

QL Today

Volume 3
Issue 2
July/August
1998

ISSN 1432-5454

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

menu_

```

moveq    #sms.info,d0    ; get info
trap     #do.smsq
cmp.l    #sysid.sq,sys_idnt(a0) ; is it SMSQ?
bne.s    ignore_check
tst.l    beq     ; job 0?
beq     ; yes, initialise

```

New Series!

680000

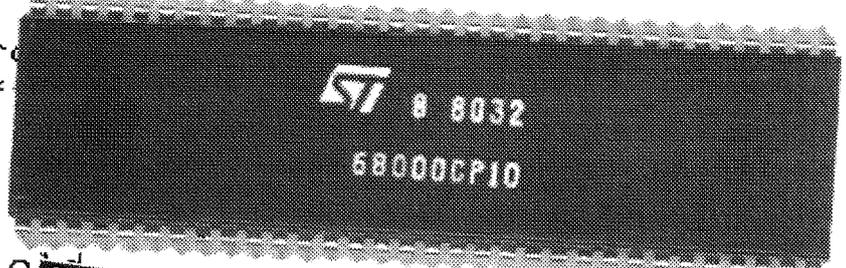
Assembler!

channel ID 0 MUST exist!

```

moveq    #0,d0
and.l    d0,d0
bne.s    start_err
moveq    #0,d0
and.l    d0,d0
bne.s    start_err
moveq    #0,d0
and.l    d0,d0
bne.s    start_err

```



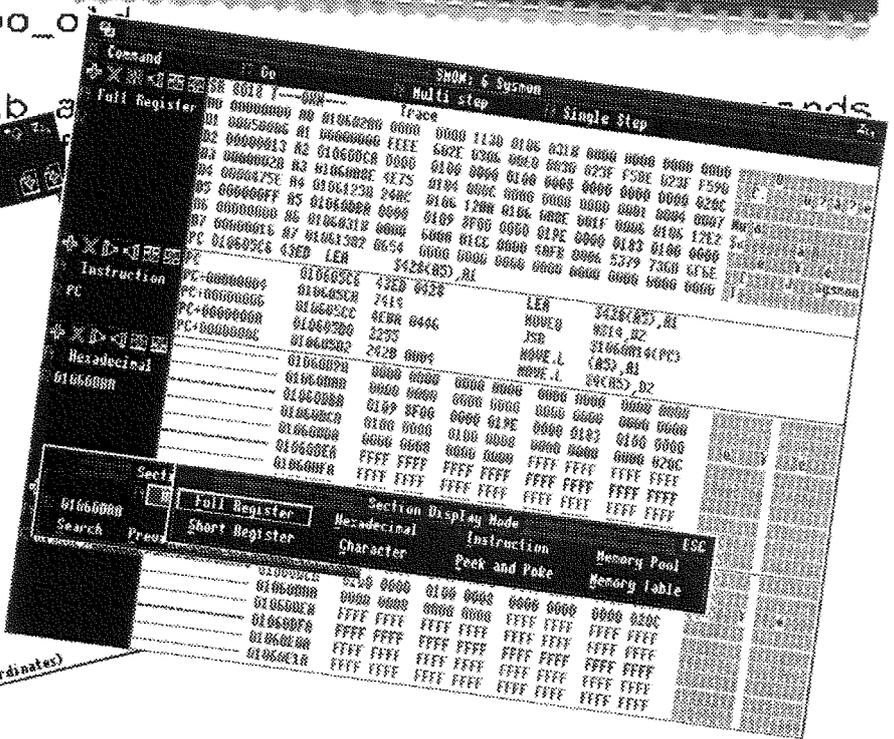
```

lea     proc_tab,a0
jsr     ut

th_menu

start
wman_
xlea   met_wver,a1

```



Contents

3	Editorial	Dilwyn Jones
4	News	
6	Snake	Dilwyn Jones
8	Harddisks - Part 2	Mark Knight
14	The Leap Back	Adrian Ives
15	MTOOLS - A Review	Timothy Swenson
16	The Pandora Case	John Peace
18	Assembly Language Programming - Part 1	Norman Dunbar
21	HP-Deskjet UNDER CONTROL	Wolfgang Uhlig
25	QLAY 0.85 - First Look & More	Al Feng
29	The 1998 USA QL Show	John Impellizeri
31	It could be You!	Geoff Wicks
34	Gee Graphics! (On the QL?) - part 5	Herb Schaaf
37	What are all these addresses? - Part 2	Nasta (Zeljko Nastasic)
39	DBEasy v1.6 - A Review	Al Feng
43	For your Eyes Only!	Jérôme Grimbert
47	Geoff Wicks Replies	Geoff Wicks
48	Parallel Printer Interface Plug Adaptation	Al Feng
50	Electronics for QL Peripherals - Part 2	Stuart Honeyball
51	Q40 - More Hardware Details	Peter Graf
53	Letter-Box	
54	Indenting BASIC Programs	Dilwyn Jones
57	Byts of Wood	Roy Wood

Advertisers

in alphabetical order

Jochen Merz Software	45,49
PROGS - Van der Auwera	55
QBox USA	13
QBranch	32,33
Quanta	27
Qubbesoft	13,17
TF Services	23
W.N. Richardson (EEC)	53

QL Today

ISSN 1432-5454

German office & Publisher:

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

English office:

Q Branch Tel. +44 1273 386030
P.O. Box 7 Fax +44 1273 381577
East Sussex email: qbranch@qbranch.demon.co.uk
United Kingdom BN41 2ND

Editor:

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

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque, Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

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

Article and Advertising deadlines are as follows:

Issue 1: 15 April
Issue 2: 15 June
Issue 3: 15 August
Issue 4: 15 October
Issue 5: 15 December
Issue 6: 15 February

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

This magazine is copyrighted and all material published remains the property of **QL Today** unless otherwise specified. Written permission from **QL Today** is required before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

This month we take the plunge into assembly language programming with a brand new series by Norman Dunbar. Many readers have asked for such a course, so we offer a big thank you to Norman for undertaking such a massive task. This will be a major series, hopefully, so I hope you enjoy it and find it useful.

It's fairly quiet on the QL front this month, as the rather short news pages will show. No doubt there are people out there beavering away as ever! So sit back and digest the useful series from our regular contributors like Mark Knight, Nasta (the cartoon wasn't my idea, honest!), Roy Wood and Stuart Honeyball. Thanks, lads, where would I be without you?

We are in a period of consolidation now. Work continues on the Goldfire, Colour Drivers, QemuLator for PC and on development of the other emulators. The Amiga QDOS emulator has acquired the ability to read WIN hard disk files, QPC may yet get the ability to read DOS hard disk files (as well as the existing QXLWIN). The Qubide may yet get the ability to handle ATAPI-IDE devices like ZIP drives and just possibly the LS-120 super-floppies. Nasta has been making noises on the QL Users Mailing List about a new soundcard, a SCSI interface is being threatened and several people on that list have been making detailed lists of how file handling and other aspects of QL development should go in the future. And Stella? A new system from Tony Tebby? Well, Arnould Nazarian gave details of this operating system core on the Mailing List, and Thierry Godefroy has kindly placed the extensive documentation on his Web site if you're interested. We've seen a lot of this type of talk and hype before and it's come to nothing. But I detect a different determination this time, that though this may all take a long time, the groundwork is being prepared for major enhancement in the future once the current crop of offerings become reality.

Welcome to a new software supplier - whether you want various Bible versions, or specialist software for Bible study, or just some reading matter or

clipart for a church newsletter, for example. George Morris is now offering modestly priced religious software to the QL scene. More details elsewhere in the magazine. Nice to see that there is still room for this sort of venture, I hope you'll support George in his venture. Our best wishes to him.

There are again a number of articles held over to the next issue, because we ran out of space for them all. If your article hasn't been printed yet, bear with us!

Last issue, we promised to print a fix to allow the use of a Prospero Pro-Pascal EPROM cartridge file to be run from RAM. Unfortunately, after we printed that, we realised that it was a copy-protection mechanism which prevented its use from RAM, and certain copyright issues pertaining to the software license came to light which mean we cannot publish David Bunbury's article, as we have not received permission from the publisher to print the "fix".

We also offered a small prize for the best Hints and Tips short listing. We had a disappointing response to this - only a small handful of entries were received, so we'll ask you again to enter this competition and hopefully publish several entries and award the prize in the next issue. My thanks to those who have already sent in the small number of admittedly quite ingenious listings.



Cartoon

RomDisq Driver Upgrade News

A new RomDisq Driver (V2.12) is available on QBBS 01442-828255 or my web page <http://firshman.demon.co.uk>

This is to correct a rare but fatal bug when writing to RomDisq. Typically if this occurred, the QL would lock up while writing to RomDisq, and then there would be a corrupted directory. Fortunately this has happened to only a few users, but it is a mandatory upgrade. It is also available from me by post - SAE and formatted disk.

We have made a minor hardware change to the board - also an essential upgrade. Early boards had a small capacitor soldered direct to pins 1 and 2 of the 8-way programming socket (or no capacitor was fitted). This capacitor should be between pin 2 and the main board. Any RomDisq not modified in this way must be returned for upgrading BEFORE V2.12 is installed.

NEWS

News from Jonathan Hudson

Richard is working hard on integrating my IP device for uqlx (tcp_, udp_) into the uqlx main release; should be done by the time you publish. The QDOS version of Lynx on uqlx can now offer full internet access; though it only does this by leverage off the underlying Unix IP sub-system, so its a bit of cheat really. Maybe as we (the emulator) can do it (TCP/IP), we'll embarrass the real QL enthusiasts into doing something?

Colour Drivers

Jean-Louis Dianoux of QLCF (QL Contact France) has written, asking us to point out that QLCF has decided to pay 17000 Francs (about 1700 pounds) to Tony Tebby towards cost of "licence of colour drivers". Jean-Louis says he thought this would be of interest to Quanta and QL Today, so that Tony could be encouraged to work on the drivers.

Religious Software

George Morris writes to tell us about a range of religious software he is making available to all QL users with an interest in this subject.

After spending some time gathering and preparing material, George Morris is now in a position to supply the following religious material, ranging from text documents to complete programs. Anyone interested in this material should give George a phone call or send a letter to him to request an A4 sheet of details of any item in particular, or a complete list of material available. The leaflets will include details of prices of the software on DD, HD or ED disks.

While some religious material has previously been available for the QL user (e.g. the text of the King James Bible is available from PD libraries), this is the first time that such a comprehensive collection has been brought together. The list includes material of interest to anyone from the layman to the professional student of religion.

The material listed is mainly Christian in origin. However, George also has a variety of material such as the Holy Koran, Buddhist material, Religions Of The World and so on. He believes that there is merit in offering these if only for the knowledge that they would give to Christians for dealing with people of other beliefs. All of these titles are available now.

- The King James Bible (Authorised version)
- The New King James Bible
- The New International Bible
- The Concordant Literal New Testament
- Hebrew Old Testament (for Quill or Text 87 - displays as Hebrew text, for the student)
- Hebrew Lexicon (Anglicised Hebrew)
- Koinee Greek New Testament (for Quill or Text 87 - displays the text as Koinee Greek)
- Greek Lexicon (Anglicised Greek)
- Bible Dictionary (with Anglicised Hebrew and Greek references)
- New Testament Word Pictures (Anglicised Greek)
- Bible Subject Index (based on words in King James Version Bible)
- Bible Cross Reference
- The Bible Companion (a major software work, covers many aspects of Bible study and Christianity)
- Hymnal Database
- Critique of Bible Translations, by Dr. J. E. Russell
- Common Bible Myths (little known facts, very interesting reading)
- Overview of KJV Bible...Construction etc.

Three new Hebrew and two Greek fonts are available - some of the above works use Greek and Hebrew fonts

In addition, the following works are currently in preparation:

- Anglicised Greek New Testament
- King James Bible (AV with J.H. Strong's Reference Numbers)
- Young's Literal Bible Translation
- J.N.Darby's Old And New Testament
- Gill's Expositor
- Thinkers' Bible Notes.
- Thompson Chain References
- 60 (yes sixty) DD disks of Religious Clipart for Line Design.
- Short Essays on many Biblical subjects

GREEK WORD Home Study Course. Suitable for understanding Greek holiday phrases or for Bible study.

GREEK WORD studies and Bible subjects on A5 leaflets.

For details, contact George Morris, 67 Wood Lane, Sutton Coldfield, West Midlands, B74 3LS, England. Telephone (+44) 0121-353 8571

Jochen Merz Software

SMSQ/E exists in Version 2.89 for all systems. It mainly contains bug-fixes, nothing major, just small things which you usually don't notice. Updates are free, as usual. You can also now debug scheduler loop tasks with QMON and SMSQ/E.

Sysmon (of QPAC 1) can now be placed outside the QL's original windows. It also handles large areas of RAM, allocated in more than 64k of chunks (could previously lead to memory corruption on systems with lots of RAM, e.g. a TT with 20MB).

Now news about upcoming things: there has been the demand for a new debugger. Look at the cover page - the right screen dump shows you SMON. SMON can debug the system you're working on, and it can debug another system via a serial port. The user interface will be the same, whether it is local or remote. SMON can handle all Motorola 68k and related instructions, PMMU, up to the 68060 etc.

You can open and close various listing-windows, re-size them, display things in various ways ... it is a completely new design which was required to debug the (new!) screen driver from a different machine and to implement SMSQ/E on new hardware, e.g. Q40 and the Milan.

It is not yet complete, the inline assembler is missing, for example.

There are thoughts about ways to allow SMON to find the source of the current modules... this would be another very useful feature.

Sandy Dillon.

Help! The rumour (which some people think was started by me) that Chas Dillons wife Sandy has died is not true. I most certainly did not start the rumour though I must admit to having told quite a few people once I was convinced of its veracity, including the editors of both QUANTA and QL Today. Sandy is alive and well, though possibly a little perturbed by rumours of her death. In fact somebody close to Sandy died and in a bad case of Chinese whispers it was passed along from person to person until by the time I heard (from more than one person as it happens) it was Sandy who had died.

I did not pass it on until I heard it from somebody I regarded as a reliable source. I will not name him, though rest assured I have made my feelings known and will be informing Chas Dillon of his name so Chas can make his own decision about whether to say (or indeed to shout) anything.

I would like to apologise to anybody who was unnecessarily upset by this rumour, especially anybody who knows Sandy.

Mark Knight

QPC not only comes with SMSQ/E V2.89 now, it also supports removable media (e.g. CD-ROMs, ZIP drives and allows you to change the QXLWIN filenames of the drives WIN1_ through WIN8_ to be re-specified while the system is running.

Also, QPC can now LBYTES files from the harddisk directly to the screen. Updates are free (as usual), just add 3 International Reply COupons if you send the master disk. You can also download the current version from the J-M-S bulletin boards.

SERNET should be available in a new version by the time you read this. You can now use SERNET via modem, i.e. access a computer running SMSQ/E via modem and use the features known from Toolkit II's network. It is also possible to execute BASIC programs on the remote machine, which means, you can do more or less everything you like (except for formatting the harddisk!).

QBranch

The price of Albin Hessler's Easy Pointer package has been

reduced. Parts 1 & 2 which contain the Menu and Sprite Generating tools, and have the SuperBasic Extensions will now be sold together for DM 99,- (by Jochen Merz) or £33.00 (by Q Branch) and part 3 which has the 'C' library extensions for DM 49,- or £16.00. In addition to this parts 1 & 2 will now include a free copy of the Pointer Environment Kit disk since this series of articles and example gives the best access to the package.

Q Branch have reduced the price of the last remaining QXLs to £180.00 and the price of Cueshell to £30.00.



Apologies that there are no screen dumps with the DBEasy review. Bill Cable did send us some 9 pin Epson style printer dumps, but as we were rather short of space we couldn't reproduce them small enough at a good enough quality to make it worth including them in the end.

Snake

Dilwyn Jones

This month, we start with a bit of entertainment brought to you in the form of a short BASIC listing. The original version of this game dates back to 1985, the listing has been updated July 1997.

Distribution status: Freeware (i.e. you can give copies of it to any QL or compatible user)

Tested on QPC V1.15, QXL, QL with SMSQ/E, JM QL etc, should work on just about anything!

This is the classic game of Snake. You have to guide the snake around the screen, eating the food points (shown by the numbers 1 to 9, or the ambiguous value '*') avoiding its own body and keeping it on the screen. As the snake grows, it gets more difficult to manage to move it without crashing into itself.

Normally, you can see the number of units by which the snake grows after seeing something, making it easier to predict whether it's safe to eat something in your particular tight corner, but the '*' values could be anything up to 9, so making it almost impossible to predict if you can get out of the situation by eating it. The object of the game is to score as high as possible. In theory, the highest possible score is 1050, but I have yet to get near this - you'll find out why as you play the game.

When the program starts, it asks for the skill level. Enter a number from 1 to 20. 1 is very, very difficult. 20 is easy. Try 10 to start with, and proceed from there. At the end of each game you can set a new skill level to suit yourself. If you find the skill level too slow, press any key to make the snake move at top speed (e.g. to get from one side of the screen to another more quickly).

The snake is controlled with the cursor arrow keys. The snake's head is shown by an arrow, pointing in the direction in which the snake moves. To quit from the program, press ESC (or enter a skill value of 0). To pause the game at any point (e.g. to answer the phone or wipe the sweat off your forehead) just press SPACE, and any key to restart. CTRL F5 will achieve the same thing on a QL.

```
100 REMark snake by Dilwyn Jones
110 RANDOMISE
120 MODE 8 : WINDOW #0,512,256,0,0 : PAPER #0,1 : INK #0,6
130 DIM board$(25,42) : arrows$ = CHR$(188)&CHR$(189)&CHR$(190)&CHR$(191)
140 REPEAT program
150 CLS #0 : CSIZE #0,2,1 : AT #0,2,7 : PAPER #0,6 : INK #0,1
160 PRINT#0,' "SNAKE" by Dilwyn Jones ' : PAPER #0,1 : INK #0,6
170 AT #0,5,7 : PRINT #0,'Enter skill level 1-20'
180 AT #0,7,7 : INPUT #0,'(1=difficult, 20=easy) > ';skill
190 CSIZE #0,2,0 : IF skill < 1 THEN STOP
200 AT #0,0,0 : PAPER #0,0 : CLS #0,3 : PAPER #0,1 : CLS #0,2
210 FOR a = 0 TO 24 : board$(a) = FILL$(' ',42)
220 FOR a = 1 TO skill
230 REPEAT loop
240 x = RND(0 TO 41) : y = RND(1 TO 24)
250 IF board$(y,x+1) = ' ' THEN
260 board$(y,x+1) = CHR$(48+RND(1 TO 9))
270 AT #0,y,x : PAPER #0,0 : INK #0,RND(2 TO 7)
280 IF RND(1 TO 10) < 10 THEN
290 PRINT #0,board$(y,x+1);
300 ELSE
310 PRINT #0,'*';
320 END IF
330 PAPER #0,1 : EXIT loop
340 END IF
350 END REPEAT loop
360 END FOR a
370 REMark set snake start position
380 REPEAT loop
390 sx = RND(5 TO 36) : sy = RND(5 TO 19)
400 IF board$(sy,sx+1) = ' ' THEN board$(sy,sx+1) = '0' : EXIT loop
410 END REPEAT loop
420 length = 1 : sx$ = CHR$(sx) : sy$ = CHR$(sy)
430 dir% = RND(1 TO 4) : REMark <> ~v
440 INK #0,7 : AT #0,sy,sx : PRINT #0,arrows$(dir%)
```

```

450 grow = 0 : REMark counts down for growth of snake
460 AT #0,0,0 : PAPER #0,0 : INK #0,7 : PRINT#0,'SNAKE by Dilwyn Jones';
470 AT #0,0,28 : PRINT #0,'SCORE: ';length; : PAPER #0,1
480 REPeat game
490   key = CODE(INKEY$(skill+1))
500   INK #0,6 : AT #0,sy,sx : PRINT #0,'0';
510   old_sx = sx : old_sy = sy
520   SElect ON key
530     =27 : EXIT program : REMark ESC
540     =32 : PAUSE       : REMark Freeze, press any key to continue
550     =192 : dir% = 1   : REMark left
560     =200 : dir% = 2   : REMark right
570     =208 : dir% = 3   : REMark up
580     =216 : dir% = 4   : REMark down
590   END SElect
600   sx = sx - (dir% = 1) + (dir% = 2)
610   sy = sy - (dir% = 3) + (dir% = 4)
620   IF sx < 0 OR sx > 41 OR sy < 1 OR sy > 24 THEN
630     INK #0,7 : AT #0,3,0 : CLS #0,3
640     PRINT #0,'Off-screen!'; : EXIT game
650   END IF
660   IF board$(sy,sx+1) = '0' THEN
670     AT #0,3,0 : CLS #0,3 : INK #0,7
680     PRINT #0,'Crashed into yourself!'; : EXIT game
690   END IF
700   IF board$(sy,sx+1) > '0' AND board$(sy,sx+1) <= '9' THEN
710     REMark increment growth counter
720     BEEP 1000,5*(CODE(board$(sy,sx+1))-48)
730     grow = grow + (CODE(board$(sy,sx+1))-48)
740     REPeat loop
750       x = RND(0 TO 41) : y = RND(1 TO 24)
760       IF board$(y,x+1) = ' ' THEN
770         board$(y,x+1) = CHR$(48+RND(1 TO 9))
780         INK #0,RND(2 TO 7) : AT #0,y,x : PAPER #0,0
790         IF RND(1 TO 10) < 10 THEN
800           PRINT #0,board$(y,x+1);
810         ELSE
820           PRINT #0,'*';
830         END IF
840         PAPER #0,1 : EXIT loop
850       END IF
860     END REPeat loop
870   END IF
880   REMark new head for snake
890   INK #0,7 : AT #0,sy,sx : PRINT #0,arrows$(dir%)
900   board$(sy,sx+1) = '0'
910   sx$ = sx$&CHR$(sx) : sy$ = sy$&CHR$(sy)
920   REMark sort out the tail
930   IF grow = 0 THEN
940     REMark move tail
950     AT #0,CODE(sy$),CODE(sx$) : PRINT #0,' ';
960     board$(CODE(sy$),CODE(sx$)+1) = ' '
970     sx$ = sx$(2 TO LEN(sx$)) : sy$ = sy$(2 TO LEN(sy$))
980   ELSE
990     grow = grow - 1
1000     length = length + 1 : AT #0,0,34 : INK #0,7 : PAPER #0,0
1010     PRINT #0,length; : PAPER #0,1
1020   END IF
1030 END REPeat game
1040 BEEP 19000,0,100,120,1 : INK #0,7
1050 PRINT #0,' Another game Y/N?';
1060 REPeat loop : INPUT #0,k$ : IF k$ == 'y' OR k$ == 'n' : EXIT loop
1070 IF k$ == 'n' THEN EXIT program
1080 END REPeat program
1090 AT #0,10,16 : CSIZE #0,2,1 : CLS #0,3 : PRINT #0,'GAME OVER.'
1100 CSIZE #0,2,0 : PAUSE 100 : REMark viewing pause before job ends
1110 STOP

```

Harddisks - Part 2

Mark Knight

The second part explains in detail how to handle subdirectories on your harddisk, how to deal with problems related to subdirectories and old QL programs which can't handle them.

