK signal behavior is compatible with the standard QL signal DTACKL for QL compatible cycles.
Phase 4: Once the -TACK signal is recognized as being low, the GoldFire bus controller latches the data supplied if it was a read cycle. Simultaneously, the -NTRQ signal is set to high to indicate that the cycle is finished. If it was a

write cycle, the data bus will still hold the data for one CPU clock cycle for peripherals that require long data hold times. The address will also be held on the address bus for at least one CPU clock cycle.
Phase 5: The peripheral must remove the data from the bus on a write cycle, as well as deactivate the -TACK signal when the -NTRQ signal goes high.

Timeout feature:
The QL compatible access cycle also has a timeout feature which prevents protocol lock-up if no peripheral is present to signal an end of a cycle by pulling -TACK low. The cycle will terminate automatically once 127 CPU clock cycles (approximately 2.5us) have passed from setting -NTRQ low. The bus will behave as if the -TACK signal was pulled low externally. Data will be lost on a write, and on a read, data will be indeterminate.

1) The Auto-terminated narrow transfer protocol. This protocol is used to access on-board peripherals, and can be used to interface ROM or ROM-like memory to the GoldFire for driver and OS expansion code storage. The recommended use for this is to access byte-wide ROM, EPROM or Flash ROM on 32-bit expansion boards, providing a cost-effective alternative to using 32-bit wide devices. This protocol is very similar to the QL compatible one, except that the timing is generated internally to the GoldFire, and a different transfer request signal is used to render the cycle invisible to both QL type and 32-bit peripherals:

```
            WRITE                        READ
Axx    XXXX_____XX....XX_____XX...

Dxx    XXXX_____XX....XXXXXXXXXXXXXXX___XXXX..

-WR    XXXX_____XX....XX_____|___|XX..

-ATRQ  ___|‾_____/_|___‾|___|‾_____|‾/_||_
          | |           | |   | |            | |
          1 2           4 5   1 2            3 4 65
```
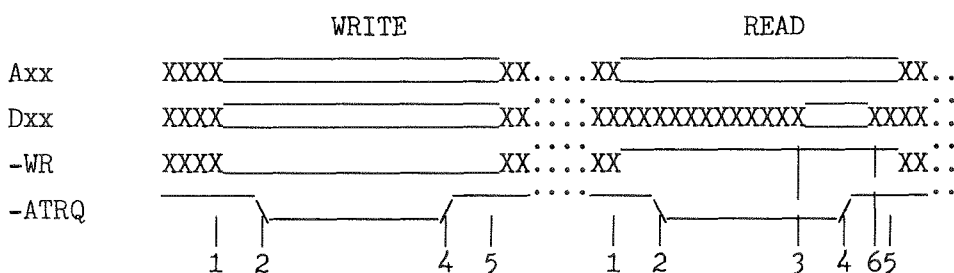
Diagram points:
1..2   Address setup time, data setup time on write cycles
2..4   -TACK active time
3..4   Input setup time on read cycles
4..5   Address hold time, data hold time on write cycles
4..6   Data hold time for read cycles

Timing data:
t(1,2) = min. 2 CPU clock cycles
t(3,4) = min. 6ns (signal MUST be stable by point 4) (final spec)
t(4,5) = min. 1 CPU clock cycle (final spec)
t(4,6) = min. 6ns
t(2,4) = 8 CPU clock cycles, fixed

1 CPU clock cycle is 1/40000000 seconds = 25ns

As can be seen in the diagram, the cycle has 4 phases:

Phase 1: The GoldFire bus controller outputs the address (and data for write cycles) on the bus, and sets the -WR line according to the direction of transfer, read or write.

Phase 2: The GoldFire bus controller sets the -ATRQ low to indicate that a auto-terminated narrow cycle is requested. This means that no response other than to take the supplied data on a write or supply data on a read. The cycle will terminate automatically after a pre-deter-

mined period of time, which is 8 clock cycles, approximately 190ns.

Phase 3: The GoldFire bus controller latches the data supplied if it was a read cycle. Simultaneously, the -ATRQ signal is set to high to indicate that the cycle is finished. If it was a write cycle, the data bus will still hold the data for one CPU clock cycle for peripherals that require long data hold times. The address will also be held on the address bus for at least one CPU clock cycle.

Phase 4: The peripheral is required to remove the data as soon as possible when the -ATRQ signal goes high

About Sprites

Bruce Nicholls

What is a sprite? A sprite, according to the EASYPTR documentation, is a small picture which may be drawn stationary or moved around with the mouse or cursor keys. Several sprite definitions can be linked together to form a dynamic sprite which can last for about 5 seconds and a sprite can be up to a maximum size of 64*48 pixels. Sprites can now be handled in QL 8, the original

mode 4 or 8 definitions, 256 palette mapped colour definition, 256 grbgrbgm colour definition and native colour (See the previous colour article on the colour definitions called GD 2). If you use any pointer environment program the sprites are the items shown on screen that usually perform a function within the program. The most obvious ones being the move, resize and sleep icons

Now there are only three programs that I know of which allow you to 'draw' the sprite

definition on screen without having to resort to hand coding the sprite definition. The QPTR pointer Toolkit contains a program called EDSPR, this is a basic program and can at present only generate QL 8 sprites. Easyptr II contains a sprite generator called EASYSPRITE which is a true executable program that can only produce QL 8 sprites at present. But a new program from Jerome Grimbert is now available called SPRITE EDITOR FOR GD2, as you can probably guess from the title this can generate sprites in all their different flavours.
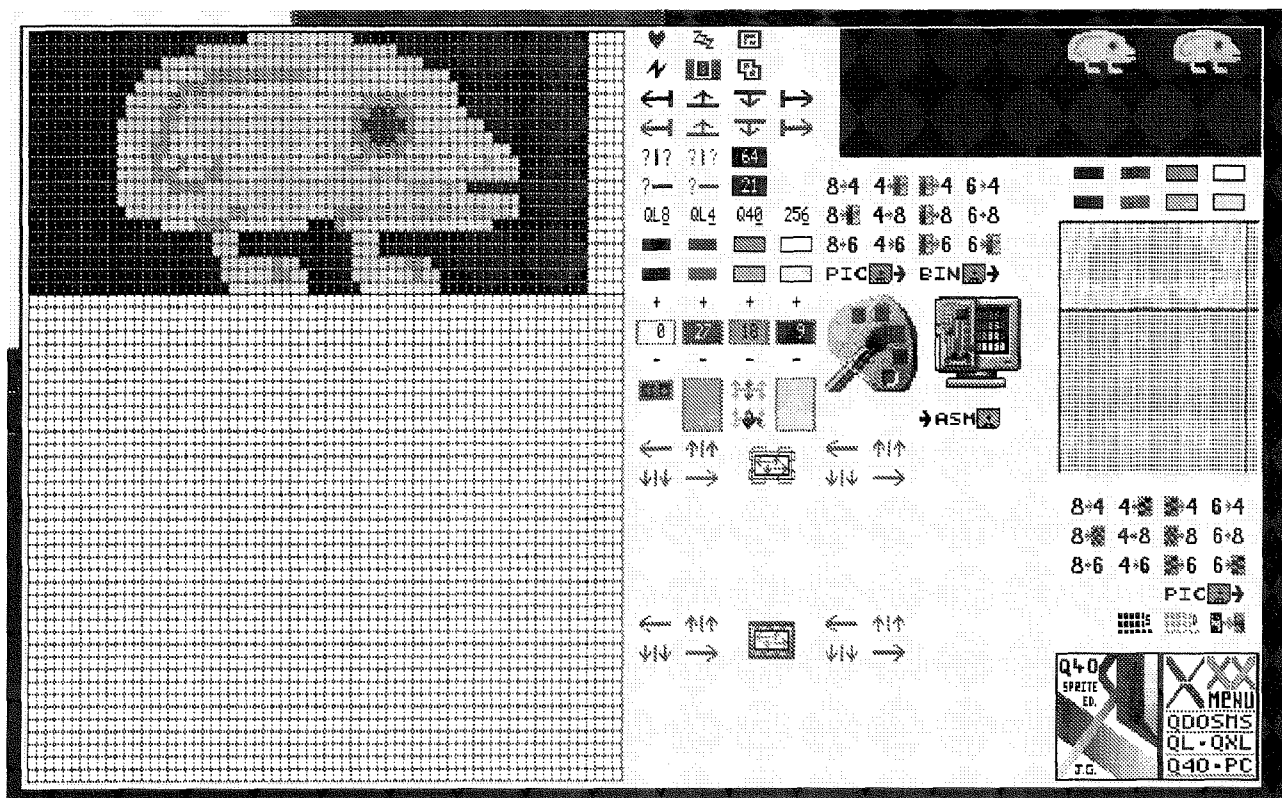
When you start up the editor you are presented with a screen similar to the ones shown on this page.