5. Using subdirectories from SuperBASIC and SBASIC

Unless you are familiar with subdirectory handling this section is best followed by having a blank floppy disk handy, and sitting with your QL ready to accept typed commands. It is also helpful to keep to hand a disk with some Quill documents on it (Text87 or Perfection documents will do just as well if that is what you use most). The examples given will all make more sense if you try them out and look at the effects on the screen and disk.

There are a number of new SuperBASIC commands in the Gold Card and Super Gold Card ROMs and also in Toolkit II which are designed to help you use subdirectories from the command line or in your programs. The Toolkit II routines that allowed selective wildcard directories now also work properly with modern QL subdirectories. Under SMSQ and SMSQ/E SBASIC has all these commands built in.

The first command is obviously the one used in the example above, the MAKE_DIR command. This allows us to make a subdirectory on a specified device or on the default data device; the default devices may need explaining. Enter the following into your QL from the BASIC command line:

```
cls
dlist
```

...and look at the result. You should see a list of QL devices, commonly it looks like this:

```
FLP1_
FLP1_
SER
```

...though your BOOT program may have changed it. So what is it? It's the list of defaults that Toolkit II adds to the system to save a little typing when loading or examining files. The first is the DATA device, which is where DIR, WDIR, WSTAT and VIEW will look for files if you don't specify a device. So enter:

```
dir
```

...on its own: you will get a directory of flp1_ unless the data default has been changed. If you type:

```
view flp2_boot
```

...you would get a look at the boot file on the disk in flp2_ if there is one, if not an error. Entering:

```
view boot
```

...instead will give you a look at the BOOT program on flp1_ because that is the data default. DATA_USE is the command to change this default, so:

```
data_use flp2_
```

...will change it to flp2_, which often makes more sense. After entering the DATA_USE flp2_ command above (try it) you will find that DLIST now gives:

```
flp2_
FLP1_
SER
```

The second item on the list is the PROGRAM DEVICE: this is used for the LOAD, LRUN, MERGE, EXEC and EXEC_W commands and others used for starting QL programs in SuperBASIC/SBASIC or machine code programs started with EXEC or EX etc. So if DLIST gives the above list, typing:

```
load flp2_boot
```

...will work as normal, while:

```
load boot
```

...will load the BOOT file from

flp1_ as it is the current program device, similarly:

```
exec devman_exe
```

...would start a program called DevMan_EXE from flp1_ if it is suitable for the EXEC command. The default devices save typing and allow users to specify where they want data and programs to be kept. Since some modern QL programs use these defaults they can be a good way to decide where such programs look for files.

The final item listed by DLIST is the DESTINATION DEVICE, which is where the SPL and SPLF commands in Toolkit II send files to be PRINTed and also where WCOPY will send files if it can. This item may be set either by the SPL_USE command or the DEST_USE command depending upon what you are likely to use it for. DEST_USE and SPL_USE work slightly differently even though they set the same default, so be careful.

DEST_USE assumes you wish to use the destination device with WCOPY or COPY and so requires a directory device like a hard disk, floppy disk or even an old fashioned microdrive. If the name you specify does not end in the underscore character then DEST_USE will add one, so:

```
DEST_USE flp2_
```

...or:

```
DEST_USE flp2
```

...would both set the destination device to flp2_ because without the underscore it would not be a full directory device name. If the data device is flp1_ and the destination device is flp2_ then WCOPY on its own will copy files from flp1_ to flp2_ without further ado.

SPL_USE is for use with the old Toolkit II background spooling commands and assumes nothing about the device you specify. SPL and SPLF are both commands to send a file, without header, to the destination device while the QL does other things. So if you have

printed a Quill document (part one of your great novel, say) to a file called "flp1_Part1_LIS", setting the destination device to ser1 would be done by:

```
SPL_USE ser1
```

...then the command:

```
spl flp1_part1_lis
```

...will send the file to ser1 in the background. While the file is printing you may EXEC_W Quill again and work on part two without interrupting the printer. By now hopefully you understand the default devices, so put a spare blank disk in flp1_ and enter the following list of commands:

```
prog_use flp1_
data_use flp1_
dest_use ram1_
make_dir TEST
make_dir TEST_DOCS
make_dir BAK
make_dir BAK_DOCS
cls
dir
```

You should get a directory of the disk something like this (assuming an HD disk, with DD or ED the sector counts will be different):

```
SPARE 2859/2880 sectors
TEST ->
BAK ->
```

...now enter:

```
cls
dlist
```

...and you should see:

```
flp1_
flp1_
ram1_
```

Now type the next command, the DDOWN command. Try:

```
ddown test
dlist
```

... and you get:

```
flp1_test_
flp1_test_
ram1_test_
```

DDOWN is a command to move "into" or down the directory tree, that is to change the defaults to look into a directory, so:

```
dir
```

...now gives:

```
BLANK 2859/2880 sectors
```

```
TEST_DOCS ->
```

...this indicated that there is a further subdirectory called DOCS in the test directory and:

```
dup
```

```
dir
```

...gives:

```
BLANK 2859/2880 sectors
```

```
TEST ->
```

```
BAK ->
```

The DUP command is to move up (or out of) the directory tree. Note that directory trees are upside down, that is up moves towards the root and down moves away from the root. The root in this case is flp1_ because we can't back out of the tree any further than that. As well as DDOWN and DUP there is DNEXT, which in theory is to move to another directory at the same level, though it works a little oddly at times. To go into the TEST directory again type:

```
ddown test
```

```
dir
```

...and check the screen, then type:

```
dnext bak
```

```
dir
```

...and you will see we have moved from TEST to the next directory, BAK, and now:

```
dup
```

```
dir
```

...shows us we are back at the root. If we type the following, EXACTLY as shown:

```
new
```

```
100 rem
```

```
110 rem
```

```
120 rem
```

```
dup
```

```
dup
```

```
ddown test
```

```
save Test_BAS
```

```
dir
```

...we should see:

```
BLANK 2856/2880 sectors
```

```
TEST_DOCS ->
```

```
test_Test_BAS
```

...this indicates that DOCS is a further subdirectory in the TEST directory and we now have a file called Test_BAS, also in the TEST directory. Now enter the following, once again exactly as shown:

```
dest_use flp1_BAK_
```

```
wcopy
```

...and when the "Y/N/A/Q" appears, press "Y". Now type:

```
dir
```

```
dnext bak
```

```
dir
```

...and you will see where the WCOPY command has copied the file but this time with the first part of the name in upper case because that is how the default was specified. Most users stick to lower case all the time for directories for ease of typing but I like to stick to upper case to separate files from directories.

In case you forget the order of devices given by DLIST there are three functions you may use to obtain the devices separately, these are DATAD\$, PROGD\$ and DESTD\$. To use them, type commands like the following:

```
print datad$
```

...or put some code in your programs resembling this fragment:

```
13750 PRINT "Programs on ";PROGD$
13760 PRINT "Data on ";DATAD$
13770 Destination$=DESTD$
13780 IF Destination$
(Len(Destination$))="_" THEN
13790 PRINT "Destination default is
";Destination$
13800 ELSE
13810 PRINT "Spooler device is
";Destination$
13820 END IF
```

6. Using subdirectories with modern QL programs

To save a file into a subdirectory call "DOCS" on flp1_ all we need to do when saving the file is to start the filename with "DOCS_", so if

the program prompts us for a filename we may type "DOCS_Go-Away_DOC" or whatever. Some programs allow you to configure a device and directory name into them and will look in the specified directory for startup or datafiles. Read the instructions carefully and experiment with a floppy disk or two, creating subdirectories from within the program if it allows and from SuperBASIC or SBASIC as well.

One of the things that is sometimes forgotten is that on the QL the directory name is part of the filename. In QDOS and SMSQ and SMSQ/E the filename is limited to 36 characters in length which means that the directory name and filename together must not exceed this length. Remember also if you have a directory name that is ten characters long this shortens the length of filename you may use in that directory by eleven characters because of the need for the underscore as a separator.

In spite of the above always try to choose names for subdirectories that make it easy to remember what the contents are. There is a lot of frustration in knowing a file is on your hard disk but not being able to find it. There are search aids available but it is obviously quicker not to lose track of them in the first place and choosing directory and filenames in a systematic manner helps a lot. You need not adopt any system suggested by others if you don't want to but make sure you have a system of some sort.

7. Using subdirectories with QUILL

This section is a "hands on" section and contains listings and instructions that should be typed into a QL to install Quill on your hard disk. I have no use for Quill since I obtained DP's Editor and

later Perfection, but I understand many still use the Psion programs. Start by making a subdirectory to keep the program files, help files and Printer_DAT file in, say with:

```
make_dir win1_QUILL_
```

So far things are simple but to access files in our new QUILL subdirectory we must prefix them with "QUILL_", so a document with the filename "BadDebt1_DOC" must be loaded by typing "win1_QUILL_BadDebt1_DOC" or "QUILL_BadDebt1_DOC" if Quill is already using win1_ for files. The trouble is Quill only allows one underscore ("_") character per filename and another to separate the device name.

There are ways to fool Quill into using the facilities however, using new device drivers which pretend to be old fashioned QL devices but actually just "point" themselves at subdirectories on modern devices. This allows Quill to see a new device, called sub2_ for example, so "sub2_BadDebt1_DOC" really means "win1_QUILL_BANK_BadDebt1_DOC".

Most advice on this subject will recommend the use of the DEV device driver in the Gold Card and Super Gold Card ROMs, but I would actually recommend something else: Use Phil Borman's SUB device. This is much better at presenting directory listings than the DEV device and doesn't mess up the Quill files display as DEV will if the path is fairly long. Look on the disk that came with QUBIDE for the text file called SUB_TXT and the SUB_BIN file.

SUB_BIN is intended to be loaded on your QL using LRESPR or the usual RESPR, LBYTES, CALL sequence. As an example I will run through an installation of Quill on a hard disk, providing a BOOT program as well. If you follow the explanation you should be able to substitute Abacus, Archive or Easel at will and decide on the subdirectory names yourself. First,

make subdirectories: here we will make four but you should see how to expand this. Remember the limit is eight as QL devices cannot have numbers higher than that, so sub1_ to sub8_ is OK but don't try to create a sub9_ you will be wasting your time. From the command line type:

```
make_dir win1_QUILL_
make_dir win1_QUILL_DOCS_
make_dir win1_QUILL_BANK_
make_dir win1_QUILL_PERSONAL_
```

OK so we've made a subdirectory called QUILL and inside it there are three more, one for general documents, one for letters to the bank and another for personal letters: what now? Assuming you already have Quill working on a floppy disk MAKE A COPY of the disk, put THE COPY in flp1_ then type:

```
lrun flp1_config_bas
```

...Then configure the quill devices: make the device for the Printer_DAT and help files sub1_, the help device the same and the device for document files sub2_. Once this is done for the copy of Quill on your floppy exit Config_BAS and type:

```
wcopy flp1_ to win1_QUILL_
```

...think about which files you want, Quill itself being obvious but Printer_DAT and the help file (Quil_HOB) being among the others. For each file you want type "Y" or press "A" if you want them all moved to the hard disk - don't do this if there are tons of document files on your Quill disk, put them in the subdirectory files later.

Once this is done type in the program listing 1 and when you have finished save it as "win1_QUILL_BOOT".

Once you have done this all you have to do to use QUILL from your hard disk is enter:

```
lrun win1_quill_boot
```

While Quill is in use simply save to sub2_ or sub3_ or any other devices you have set up. To back

Listing 1:

```
100 TK2_EXT : REMark Make sure Toolkit II is ready for use.
110 WTV 4 : REMark Select TV window arrangement but in mode 4.
120 DATA_USE "" : REMark Cancel Toolkit II defaults, Quill can be confused.
130 PROG_USE ""
140 :
150 REMark Next few lines loads SUB_BIN unless it is already loaded.
160 REMark POP_OVER is a Toolkit II function (Function OPen for OVERwrite)
170 :
180 Testing=POP_OVER(#3,"sub1_Test")
190 IF Testing<0 THEN
200   REMark if using HOTKEY system next line should have REMark removed
210   REMark HOT_STOP
220   :
230   LRESPR "win1_QUILL_SUB_BIN"
240   :
250   REMark if using HOTKEY system next line should have REMark removed
260   REMark HOT_GO
270   :
280 ELSE
290   CLOSE#3
300   DELETE "sub1_Test"
310 END IF
320 :
330 SUB_DRV "win1" : REMark Select which drive you wish to use with SUB
340 REMark Don't use "win1_", SUB_DRV needs devices without the underscore.
350 :
360 SUB_USE 1,"QUILL_" : REMark Now sub1_ = win1_QUILL_
370 SUB_USE 2,"QUILL_DOCS_" : REMark Now sub2_ = win1_QUILL_DOCS_
380 SUB_USE 3,"QUILL_BANK_" : REMark Now sub3_ = win1_QUILL_BANK_
390 SUB_USE 4,"QUILL_PERSONAL_" : REMark Now sub4_ = win1_QUILL_PERSONAL_
400 :
410 REMark add/use your own subdirectories up to sub8_ if you wish.
420 FORMAT "ram1_" : REMark Clear a ramdisk
430 FORMAT "ram1_200" : REMark Now reformat to a fixed size of 200 sectors
440 :
450 AT 2,28:PRINT "LOADING QL QUILL"
460 AT 4,30:PRINT "Version 2.3"
470 AT 6,23:PRINT "Copyright 1984 PSION LTD"
480 AT 8,29:PRINT "Word Processor"
490 BAUD 9600
500 EXEC_W "sub1_QUILL"
510 REMark After Quill quits WCOPY documents on ram1_ to win1_QUILL_DOCS_
520 WCOPY "ram1_DOC" TO "sub2_DOC"
530 STOP
```

up use your normal hard disk backup method or save files twice, once to a SUB device and once to flp1_ as well. Using the above example most people should be able to write suitable BOOT programs for Abacus, Archive or Easel if they want. A significant advantage is that each of the Psion suite can have a separate Printer_DAT file in their own subdirectory, so switching from one to the other is not such a hassle if you want Quill to use 10 or 12 characters per inch and

Abacus to use 20 characters per inch to fit bigger spreadsheets in.

These methods can also be adapted for any program which refuses to recognise subdirectories because it is old software or badly written. Modern QL programs use subdirectories and you should really learn to use these facilities properly rather than adopting the above methods when they are not needed. It may take time and effort - I know some people aren't keen on effort (most people perhaps) but it is worth it.

8. A nifty trick with a hard disk drive

One of the troubles that might come with having a hard disk drive is remembering where everything is, especially if you have dozens or hundreds of EXECutable files. It may surprise you to know that in order to EXEC a file on a hard disk you don't need to know where it is. As long as you keep the program files in a set of directories no longer than eight items long you can just EXEC them, for example to EXEC Perfection and Editor SE you would just enter:

```
exec perfection
exec edt_bin
```

...even if Editor is in win1_EDT_ and Perfection is in win1_PERF_ and you would not need to change the program default in between. As well as this you might know that you have my PopCalc calculator in EITHER win1_UTIL_ OR win1_SYS_ and yet to exec that to multitask alongside Editor and Perfection you would simply type:

```
exec popcalc
```

How is this trick done? Using the DEV device mentioned earlier: DEV is not

the best way to work with Quill or the other members of the Psion suite but it is superb for use with EXECutable files. DEV and SUB are rather similar in what they aim to do, so I will revisit SUB to explain the differences. Assuming there are subdirectories on win1_ called EDT and DOCS if we type:

```
sub_drv win1
```

```
sub_use 1,_EDT_
```

```
sub_use 2,DOCS_
```

...then loading sub1_ReadMe.TXT will fetch win1_EDT_ReadMe.TXT and loading sub2_Read

Me_DOC with Quill will fetch win1_DOCS_ReadMe_DOC, fooling Quill into loading the document without breaking its outdated rules about filenames. Similarly we can set up DEV to do the same using the DEV_USE command thus:

```
dev_use 1,win1_EDT_
dev_use 2,win1_DOCS_
```

If we load dev2_ReadMe_DOC into Quill we will fetch win1_DOCS_ReadMe_TXT just as with sub2_ReadMe_DOC earlier. However if we type:

```
dir sub2_
dir dev2_
```

...the results are rather different. With sub2_ we might get something like this:

```
SAUSAGES 58052/83044 sectors
ReadMe_DOC
HardDskS_DOC
Damage_DOC
```

...but with DEV we will get the full filenames; e.g:

```
SAUSAGES 58052/83044 sectors
DOCS_ReadMe_DOC
DOCS_HardDskS_DOC
DOCS_Damage_DOC
```

...this makes DEV less suitable for use with the Quill directory listing as the names are sometimes long enough to overlap names in the next column or wrap around onto the next line. With SUB this long filename problem does not occur but DEV has a useful trick up its sleeve that SUB can't match. DEV will chain, that is it will store a search path for a file as opposed to just one directory. For example if we type:

```
dev_use 1,EDT_,2
dev_use 2,DOCS_,3
dev_use 3,TXT_,4
dev_use 4,LETTERS_,5
dev_use 5,LETTERS_WORK_,6
dev_use 6,LETTERS_STUDY_,7
dev_use 7,LETTERS_FAMILY_,8
dev_use 8,INFO_,1
```

...we set up DEV so that it will search all the listed directories in the given order. If we ask for dev1_ReadMe_DOC using Quill

then dev will first try win1_EDT_ReadMe_TXT and then win1_DOCS_ReadMe_DOC and so on down the chain. Note that this chain wraps around, so starting with dev8_ReadMe_DOC will try win1_INFO_ReadMe_DOC then win1_EDT_ReadMe_DOC and then win1_DOCS_ReadMe_DOC as before. When DEV has tried all the devices from dev1_ to dev8_ it will give up, even if the chain wraps around like this one.

Assuming we have the directories present on a hard disk the following program will set up the system so that EXEC will search all those in the listing. It is only necessary to keep all our program files in one of the eight listed directories to allow us to EXEC them even if we have forgotten which directory they are in. As BAS_ is on the list, presumably where SuperBASIC/SBASIC listings are kept, you can also LOAD a program just by typing the filename, e.g:

```
load FileMan_BAS
...would load win1_BAS_
FileMan_BAS. You should naturally
substitute your own directories for
those in the listing.
100 TK2_EXT
110 DEV_USE "hdk"
120 DEV_USE 1,"win1_UTILS_",2
130 DEV_USE 2,"win1_SYS_",3
140 DEV_USE 3,"win1_PERF_",4
150 DEV_USE 4,"win1_EDT_",5
160 DEV_USE 5,"win1_QPAGE_",6
170 DEV_USE 6,"win1_IDE_",7
180 DEV_USE 7,"win1_PSION_",8
190 DEV_USE 8,"win1_BAS_",1
200 :
210 DATA_USE "win1_"
220 PROG_USE "hdk1_"
```

Note that you don't have to keep a program and its data in the same place, for example Quill, Archive and Abacus could all be in the PSION subdirectory while there is a directory called QUILL for the Quill Printer_DAT and Quill_HOB files, another called

SPREADS for the Abacus files etc. You can configure Quill to use sub2_ which is really win1_QUILL_DOCS_ or whatever, while Quill itself is in the file win1_PSION_QUILL so the DEV search will find it. I use a version of this listing to set up Turbo so all I have to do is type LOAD FileName_BAS then use the CHARGE command and Turbo loads win1_TURBO_Parser_TASK and win1_TURBO_Codegen_TASK. QLiberator and Supercharge can be persuaded to do similar tricks using DEV.

If you decide to buy QUBIDE (or one of the other QL compatible systems that can use a hard disk like a QXL, QPC or an Atari QL emulator) I would suggest you don't load much software onto the hard disk right away. Instead spend some time experimenting with DEV, SUB and the various ways of setting up subdirectories before you design a system for using your hard drive that will suit you. As soon as you start keeping real work on your hard drive buy a backup program and use it.

■

Have YOU
Sent in
YOUR
useful hot
tips yet? It'S
not too late -
YOU
COULD
WIN A
PRIZE...

QL/Aurora
Public Domain
and Sharware
Software

£1 per Disk

Aurora
Replacement QL M/board
£100 each

QL/Aurora
New & Used
Hardware,
Software,
Books &
Lots More

QUBIDE
AT/IDE Interface
£55 each

QPLANE
3 Slot Back Plane
£25 each

QUBBESoft P/D 38, Brunwin Road, Rayne, Braintree, Essex. CM7 5BU. UK

Tel/Fax: +44 (0)1376 347852 Email: QUBBESoft@aol.com

All goods inc P&P (UK only) except Keyboards + £2 P&P

WE SUPPORT SINCLAIR

QBOX USA

COMPUTER BULLETIN BOARD SERVICE

(810)254-9878

We support *all* SINCLAIR COMPUTERS
(QL, SPECTRUM, ZX81, Z88, Thor, QXL)

- Now in our 4th year on-line round the clock since October, 1993
- Full message area and File Download areas
- We carry all popular SINCLAIR message areas from Europe
- Calls from 14.4k--300 baud are welcome
- QBOX - USA runs on a SINCLAIR QL with Super Gold Card, Hermes, QUBIDE, 200MB drive, USR sportster 14.4 modem

→ NO FEES CALL US

The Leap Back

Adrian Ives

Come back from PC to QL? Some people did it recently, and Adrian writes his experiences down for you - more reasons to stay with the QL.

Until some time around October last year, I hadn't touched a QL in almost five years. My job had made it impractical to continue using the QL as more than just a curiosity, and so my 'kludged' QL + Trump Card + Miracle Hard Disk was consigned to the spare room.

With a change of job into project management, I had less need to maintain an in-depth knowledge of the PC and its development tools, and, frankly, I was getting very bored with WinSLOTH 95 and the "MicroHard" range of disk-gobbling behemoths.

When I was allocated a rather nice Pentium Lap Top, which came as part of the job, I no longer felt the need to keep my "own" PC at home. So, off it went - obsolete, anyway, just five months after I had built it from "state of the art" components!

When I look at the PC today, I wonder where it will all end. The entry-level machine now is a Pentium 166 (Sorry, just heard, make that a 266 so that you can run NotePad) with 64MB RAM and 2GB of Hard Disk! And don't forget the mandatory Hextuple speed CD-ROM! I mean ... give me strength!

And just look at some of that software - sure, it looks nice; it comes in a nice box on 15 CD-ROMs; plays music; warps the fabric of sub-space; displays dialog boxes in three dimensions ... and locks up more times in 30 minutes than my QL has done in a lifetime!

Anyway, enough of irrelevancies. I had been thinking about a short article for QL Today for a couple of months now, so I thought it was about time to load Text87 Plus4 (in just 1.9 seconds from my QUBIDE Hard Disk - Word

95 takes a staggering ten times as long on a machine several orders of magnitude more 'capable').

The question is: What should I write about? And who would be interested anyway? I suppose a good starting point is probably to describe what hardware I run - there seems to be an immense variety across the QL community ...

I run Aurora, Super Gold Card, superHermes, and QUBIDE driving a Western Digital 1.2GB Hard Disk. All of it is packaged in a PC mini tower case, with an IBM 15X SVGA monitor, Canon BJC250 printer, CisCom 33600 Voice/Data FAX Modem, and PC keyboard and mouse. Yep, it looks just like a PC - except that it boots up in about one thirtieth of the time! I do like seeing the look of surprise on people's faces when they DON'T see an improperly rendered bit-map of a cloudscape appearing two minutes after switching the machine on.

In the spare room is my original QL (what I call the "QL Classic"). In fact, The Classic is barely recognisable as a QL. A long time ago, the black plastic "2001 Monolith" housing was stripped off, and the main board mounted in a heavy duty steel case that formerly housed a very old CP/M micro-computer.

The multi-rail power unit in this case was ideal for driving the QL, and banished forever the infamous power glitch problems of yesteryear. The Trump Card was a problem, though, extending about four centimetres out of the end of the case - so I had to construct a housing out of a Maplin project box to protect it. Like I say, a strange looking beast indeed. But it worked, and it was (is) very robust and reliable.

In fact, it's from the QL Classic that my current system was inspired. After five years of neglect, the Miracle Hard Disk had stopped performing miracles, and was siezed solid. I could not find an XT-style Hard Disk for love nor money. Was the Quantum Leap doomed to end its days gathering dust in a corner?

Well, almost. Jochen Merz, for some strange reason, had continued to send me his catalogues - even though I hadn't placed an order with him for a very long time. I noticed the QUBIDE interface, did a quick conversion of Sterling to DM and thought "why not?".

I was very pleased to find that the QUBIDE worked first time. While I was at it, I decided to give QL Today a try and ordered a year's subscription. Without boring you, this was the start of some frantic spending (Jochen, Roy Wood et al - I think you're all a bit better off as a result!) that resulted in my current system.

I had set out just wanting to get the old QL up and running as a hobby, but I was amazed by the number of loyal QLers still out there, and by the hardware and software, much of it of a very high quality, that was still under development. Once I started fiddling around with S*BASIC extensions and completing some half-finished utilities written in QLiberator, it all came flooding back.

I can say, without a doubt, that the QL is by far the easiest machine I have ever developed software on. It is clean and elegant, and avoids the 'BloatSoft' approach that dogs the poor PC. Long may it continue.

It's good to be back. For my money, the move from the PC to the QL really is a "Quantum Leap".

p.s: If anyone wants my old Miracle Hard Disk interface card (no case and no guarantees) send me an E-Mail: ADIves@aol.com

■

MTOOLS - A Review

Timothy Swenson

Jonathan Hudson has ported the Unix utility, Mtools, to QDOS. In the Unix world, Mtools is a set of utilities that read, write, and navigate MS-DOS disks. With Unix, a single program can be called by different names and have different behavior. In QDOS this is not possible, so the full set of these utilities is the program 'mtools'.

Mtools is essentially an implementation of a number of MS-DOS commands, such as COPY, DIR, DEL, TYPE, CD, MD, RD, DELTREE, ATTRIB, etc. Mtools can handle both DD and HD 3.5" disks. I have not tested it, but it should handle 5.25" disks also.

The mtools.zip file contains the Mtools executable, the Mtools manual, a QDOS README file, and a number of Unix man (manual) pages for each of the commands. The manual covers all of the commands. The man pages are added only for completeness.

Besides the Mtools executable, a configuration file and environment variable are needed. This means that you must have environment variables (ENV_BIN) support loaded.

The Mtools configuration (mtools.conf) defines how your disk drives are setup and how you want to use Mtools. The configuration file is a text file and can be created with any text editor. The main thing that is needed in the configuration file is the definition of your disk drives. You must define what drives you have, what format they are, and what they are called in MS-DOS terms. In my case I have a single HD disk drive, so use the following configuration entries:

```
drive a: file="flp1_*d2h" 1.44m
drive a: file="flp1_*d2d" 720k
```

The first line tells mtools that I am using FLP1_ as the MS-DOS drive A: and that it is an HD (1.44 MB) drive. The second line says that the same drive can also be a DD (720K) drive. The items 'flp1_*d2h' and 'flp1_*d2d' come directly from the ToolKit II manual and are the direct sector reading devices. This is how Mtools knows how to read the disk drive. If you only have 1 DD drive, then you would use only the second line. If you have a second disk drive then you would just add lines with 'drive b:'. For ED disks, I believe the device needed is 'flp1_*d2e' and it is only 2.88 Meg. MS-DOS did not support the ED drives at 3.2 Meg.

Once I had a configuration file created, I needed to tell Mtools where to find it. I used the SETENV command like this:

```
SETENV "MTOOLS_RC=ram1_mtools_conf"
```

Since I only have one disk drive, I have to keep all my QDOS files on RAM1_, FLP1_ is used as the MS-DOS disk. You would use whatever device you are

using for your QDOS files (FLP2_, WIN1_, etc.).

The next thing I needed to do is to set the DATA_USE and PROG_USE devices to the device I am using for QDOS files (RAM1_). Mtools has a problem using QDOS device names, so it is better not to use them and let DATA_USE/PROG_USE handle them.

Now I put an MS-DOS disk in the drive and execute Mtools:

```
exec mtools;"-c mdir a:"
```

Because of how Mtools was written and how QDOS is, Mtools must be given a '-c' argument and then the command to run. Mtools also puts a 'm' before each standard MS-DOS command name. What I am doing is 'DIR A:' in MS-DOS terms. Once Mtools executes, a window pops up and after a bit of reading the disk, a MS-DOS-looking directory of the disk appears, including any directories on the disk.

To copy a file from MS-DOS to QDOS Mtools is executed like this:

```
exec mtools;"-c mcopy a:file.ext file_ext"
```

I am telling Mtools to copy the file 'a:file.ext' to my QDOS device (set by DATA_USE) as the name file_ext. Mtools recognizes the 'a:' as an MS-DOS device and knows where to get the file.

Copying a file from QDOS to MS-DOS would go something like this:

```
exec mtools;"-c mcopy file_ext a:"
```

Mtools knows to convert the _ in file_ext to a standard MS-DOS period. Now if I had run Mtools like this:

```
exec mtools;"-c mcopy ram1_file_ext a:"
```

then a file called 'ram1_file.ext' would have been copied to the MS-DOS disk. Remember, Mtools does not know how to handle QDOS device names. This is why I set DATA_USE to RAM1_.

Mtools supports the newer MS-DOS file systems that allow lower case and longer file names, now used by Win95 and NT. Any files copied to MS-DOS in all caps will remain in all caps.

Because of a problem with C68, Mtools has a problem handling the MS-DOS device/file separator back slash (\). In my testing I tried to create a directory TEST. I used the Mtools command:

```
exec mtools;"-c mmd a:\test"
```

Mtools seemed to translate the \t into the Tab character and wanted to create a subdirectory 'a: est'. To get around this problem you have to use two back slashes (\). So this command would work:

```
exec mtools;"-c mmd a:\\test"
```

Mtools also supports the standard Unix separator slash (/). So this will also work:

```
exec mtools;"-c mmd a:/test"
```

Mtools supports the following commands:

Mattrib - Used to change MS-DOS file attribute flags.

Mbadblocks - Scan an MS-DOS floppy and mark unused bad blocks as bad.

Mcd - Change directory.

Mcopy - Copy a file.

Mdel - Delete a file.

Mdeltree - Delete a whole directory tree.

Mdir - Directory of an MS-DOS disk.

Mformat - Format an MS-DOS disk.

Minfo - Print parameters of MS-DOS file system.

Mlabel - Label disk.

Mmd - Create a subdirectory.

Mmove - Move a file (Rename).

Mrd - Delete a directory.

Mren - Delete a file.

Mtype - Copy a file to the screen.

Xcopy - Copy a directory tree to another.

I did have one problem with Mtools that Jonathan and I have not sorted out. Mtools seemed to have problems formatting MS-DOS DD disks on a HD drive. It formatted the HD disks just fine. Since I only have the one disk drive, I can't rule out my drive as being the problem.

I have also noticed that NT seems to have a problem formatting 720K disks also. I formatted a 720K disk on an NT machine at work, brought it home and tried to read it. Mtools failed to read it, along with another MS-DOS disk reading utility I've been using for years. I doubt an old MS-DOS 3.3 system would also be able to read the disk.

Mtools is a fairly powerful set of utilities. All of the commands and the configuration file have a lot more options that I have discussed. At first it may seem a bit overwhelming to have to many configuration options, but the simple configuration file like the one listed above should suit most uses. The manual is fairly thick but it is very thorough and between it and the QDOS README file, most of your question should be answered. I wrote this article in such a way that it can be used as a "quick start" document and get you up and running.

Mtools may not be for everyone and most who use it will only need to use just a few of the commands (DIR & COPY), but I am finding it very handy. Now that 720K disks are going out of style, I needed a way to read HD disks on my QL. Mtools fills this need very well.

Mtools can be found from most QL Freeware sources, includes QL BBS's, and on Jonathan's Web page.

[Editor's note: Tims article did not make it into the previous issue and was waiting here for three months. As we all know, Jonathan is quite productive and therefore we asked him if he had any comments about improvements. Here is his reply:

"... I think Tim's article misses (or only covers very superficially) the two most important things about mtools.

- 1. It supports VFAT (long file names in W95 and NT)*
- 2. It supports sub-directories on DOS media*

Tim and Adrian Ives were very helpful in debugging mtools; Adrian wrote to me: "One thing I will say: MTOOLS is an order of magnitude faster than the SMSQ copy routines. I do a hell of a lot of copies between PC and QL, and I have been using QPAC for this. Copying a 500K file off an MSDOS disk can sometimes take several minutes! MTOOLS does it in about ten seconds!"

...

Adrian's SHELL program supports aliases for the full mtools command set, so using that you can do

```
mdir a:
mcopy b:
etc.*]
```



The Pandora Case

review by John Peace

About a year ago I decided to upgrade my QL. My equipment at the time was the usual black box with standard Hermes, Minerva, Super Gold Card and double ED Disc Drives. I was still using my old Phillips monochrome monitor and decided it was time to redesign my system completely. I opted for an Aurora board with a 1.6Gb hard disc and SMSQ/E plus whatever I could salvage from my existing set-up. I also bought a 15" SVGA monitor and Acer keyboard.

John Southern's offer to build systems using the North Manchester Group's Pandora Box persuaded me to use this rather than a converted PC case. I have now been using my new system for about a year.

The case itself is made from fairly heavy gauge steel sheet sprayed black inside and out. It is very robust and consequently a little on the heavy side. The case is 50 cms. (20 ins.) long and the ends are 17cms. (seven inches square, i.e. just wide enough to take 5.25" drives). A kind of bonnet, retained by six screws at the base, lifts off leaving the sides and most of the top open for access.

QL/Aurora
Public Domain
and Sharware
Software
£1 per Disk

**superHermes Compatible
102 Key PC Keyboards
£10 each**

QL/Aurora
New & Used
Hardware,
Software,
Books &
Lots More

**3 Button Serial Mice
for superHermes
£6 each**
includes mouse-mat

**Seagate 545mb IDE
Hard Drive
£60 each**
QUBIDE Compatible

QUBBESoft P/D 38, Brunwin Road, Rayne, Braintree, Essex. CM7 5BU. UK
Tel/Fax: +44 (0)1376 347852 Email: QUBBESoft@aol.com
All goods inc P&P (UK only) except Keyboards + £2 P&P

A recessed retractable handle is fitted in the centre of the case top and although the Pandora is rather heavy it is very well balanced and easy to carry.

As supplied the unit has bays for three 3.5 inch and two 5.25 inch drives. There are holes to carry five 9-pin and two 25-pin PC type connectors plus a 5-pin DIN socket for the keyboard. Although the the number of I/O leads fom QL peripheral boards can greatly exceed this, there should be enough sockets for most requirements. Certainly there are enough for mine.

The Aurora board is fixed to the base with four bolts fitted with 0.5cm. spacers and has just enough clearance at the back of the case to allow a QPlane to be fitted.

The Super Goldcard fits in upside down and the large heat sink prevents it from occupying the lower slot of the QPlane, so the Qubide went there with the Goldcard immediately above. Since the Qbide is very short it doesn't get in the way of the heat sink.

The location of the various drives needs to be carefully worked out to avoid too much twisting of the connecting ribbon cables - there isn't a great deal of spare space and there are quite a lot of cables going back and forth. Ribbons going to sockets at the rear

of the Pandora will probably need to be routed underneath the Aurora.

One of the main selling points of the Pandora is its portability. Over the past year I have taken the equipment to QL group meetings etc. and have found it very convenient to carry around, although I still seem to have as many loose bits as ever. I suppose I am just acquiring more of them.

The one serious problem is not directly connected to the Pandora but is caused (I think) by the large heat sink on the Goldcard. When carried in the car boot it bounces up and down and the bottom connector on the QPlane has become loose. A little judicious packing will hopefully have solved the problem and I am keeping my fingers crossed. I understand John Gilpin is already looking into the problem.

Overall I am quite happy with the Pandora Box. The case will stand fairly rough handling without harm, it can be moved around with one hand and the top presents a flat surface for external peripherals etc. Three of the 9-pin sockets are on the front of the case where they can be used for the more frequently changed equipment and generally I find it more convenient than, say, a tower case.

■

Assembly Language Programming - Part 1

Norman Dunbar

This series of articles is intended to let you in on the basic secrets of programming your QL in its own natural language - machine code or assembly language. It will not just be a summary of 68000 instructions, no, this series will teach you how to program your own machine code programs under QDOS.

Assembly language is very, very simple. Not many people will agree at first, but if you think about it, it is. You have to tell the processor what you want it to do in very simple steps. In SuperBasic, you can multiply two numbers together easily - you can do it almost as easily in machine code too.

This series of articles is intended to let you in on the basic secrets of programming your QL in its own natural language - machine code or assembly language. (Actually, machine code is what the QL talks, we use assembly language which is 'English' sounding 'words' that get converted to machine code by an assembler - I will tend to use the two terms as one.)

To talk directly to the QL, you must learn its language. This series should hopefully teach you how to do just that. I make no assumptions about how much or how little you may already know - I will start very simple and continue from there. Hopefully you will have an assembler, but if not, I will be presenting each program or code fragment - at least the ones that you can run - as both an assembly language listing and as a hex listing along with a small SuperBasic program to convert the hexadecimal numbers into a machine code file ready for use.

I was going to base the series on George Gwilt's GWASM assembler, which is free and can be distributed easily. Unfortunately, it won't run on anything less than a 68020 which is no good for those of us who are still running on an original QL - anyone out there got a free assembler for the masses?

Most assembly language books tend to give little example programs as they go along to try to show the bits of the instruction set that you have just learned about. I will attempt to do likewise, but in addition, we will develop a useful application as we go along as well - the project!

The Project

As we go along, we'll be developing a small but hopefully useful program, which may not have too

many bells and whistles, but should just work!

Now that I have said that, I suppose I will have to put my machine code where my mouth is!

The 6800x Processor

The processor we are programming is one of Motorola's 68000 series. Be it a 68008 or a 68060 (if you are lucky) all of them have the same basic instruction set, although some of the more powerful processors have additional instructions. Partly because we have to cater for those on an original QL but mostly because I don't have a clue about these additional instructions, we will be dealing with the basic instruction set - there is enough there to keep us happy for a while.

Inside the processor there are a few different parts, but we are only concerned with the registers - the rest just does the work and puts the results somewhere, setting a few flags along the way. Talking of flags, we will also take a look at the status register - a very important part of programming.

Registers

Registers are where numbers get loaded into, manipulated and written out from. Some instructions operate directly on memory locations, but to all intents and purposes, memory is just another register but outside of the processor and a lot slower.

The 68000 - which is the term I shall use from now on to describe the entire family of processors - has different types of registers - data, address, status and program counter.

Data held in registers and in memory is held in High Order format. This simply means that the numbers are stored in a similar manner to the way in which we would expect them to be - the 'rightmost' end holds the most significant bit and the 'leftmost' the lowest - just the way we write numbers down.

Data Registers

There are 8 data registers named D0 to D7 and these can be used to perform manipulations on the numbers that are held in them. Each register can hold 32 bits of information. (A bit is a single binary digit - basically a one or a zero). What these bits actually represent depends on the program running at the time. Data registers are normally used for manipulating data in the form of bytes, words and long words - these being 8, 16 and 32 bits long respectively.

Address Registers

There are 9 address registers named A0 to A7. A7 is sometimes known as the stack pointer or SP

register. What about the other address register then? The ninth address register is a duplicate of A7 and is the SSP or Supervisor Stack Pointer. When coding the chip, you only have access to 8 address registers at any one time - you are either using the SP or SSP version of A7 but never both at the same time.

Address registers are normally used to hold memory addresses, stack pointers etc and cannot be used for byte sized manipulations.

Status Register

The status register holds a list of flags to tell the processor what is happening or has happened internally. The status register is a 16 bit register in two 8 bit halves. The user byte is held in bits 0 to 7 (the lowest end) and the system byte is held in the upper half or bits 8 to 15. The layout is as follows, first the system byte:

Bit 15	14	13	12	11	10	9	8
T		S			I	I	I

'T' is the trace flag - this defines whether the processor is in 'single step' mode or running normally. If set to 1, the processor is tracing and if 0, is running normally.

'S' is the supervisor flag - this defines whether the code being executed is running in user or supervisor mode. If set, the processor is in supervisor mode otherwise it is in user mode.

'I' is the interrupt mask and represents a value between 0 and 7 and indicates which of the seven interrupt levels are enabled.

The other bits are not used.

Then the user byte :

Bit 7	6	5	4	3	2	1	0
			X	N	Z	V	C

The user byte contains the 5 condition code flags which are set or reset by certain instructions and then used by arithmetic or comparison instructions. They are used to tell later parts of a program what happened recently. The program can adjust its operations to suit. The flags are:

'X' is the extended flag. Which is very similar to the 'C' flag bit is affected by fewer instructions than 'C' is. This is used when carrying out very large sized arithmetic instructions - such as 64 bit adds, for example. When affected it is set exactly like the 'C' flag.