The main area on the left of the screen is where you 'draw' the sprite you wish to define. In the top left hand corner of the screen are four smaller images of the sprite you are currently drawing. Why four? well this program allows you to copy images between the four different types of sprites as described above. You can design a sprite for the native colours and then with one button press on a conversion icon (4-›8 etc situated below the four boxes) convert it automatically (colour loss accepted) to another mode. This is simplicity itself and will allow for easy conversions of old sprites to the new formats. If you are adverse to redrawing the sprites this program has two methods for reading in old sprites. You can give the program a _pic file and this will automatically be loaded in and converted to a sprite. The alternative method is to load a program containing the sprite you want to use into the editor. The editor reads the file looking for sprite definitions and prompts you if you would like it to be loaded.

Once loaded into the program various options are available for manipulating the sprite image. Colour can be chosen from the standard QL 8 range shown on the screen. If you are working on 256 colour sprites by clicking on the icon looking like a monitor you can choose from the 256 colour palette which presents itself. If you are working on a Q40 (native) colour sprite clicking on the painters palette icon bring up a lovely colour wheel showing all the possible colours available, by moving the pointer around within the colour wheel you can choose the colour you want to use. Once the colour is chosen you are given the opportunity to decide on the intensity of the colour simply by clicking on the intensity chart next to the colour wheel. When a colour has been chosen you can easily change it by pressing on the + or - symbols above the white/red/blue/green icons which will change its native colour number.

The sprite image can also be manipulated by splitting the screen into areas with red and black lines which then allow the sprite image to be shifted to the left/right/up and down by using the arrow keys. There are various other features that I haven't had time to explore but my first Q40 sprite was generated in about an hour of use of the program without a manual.

Once you are satisfied with your sprite design you can either save the image as a _pic file or convert it automatically to an assembly file for use by QMAC or AS86.

This program is a delight to use and a lot of thought has gone into the design to make it as easy as possible to create a sprite definition, all credit must go to Jerome for producing a much needed piece of software that works so efficiently.

The zip file containing the editor is available from **www.crosswinds.net/~grimbert/**
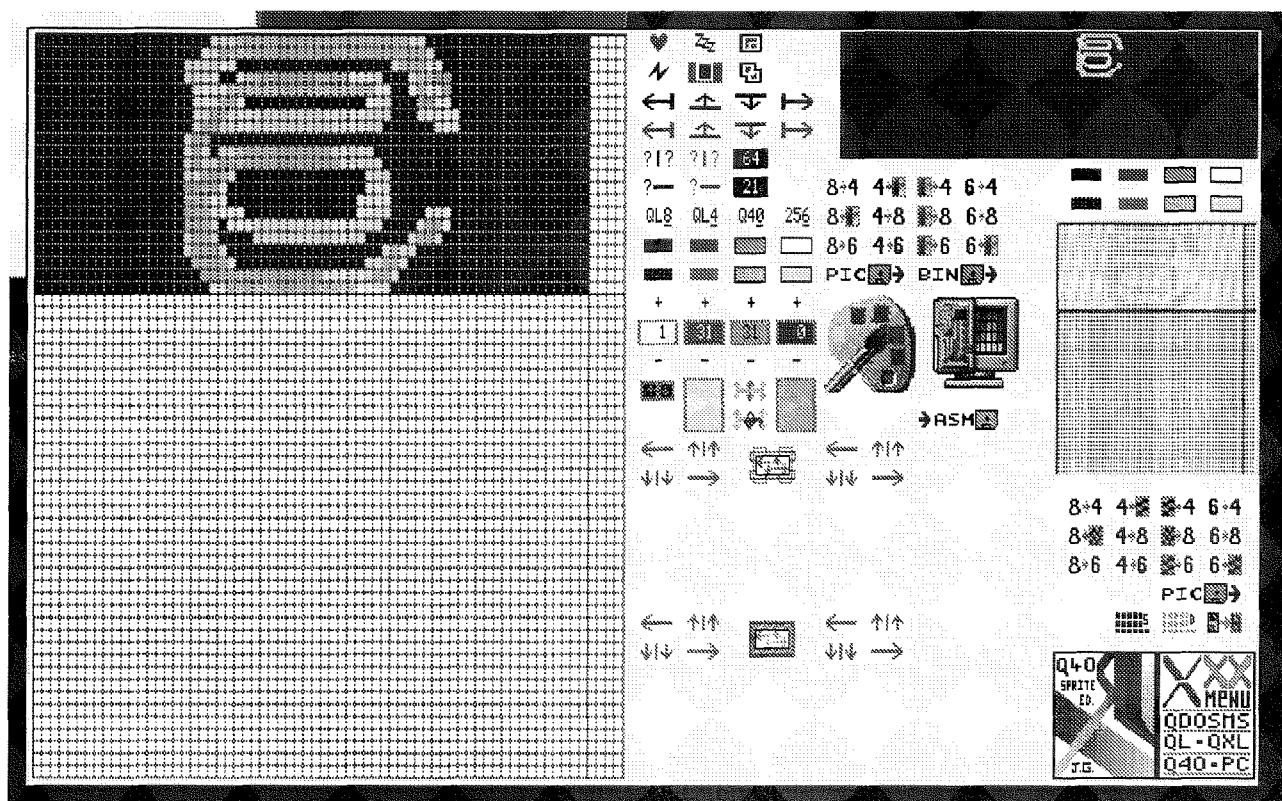
A Useful Windows Program

Norman Dunbar

I could get seriously hung drawn & quartered for this article, but here goes ....

A long time ago, in a magazine far far away, a program was printed which was written by Dilwyn Jones. (I think I have got the time sequence correct here, but I am sure to be informed if I have not!) This program was written in SuperBasic and converted a Quill doc file into a plain text file which had all of the Quill control codes replaced by any text you wanted. Bold on could be replaced by '‹BOLDON›' and bold off by '‹BOLDOFF›' etc, leaving something like this:

```
This is ‹BOLDON›Bold
text.‹BOLDOFF› This
is‹ITALICON› Italic
text.‹ITALICOFF›
```

and so on. This file could then be imported into another word processor. This was useful if the new word processor didn't allow direct importing of Quill files.

Then Phil Borman took Dilwyn's program and converted it into C using C68. This was a much faster version but did require that you had to have C68 in order to make any changes if, for example, you wanted to change ‹BOLDON› to just ‹BOLD› etc. Then I came along with a need to convert a number of Quill doc files back into plain text with all the control codes removed, and

Stripper was born. This has been available for some time on various BBS and most recently on Dilwyn's Web site **(http://www.soft.net.uk/dj/software/other/striperc.zip)** This is the QDOS version and can strip control codes from any Quill document.

Recently, I was at work having my lunch break and I had a few zip files on my drive which had QDOS software in them. I fancied a bit of light relief so I opened up one of the Quill doc files to read in my editor (PFE - excellent) but of course all the junk from the header was there and the lines spanned over the width of the screen and when I re-formatted them, I got everything in one big lump.

Totally fed up of this, I had a thought, I have Borland C++ Builder on my PC at work as I write utilities for our development team to use, and I wrote Stripper in C way back then. If I could combine those two thoughts, then I could have a

useful (to me anyway) utity that I could use when I needed to read through a Quill doc file which I was at work. Stripper for Windows was born.

I downloaded my own source from Dilwyn's web site - soon to be renamed Christine's Web Site ;o) - and converted it to run under Windows and to produce a text file. This was fairly easy. Then I decided that it would be nice to be able to print the Quill files as well. Unfortunately, printing from Windoze is a nightmare but the joy of Borland's C++ Builder is that being component based, you can write your own components and add them to the environment. This I had to do anyway as one of our work utilities also needed some printing abilities - if the same piece of code is needed in two or more applications, thyen it is crying out to be 'standardised' and built into a component. So it was.

Now we have a Quill conversion program that prints and saves the Quill file to a text file. So far so good. I tentatively mentioned on the QL List if anyone else would be interes-
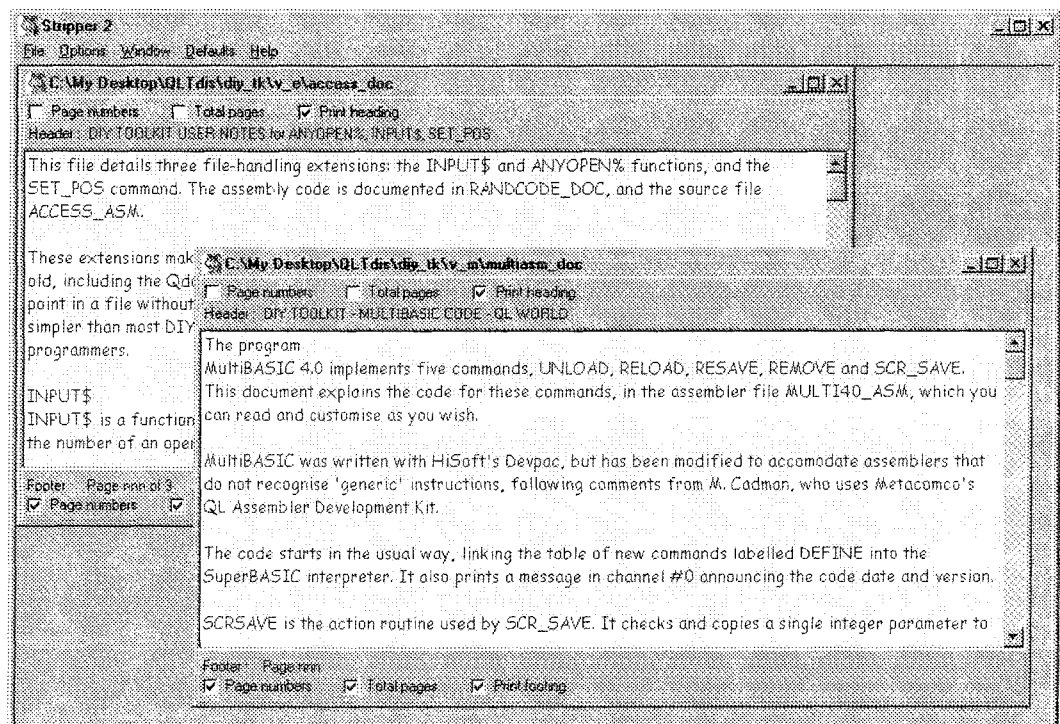
TF Services