'N' is the negative flag. It gets set to 1 if the last instruction created a negative number.

'Z' is the zero flag and is set to 1 if the last instruction generated a result of zero.

'V' is the overflow flag and is set to 1 if the last instruction generated an overflow during 2's comple-

ment arithmetic. See later for details.

'C' is the carry or borrow flag. And is used when a subtraction operation is carried out - be it an actual subtraction or an implied one.

The flags are used by the branch on condition (Bcc) instructions, the Decrement and branch (DBcc) instructions or the Set (Scc) instructions. These will be explained later.

Program Counter

The program counter does just that, it keeps track of where exactly the processor is within a program. The program counter always points to the address in memory of the next instruction to be executed. The program counter can of course be changed by a JMP (jump) instruction or a BRA (branch) but it is always ready with the next instruction to be executed.

Addressing Modes

The 68000 has a large number of addressing modes and these can often become overwhelming to a new machine code programmer - I know. It takes some time to understand each and every mode, what it does and why it is used. Having said that, you do not need to remember all of their names, just what they look like in source code and of course, what they do.

From here on, you need to be aware that numbers may be in decimal format or hexadecimal. All hexadecimal numbers are prefixed with the dollar sign (\$) and wherever this is seen in front of a number (or in some cases, what appears to be a word) will be a hexadecimal number. (I will assume that you are familiar with hex. *Editor's note: it has been explained in QL Today Issue 6 Volume 1 "Hardware or not Hardware"*) A couple of examples of hexadecimal numbers are:

\$100

\$COFFEE

which are equivalent to 256 and 12,648,430 respectively. Without any further hesitation, let's dive right in with the addressing modes ...

Register Direct

This is an easy one to start off with. Register direct addressing mode simply means that both the source and the destination in the instruction are registers either data, address or a mixture of both.

Simple examples are :

MOVE.L A2,D1

MOVE.W D0,D1

MOVE.L A1,A3

These simply move (actually, they copy) data between various registers. The full meaning of the actual instructions will be described later on.

Absolute

In this mode, the operand of the instruction is simply a memory address. This is also quite simple. For example to 'zeroise' the contents of the first byte of screen memory (assuming a standard QL and this is the last time that I will assume anything!) :

```
CLR.B $20000
```

There are two variations to this mode, absolute short and absolute long. If the address given is a 16 bit word (ie 0 to 7FFF hex or 32767 decimal) then it refers to addresses in the first 32K of memory. If the address given is 8000 hex or 32768 decimal and upwards it refers to address FFFF8000 and upwards due to **sign extension** of the address word. This is absolute short, best used for addresses of 0 to 7FFF hex only - to avoid confusion.

```
MOVE.L $1000,D1 - gets the long word at  
                  address $1000
```

```
MOVE.L $9000,D1 - gets the long word at  
                  address $FFFF9000
```

The other variation is absolute long, in this case, the address given is a full 32 bits long and refers to the actual address in memory - there is no ambiguity with absolute long.

```
MOVE.L $123456,D1 - gets the long word  
                  at address $123456
```

Relative

This mode will probably be the most used with QL programs as all code should be relocatable. This means that it never assumes that it is running at a specific location in memory. Some early QL programs were written to run at a specific location in memory and this caused no end of problems when memory expansions became available. I think Psion chess was one of the guilty ones.

However, relative addressing simply means, relative to where the program counter is. The program counter is always pointing at the address of the instruction in memory after the current one. An example of relative addressing is this small loop and the jump back to the start of the loop:

```
Start MOVEQ #1000,D0  
Loop  SUBQ #1,D0  
      BNE.S Loop(PC)
```

This is a small and totally useless fragment of code. The relative address mode is in the BNE.S LOOP(PC) instruction - it says - branch to the label called loop, relative to where the program counter is currently pointing, if the result of the subtraction was not zero.

The jump is specified in the code as a negative number, not the actual address of where the label 'loop' is at.

This negative number (in the example above) is how many bytes are to be added to the program counter to get the address of the next instruction to be executed. The jump can be forwards as well as backwards. Using relative addressing means that the program can be loaded anywhere in memory and still work. If absolute addressing was used, the program would always have to be loaded at the same address if a crash was to be avoided.

The example above is the equivalent of the following SuperBasic code :

```
1000 REMark Start  
1010 LET DO = 1000  
1020 REMark Loop  
1030 LET DO = DO - 1  
1040 IF DO <> 0 THEN GOTO (1040 - 10)
```

Address Register Indirect

This mode is called 'indirect' because the address register in question is not the operand in the instruction. It simply serves as a pointer to the operand. In an earlier example we cleared out the first byte of screen memory by using absolute addressing like this:

```
CLR.B $20000
```

This instruction could have been carried out using address register indirect mode as follows:

```
MOVEA.L #$20000,A1  
CLR.B (A1)
```

All that this is doing is setting address register 1 with the value 131072 (decimal) which is 20000 (hexadecimal). It then clears out the first byte at that address. This is the same as this SuperBasic example:

```
1000 LET A1 = 131072  
1020 POKE A1,0
```

The register's name is put in between a pair of brackets to signify that it is the memory address held in the register that will be acted upon and not the register itself.

Register Indirect With Displacement

This mode is similar to the above, except that a displacement is added or subtracted from the address register to give the final address to be operated upon. Using the above example again, we can zeroise the first 4 bytes of screen memory as follows :

```
MOVEA.L #$20000,A1  
CLR.W (A1)  
CLR.W 2(A1)
```

This time we use word sized operations, these simply affect 16 bits instead of 8 as with the byte

sized operations. The displacement is the number outside of the brackets and it is added to the address registers contents to create the address to be operated upon. The displacement can be any signed number that will fit into 16 bits (-32768 to +32767).

Register Indirect With Displacement And Index

It's starting to get complicated now. This is another mode where we have an address register and a displacement to consider, but this time we have an index as well. In this case the displacement has been reduced to 8 bits only giving a range of -128 to +127. The format of this addressing mode is :

```
CLR.W 2(A2,A0.L)
```

The contents of A2 is added to A0 to get the first address then the displacement is added to give the final result. The 16 bits of memory at the final address is cleared out. (Like `POKE_W A2 + A0 + 2, 0`).

In this case the entire 32 bit value of A0 is added to A2, this is indicated by the 'L' after the second register - the index. If the suffix had been omitted or was 'W'. (which is the default if omitted) then the lower 16 bits of A0 would have been used instead of the whole 32. Take note that the 16 bits will be 'sign extended' to a full 32 bits and this can have unpleasant side effects if the value in bit 15 is a 1 as this will cause a negative index to be generated. There will be more on sign extension later.

The first register specified is always treated as 32 bit (L) and does not require a 'L' suffix - most, if not all, assemblers will reject it anyway.

Register Indirect With Pre Decrement Or Post Increment

These addressing modes are used for stack operations, usually. The format of the pre-decrement instruction is :

```
MOVE.L D0,-(A7)
```

And for post-increment it is:

```
MOVE.L (A7)+,D0
```

The actions carried out are as follows for pre-decrement:

The value in A7 is decremented (reduced) by the size of the data to be stored (byte, word or long) then the contents of D0 are stored at the location pointed to by A7.

In SuperBasic this equates to the following code fragment:

```
1000 LET A7 = A7 - 4
```

```
1010 POKE_L A7, D0
```

The opposite action takes place with post-increment, as follows:

The contents of the memory address pointed to by A7 is copied into D0, then the address held in A7 is incremented by the size of the data just copied (byte, word or long).

Again, this equates to:

```
1000 LET D0 = PEEK_L(A7)
```

```
1010 Let A7 = A7 + 4
```

Immediate

This is probably the simplest of all the addressing modes. It simply means that the data specifies the address value. For example:

```
MOVE.L #100,D0
```

This copies the value of 100 into data register 0. The hash sign indicates that the data is copied directly into the register. Do not get confused between this instruction and:

```
MOVE.L 100,D0
```

Note that there is no hash. This instruction means load the CONTENTS of address 100 into register D0. This is not the same! This is a good source of confusion for beginners - I know all about it, and sometimes still make this mistake!

The next part will deal with the instruction set.



HP-Deskjet UNDER CONTROL

Wolfgang Uhlig

No idea how to program your HP Deskjet printer? Do the control codes (if you ever found a description) confuse you? Wolfgang tells you how to drive your HP from BASIC or QD, for example.

When I bought my Deskjet 690C I soon got very dissatisfied with what I could get out of it using the HP printer-driver for Text87plus4, my favourite word processor on the QL (QXL,QPC,

whatever). This printer-driver pretends to support more fonts than the HP 500 actually has - but prints the ones there not in all the possible widths and heights!

So I tried to find out what could

be programmed from BASIC or to integrate into an editor (in my case: QD).

This wasn't quite so easy: along with my 690C came a beautiful small book with lots of tips and tricks for WIN95, lots of advertisements for Hewlett Packard products and so on but unfortunately not a single line of printer commands. The only very small hint I could find were the names of the 4 built-in fonts and the possible

height and width of them, which in the end appeared to be rather important.

A friend of mine had an older manual of an HP 500, in this manual there were printer commands: a start!

After a couple of telephone calls with the HP help-line in Amsterdam - the people there are really very friendly and helpful, but they are completely unable to imagine that there are people with anything other than Windows; even for DOS there are still 'drivers' available!? - I finally got two other manuals, one was the HP-PCL, GL/2 and PCL Tables for Laser printers and the other one was a Printer Command Comparison Table for all HP printers.

What I've found out is that we (QLers) can use HP printers very well except for colours. I've tried a lot, but up to now I have not had any success at getting my printer to print in colour. So, if anyone knows a solution for this: please write to me and/or to QL-Today!

Getting HP printers to print exactly what you want them to is not as easy as it is with Epson printers.

The PCL (Printer Commands Language) commands are longer than Epson commands AND have to be given in a special order (which by the way is not very logical) AND you have to take care of the values for width and height AND not all heights go together with all widths (and vice versa) AND italic doesn't go together with some heights or widths AND italic doesn't go together with Letter Gothic anyway AND proportional has other heights than non-proportional AND ...

You see, it is not at all difficult to get strange results.

When testing the commands, I found it very annoying that there is no chance to test different commands on one page and look at

the result immediately, because the printer doesn't begin its work before you've given a Form Feed Command or before the 'pagebuffer' is full. Thus you waste a lot of paper with only some lines of text on it.

After all that 'frustration' now the good news: if it all works, it works fantastically well!

In the following examples, I'll give the commands in two versions, one with CHR\$(27) which is the start of any HP command followed by a combination of letters and numbers. The other with BPUT and the decimal values of the above mentioned letters and numbers.

The most important commands (in the form of a BASIC-program) in the correct order are:

Open the print channel
100 OPEN #3,par

Printer reset
120 BPUT #3,27,69
Or PRINT #3,CHR\$(27)&"E"

Line termination command, absolutely necessary!!
130 BPUT #3,27,38,107,50,71
Or PRINT #3,CHR\$(27)&"&k2G"

Portrait or landscape: portrait is default, for landscape change the last but one number to '49' or '1' respectively.

140 REMark BPUT #3,27,38,108,48,79
Or REMark print #3,CHR\$(27)&"&100"

[Achtung: Null und Buchstabe O, nicht Doppel-Null!]

Spacing: proportional has priority, which means, if you choose proportional, you won't be able to print Courier (or Univers with 690C) because they are fixed spacing fonts. Here, proportional is ASCII '49' or decimal '1' and fixed spacing is '48' or '0'

150 BPUT #3,27,40,115,49,80
Or PRINT #3,CHR\$(27)&"(s1P"

Pitch (characters/inch): only for non-proportional of course. Take care: depends on font and height of it and above, whether it is italic or not.

160 BPUT #3,27,40,115,50,52,72
Or PRINT #3,CHR\$(27)&"(s24H"

Height (in points): same problems as before

170 BPUT #3,27,40,115,49,50,86
Or PRINT #3,CHR\$(27)&"(s12V"

Style: Upright is ASCII '48' or decimal '0' italic is '49' or '1'

180 BPUT #3,27,40,115,48,83
Or PRINT #3,chr\$(27)&"(s0S"

Stroke weight: normal is ASCII '48' or decimal '0' bold is ASCII '51' or decimal '3'

190 BPUT #3,27,40,115,48,66
Or PRINT #3,CHR\$(27)&"(s0B"

Typeface family:

1. COURIER

200 BPUT #3,27,40,115,51,84
Or PRINT #3,CHR\$(27)&"(s3T"

2. CG TIMES

210 BPUT #3,27,40,115,52,49,48,49,84

Or PRINT #3,CHR\$(27)&"(s4101T"

3. LETTER GOTHIC

220 BPUT #3,27,40,115,54,84
Or PRINT #3,CHR\$(27)&"(s6T"

4. UNIVERS (not with HP 500)

230 BPUT #3,27,40,115,53,50,84
Or PRINT #3,CHR\$(27)&"(s52T"

Now you can write your text or whatever you want to print:

240 PRINT #3,"this is mytext"

FORM FEED: is necessary to force a print out

250 BPUT #3,12
Or PRINT #3,CHR\$(12)

Close printer channel
260 CLOSE #3

TF Services

superHermes

A major hardware upgrade for the QL

- All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound
- IBM AT keyboard interface (plus foreign drivers)
- HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps) Driver for SERIAL MOUSE supplied. Other uses include RTTY/graphics tablet etc
- THREE spare I/O lines (logic) with GND/+5V
- Capslock/scrolllock LED connector
- Turbo/keylock connectors
- 1.5k user data permanently storeable in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software) £90 (£92/£87/£90)
 IBM AT UK layout Keyboard £22 (£24/£23/£27)
 Serial mouse £11 (£13/£12/£14)
 Capslock/scrolllock LED £1 (£1.50/£1/£1.50)
 Keyboard or mouse lead..... £3 (£3.50/£3/£3.50)
 High speed serial (ser3) lead £4 (£4.50/£4/£4.50)

Hermes available for £25 (£26/£24/£27) Working ser1/2 and independent input, debounced keyboard & keyclick.

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.
 Cost (incl keyboard lead)...£53 (£55.50/£51/£53.50)

Minerva

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

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

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

First upgrade free. Otherwise send £3 (+£5 for manual if reqd).
 Send disk plus SAE or two IRCs

MK1...£40 (£41/£40/£43) MK1L...£65 (£66/£63/£67)

QL REPAIRS (UK only)

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

£27 including 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the Sinclair QL
NOW BEING SHIPPED

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent FLASH memory (ie there when the QL is switched off) which can be written to by the QL.

The software to access it is loaded automatically at power up/reset. It uses a directory driver written by Tony Tebby, and logic code from Stuart Honeyball. You can even load ROM images.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed (reading at over 1mbyte per second).

It is an extremely small and compact circuit board, and has hard gold edge connectors, eliminating contact problems and corrosion.

2 mbytes RomDisq.....£39 (£41/£37/£40)
 4mbytes RomDisq.....£65 (£66/£63/£67)
 8 mbytes RomDisq.....£98 (£100/£95/£99)
 Aurora adaptor.....£3 (£3.50/£3/£4)

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/£38/£44)

4 amp total (for motors etc) £45 (£48/£43/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... £25 (£28/£23/£27)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.. £25 (£28/£23/£27)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temperature measurements, sound sampling (to 5 KHz), x/y plotting..... £30 (£31.50/£29/£30)

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

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

Data sheets..... £2 (£2.50/£2/£3)

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

QL SPARES

Keyboard membrane..... £12 (£12.50/£12/£13.50)

1377 PAL £3 (£3.50/£3/£4)

Circuit diagrams £3 (£3.50/£3/£4)

68008 cpu or 8049 IPC £8 (£8.50/£7.50/£9)

8301/8302 or JM ROM or serial lead.. £10 (£11.50/£10/£11)

Power supply (sea mail overseas)..... £12 (£17/£16/£21)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH (No Eurocheques). SAE or IRC for full list and details

8 MAR 98

VISA

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: 01442-828254

Fax/BBS: 01442-828255

tony@firshman.demon.co.uk

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

MasterCard

According to the HP-manual (HP 500) the command for stroke weight has to be sent in the correct order (see above) but it seems to be no problem if you send it alone in order to make a single word or a sentence bold for example.

The commands for underlining are independent from order rules. You have four sorts of underline: single fixed, single floating, double fixed and double floating, the latter two being really very attractive.

Here the commands:

single fixed:

27,38,100,49,68

Or CHR\$(27)&"&d1D"

single floating:

27,38,100,50,68

Or CHR\$(27)&"&d2D"

double fixed:

27,38,100,51,68

Or CHR\$(27)&"&d3D"

double floating:

27,38,100,52,68

Or CHR\$(27)&"&d4D"

disable underline:

27,38,100,64

Or CHR\$(27)&"&d@"

Related to the height of the fonts, it may be interesting to alter the line spacing. The command for that is:

27,38,108,#.#,68

Or CHR\$(27)&"&1#D"

whereby '#' is a number out of: 1,2,3,4,6,8,12,16,24,48 and '#.#' the decimal value(s) of this number. It means: # lines per inch

So for example

27,38,108,54,68

Or CHR\$(27)&"&16D"

means: 6 lines per inch (default) and

27,38,108,50,52,68

Or CHR\$(27)&"&124D"

means 24 lines per inch (which is very narrow)

Sometimes you want to print in columns but you would like to do this writing with a proportional font. Then you will have to position

the (virtual) cursor horizontally. For example, if you define:

tab1\$=CHR\$(27)&"&a40C"

tab2\$=CHR\$(27)&"&a80C"

(40 and 80 being at your choice) then

PRINT #3,a\$;tab1\$;b\$;tab2\$;c\$

will print a\$.b\$.c\$ in three columns.

To get the right position, you must experiment a little, surprisingly the HP 500 and the 690C give different results on the same 'TAB'!

Of course everything only works with TRA 3 (SMSQE) or with IBM_TRA on Minerva or others.

The fonts can be printed in the following combinations:

COURIER:	5, 10, 16.67, 20 cpi	- point 6, 12, 24
	italic 5, 10, 20 cpi	
CG TIMES:	proportional	- point 5, 6, 7, 8, 10, 12, 14
LETTER GOTHIC:	6, 12, 24 cpi	- point 6, 12, 24
LETTER GOTHIC:	16.67 cpi	- point 4.75, 9.5, 19
UNIVERS:	proportional	- point 5, 6, 7, 8, 10, 12, 14
		- italic 5, 6, 10, 12 point

From BASIC the BPUT commands are most comfortable (except for the 'TABS'). With a little testing you can design fantastic pages!

From within QD or other editors that will accept CHR\$(27), the string-commands offer an interesting possibility of formatting a text. You could give your preferences first, then write your text. Within the text you can swap stroke weight or change to underline and back, you can play with different heights, italic and so on.

Simply write one of the above described commands where CHR\$(27) has to be the unprintable ESC-code. In QD you find this one in the code-table. That looks a little bit strange (CHR\$(27) is '█' in QD), for example:

This is normal text, █(s3B now it's bold and █&d3D now it's double underlined, too but it works fine. Of course you have to take care of what you write in the printer preamble of QD.

(I don't know whether this is also possible in Quill, because I haven't used it for years, Text87plus4 won't accept CHR\$(27)) *It might be possible if you use a translate sequence to convert an unused character to be output as CHR\$(27) instead - Editor!*

As already said, I've tried to print in colours, I even have found a command for that, but up to now it refuses to work correctly.

If anyone is interested in a small PE printer driver for HP 500/690 in which most of the above features can easily be arranged by simple mouseclicks, he/she should get

my "HP-Driver.zip" from Jochen Merz's BBS. In it there is the Basic-sourcecode, the menu, created with the EASYPTR-suite, one version compiled with and another without runtimes, this text and a small text concerning the Basic program itself. You can get everything, also apart, from me by e-mail, too.

And now, have fun with your HP printer!

Email: wolwol@compuserve.com



QLAY 0.85 – First Look & More

Al Feng

Jan Venema (<http://www.inter.nl.net/hcc/A.Jaw.Venema>) recently made a new version of QLAY available (03 May 1998). QLAY is a freeware QL emulator and Al has had a closer look at the recent version for you.

QLAY is a freeware QL emulator; and with the help of others, I received a copy of the most recently posted ZIP files -- first, version 0.84; and subsequently, QLAY 0.85. For all practical purposes, QLAY requires a 486-or-better running at 66 MHz or faster with 8 Meg of memory. The emulation creates a 640K QL.

The most recent release of QLAY actually comes in two versions - the traditional DOS program and one that can be launched directly from within WINDOWS 95.

That's the good news. The better news is that Jan Venema has indicated that he is already working on the next version and it should be ready by the time you read this.

QLAY 0.85a

The DOS based code functions in a manner similar to the previous versions. First you load the CWSDPMI.EXE file, and then the QL code and extensions are loaded, and then the NFA (Native File Access) code is loaded. The NFA code give QLAY the ability to access the host PC's hardware and is a remarkable piece of coding.

Depending on the speed of your computer, the code is processed, and then the emulation begins.

The obvious change is the inclusion of the QLAYRC file. This file is an ASCII script which is essentially a dedicated BATCH file which both version 0.85a & 0.85b look for when the EXECutable file is loaded. This is how the (stock) QLAYRC file looks:

```
-r js.rom
-c c000@nfa.rom
-f 1000
```

The syntax appears to be just different enough that I was not able to successfully load the TK2_EXTensions on the first several tries.