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 requd). Send disk plus SAE or two IRCs

MKI...£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)

QL SPARES

Keyboard membrane ....£24 (£25/£26)
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 (£11.50/£11)
Power supply (sea mail overseas)....................£12 (£17/£21)

ted. The response was almost phenominal and many requests were furnished. I even had to start a stripper mailing list to make sure that everyone got the latest versions.

There were lots of favourable comments received by the initial beta testers, I mean, users and I suggested that maybe MDI (Multiple Document Interface) would be a good idea - again, I was inundated with almost 4 requests for an MDI version! (MDI is Microsoft speak for 'having lots of documents open at the same time in one application).

On my next lunch hour, I converted Stripper to MDI format and released it to my happy users. Then I added more bits and pieces - as is the need with Windoze programs - and released these as well.

The final version, at the moment, has been in use by me for some time (almost daily) and I have not yet had any complaints from my user group - so it must be ok!

This latest version allows a Quill doc to be loaded, and saved as text. The text has been set to 80 characters per line with a DOS format end of line (CR/LF) rather than a QL (LF) one. This was requested so I had to do it.

Printing is allowed, and due to differences in the Quill page length, you are allowed to edit headers & footers to add the page number (and other bits) so that regardless of the original format, you get the correct page numbers for whichever printer you happen to be using under Windoze. Fonts can also be changed and saved as default etc etc. There is even a help file (gosh!) Background colours (or even color for our American cousins!) can be selected but don't get printed - there is no point really. And, as they say, 'and lots lots more!'

I have enclosed a screen dump of the program in action (at work) as a gif file, which I think Jochen can convert and include in the magazine. Hopefully, you will like what you see and take a trip to Dilwyn's web site to download it. Remember, this is a Windows program, not a QL program.

One problem that we had was when someone in my user group - sorry, I can't remember your name :o(- had fairly major problems with the program. I was puzzled, but suggested turning off 'Active desktop mode' if it was turned on. This cured all the problems. I have also had problems at work with other programs (some mine, some Microsofts) with Active Desktop and turing it off always fixed the fault. I suggest turning this option off if you have it on!

I hope you will all forgive me for taking up space in a QL magazine to discuss my latest masterpiece, but if you ever use a PC to do your downloading etc, then why not give this program a try - you might even like it!

I have deliberately NOT revied my own program, just told you enough to whet your appetite. If anyone wants to do a full review, then download the file from Dilwyn (or Tony Firshman's web site) and use it. Then write an article and send it to **Editor@QLToday.com** where it will be well received.

Once again, apologies to readers who don't want/like/need to see anything at all about other platforms in our magazine.

Small ads

For Sale

First off this issue I think I should explain the cryptic end to the last column. I got as far as the honourable mentions in dispatched section and then fell foul of the 'too many computers' syndrome. I had the finished column on my laptop but then sent the one on the desktop machine which did not have the last section on it.

Not only that but, being the modest person that I am :-), I did not read my own column in the printed issue so I was very puzzled by the various comments on the user group when Norman asked why he got the accolade.

So here, for those of you gasping for the final part, and, of course, to give Norman his due, is his Honourable Mention......

Last issue the accolade goes to Norman Dunbar for a little piece of code aptly called The Stripper. This actually runs under Windoze (either 95 or 98) and can be used to extract all of he control codes and other junk from a Quill file and displaying it in plain old ASCII format. This is a very good addition to the growing armoury of files available to those who use both the QL and other systems and, best of all for many of you, it is free. It does not have the facilities or flexibility that Geoff Wicks programs have and it will only convert files from Quill to ASCII whereas Geoff's programs will convert both ways and preserve such things as underlining and bold/italic styling as well as handling Text 87 and

Perfection files.

None of this is in any way meant to detract from the appreciation of this program. The program comes as a zip file which can be unzipped by the standard Windoze programs. It is a fairly standard executable without any need for an installation program or other external support files. Once you have unzipped it all you have to do is to double click the '_exe' file and off you go. If you are a QPC 2 user you can use the QXL.WIN Explorer to take a file from your QXL.WIN file and save it somewhere on the Windoze hard disk. Even if you are not using QPC 2 you can always use SMSQ/E (Now you have to be using that!) or Discover to copy a Quill file onto a PC Format disk. Once you have opened the file in 'The Stripper' the program goes to work removing the codes and the text scrolls past a a great speed - even on my cranky old 233MHz laptop. The resulting code is presented in a format as good as the original Quill document and can be saved in plain text format ready to be imported into a Windoze Word processor.

The program has a number of option including the choice of font for the text and the choice of background colouring. All in all this is a boon to people who have only a QL at home and want to take documents to work all you need to do is to install this little beauty on your work PC and the text is there at your fingertips (so long as you can write to PC disks that is).

The program also has a nice line in saucy icons with a twist in its tail.

Norman said that he did this in his lunch break at work - maybe he should go on a diet and produce a few more little gems like this.

Normal Service Resumed

Right then, on to the business in hand. A couple of new items were announced on the user group news list. The first is a new C compiler ported over from the LINUX platform by Richard Zidlicky. This was the source of frantic activity by some of the user group regulars. So far the jury is out about this new addition to the C compilers armoury but, in the ensuring debate a few people seem to question its speed. I still feel that a C compiled program is inherently slower than a machine code one but saying this always seems to create a loud discussion.

Programmers on other platforms use C a lot but then they have oodles of excess processor speed and memory to hand they don't really care about making it too tidy. I am no programmer so I cannot speak for my own experience but I would like to see more native machine code stuff available for us. This, of course, brings us to the problem of programmers.

These days, it would seem, we have a distinct lack of people writing QDOS/SMSQ programs. There is a lot of activity going on porting programs and utilities from other platforms and I am in no way knocking that but we also need to get more programs written directly for our own platform.

Don't get me wrong here, there is a lot of good work being done, especially by people like

Theirry Godefroy with his excellent utilities and Micro-Emacs and by people like Wolfgang Lenerz, Geoff Wick, Rich Mellor, Simon Goodwin, and George Gwilt among others. There is. however, a feeling around that we are missing out and we should be able to have more. True, the PC has all of the flash and verve that our platform lacks but then it has a multimillion dollar industry behind it. In our case we have a handful of dedicated programmers and talented amateurs.

The Hardware Store

Right now the pace of hardware developments seems to be overtaking the software one but that is probably more down to a lack of a proper framework for the colour Pointer Interface. Nasta is bubbling over with enthusiasm for the Goldfire - so much so, in fact, that we may have to pin him down to stop him running into too many redesigns and get the prototype off the ground and the Q40, in spite of its teething problems, is proving to be a very popular machine.

Of course all of this is for nothing if the software is not there and that becomes increasingly difficult. The things that most people seem to want involve some sort of communication with the outside world. Even things as simple as printer drivers are making life harder and harder for the writers of QL software as the manufacturers put less and less software hooks into their machines and leave more of the functionality to the O/S. This would be OK if they allowed the information out freely but many of these manufacturers want to keep this all close to their chests.

In the course of my current job I had to call a chip manufacturer to ask about an I/O chip used

on a new PC motherboard and they would not give me any solid information. Try buying a new printer to use with your QL and you will be amazed to find out that there are very few that will do the job.

Ink Squirters - again!

Even Epson have very few models which still have the vital ESC/P2 driver interface. (I was recently told that the company that makes Epson printers made the first daisy wheel printers and, when the dot matrix first came out decided to call the company Epson - Son of Electronic Printer - can anyone confirm this?)