I finally e-mailed Jan Venema to get the "proper" syntax:

```
-r js.rom
-c 0c000@f:\tk2.ext
-c c0000@f:\nfa.rom
-d 2
```

The new version comes with an image of the Dynamic RAM code; but, I have not been able to load it to date. This code (or, similar) is necessary if you want to use the XCHANGE program which sets up temporary files in RAML...

All other tested programs ran, including the TURBO compiler.

Exit the emulation via CTRL-ALT-DEL key combination when loaded from within WIN95.

To exit from a plain DOS load, I believe you have to reset the computer.

QLAY 0.85b

For reasons that are always suspect, I upgraded my IBM compatible PC recently and the OS with a copy of WIN95. While subjecting myself to the OS upgrade was semi-traumatic (i.e., I inadvertently wiped out all my files!), having WIN95 allows me to give the new QLAYWEXE (aka QLAY 0.85b) a try. The QLAYWEXE is designed to run directly from either a WIN95 icon or from the WIN95 DOS prompt.

When you load QLAYWEXE, you will see a modest window with the

following options: File, Screen, Language, Help.

Using the 'File' selection, you essentially verify (or, load?) the equivalent of the CWSDPMI.EXE code by selecting 'Init!'. You then select 'Go!'. Very shortly after that, you will see the familiar speckled screen followed by the QL's startup screen.

I was not able to load an image of my MINERVA ROM code. I was told that this is only viable with the JSL1 (i.e., older) code (which I have). I have to presume that my MINERVA code file was corrupted in the transfer.

Your language choice is English or German, with the default now being English.

I believe that all my problems can be related to implementation of the QLAYRC file.

If your QL program uses ALT keys, then you will want to use the DOS version due to a key-stroke conflict which will exist with WIN95.

HEY, I KNEW IT COULD BE DONE!

Many of you may recall that one of my biggest complaints about SMSQ [the OS used for the QXL & QPC emulators] is that the display that is generated is less than I think it could be.

Jan Venema modestly states that "the Windows95 version has better display support."

With the WIN95 version of QLAY, Jan Venema has demonstrated that the display is not only scaleable, but that it can be done in an efficient manner. In version 0.85b, the screen can be selected before the emulation OR during the emulation. The five display sizes which he has provided are:

```
Size 1: 512 x 256
Size 2: 512 x 341
Size 3: 768 x 512
Size 4: 1024 x 683
Size 5: 1024 x 768
```

The great news is that 'Size 3' actually fills the usable screen space (9.5"x6.75"; or, about 11.75" diagonal on a 14" VGA monitor). I have to mention that the first several times I switched to 'Size 3' that it was just a little larger than the available screen space. Initially, I would estimate that 'Size 3' was being displayed as one line higher than a normal screen, and one character wider.

This over-fill stopped after I added the CWSDPMI.EXE file to the same directory as the QLAYW.EXE program. Coincidence? I'm not sure. I haven't removed the file to verify this.

'Size 4' & 'Size 5' definitely over-fill a standard VGA screen.

'Size 1' is similar to the image created by SMSQ, but the vertical is slightly compressed due to the "top line" being occupied by the control bar (7.5"x3.75").

'Size 2' is taller than the standard QLAY (0.85a) display and the SMSQ display (7.75"x5")

The font quality varies, with the font created in 'Size 3' as best described as being a compromise (1/4" font height). The font in 'Size 1' & 'Size 2' are similar to CSIZE 0,0 (1/8" & 3/16" font height, respectively); and, the font in 'Size 4' & 'Size 5' appear to be similar to CSIZE 2,1 (5/16" & 3/8" font height, respectively).

As one might suspect, there is a price to pay for the scalability of the screen display with the larger displays using more of the host's resources. Ergo, the larger the display selected, the slower the emulation.

As I mentioned, the size can be switched at any time during the emulation when using version 0.85b; so, the smaller size can be selected when greater speed is required, and to the full screen mode at other times.

TASK SWITCHABLE

Both versions are easily task switchable from within WIN95 by using either the ALT-ESC or the ALT-TAB key combination; or, the pointer arrow with QLAYW.EXE if your task bar is visible.

Of course, use the CTRL-ESC key combination to load other programs; or, the pointer arrow with QLAYW.EXE.

QLAY 0.85b is unstable if other programs are dormant in the background; but, Venema assures me that this problem will have been fixed with subsequent releases.

MINERVA COMPATIBILITY

Only the older (JSL1) MINERVA code works. Even so, I was not able to load the MINERVA ROM image that I made. I will try this again at a later date.

QLAYT

QLAYTEXE is currently an essential DOS utility for effectively using QLAY.

The QLAYTEXE is found on earlier ZIP files, so you should ensure that if you are downloading from Venema's web site.

There is a QLAYT program which I believe is for use with LINUX, and this should not be confused with the QLAYTEXE program.

QLAYT allows you, first and foremost, to add programs to the active directory file (QLAYDIR).

With the anticipated addition of QDOS floppy access, you should be able to add files to your hard drive directly from floppy.

The most important thing to know is that if you want to INSERT a file into the QLAY dir, the DOS syntax is as follows:

```
C:\ QLAYT -i QLATter -d 31790
```

QLAYT switches are case sensitive. Note the file being inserted into the existing QLAYDIR file is

'QLATter' which has a size of '31790 bytes'. The file size on some files does NOT need to be declared.

LIMITATIONS

Some of the documentation presumes you know more than I do.

The QLAYRC file requires the syntax be written a bit differently than in the past.

TK2_EXTensions must be provided by the user.

QLAYW.EXE is designed to run without the CWSDPMI.EXE file, but it appears to be more stable when it is accessible by the QLAYRC file.

Sub-DIRECTORIES are not currently supported.

While I don't know how to use the virtual "mdv" drives for storage (this is probably something that I should learn how to do in the near future because I believe these can be used as effective substitutes for sub-DIRECTORIES), and QLAY's I/O is still limited,

While floppy access should be implemented by the time you read this, QLAY still does not have SERIAL access; but, presumably this will be a future function.

WHY QLAY?

I have been told that there is another freeware, QDOS emulator (WinQL -- written by the author of the emulator that runs on the 68000-based MACs); but, I haven't seen a copy, yet; so, I do not know what limitations exist with it (if any).

[Do you mean Q-EmuLator? - Editor]

The question that I am sure that some people will ask is "Why would I want to use QLAY?" (or, any other emulator).

There are actually reasons that QLAY, as tested, is more than adequate for most people who need to use a QDOS emulation.

The fact that files are created saved to your hard drive in DOS format may be a benefit for some

QUANTA

Independent QL Users Group

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

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

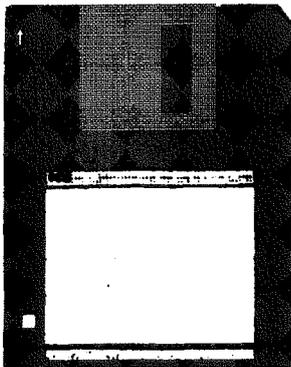
Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our FIFTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407



Some speed comparisons

Native 68k

System/CPU	print	function	String	Configuration/Emulator
QL original	980	840	1100	128k, JS-ROM
SandyQL	1080	1040	1360	512k, Floppy, Par. Port
GoldCard	2580	4280	5680	
Super GoldCard	5680	10260	13820	
Amiga 68040/25	9480	16540	23000	AmigaQDOS (JS/3.24)
QXL-20	14300	27900	70640	
Amiga 68060/50	10480	48750	73880	AmigaQdos 3.24 Beta (Blitter)

Emulated 68k

System/CPU	print	function	string	Configuration/Emulator
486/DX2-66	660	980	1300	QLAY081 -d2 (DOS)
486/DX2-66	1080	1360	1820	QLAY082 -d2 (DOS)
486/DX2-66	920	1440	1840	QLAY083 -d2 (Linux)
Amiga A1200	2500	1800	2000	QL emulator, with 68060/50MHz
Pentium/150	2360	3680	4960	QLAY082
CyrixPR200MMX	3780	5800	7240	QLAY085a
CyrixPR200MMX	3020	4860	6120	QLAYW085b size 1
CyrixPR200MMX	2560	3640	4560	QLAYW085b size 2
CyrixPR200MMX	2340	3280	4100	QLAYW085b size 3
CyrixPR200MMX	1920	2540	3180	QLAYW085b size 4
CyrixPR200MMX	1900	2500	3100	QLAYW085b size 5
PentiumPro/200	6000	8000	11000	QLAY082 -f2900
Pentium/200	5260	9340	12660	QLAYW085 size 1
Pentium/150	7120	10840	28120	QPC

people because it eliminates the file conversion step.

For example, I recently prepared a manuscript which began in Quill, but which I later imported into WORKS for formatting and printing; and, I was glad for the better fonts available through WINDOWS (having said that, I have to admit that I have not used or seen any of the output generated by the programs from PROGS; so, I don't know what I am missing with regard to QL-to-printer output).

If you have a proprietary pro-

gram that was written to run on QDOS, you no longer need to be concerned with porting it over to run on a PC since you can simply run it using the QLAY emulation. Or, for example, if you do not have the PC version of the PSION Suite, but prefer to use DBEasy (as I do) for your database needs, then QLAY will allow you to continue to do so.

If a program will run on a standard QL, then more-than-likely, it will run with the QLAY emulation; but, the exception might be older

games which require a key.

Unlike the QPC emulator, all versions of the TURBO compiler work.

Unlike the QPC emulator, QLAY is (currently) freeware.

All in all, I think that QLAY is excellent QDOS emulator for a person with a suitable host PC who has a need to access existing QDOS programs and files.

**HAPPY TRAILS, AND
COMPUTING, TO YOU**

The 1998 USA QL Show

John Impellizeri

You haven't been to a QL show in the USA? Definitely a mistake! This year's show was very well attended. If you read this review and hear about the plans for next year's show (... more on this in a few months) then you may find that it's the right time to save your money for a flight to the USA next year.

On Saturday, May 23rd, the sixth US QL show was held once again in Bedford, Pennsylvania, the same place as last year's show. Held in the banquet room of the Carriage Inn restaurant, by all accounts the show was another success. Traders present included the show's organizers, Frank and Carol Davis of FWD Computing, Jochen Merz of Jochen Merz Software, Tony Firshman of TF Services, Roy Wood of QBranch, Albin Hessler, Paul Holmgren, Francois Lanciault, Zeljko Nastasic (Nasta of Qubide and Aurora fame), Bill Cable of Wood and Wind Computing, and the NESQLUG QL users group. Don Walterman represented Quanta selling memberships and Quanta merchandise. I brought a working copy of the QBox-USA BBS.

While the official show wasn't until Saturday, things get going Friday night as everyone who has arrived in town gathers at the restaurant for dinner, drinks and conversation. Friendships and acquaintances are renewed, new faces are introduced and the excitement for the show really starts building.

Afterwards, and before it gets to be too late, the banquet room is opened to allow the traders some time to set up.

Trying to get to sleep Friday night reminded me of Christmas

30 show visitors, not a lot but about the usual amount at the past few US shows. It's mostly the same people every year but I did see a few new faces this year. Unfortunately, I don't remember their names. I hope we'll see them again next year.

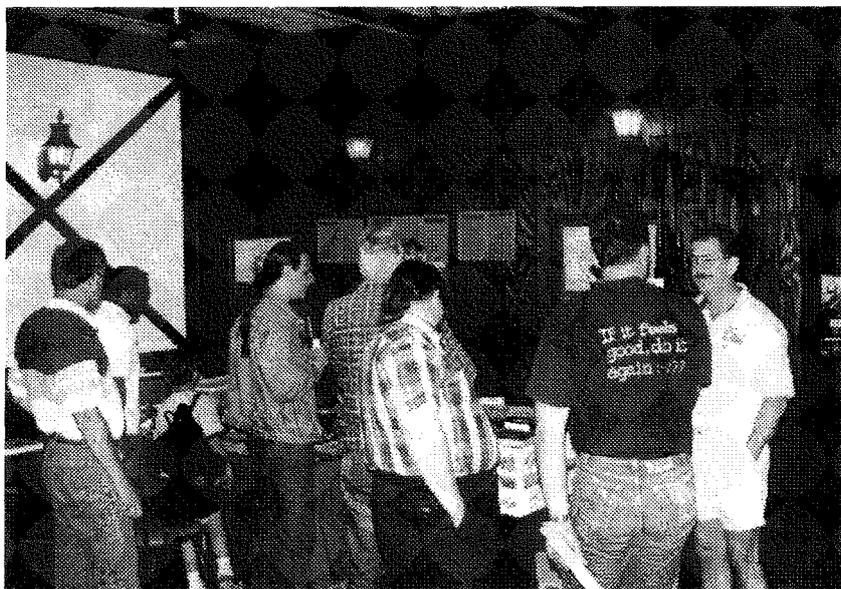
From my viewpoint it looked like most of the traders tables stayed busy all day. Jochen was busy like always updating disks and promoting QL Today. Back issues were available as was the new copy of volume 3 issue 1. Albin Hessler had

sales and updates to Cue-Shell and the EasyPTR series. Frank and Carol Davis had their wide range of QL and Z88 software and hardware. In addition they also had other types of 'software': T-shirts and mouse pads printed with a number of different QL and Z88 logos and pictures along with a special show shirt. I waited too long and missed out on one of these. Roy Wood was also busy updating disks and selling all sorts of QL goods from his colorful new catalog (produced and printed using QL software!).

Tony Firshman had the new ROMDisq available along with Minerva, Hermes, and super-Hermes. Also on display were the Lego models controlled by Minerva and I2C interfaces. I have read about these



even as a kid, with lots of anticipation for the next day. Saturday morning found the usual crowd waiting for the show to open. Not including traders, I counted about



at other QL shows but I believe this was the first time they appeared at a US show. Paul Holmgren was there with boxes of older Sinclair and Timex computer items. For those of us who started out with Sinclair ZX81s and Spectrums and Timex 2068s, this was a good chance to complete your collection of older hardware, software and books. He had a pretty good supply of QL motherboards and spare parts too.

NESQLUG, a Quanta subgroup, was there offering memberships along with current and back issues of their very well done newsletter. Bill Cable of Wood and Wind Computing had available his line of software designed to make Psion's Archive more friendly and easy to use. Don Waltermann represented Quanta, signing up new members and selling some of the merchandise that is offered in the Quanta newsletter.

I brought along a working copy of the BBS and took care of any requests for copies of PD software and shareware. Not every QL user is 'wired' (access to the internet) or cares to call BBS's, so this was an opportunity to get copies of some of this stuff.

One nice feature, I feel, at both last year's and this year's show is the 'forced' lunch. By forced I mean that at around noon, the show closes

(everyone is kicked out and the room is locked), and a buffet style lunch is served the cost of which is included in the show admission. What's nice is that it was held in the restaurant where the show was held so there's no travelling. But the best part is that if you're anything like me, you can get so wrapped up in all things QL that



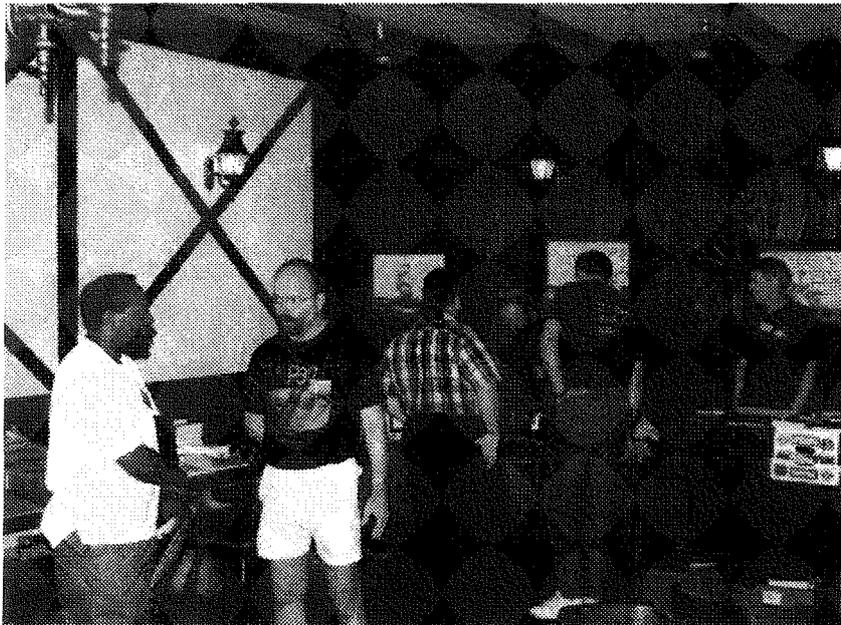
you forget to eat and after a while you start to wonder what that odd pain is in your stomach and then you realize that it's hunger!

As is usual there were a number of talks and demos throughout the day given by the various traders

with others trying to do business. I think this worked rather well. Nasta gave a talk about the Goldfire, the replacement for the Super Gold Card. He brought along a prototype to show. A number of people wanted to buy it from him on the spot! It is a very impressive looking board, even for a prototype. Some of the reasons and decisions for its various features were given. It was a thrill for me to get to meet the man behind such things as QUBIDE and Aurora. Tony Firshman gave a talk and demo of the new ROMDisq, the flash memory card for the QL. Available in 2 meg, 4 meg and 8 megabyte versions, it is very small, fast and just plain

works! Depending on your needs, you could put all your software on it and just carry it around, plugging it into any available or borrowed QL which is what I think Tony did for the US show.

Francois Lanciault showed Paragraph, a new word processor for the QL that he is developing. It is not a finished product yet, but what he has so far looks very promising. It will run under the PROWESS window manager and be fully WYSIWIG. It is planned to have advanced features such as tables and drawing tools in addition to the usual word processing functions. I think there



and developers. A change this year was to hold these in a different room so as not to compete

may have been a few other talks or demos but I missed them being busy copying files and trying to

get some of my own personal purchases and updates done. I apologize if I left anyone out. Last year Bill Cable, Al Boehm and Herb Schaaf all gave demos, maybe they did this year too? Sorry if I missed any of you guys.

The show seemed to end too quickly (don't they always!?) and we all had a little time to dismantle displays and go back to the hotel rooms to relax for a short while. Then we were back at the Carriage House Inn to have dinner in the banquet room where the show was just held. With good food and drink and lots more conversation, it is a great way to end the day, although not everyone ends it here. Some of the talking and

drinking goes on until they kick the group out so they can close the restaurant. Others go back to the hotel to start checking out new purchases or to help with on-the-spot upgrades and repairs. Sunday morning finds most people in the hotel lobby enjoying the continental breakfast and saying goodbyes and getting ready to depart.

I think this year's US show was another success, I certainly enjoyed it, and rumour has it that next year may be on two consecutive weekends, one perhaps in Bedford again and the following weekend in California to accommodate the QL users on the opposite side of the nation who can't always make it out east. I can't wait!



It could be You!

Geoff Wicks

Geoff Wicks' tales of horror with harddisk crashes.

A giant finger points from the sky as a deep voice booms "It could be you".

UK readers will recognise this as the advertising of the National Lottery. Recently, when a giant finger was pointing in my direction, it was decidedly off-message. It was not telling me I was about to become Britain's next millionaire, but that I was next in line for a hard disk failure. Fortunately it was only a 3-day wonder, but it could have been different.

Our chances of a hard disk failure in the next year are far greater than those of winning a decent prize in any lottery, but we still continue to buy our lottery tickets and to neglect our hard disk maintenance.

As computer users we have a naive faith in magnetic media. I learnt my lesson way back in the Spectrum days, when I bought some "own name" tapes from a reputable department store. They caused many hours of frustration with aborted attempts to load

programs, lost data and frequent cleaning of tape heads covered with an oxide deposit.

When disk drives took over I resolved never to use cheap disks, because I knew that the tape heads of drives are less accessible and more delicate than those of cassette recorders.

In his QL Today articles Mark Knight warns us against cheap disks and it is a warning I can reinforce. If you ever have a floppy failure, do not throw the disk away. Open it up and have a look inside to see on what a flimsy thing your precious data is preserved. It will make you think twice about such practices as boring holes in disks to increase their capacity.

Over the three years I have been running JUST WORDS! I have used more disks than the average QL-user will use in a lifetime. About 2 years ago DD disks became almost impossible to obtain in the Netherlands, and I had to use what disks I could get. They were by no means the cheapest,

but were a quality less than what I would ideally use. I am getting many of these disks returned as clients upgrade, and a high proportion have damage to the sliding shield that protects the magnetic surface. Some are no longer usable and go straight into the dustbin. As the Dutch say, (but don't always follow), "Goedkoop is duurkoop" or "Buying cheaply is expensive".

But back to my hard disk. My working QL system is a QXL card in a 286 PC. I took it over from a family member when she upgraded to a Pentium, who in turn had taken it over from another family member when he had upgraded. This is a complicated way of saying it is very old.

The computer's hard disk had been upgraded from 20Mb to 120Mb, and so was non-standard, and being non-standard had its parameters stored in CMOS. When the computer battery ran out, all this data was lost. To make matters worse an obsolete computer uses an obsolete battery, and it takes a few days to find a shop selling obsolete batteries.

We become surprisingly dependent on our hard disks. Three years ago I worked happily on my computer with a pile of floppies beside me. Now I just have two or three containing word processing documents or other files on which I am currently working. All programs are loaded from the hard disk. Suddenly those programs were inaccessible. I knew they were there safely on the disk, but I could not load them. I felt like the alcoholic in front of the locked grille of a closed off-licence, or the heavy smoker whose lighter has just run out.