There are currently only two models of Epson Printers which have the vital ESC/P2 drivers. The Stylus Colour 900 is one of the better inkjets with an impressive speed and a very high resolution. All of the models below this are driven by the ESC/P raster driver which is wholly dependant on the O/S for its function. You may be interested to hear that there is also an EPSON-laser printer with the ESC/P2 driver, the EPL N5700. This is not too expensive (well it does have a recommended retail price of +VAT, but you get it cheaper than that) so if you want a good quality mono printer for your ProWesS output then check this one out. Don't forget, though, that programs that do not utilise the full possibilities of the ESC/P2 codes will only produce an output similar to that of the driver you are using. If you are printing your files from Quill, say, and using the FX 80 driver you will only get an output based on the 24 pin driver - not a good compromise for a laser printer. Although you can change a lot of the attributes of the driver itself the basic underlying code will only give a 24 pin type resolution.

Text 87

Text 87 does have a very good driver for the EPSON ESC/P2 drivers and, with this, you get a very good selection of fonts. There are, however, other problems which will come to light more and more with Text 87. When Fred Toussi wrote the program (in 1987 - believe it or not) there was no such thing as extended resolution so the program was stuck in 512 x 256 resolution. Later, when the Atari QL emulators came out, he updated the program to allow the use of the new high resolution screens. When the Aurora was released a few years ago it brought another problem to light. The way in which Text 87 calculated the screen size was fine for both the Atari and QXL emulators but fell over spectacularly when the new high resolution Aurora screens appeared. Phil Borman popped up and wrote a very neat patch to allow Text 87 to run on the Aurora so all was well but, now that the new colour drivers are beginning to appear the problem is arising again.

There is not a problem with the program's operation, as such, it is just a display corruption but that is just as bad as the program not working at all. It occurs because the old 'mode 4' resolution now only exists in high resolution/high colour mode. Mode 8 still exists but this is not really useful for this purpose.

There have been a couple of suggestions on how to deal with this problem. The most obvious way to deal with it would seem to be to get Fred Toussi to re-write the interface to use the new drivers but this does not seem to be happening so we will have to fall back on a different solution. In the absence of someone to write a

patch for the program what we need is a special mode for Text 87 to run in. The obvious answer is to get rid of mode 8 because I am reasonably sure that very few of the people who will be buying the new versions of SMSQ/E will either, not be running mode 8 applications or willing to give up this option for a better, four colour, high resolution, mode 4 for the programs that will not use the newer modes. This is something that we need to address as soon as possible.

How Soon We Forget

Another recent exchange on one of the internet user group lists was between Theirry Godefroy and Jonathan Hudson. Theirry was expressing the view that UQLX was 'full of bugs' and Jonathan leapt to its defence. As usual, in situations like these, the old hobby horses get wheeled out for jousting and the efforts of some people to extend the basic QL's abilities came under fire. UQLX, like most of the free QL emulators runs as a JM or JS rom with the additional ability to use the Toolkit 2 extensions if you can extract them. I cannot say anything about UQLX because I have no experience of it but I do know that given the choice I would always use SMSQ/E over any other QL O/S.

The issue arose when Theirry said that using the 'MAKE_DIR' command in UQLX caused an error. Theirry said this was a bug and Jonathan said that it was not. As always it was a 'six of one and half a dozen of the other' situation. The ability to create subdirectories comes not from the Toolkit 2 routines but from the extended level 2 drivers that appeared on the later versions of the Trump

Card, and the ATARI emulators. It has existed and been part of our environment for so long that most of us have forgotten where it came from and when it arrived. The ironic thing is that there is a bug in UQLX which also relates to the same area and this argument between two QL giants pinpointed it.

The other day I had to set up a machine for a QL user who was really still in the Quill and Archive stage of life. I was amazed how much I had forgotten about the programs I had started out with.
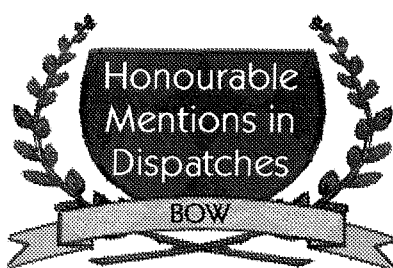
Honourable Mentions in Dispatches

This issue the accolade goes to someone whose work you all have yet to see. The only evidence that it exists at all is an email sent to the user group on the internet a few weeks ago. Jonathan Dent has sent the first email to be generated on a native QL with its own TCP/IP stack. This may not look that momentous I grant you but it will open the door to many QLers who want internet access. These are just the first steps but, in my opinion, very important ones and very deserving of this issue's accolade.

Once the TCP/IP stacks are released to the public we can get to work on a dedicated Email client and possibly a web browser. Email ability should be very easy and I believe that Jochen has suggested that QD could form the basis of the

package for this but web browsing is more problematic. There are still many hurdles to be overcome before we are all online with our QLs and, for many of users it will mean investing in a superHermes and maybe a Super Gold Card (if you can get one) but this was 'one small text message for a man - one giant Quantum Leap for QLers.