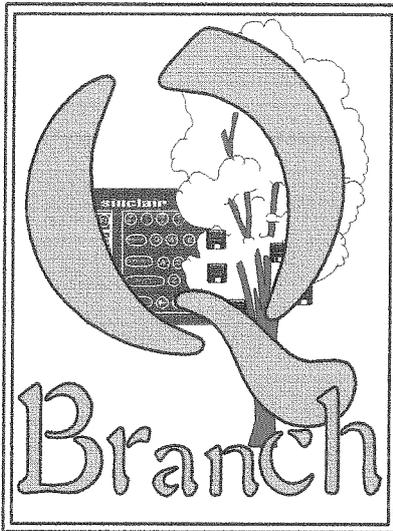
Having worked professionally with people who display problematic substance abuse behaviour - "addicts" to the politically incorrect - I know a few tricks. It is amazing what you can hide in

Q Branch

Feeling out on a limb ?

Reach out for Q Branch

Suppliers of Quality QDOS/SMSQ products
Hardware and Software



Hot news this issue is that we are reaching the end of the Super gold Cards and QXLs. There are only a few of these items left and they are disappearing fast. We will be offering trade in prices for The GoldFire when it is released later this year and having a backup device is essential if you use your system a lot. So don't delay - order now before our branches are bare !

Don't forget that a QXL in a 386 or 486 PC will perform faster than a Super Gold Card or QPC in anything slower than a 200 MHz Pentium PC. There are a lot of cheap PCs around these days so there has never been a better time to snap up a QXL. Look at the special QXL + SMSQ/E price below !

Q Branch Programs

The Knight Safe - Safeguard your hard disk data !

The Knight Safe 1 - standard version £ 30.00

The Knight Safe 2 - with compression £ 35.00

Q - Count £ 25.00

Pointer Driven Home accounting

Q - Route v1.06 £ 25.00

Route finding programme

Page Designer 3 S £ 40.00

Pointer driven desktop publishing for big screens

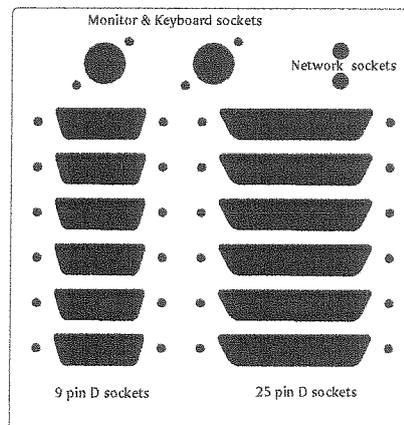
Hardware

QXL II	£ 180.00
Super Gold Card	£ 160.00
Recycled Gold Card	£ 60.00 *
Aurora	£ 100.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
The 'Braquet'	£ 16.00

* when available.

NEW PRICES !!!

The MC - Plate



All the socket holes you will ever need on the back of your tower case. Just rip off the silly PC thing and screw on this one. No cutting needed. What could be easier ?

Package deals

QXL II + SMSQ/E	£ 220.00
Aurora + SMSQ/E	£ 160.00
Aurora + SMSQ/E + Super Gold Card	£ 330.00
Above package + Monitor	£ 380.00

P.O. Box 7, Portslade, Sussex. BN41 2ND. UK.

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577

email qbranch@qbranch.demon.co.uk web : <http://www.qbranch.demon.co.uk>



We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.



The MinisQL is
Coming !
Watch these
or visit a local
QL Workshop !

Q Branch

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

ProWesS

ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

PROGRAMMING

QD v 9.15	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1 & 2 (together)	£ 33.00
Easyptr pt 3 (C library)	£ 16.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3
£ 15.00 ea / ANY 2 PROGRAMS £ 25.00 / ALL 3 PROGRAMS £ 35.00

!!! NEW VERSIONS !!!

THESAURUS V4.01 / SOLVIT PLUS 3 v 3.00

Pointer drivens versions of Geoff Wicks popular programs.
(needs Hot_text, WMAN and PTR_GEN or SMSQ/E to run)
Still the same price

£ 15.00

Upgrades from previous versions £ 2.50 + S.A.E.

SMSQ/E

Gold Card / Atari / QXL Version

£ 76.00

Various Atari versions : call for details

QPC v 1.43

£ 90.00 (£ 70.00 SMSQ/E owners)

We can now supply colour laptops with QPC already
installed ! The systems are secondhand and vary
considerably so you will need to call for details but a
typical starting price for a 486 DX 70 Toshiba with
350 Meg hard disk and 12 Ram complete with QPC
and Carrying case is

£ 500.00 (+ postage)

NEW PRICES !!!

Text 87

£ 79.00

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

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

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00
LDUMP	£ 26.00
Q Count	£ 25.00

You've been asking for it and here it is ! Due out late June.

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including
three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500
pages !

Call for
pricing details

lavatory cisterns, just in case. What you can get away with by opening a window, lighting a candle and placing a wet towel by the door. What you can do with a piece of silver paper and a light socket (Don't try that at home, kids). Neither of these was appropriate in my case, but I was soon trying every trick in the book to get that hard disk back in operation, but all to no avail. I had no choice. I had to face life without my hard disk. This was cold turkey for real, man.

"Aha", I hear Mark Knight say, "Did you back up your hard disk? I have this nice little program..." Yes Mr. Knight, I do back up my hard disk. I back up everything. In fact you need two back ups. One of the hard disk information as it is, and the other in a form that can be used directly from the disks. Never forget we customise our programs to run from hard disk.

Once I started to think about the problem rationally, I was soon devising a survival strategy. I assessed my computer needs for the next few days and realised I could keep these simple. I could load SMSQ from floppy, and had pointer environment files on disk that could be quickly adapted for use on the QXL. I made up a disk to load the essential DOS files, then SMSQ, the pointer files and QPAC2. I could continue word processing and program development. Even the use of Line Design would still have been possible, although I decided against it. I had a system that was cumbersome to use, but which worked.

In fact I had a QL in a PC that was unusable as a PC but usable as a QL. Most QL programs can still be run from floppies, but almost all PC programs have to be run from hard disk. PC wordprocessing would have been impossible unless I had reverted to using WordPerfect 4.2. We are fortunate as QL users that when we have a

hardware problem, we can usually improvise a way round it. Our PC colleagues are not so lucky.

Even when I had fitted a new battery my troubles were not over. I needed to know the parameters of the hard disk. Fortunately I had made a note of them. Then it proved impossible to enter these. After many attempts and some experimentation, I discovered that both the computer's manual and help screens gave incorrect installation information. Again an advan-

tage of being a QL user. When you have a QL problem, there is usually help at hand, either through a Quanta subgroup or even the trader. Where do you go for help as a PC user?

My hard disk problems are solved. That big finger in the sky is no longer pointing in my direction. So look after your hard disk and back it up regularly. The big finger is pointing at someone else. It could be you!



Gee Graphics! (On the QL?) - part 5

Herb Schaaf

Ellipses, Ellipsis, Eccentricity, Extencidity?

Thanks again for all the kind words and encouragement about this series from the people at the Bedford USA '98 QL show. Because of this series I'm learning a lot of things I never knew, and learning again that there's a lot more I'd like to learn about!

ELLIPSE is just another way to spell CIRCLE as far as the QL knows, and you may use the two terms indiscriminately for all it cares. If given only three parameters it wants to do a circle; if given five parameters it tries for an ellipse. Then too there are the corresponding ELLIPSE_R and CIRCLE_R that Remember where the graphic cursor was so that Relative shifts of the first two parameters can be made.

The QL ProCedure ELLIPSE uses 5 numerical parameters:

- Graphic Horizontal Center
- Graphic Vertical Center
- Graphic Vertical Radius
- Ratio of Horizontal Radius to Vertical Radius
- Inclination to Horizontal axis in Radians

The 4th parameter, is called eccentricity in some QL guides, but is really the 'aspect ratio' (width/height) of the ellipse.

"Eccentricity" is an example of a word that has different meanings depending upon context. When some QL guides refer to the eccentricity of an ellipse, it is different from the usual mathematical meaning of eccentricity as applied to the conic sections. The mathematical usage involves the ratio of the distance from a point on the curve to a foci as compared to the distance from the same point on the curve to a line known as the directrix. This ratio uses values from 0 (0 is a circle) to a little below 1 (1 is a parabola) for ellipses, and values greater than 1 produce hyperbola. We can think of the QL's 4th parameter as a measure of how much a circle having a fixed vertical size is stretched or extended in the horizontal dimension; perhaps a better term would be "extencidity". The QL permits this parameter to be negative or positive; if the ABSolute value of this ratio is 1 you get a circle, if the ABSolute value is less than 1 the circle is compressed in the horizontal dimension, and if the

ABSolute value is greater than 1 the circle is extended in the horizontal dimension.

There are limits when the QL refuses to draw the ellipse, as the shape approaches a line. Very small ratios and very large ratios just don't get done. The effect can be viewed in the program "Ellipse_gone_bas" where you see the QL trying, and then eventually giving up as the values become extreme.

The 5th parameter turns the ellipse about its center before drawing it, for example $\pi/4$ will rotate it 45 degrees counter-clockwise, and $-\pi/4$ will turn it 45 degrees clockwise.

Now have a look at listing 1 please.

Ellipsis in the form of '...' is used in writing to show that something has been left out of a quotation. I wish I knew the word for showing that something has been changed within a quotation (alteration, or perhaps the opposite of emendation?). I see that the program Show_ARCcircle_bas in QLToday May/June 1998 had a few changes along the way from me to the printed page. The changes involve non-ascii (3 keys at once!) QL characters that I've found convenient, but seem to have been translated before the final printing, perhaps by some printer translates. *[More likely by PC-QL email translations - Editor]*

I find that the symbol for degree is available on the QL as (CTRL SHIFT Z) = CHR\$(186). It came out as an acute accent in lines 430, 1110, and 1390. The QL provides an Up Arrow with (CTRL SHIFT 6) = CHR\$(190), on line 1390 it came out as an R in an oval. (How does an oval differ from an ellipse? - more about this further down.) The QL has a Down Arrow with (CTRL SHIFT -) = CHR\$(191) which came out as a blank on line 1390.

Listing 1

```
100 REMark Ellipse_gone_bas
110 REMark HL Schaaf June 3, 1998
120 :
130 WMON : PAPER 0 : INK 6 : MODE 4
140 SCALE 100,0,0 : CLS
150 CSIZE#0,3,1 : CLS#0
160 PAPER#2,4 : INK#2,0 : CSIZE#2,1,0 : CLS#2
170 PRINT#2\\,"E l l i p s e s"
180 PRINT#2\\ " A series of ever decreasing"
190 PRINT#2;" ratios is used, the vertical"
200 PRINT#2;" dimension staying the same."
210 PRINT#2\\ " the ellipse is shown normal,"
220 PRINT#2;" then turned 1/8, then 1/4"
230 PRINT#2;" of a circle"
240 prompt
250 FOR i = 1 TO 24
260 extendicity = 1/(i^2)
270 CLS
280 ELLIPSE 40, 50, 40, extendicity, 0
290 PRINT ' ratio = ';extendicity,
300 prompt
310 ELLIPSE 40, 50, 40, extendicity, PI/4 : PRINT'1/8',
320 prompt
330 ELLIPSE 40, 50, 40, extendicity, PI/2 : PRINT'1/4'
340 prompt
350 END FOR i
360 :
370 PAPER#2,4 : INK#2, 0 : CSIZE#2, 1, 0 : CLS#2
380 PRINT#2\\,"E l l i p s e s"
390 PRINT#2\\ " A series of ever increasing"
400 PRINT#2;" ratios is used, the vertical"
410 PRINT#2;" dimension staying the same."
420 PRINT#2\\ " Scaling is adjusted to fit."
430 PRINT#2\\ " the ellipse is shown normal,"
440 PRINT#2;" then turned 1/8, then 1/4"
450 PRINT#2;" of a circle"
460 prompt
470 FOR i = 0 TO 12
480 extendicity = 2^i
490 SCALE 100 * 2^i, 0, 0
500 CLS
510 ELLIPSE 40*2^i, 50*2^i, 40, extendicity, 0
520 PRINT ' ratio = ';extendicity,
530 prompt
540 ELLIPSE 40*2^i, 50*2^i, 40, extendicity, PI/4
:PRINT'1/8',
550 prompt
560 ELLIPSE 40*2^i, 50*2^i, 40, extendicity, PI/2
:PRINT'1/4'
570 prompt
580 END FOR i
590 WMON
600 STOP
610 :
620 DEFine PROCedure prompt
630 CLS#0
640 PRINT#0;" Any key to continue"
650 PAUSE
660 END DEFine prompt
670 REMark end of listing
```

Listing 1

```

100 REMark Extendicity_bas
110 REMark HL Schaaf June 4, 1998
120 REMark takes eccentricity as input
130 REMark and converts to ratio for QL
140 REMark displays ellipse and foci
150 REMark and circles for major and minor axes
160 :
170 REMark lines 190 to 230 allow this to be run
175 REMark as an executable job in SMSQ/E
180 IF VER$<"HBA" : GO TO 230
190 JOB_NAME "Extendicity"
200 OPEN#0,con
210 OPEN#1,con
220 OPEN#2,con
230 WMON
240 :
250 PAPER 0
260 INK 4
270 SCALE 100,0,0
280 CSIZE #0, 1,0 :CLS #0
290 CSIZE #2, 1,0
300 CLS
310 explain
320 REPEAT query
330 CLS#0
340 INPUT#0, "eccentricity ? ",ecc
350 IF ecc <0 OR ecc >=1 :PRINT #0;"0 to almost 1 ?":GO TO
340
360 REMark major axis will be 70 units
370 a = 35 : REMark semi-major axis
380 c = a*ecc :REMark foci distance from center
390 INK 4
400 CLS
410 PRINT "eccentricity ";ecc
420 REMark red foci
430 INK 2
435 REMark do both with only 1 call to CIRCLE with
440 REMark the semicolon in the line below
450 CIRCLE 40-c, 50, 1 ; 40+c , 50, 1
460 b = SQRT(a*a-c*c): REMark semi-minor axis
470 REMark major, minor circles
480 INK 4
490 CIRCLE 40, 50, a ; 40, 50, b
500 PRINT "ratio ";a/b
510 REMark white ellipse
520 INK 7
530 ELLIPSE 40,50,b,a/b,0
540 END REPEAT query
550 :
560 DEFINE PROCEDURE explain
570 PAPER#2,4:INK#2,0:CLS#2
580 PRINT #2\ " Here are some values for"
590 PRINT#2;" the eccentricity of a few"
600 PRINT#2;" heavenly bodies."
610 PRINT #2\ " shape of Earth .0818"
620 PRINT #2;" orbit of Earth .0167"
630 PRINT #2;" orbit of Mars .0934"
640 PRINT #2;" orbit of Mercury .2056"
650 PRINT #2;" orbit of Venus .0068"
660 PRINT #2;" orbit of Pluto .2502"
670 PRINT #2;" Halley's Comet .9675"
680 PRINT #2\ " enter values of choice"
690 PRINT #2\ " foci will be shown in red"
700 PRINT #2;" circles of major and minor"
705 PRINT #2;" diameters will be in green"
710 PRINT #2;" and the ellipse in white."
720 END DEFINE explain
730 REMark end of listing

```

A friend who knew I was trying to get a better understanding of 'ellipse' suggested I check out a recent (1996) book from the library: 'Feynman's Lost Lecture' by David L. and Judith R. Goodstein, ISBN 0-393-03918-8. I found it of interest, and you may also. The amount of eccentricity that Kepler puzzled over in trying to find the orbit of Mars is indeed small, so it is all the more marvelous that he was able to work it out. I've included a small program 'Extendicity_bas' which displays the eccentricity of a few 'heavenly bodies'; the shape of the earth, the orbits of selected planets and Halley's comet. It converts the mathematical 'eccentricity' into the QL's 'extendicity' ratio, and draws the ellipse, the foci, and the circles for the minor and major axes.

Please look at listing 2 now.

The planetary orbits look very much like circles don't they? Ovals look a lot like ellipses too. The difference between an oval and an ellipse is that the oval is made of lengths of circular arcs joined together, while the ellipse is its own shape. Ovals can be made that approximate the ellipse, and perhaps we will do some simpler ones in a future article.

Next time I hope to explore the question: "how far is it around the ellipse, or what is the perimeter of an ellipse".



Remember:

**we need
your hot
tips by the
15th of
August!**

What are all these addresses? - Part 2

by Nasta (Zeljko Nastasic)

Things have substantially improved with the SGC. Not only did it expand RAM, but an IO area was again reintroduced, and even the QL ROM can be still read, albeit at a different address. Notice that on the GC, the usual QL addresses, like the IO and the screen RAM areas, appear at the same places as usual. They also do in the SGC, except that the address of the IO area appears the same as on the QL on the expansion bus, although to the CPU it is very different. The same applies to the QL ROM, too:

designed, the obvious solution was to use the IO area for this screen RAM, much as the TC used it for expansion RAM. Unfortunately, with the Qubide, the upcoming RomDisq and the announced UGC, things have again become very much plugged up. The Aurora uses 240k of the IO area out of 256k available. Only the top 16k of IO area is actually available for use by peripherals. The idea was to set Qubide to operate at this address (bus address FC000h) to keep the ROM slot free again, in SGC + Aurora systems. In fact, The Aurora introduces yet another, small IO area of 15.5k, by additionally decoding the original QL internal IO, in a somewhat desperate attempt to free some address space. Table 8 shows what the Aurora address map looks like (compare it with the original QL).

Table 6: SGC bus address map

00000h...0BFFFh	QL ROM	Read only, remapped
0C000h...0FFFFh	ROM slot	
18000h...1BFFFh	QL internal IO devices	
20000h...27FFFh	QL RAM, screen 0	Write only (WO)
28000h...2FFFFh	QL RAM, screen 1	WO, Appears only if scr1 enabled
C0000h...FFFFFFh	IO area, 256k	

Table 7: SGC internal address map

000000h...00BFFFh	QL ROM image in SGC RAM	write protected during operation
00C000h...00FFFFh	ROM slot	uses 8 bit bus sizeing
010000h...017FFFh	SGC RAM	Toolkit II and SGC software image
018000h...01BFFFh	QL internal IO and SGC IO	uses 8 bit bus sizeing
01C000h...01FFFFh	SGC RAM	
020000h...027FFFh	SGC RAM R/W, QL RAM write	Anything written copied to QL RAM
030000h...1FFFFFFh	SGC RAM R/W, QL RAM write	QL RAM write only if scr1 enabled
200000h...3FFFFFFh	SGC RAM	
400000h...40BFFFh	QL ROM (actual, not image!)	Remapped
40C000h...4BFFFh	Not used	
4C0000h...4FFFFFFh	IO area, 256k	Remapped
500000h...FFFFFFh	Not used	

Since the SGC offers a 256k IO area, things like extended graphics were finally possible. But, users wanted a lot of colour and high resolution, which would mean a lot of screen memory. Already available RAM cannot be used for this, because it's on the SGC - it doesn't have the necessary circuits to display its contents onto the screen! When the Aurora was

Although this has so-far been neglected, the Aurora has the capability to use a larger size EPROM, anything from 64k up to 512k. Access to it is provided by paging any 32k block of the EPROM (up to 16 in the 512k EPROM) into the first 32k of the space used by the original QL ROM. This feature can be utilised on the SGC since the QL ROM space is accessible even

Table 8: Aurora address map

00000h...07FFFh	QL ROM, lower 32k	Bank swiched, see below
08000h...0BFFFh	QL ROM, upper 16k	
0C000h...0FFFFh	extended ROM slot, 16k	Write signal added to ROM slt
10000h...17FFFh	Unused, 32k	Not accessable by GC/SGC
18000h...180FFFh	QL IO (8301 chip) 256 bytes	Aurora control registers included
18100h...1BEFFFh	Extra IO area, 15.5k	Select signal added to ROM slot
1BF00h...1BFFFh	Integrated QIMI, 256 bytes	Partial decode, 2 addresses used
1C000h...1FFFFh	Unused, 16k	Not accessable by GC/SGC
20000h...2FFFFh	Original Screen RAM, 64k	Remaps to extended screen RAM
30000h...3FFFFh	Unused, 64k	Not accessable by GC/SGC
40000h...7FFFFh	Copy of C0000h...FFFFFFh	Not accessable by GC/SGC
80000h...BFFFFh	Copy of 00000h...3FFFFh	Not accessable by GC/SGC
C0000h...FBFFFh	Extended screen RAM, 240k	
FC000h...FFFFFFh	Not used, 16k	Remaining part of IO area

after its copy has been stored into protected SGC RAM and pached. Because of this, once the SGC has started up, the ROM may freely be page-switched because it is not used any more by the system. If anyone wants to use this, details are available.