Apology

Finally an apology for the shortness of this column this issue. The problem was partly that I had to get it done before Jochen set out for the Italian show by car and partly due to the fact that my wife is now just over six months pregnant with our new daughter (yes we opened our Christmas presents early and looked at the pictures). In her current 'large' condition she cannot do too much so I am doing a lot of 'domestic stuff'. By the time the next magazine comes out we should be a bigger family and my wife back to her sylph like figure.

Editor's final word

You may wonder about the fact that Roy pinpointed most of the issues which matter to many QLers now - and that we already mentioned many of them in other articles in this issue of QL Today.

We must be psychics, because Roy did not know any of the articles (neither the contents, nor the subjects) which you find in this issue - or, alternatively and more likely - we all think about the current problems and their solutions. We can't solve all of them immediately, but YOU may be able to help - e.g. by dealing with the printer situation.
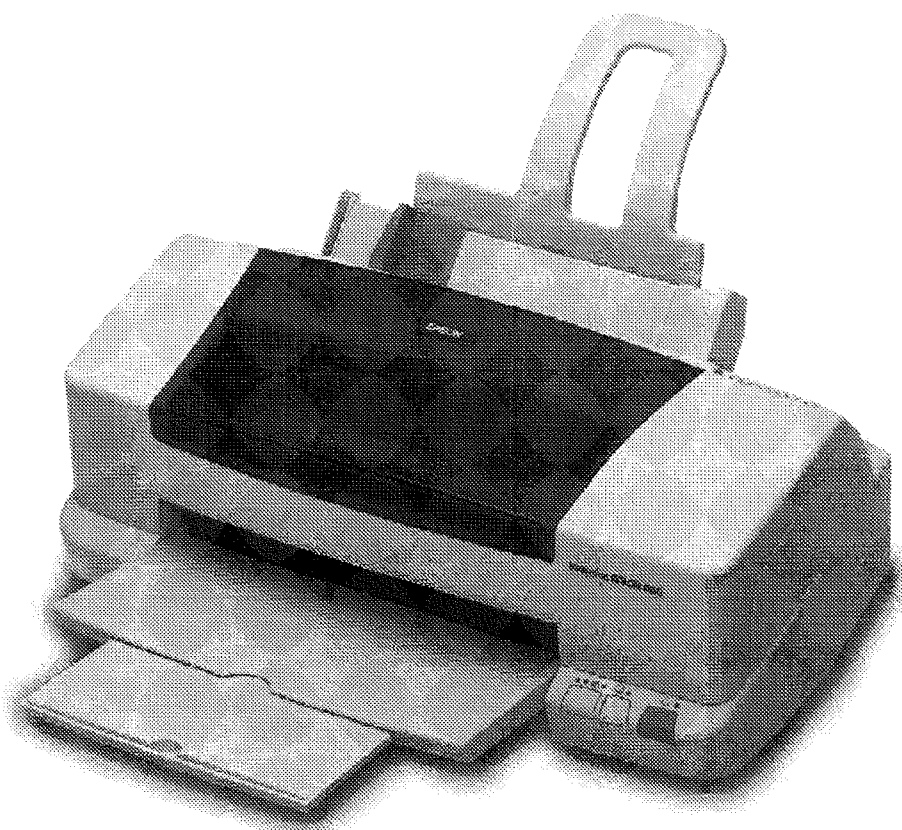
The QL Show Agenda

Latest News: EPSON Stylus Color 880

Look what I just discovered on the EPSON Website! A brandnew model! With a resolution of 2880 dpi, a printing speed of up to nearly 12 pages per minute (b/w) and a price of - yes, you read correctly - about £130, is there anything else you can wish? Yes, of course, we would like it to support full ESC/P2. EPSON may have heared our wishes - it does say on the website and on the fact sheet that it supports ESC/P2. What surprises me a bit is, that it only supports 2 bitmap fonts and one (scaleable) - so it has to be ESC/P2, but with a limited range of fonts. Anyway, this is pretty good news because we should be able to connect it to the QL and use it with Quill, graphics programs etc. without the need to change the driver. However, this does still mean, that an ESC/P2 Raster driver would be more than useful too!

To all readers, show organisers, authors!

Please inform us as soon as possible about forthcoming QL shows next year. The earlier we can inform our readers, the better.
The deadlines have moved more and more towards printing time - this puts a lot of additional on me, it really does. To help us in providing you with this magazine with the same quality and to make sure we can deliver QL Today to you in time, then please try to send material to the editors as soon as possible.