Another Aurora introduction is the Extra IO area. As such, it is decoded on the Aurora, and a special signal is provided on its extended ROM slot connector, which becomes a logic 0 when any address in this area is accessed. Provided you are not using the ROM slot for something already, a simple piece of hardware could be interfaced to it without too much difficulty, or the need to manage the usual QL signals - a simple address decoder is enough. This area has been chosen because it is not used by anything else, and the GC and SGC can access it. Of course, the GoldFire will be able to do so as well. The idea behind the Extra IO area is to have peripherals use multiples of 256 bytes of it. 256 bytes is usually more than enough to map in the various registers used by a great variety of IO devices. Unlike normal QL peripherals, the idea is not to have the accompanying ROM, which usually takes up 16k. So, in order to use those peripherals, the drivers have to be loaded from disc, or a similar device. Several 256 byte areas are already reserved:

Table 9: Extra IO area

18000h...180FFh	Motherboard IO	Original QL IO and Aurora registers
18100h...187FFh	Reserved	Future peripherals
18400h...1BEFFh	Free for use	
18B00h...18BFFh	On-board QIMI	

The extended ROM slot provides 5 additional signals compared to the standard QL ROM slot. Those are -RESET, -EXTINT, -WR, -DS and -EXIO (Aurora users will find the locations of these and other signals in the Aurora manual). The first four correspond to the similarly named signals on the QL's expansion bus. The signal -EXIO is generated by the Aurora, and becomes a logic 0 whenever an address in the range 18100..1BEFFh is detected, to aid simple IO devices to be interfaced with minimum additional circuitry.

The Aurora also has the decoder disabling input, DSMC, implemented, just like the original QL. Like on the original QL, it is possible to disable any or all of the Aurora address decoding by pulling this signal high when an address of interest appears on the address lines. This address can then be used by external devices.

GoldFire

Just a quick preview of changes which will be brought on by GoldFire. The GF has a pretty complex address map, since it introduces many new features. The existing bus features are expanded to include a 16 Mbyte address range (24 address bits), and selectable access speed. A 32-bit access mode is added, as well as a dual CPU feature which needs to be catered for. See table 10 for the GoldFire address map.

The GF expands on the existing 8-bit QL bus in three ways, of which two are relevant for this article.

Firstly, the GF provides for 24 address lines, extending the range of addresses on the bus to 16 Mbytes. The additional 4 address lines appear instead of signals which have ceased to have a useful meaning with the evolution of QL hardware. The bus lines are FC0, FC1, FC2 and E.

Secondly, it is possible to generate bus addresses in

two ways. The first way is accessing them as the usual QL addresses, meaning the ones which have to appear at certain places because of compatibility. These are the ROM, the on-board IO and the screen 0 and 1. The second way is accessing the multiple 8-bit IO areas which all map to bus addresses 000000..FFFFFFh, but with subtle differences depending on which IO area is accessed.

The 32-bit bus protocol is designed so that old 8-bit peripherals and future 32-bit peripherals can co-exist on the same bus.

Table 10: GoldFire internal address map

00000000h..0000BFFFh	QL ROM image in RAM	write protected
0000C000h..0000FFFFh	ROM slot	Appears on the 8-bit bus
00010000h..00017FFFh	RAM	write protected
00018000h..0001BFFFh	QL motherboard IO	Appears on the 8-bit bus
0001C000h..0001FFFFh	RAM	write protected
00020000h..00027FFFh	RAM R/W, QL bus write	Shadowed Screen 0 area
00028000h..0002FFFFh	RAM R/W, QL bus write	Shadowed Sreen 1 area
00300000h..07FFFFFFh	RAM	Up to 128 Mbytes
08000000h..08FFFFFFh	On-board 512k Flash ROM	Partial decoding
09000000h..0BFFFFFFh	8-bit IO area	Appears as 000000h..FFFFFFh on bus
0C000000h..0CFFFFFFh	On-board IO	Partial decoding
0D000000h..0FFFFFFFh	shadowed 8-bit access	Write only 000000h..FFFFFFh on bus
10000000h..17FFFFFFh	copy of RAM area	Caching disabled for dual CPU mode
18000000h..1FFFFFFFh	32-bit IO area	

Summary

Presented here are two tables. Table 11 is the address map of the two possible QL type motherboards that are now in circulation - The original Sinclair one, and the Aurora:

Table 11

Start	End	Size	Original QL	Aurora
00000h	07FFFh	48k	QL ROM	Paged ROM #
08000h	0BFFFh	16k	QL ROM	QL ROM
0C000h	0FFFFh	16k	Free (ROM slot)	Free (ROM slot)
10000h	17FFFh	32k	Unmapped	Unmapped
18000h	1BFFFh	16k	Motherboard IO	Extra IO area ##
1C000h	1FFFFh	16k	Unmapped	Unmapped
20000h	27FFFh	32k	RAM, Screen 0	RAM, Screen 0
28000h	2FFFFh	32k	RAM, Screen 1	RAM, Screen 1
30000h	3FFFFh	64k	RAM	Unmapped
40000h	BFFFFh	512k	Partial decode *	Partial decode *
C0000h	FBFFFh	240k	Partial decode *	HR Screen RAM
FC000h	FFFFFh	16k	Partial decode *	Unmapped

* - Due to partial decoding on the QL motherboard, the devices present in the 00000..3FFFFh are repeatedly mapped and appear at 40000..7FFFFh, 80000..BFFFFh and C0000..FFFFFh.

- Depending on the programming of Aurora control registers the extended ROM socket on the Aurora can page any 32k of the EPROM in it to this area. Initially the first 32k is paged here to allow correct emulation of the QL ROM.

- This area is further decoded on the Aurora. The usual QL motherboard addresses are provided together with an extension catering for the on-board QIMI interface. The rest is decoded and a pin is provided on the extended ROM slot for easy interfacing to simple IO hardware, see table and explanations below.

The following table summarises the address maps of the various expansion boards. To create a complete bus address map of your QL set-up, concatenate the column with the motherboard you use from the table above, with the column of the expansion you use from the table below, by replacing the 'Undecoded' or 'Read only' or 'Read/Write' entries by the ones from the motherboard map:

Table 12

Start	End	Size	Expanded QL	TC	GC	SGC
00000h	07FFFh	48k	Undecoded	TC ROM initial +	Initial read ++	Initial read ++
08000h	0BFFFh	16k	Undecoded	Undecoded	Initial read ++	Initial read ++
0C000h	0FFFFh	16k	Undecoded	Undecoded	Read only	Read only
10000h	17FFFh	32k	Undecoded	TC ROM and IO	Inaccessible ***	Inaccessible ***
18000h	1BFFFh	16k	Undecoded	Undecoded	Read/Write	Read/Write
1C000h	1FFFFh	16k	Undecoded	Undecoded	Inaccessible ***	Inaccessible ***
20000h	27FFFh	32k	Undecoded	Undecoded	Write only ****	Write only ****
28000h	2FFFFh	32k	Undecoded	Undecoded	Write only ****	Write only when SCR2EN
30000h	3FFFFh	64k	Undecoded	Undecoded	Inaccessible ***	Inaccessible ***
40000h	BFFFFh	512k	RAM	RAM	Inaccessible ***	Inaccessible ***
C0000h	FBFFFh	240k	16k ROM and IO \$	RAM	Inaccessible ***	Read/Write
FC000h	FFFFFh	16k	Undecoded	RAM	Inaccessible ***	Read/Write

\$ - Normally, the floppy disc interface maps itself to C0000..C3FFFh. On an expanded QL the rest of the IO area, C4000..FFFFFh is undecoded and in fact usually a copy of 04000..3FFFFh can be found there.

+ - The TC has its on-board ROM mapped in a non-standard place which the OS will not automatically initialise. Because of this it is also initially mapped at 00000..07FFF, where the CPU will start execution on reset. The code forces the OS to recognise the TC ROM at its normal place and removes the initial mapping of the TC ROM.

++ - The GC and SGC initialise from their on-board ROMs on reset, copying the QL ROMs to on-board RAM for faster execution. Therefore the usual ROM area at 00000..0BFFFh is only read, and once execution is resumed from the copy, that address block becomes inaccessible on the GC. It can be still accessed if a SGC is in use. On a SGC the bus addresses 00000..0FFFFh can still be accessed as memory addresses 400000..40FFFF

*** - The GC and SGC never generate these addresses on the bus.

**** - The shadowing used by the GC and SGC generates writes to these addresses expecting a write to screen memory to be the result. A write to the same address in the GC/SGC on-board RAM is also performed. Reading is always done from GC/SGC RAM as it is several times faster. Read accesses are never generated on the bus at these addresses. Also, writes to the Screen 1 area (28000..2FFFFh) are not generated on the SGC if the second screen is disabled by the SCR2DIS command.

Next time: How to use the expansion bus, Rom slot and Aurora enhancements in practice, with examples.

DBEasy v1.6 - A Review

by Al Feng

Some programs are so well designed that it sometimes takes years for the user to appreciate the many features which the program author has thoughtfully incorporated into them. DBEasy [Wood and Wind Computing, RR3 Box 92, Cornish, NH 03745] is certainly a well designed program that has rightfully been well received. With the program undergoing an

extensive re-writing, it deserves a renewed look.

For those not familiar with DBEasy, it is a front end program for the Archive [Psion] database program which is available either for QDOS/SMSQ (v1.6) or for DOS (v1.3). While there were minor feature differences between the two versions in the past, they now appear to be almost identical in function and operation.

Those not familiar with DBEasy may think that it is simply a menu program for loading and viewing databases. It is more. It is a program for creation and management of any database you may wish to

create, use, and maintain (for example, altering/updating fields and maintaining by adding/deleting records).

General Database Design Philosophy

From what I have gleaned over the years, there are two types of databases - hierarchical and relational. While the hierarchical database came first, most modern databases are now relational. Archive is a relational database with an integrated, procedure based programming language.

Hierarchical databases usually seem "better" to the novice because they are designed with a non-discriminating, "flat" field structure. Essentially each field (or screen) is a record, and conversely each record is a field, thus allowing any piece of data to be put anywhere in a given record. In essence, a hierarchical database record can be thought of as one in a set of electronic file cards. Thus, to use a hierarchical database, you just name (create) the database file, open it, and then start inputting whatever you desire into any record.

In theory, you could simulate a hierarchical database by creating a single field database within a relational database. In Archive, you would:

```
CREATE "flat" <ENTER>
flat$ <ENTER>
<ENTER>
```

After this, you would use the SEDIT (i.e. screen edit) function to create the screen with a single field essentially taking up as much of the usable screen as you wanted; and then, SSAVE (i.e., screen save) that screen. You would still OPEN, SLOAD (i.e., screen load), ALTER, and CLOSE the database as you would a normal Archive database. If you were inspired, you could EDIT a program to expedite navigating your "flat" field database.

Because a hierarchical database does not have fields, per se, it is not possible for the user to easily, or consistently manipulate the data of individual records. For example, it would not be easy to use a "flat" field database for printing names and addresses on labels unless the entire contents of each record was comprised of consistently placed names and addresses.

My understanding is that a simple FIND query is not handicapped by searching a hierarchical database.

The HYPERCARD database (APPLE Macintosh) appears to be a hierarchical database which has user linkable elements. The selectively "programmable" links and inherent graphical nature of the database provide an extended versatility not available with a simple, hierarchical database.

Relational databases differ in that the individual fields (text or numeric) must be pre-defined BEFORE data can be input. The need to pre-define the fields

can be particularly discouraging for the novice; and, the same novice may wonder why or how relational databases have become so prevalent.

In a relational database, the OUTPUT of various, individual fields can be "manipulated" so that it suits multiple needs; thus, one set of data can be used and output in more than one way with differing fields included-or-excluded in a particular output. The unqualified advantage of a relational database becomes most apparent when you have the opportunity to use a front-end program such as DBEasy. As designed, DBEasy allows you to print:

- 1) the individual record;
- 2) an address label (if applicable);
- 3) a line of user-specified fields; and/or,
- 4) a "custom" output (a future article).

The output can be to your printer, the screen, or a file (for import to QUILL, for example) when the request is initiated from the MULTIPLE MENU screen and to the printer only from the SINGLE MENU screen.

The Design

DBEasy uses a MAIN MENU which allows the user to branch to ten other functions:

Single menu	w/ find, search, insert, alter, delete
Multiple menu	w/ select, order, print, copy, export
Log menu	w/ add, change, remove, import, capture
Backup	backup the current database
sWitch	switch the active database
Help	access the DBEasy help system
Directory	w/ copy, delete options
Configuration	set video or date style
Rescreen	check database / redraw screen
Quit DBEASY	all done

In earlier versions you would navigate this screen by pressing a number key (1 to 9) to put the pointer arrow on a given line, or by using the up-or-down cursor key to move the pointer up or down from line to line. '0' would provide a fast branching to the exit screen.

With the most current version, the HOT KEYS developed in QLERK are now available for use in DBEasy - generally, pressing the first letter of the KEY word will move the pointer arrow to the line or option; or, activate the options within the secondary menus. Each version change over the years has made DBEasy more convenient to use; and, I would guess that it only took about five minutes to become comfortable with the new HOT KEY feature.

A welcome change is the increased facility to quick "sWitch" the active database from one to another. A "sWitch" can also be executed from the SINGLE and MULTIPLE MENU as well as from the LOG MENU (the LOG MENU was previously named "Switch mode"). When using the "sWitch" routine, you do not have to type in the entire database name, only enough

characters to distinguish one database from another. Thus, when I want to access my 'PLATYPUS.dbf' I do need to type in at least 'PLAT' since typing in 'P' or 'PL' or 'PLA' will always locate the 'PLANNERX.dbf' first. If I were to remove the 'PLANNERX.dbf' from the drive, then simply typing 'P' would result in DBEasy querying me whether or not 'PLATYPUS.dbf' was the one I wanted.

In the past, if you wanted to change the date format at the top of the MAIN MENU screen, you needed to edit and re-save the DBEMM code. This is now an option which can be done from the main menu. There are now three formats for displaying the date on the MAIN MENU [YYYY/MM/DD, MM/DD/YYYY, & DD/MM/YYYY].

The HELP.dbf file is now available as a HOT KEY from almost any point in the program by simply pressing 'H' <ENTER>

The Design in Practice

All the databases which are CREATED have the same fields - twelve text fields (date_\$, key_\$, s1_\$, et cetera) and six numeric fields (n1_, n2_, et cetera). While I originally thought this would be limiting, I have found that this is a good compromise for most databases.

In day-to-day use, one of the apparent benefits of using DBEasy is that it eliminates the need for pre-defining the fields of each database. If you don't like the labels, you can change them later.

The 'date_#' field is automatically filled when you INSERT a new database; but, any other text data can be inserted therein. The 'date_#' was automatically updated to the current date in prior versions, but this has been changed with the current version whereby the current version maintains the record's 'original' date - as I generally did NOT want this field changed in the past, I did find the automatic updating of the field disconcerting; so, I'm glad that the newest version of the program keeps this unchanged.

If necessary, the actual DBEasy program (i.e., DBEMM_pro) can be modified to accommodate more or less fields; or, as demonstrated with the newly incorporated, ADDRESSX.dbf, the specific screen can be customized (using SEDIT & SSAVE) to accommodate different field_lengths.

Of course, the labels assigned to the fields can be changed at any time; and, a new option lets the user create new databases by "capturing" data from existing DBEasy database files.

When you choose the MULTIPLE MENU you can view up to ten LINES of truncated RECORD data per screen page. In the past, the LINE represented three user selectable FIELDS of data. This has been greatly improved with the current version being capable of displaying at least seven fields of variable length. Each LINE format can be exclusive to a particular database.

The "Capture" Pitfall?

There is a very small inconvenience that you have to suffer if you are upgrading from a previous version - the DBELOG.dbf contains different data than in previous versions. Presumably, it is the difference in structure which now allows the user to 'capture' data from already logged databases when creating new database files.

It seemed that the only files that did transfer were my VARIOUS.dbf files and the DBEASY.scn that I use!

However, after you get over the shock of not being able to simply the transfer various ancillary files along with your VARIOUS.dbf files you will quickly appreciate the 'capture' function; and, the setback is minor compared with the gain.

Customization and Cosmetic Modifications

You may not have to make any changes to the program to use DBEasy; but, if you do, most customization is now easier, or faster, to implement. Of course, the single RECORD screen can simply be copied from your previous DBEasy program disk (if it is different from what is provided) - for example, the colors of my single RECORD screen complement the MAIN MENU colors I had chosen to suit my personal sensibilities.

Okay, as good as DBEasy is, it probably isn't 'perfect' for you as it comes out-of-the-box any more than it was for me.

Based on the earlier DBEasy versions, I knew that I would have to 'go inside' and modify the print option to add a few 'lprint' statements to adjust the printer's vertical spacing for the labels which I currently use (4' x 1 7/16"). Of course, the first thing I had to do was to (esc)ape to Archive's command_line and enter the edit mode and then edit the proc(s) [consult the QL USER GUIDE if you don't know how to do this].

```
proc easy_out_lbl

if ns: print at ml,12;rs:: endif : let
i$=s1_#: if s2_<>nul$: let i$=s2_#+sp#+i$
endif

lprint : lprint i$: lprint s3_#: if
s4_<>nul$: lprint s4_#: ndif : lprint s5_#,"
";s6_#;" ";s7_#: lprint : lprint : lprint :
lprint tab 15;" ": if s4_#=nul$: lprint :
endif : let line=6

endproc
```

In addition to adding 'lprint' statements, I also omitted the statement which printed field 's9_#' which is used for 'country' since I generally use 's9_#' for other comments.

All of the remaining changes described here are cosmetic to suit my own sensibilities and do not enhance or improve the operation of the program.

The least significant change I made was to the QUIT page. This was the easiest to implement since it essentially involved adding a long enough string of periods ('.....') to tidy up the appearance.

```
proc quitter
```

```
heady;"Quit DBEasy":yorn;0, " Quit DBEasy
program .....": if ans$="n":start: stop :
endif :yorn;0, " Quit ARCHIVE, also .....":
if ans$="n": new : else : quit : endif
```

```
endproc
```

After that, there were some other cosmetic changes that I made to the MAIN MENU and LOG MENU because I liked the 'look' of the previous versions (at least, the way I had them) better than the current appearance.

The changes to the MAIN MENU were primarily done to the header related procs.

```
proc arrow_mess;l,m$
```

```
let i=(80-(36+len(m$)))/2
```

```
paper sink
```

```
ink spap
```

```
print at 1,1;rv$; tab i; "Use arrow keys or
Hot Key to choose "+m$; tab 79; print at
1+1,1; tab 25; "<ENTER> to select or 0 to
exit"; tab 79;rv$;
```

```
paper spap
```

```
ink sink
```

```
endproc
```

Changing the 'arrow_mess' tidies up the MAIN MENU box by coloring the related lines to the same color as the outer field. Modifying the 'heading' was mostly rearranging the text and the 'heady' made the banner the full width of the screen, as on earlier versions.

```
proc heading
```

```
heady;nul$: paper hpap: ink hink:runtime: let
j=ans: timey
```

```
print at 1,2;upper(sys$);" MAIN MENU"; ink
sink; at 0,36;today$; tab 54;"Day ";day;" of
"; val(date(0)); ink hink; at 1,37;ans$; tab
54;j;" minutes since start"
```

```
paper spap: ink sink: if begent=-1: let
i$="-": else : let i$=str(begcnt,2,zer): endif
```

```
print at 4,2; ink spap;desc$;
```

```
print at 4,52;"Module : DBEmm"; at 5,52;
```

```
"Device : ";pd$; at 6,52;"Database : "; sf$
```

```
print at 7,52;"Records : ";i$
```

```
endproc
```

```
proc heady;i$
```

```
paper spap: cls : paper hpap: ink hink: print
at 0,0; tab 80; at 1,0; tab 80; at 2,0; tab
80; at 1,2;i$;; paper spap: ink sink: print at
4,0;
```

```
endproc
```

Still in a 'retro' mode, changes to the LOG MENU were next. The changes to the LOG MENU were done to, you guessed it, log related procs; and again, the changes were made to resemble the appearance of earlier versions.

The simple part was to remove the statement which indicated the 'active database' in the 'log_menu' procedure by truncating the fifth line.

```
proc log_menu
```

```
....
print at 17,3;"of ";npg; tab 79; at 18,1;rv$;
tab 79;
```

```
....
endproc
```

The real trick was to highlight the active line to echo the previous versions. Of course, the hardest part was to actually find which procedure performed this function! Not surprisingly, it is in the 'log_show' procedure. The asterik (*) has been replaced with the greater_than sign (>), but can be eliminated since it is now superfluous for most video configurations.

```
proc log_show;pg
```

```
let k=10*(pg-1):position;"1",r1,k: let j=5:
print at j,zer::usel
while not eof("1") and r1<k+10:usel
```

```
let i$=desc_$: if len(i$)>33: let i$=i$(1 to
33): endif : let j$=user_$: if len(j$)>10: let
j$=j$(1 to 10): endif : if act_: paper sink:
ink spap: let k$=" >": else : paper spap: ink
sink: let k$=nul$: endif
```

```
print sp$;r1;k$; tab 4;sp$;sd_$. t ab
5+dl;sp$;sf_$. tab 23;sp$;i$; tab 58;date_$.
tab 68;j$
```

```
tab 80: ink sink: paper spap: let
j=j+1:plnext: endwhile
box;j,zer,14,80,nul$: let j=16
```

```
paper spap: ink sink
```

```
print at j,6;pg;" "
```

```
endproc
```

After you make each change, you will want to see if it either looks right, or "works" the way you want it to. Of course, you will also want to save it to a working copy of the program by inputting:

```
save object "DBEmm" <ENTER>
```

I made the same changes for both the QDOS and the PC version of the program. In each case, you will want to compare and insert/change lines to match what you see listed. After you have made changes, you can further modify the procedures to fit your sensibilities instead of mine.

GUI - Phooey!

There are alternatives to Archive (more so for the DOS world).

The essential thing to know is that DBEasy is NOT handicapped by being tied to a text-based program. Now, I realize that a LOT of people LIKE using a mouse - you know, trying to move the pointer to a viable spot on the screen as if it were a game; but, I'm not one of them.

In day-to-day use, you only need the cursor keys and ENTER key, or a mouse, to navigate the program.

Generally, the only time you need to use the alphanumeric keys is when creating/deleting a new/old database, or when making changes within an existing database.

If there are some situations when you need a secure environment, then a copy of DBEasy running on a QL without a keyboard (of course, with a pointer input device) would probably be more secure from tampering than a GUI database which would probably remain vulnerable to change or removal.

The bottom Line

If you aren't using Archive, yet, then you should consider getting a copy of DBEasy since it will make using a D(ata)B(ase)Easy.

Because I come from a long line of read-the-instructions-last people there are certainly features and functions which I have failed to discuss adequately (or, at all) because I probably have not used them, yet. For those who like to read, there is an extensive DBEASY_doc which is obviously a good reference tool for using DBEasy, too.

The QL disk contains an XCHANGE compatible version. Other than to say that the XCHANGE version is smaller than the "normal" version, I cannot indicate what the difference is. The "normal" QL and PC versions are essentially the same.

An additional "_dbf" file is now included (PLANNERX_dbf) which can be used as a daily planner.

Unlike PC users who currently do not have access to Archive, DBEasy is reason enough to use Archive, and would be reason enough for database users to use a QL or QXL as their hardware platform.

Before I started to use DBEasy, I would use the back pages of my check book register for names and addresses! At the time, it just seemed to me that with less than 50 names/addresses/phone_numbers of various sorts that it was far easier transcribing those names (as needed) than to learn to use Archive. Were it not for DBEasy, I am certain that I would still not be using the Archive program, or any database program for that matter.

If you are using a previous version of DBEasy, I am sure that you will not regret upgrading to the newest version. The HOT KEYS, alone, make it worth the effort to transfer files and make whatever changes need to be made.

New users need to know that DBEasy is both EASY to use and EASY to modify if need be; that, "sample" databases are included that can be amended with your own data; and, that your QL requires memory expansion beyond 128K to run DBEasy.

The DBEasy program costs \$24.00 (US); upgrades cost \$7.00 (US).



For your Eyes Only!

Jérôme Grimbart

Subtitle "or Time for 16 colours, NOW!" - and the author gives you some very interesting aspects and suggestions on how to use more than eight colours with existing hardware, or more precisely, how to refer to them in our current system of "naming" colours 0 to 7.

As everybody knows, the QL has two basic screen display: either 512x256 in four colours, or 256x256 in eight colours with some flashing capabilities.

Then came the Atari emulator, only on mode 4 but with a bigger resolution. And now we have many other QL compatible system, usually providing the four colour

and also the eight colour mode, but without the flashing, because it would have required specific hardware chips. Thus, on every platform, except the original QL one, in mode 8, there is a waste of one bit for every pixel. As usually the hardware on these platform is not limited to a True/False logic for the RGB lines, this means we can easily have 16 colours instead of 8.

In fact, the problem is (or was) not to have 8 new colours, but to choose which one. The original

colours of the QL were arranged, from 0 to 7, in order of increasing luminosity (or perceived luminosity). Hence, when adding new colours we should have tried to keep these properties. But, as the programming interface does not allow us to go any further than the original range, due to the fact that stipple and complex combinations, to remain compatible, must be specified in only 8 bits, with 3 bits for the colours.

That means that the 8 new colours won't be numbered!

- Uh? How do you use them if they have no number???

Well, in the current systems, they won't be. Maybe later, but you don't have to rewrite the whole system just to see 8 new colours. As stated earlier, the problem is not to have them (on the standard 32K of screen memory, we already have the needed bits), the problem is to choose them.

Actually, I have seen three different systems QL-compatible displaying more than eight colours (true colours, not counting an old hack for QL based hardware

which switched the screen displayed between the two memory areas to create the illusions of more colours...). For the curious, they were:

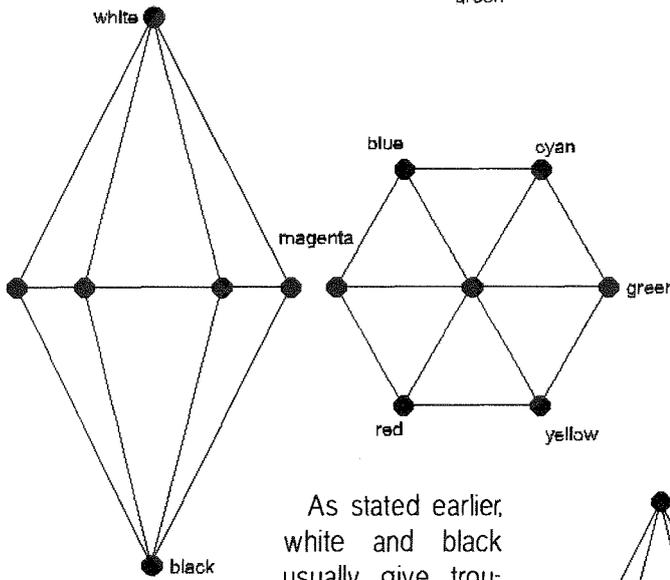
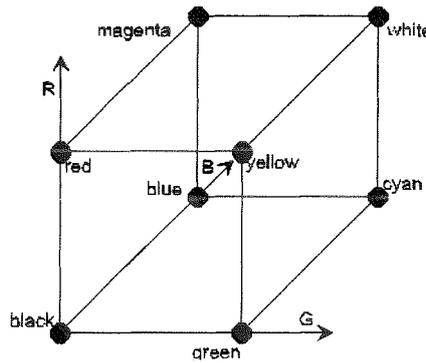
- the demonstration of the 256 colours with the prototype of Aurora, at some Show

- uqlx, modified by myself

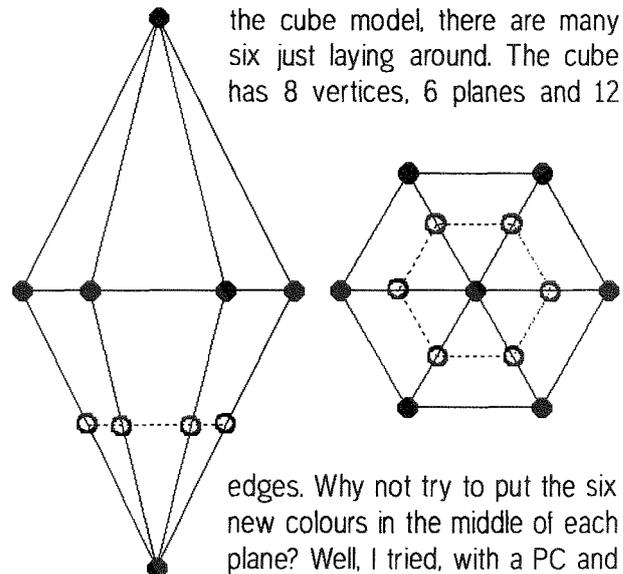
- QPC, with a specific program also written by myself

Ok, back to the choice of colours: the problem with 16 colours is that you start with a

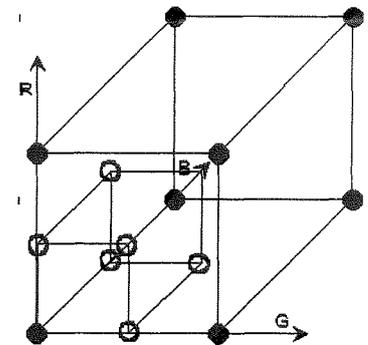
base of 8 colours and should try to have some kind of logic to produce the 8 new ones. Alas the black and white colours usually mess things up. If you darken all colours then what is darker than black? If you lighten all the colours, what is brighter than white? If you try to represent somehow the colour in the space, you usually end up with a 3D model, either the RGB cube, or the Hue/Saturation/Value double pyramid. Look at the location of the 8 basic colours of the QL on these models.



As stated earlier, white and black usually give trouble when trying to keep a logic description of the variation for the new colours. And what we usually want are two different shades of grey. So this left us with six colours to specify. Let's see how the standard QPC (but it's easy to change



that with Thierry Godefroy's extensions, thanks Thierry) behaves: it has two shades of grey (we already agree on that...) and then darken every other colours. Well, let's face it, I don't like it too much. If you take a look at the cube or the double pyramide, the new colours are concentrated on a tiny part of the model. I do not ask (yet) for photo-realism, but if we are gonna have 16 colours, I would rather have a wider spectrum. By just darkening the original one, we do not add any new tincture, we just replace a stipple by a full colour. We augment the resolution, but not the number of perceived colours.



The colour codes are (full 24 bit, VGA should divide by 4 for direct programming): 00007F, 7F0000, 7F007F, 007F00, 007F7F, 7F7F00.

Of course we could choose to lighten the six colours, but with this we would have the same trouble (just a symmetry in either model). But, if we look carefully in the cube model, there are many six just laying around. The cube has 8 vertices, 6 planes and 12

edges. Why not try to put the six new colours in the middle of each plane? Well, I tried, with a PC and

JOSIMM MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
 ☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)
<http://www.j-m-s.com/smsq/> email: smsq@j-m-s.com

News!

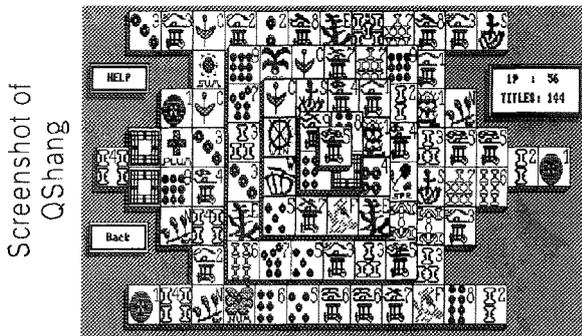
The RomDisq! Fully functional! See also previous issue of QL Today! Extremely small board for the QL's ROM-Port, works like a permanent RAM-disk, even if the power is taken away contents are not lost!

RomDisq 2MB	DM 129,00
RomDisq 4MB	DM 199,00
RomDisq 8MB	DM 299,00
Aurora Adaptor for RomDisq	DM 9,90

Upgrades!

Two games Upgrades: **Diamonds** and **QShang**. They are playable on all systems (including Minerva and SMSQ/E) and work in higher screen resolutions too. Other advantages are, for example, the ability to configure the file in which Qshang saves the highscores..

Diamonds^s costs new **DM 36,-**
 or as an **Upgrade DM 16,-**
QShang^s costs new **DM 36,-**
 or as an **Upgrade DM 16,-**



Screenshot of QShang

A new, complete and detailed catalogue is in preparation!

Special Offers!

The colour drivers are coming. In order to be able to upgrade to the colour drivers, you need to have SMSQ/E. The "normal" SMSQ which is shipped with the QXL card will not be able to handle more colours. To benefit from the colours in the future, and benefit NOW from the extra features of SMSQ/E, here a special offer:

SMSQ/E for the QXL
DM 169,-
 only

As Aurora owners will be able to use more colours when the colour drivers are available, another offer for SMSQ/E for the Aurora plus GoldCard/SuperGoldCard:

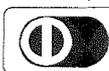
SMSQ/E for the (Super)GoldCard
DM 189,-
 only

Offer now ends 30th of August 1998!!!

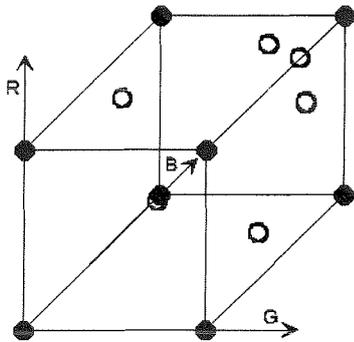
Verbatim 3,5" DD Disk 720k
Each only -,69 DM!
Noname 3,5" DD Disk 720k
Each only -,49 DM!
Various colours!!!

TERMS OF PAYMENT

Postage and package [Germany] DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). [Europe] DM 14,50 (if total value of goods is up to DM 50,- then only DM 9,50). [Overseas] between DM 14,50 (1 item) and DM 35,- (maximum). All prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.



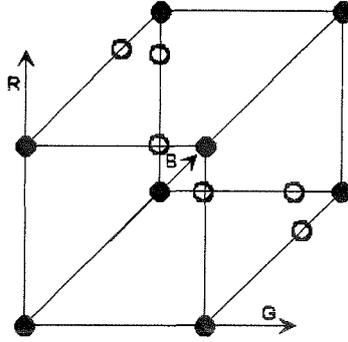
an HTML browser (easy way to make a colourful page, as HTML allows us to specify the colour with a #RRGGBB format). Maybe it was my screen or the graphic cards, but right in the middle value is not enough to completely discern the new colour from the old one. It seems that 3/4 of the full range works better.



The colour codes are (still 24 bit, VGA divide by 4 each bytes): BFBFFF, FFBFBF, BF00BF, BFFFBF, 00BFBF, BFBF00.

I also tried to move the three top colours near the white back to the other colours, but with a value of 1/4 instead of 3/4, they looked the same as their nearest colours. Using the model of the cube plane for the new colours, I get a clear blue, a light pinky, a dark violet, a very light green, the 'standard' dark cyan, and a dirty yellow. May I say I do not like them too much... may be because of the dirty yellow and the very light green. But there is another 6 lying in the cube model: we have six edges connecting the six vertices of the normal colours, so what about setting the colour on the edge? One more time, half values didn't display well, but 3/4 was just perfect (I also tried 2/3, not good enough). I get a light blue (darker than cyan), a pink, a violet (dark magenta), a green going yellow (yellowish? like the young grass, well sort of), a green going cyan (or a cyan going green? well, in french the word would be 'glauque' but nobody will define it as a colour, due to a cultural misinterpretation, it's a bluish green, as can be seen on the sea

near the beach of paradise island, usually in the lagoon...), and a gold-yellow. Probably due to the fact that you can always group two new colours with two old ones and still have a smooth gradation, hence allowing nice shading effect on things like buttons, this model for the time being has my preference.



The colour codes are (always in full 24 bits): 00BFFF, FF00BF, BF00FF, BFFF00, 00FFBF, FFBF00.

I would gladly try any other repartition, so if you think that you have a better idea, let's share it with everyone! I think it's time for at least the 16 colours!

Ok, so, now, how can you use the new colours, if you do not number them... Well, that's the tricky part. You can, of course, assign them with a number, but as long as the QDOS call will only

allow a plain colour in the range 0-7, you won't be able to use them as the standard ones. But just remember that we are using the flash bit on the screen map. So, all you have to do is set this flash bit when drawing your pixel. There are two ways to do this: either you directly write to the screen map (Bad, very bad, where is the screen map? what is the current resolution ... hey, is the application currently displayed... well you go back to the very hard days without the PE), or you use the PE! Yes, it's astonishing, but the PE fully supports the 16 colours. Well in fact the basis of the PE is to use a direct encoding of the value to put through a mask on the screen. All you have to do is to raise the flash bit in your sprite/pattern (and also using a large mask, so as to let the flash bit go through). And to request mode 8, as I do not know of a lot of applications that do not use mode 4 (well, it's understandable on a limited display of 512x256(4), but on a 800x600(4) this still allows mode 8 with a great 400x600 resolution!)

Email: jgrimbert@atos-group.com



Geoff Wicks Replies

Opinions, even controversial opinions, are permitted in reviews, but facts are sacred. Your review of SOLVIT-PLUS 3 was marred by several factual inaccuracies.

"The biggest failing I saw was the lack of any update to the manual".

Not true. An upgrade sheet is provided with the program, and this sheet was enclosed with the review copy I sent to QL-Today.

With upgrades I never produce a new manual, but a two page upgrade sheet reduced to fit on one side of A4. As any software producer will tell you, the manual is the highest single direct cost of a software product. Each copy of the Solvit-Plus manual costs about

half my upgrade price to produce. It is also a capital cost. Disks can be made up on demand, but the manual cannot. I have to estimate the possible demand for a product, and order the requisite number of manuals. If I produce a new manual with every upgrade, then the cost of any unused old manuals would have to be charged to the upgrade.

In practice this would mean that an upgrade would be double or even treble its present price. Do

we really want to return to the days when upgrades were sold as "Special Editions" with a 25% to 50% price increase?

SOLVIT-PLUS 3 "does not allow you to import a text file as a dictionary".

Not true. An ASCII word list can be imported into SOLVIT-PLUS 3. In fact this part of the program was extensively rewritten to provide additional error trapping and more on-screen progress messages.

Loading and importing dictionaries, which were two separate commands in SOLVIT-PLUS 2, are combined into one command "WORD LIST" in SOLVIT-PLUS 3. This is not only documented in the upgrade sheet, but also mentioned on a help screen that your reviewer has clearly not read.

"No new features were added".

Not true. If full mouse compatibility, repositioning on high resolution screens and backwards movement through word lists are not new features, what is? In addition there are user configurable screen colours, a simpler user interface and the optional use of a forward slash in wild card searches, a feature many users have requested.

"There are a few items that need to be done to get Solvit-Plus 3 running."

Your reviewer then goes on to describe a complicated installation procedure. This is a fair criticism of the program, and I accept that insufficient help is provided on this point. The reviewer correctly identifies part of the problem as being due to the program and default files being on disk 2 and the word list on disk 1. He is also correct in his assumption that you could make a bootable disk by copying these three files to one disk. The procedure for installing on hard disk is:

1: Copy files Solvit3_obj and

Solvit3_def on disk 2 and English_dic on disk 1 to your hard disk directory. Let us assume this is 'win1_solvit_.'

2: With disk 1 in your disk drive start the program via the command

EX win1_solvit_solvit3_obj;
'win1_solvit_'

3: Press L for WORD LIST, and reload the dictionary from your hard disk directory.

4: Press P for Parameters, make any changes you wish, and then press D for DEVICE to save the new values.

5: Use the QJUMP configuration program to change the default drive to 'win1_solvit_.'

"To keep development time down, it looks as if Geoff Wicks took the core part of Solvit-Plus 2 and moved that part to the Pointer Environment version".

This is fair comment within the rules of reviews, but the reviewer is wrong in his assumption.

The design of SOLVIT-PLUS 3 was the result of extensive market research, which is a pretentious way of saying I talked to users of SOLVIT-PLUS 2 for three years. Two themes came out of these discussions. Firstly most users are using the program only for wild-card searches, and secondly there is a small group of purchasers who were overwhelmed by the number of commands. There was obviously a need for a simpler front end.

Another factor affecting the design of SOLVIT-PLUS 3 were the demands of the pointer environment. For ease and speed of use, particularly with a mouse, pointer programs are most effective if the menu items are displayed on screen alongside the application window. There was no room for all the SOLVIT-PLUS 2 commands and sub-menus would have been necessary. Each sub-menu would add to the complexity of the program, and its memory requirements. The screen and

menu routines already form over a quarter of SOLVIT-PLUS 3's memory requirements.

I therefore took a decision to cut down the program to make it simpler and faster to use. Three features, Backwards, Palindrome and Figures are "once only" routines for any one word list, and would remain available in SOLVIT-PLUS 2. The spell checking routine was never intended to rival the dedicated QL spell checkers, and there was no evidence of widespread use. In addition a word list suitable for spell checking, especially in some non-English languages, has some differences from a word list for word puzzles.

In short SOLVIT-PLUS 3 was written more for existing users than new users. It modernised the program to make use of some of the hardware developments of the last few years. SOLVIT-PLUS is over three years old, is a specialist program, and although it still sells slowly but surely, I am not expecting many new users.

If your reviewer is feeling a little bruised by this rebuttal, let me finish by saying that I agree with most of his last paragraph:

"If you are a current user of Solvit-Plus and use the Pointer Environment, I would recommend getting version 3. Using the same consistent PE interface speeds up the program. If you do not use the PE then you are not really missing out on the new features."

Tim Swenson responds to Geoff Wicks' comments regarding the Solvit Plus 3 review:

>"The biggest failing I saw was the lack of any update to the manual".

>Not true.

There is a good possibility that the Update sheet got lost when the package was sent to QL Today and then to me. Since Geoff says there is a sheet, my comments can pretty well be considered null and void.

>SOLVIT-PLUS 3 "does not allow you to import a text file as a dictionary".

>Not true

I must have missed this one. I think having the Update sheet would have helped, as Geoff mentioned.

>"No new features were added".

>Not true.

I meant this to refer to the general use of the program and not the inclusion of the PE and user interface.

>"To keep development time down, it looks as if Geoff Wicks took the core part of Solvit-Plus 2 and moved that part to the Pointer Environment version".

>This is fair comment within the rules of reviews, but the reviewer is wrong in his assumption.

My assumption was based only on what I saw and a whole lot of guessing. It can be hard to guess the WHY only seeing the WHAT.

Tim Swenson



Connect 'A' to 'A', 'B' to 'B', and so on.

I used a solder-type D-connector and eleven short (about 6" long) pieces of multi-strand wire.

If you are simply making an adapter then you may note that using IDC (ribbon-type) plugs should be much faster - only the ground ('K') connection needs to be "off-set" with a jumper.

If you are hard-wiring, look at the circuit board carefully, and then decide where you want to make your connections - I chose to hard-wire directly to "pins" on the circuit board - visually tracing back from the CENTRONICS plug to the first available pin. A hard-wire implementation will minimize components.

I twisted the wire together (loosely) and then wrapped the cluster (tightly) with electrical tape.

Finally, I used my soldering iron to melt-away a small "cut-out" in the back edge of the base of the interface's black plastic box so that my short pig-tail adapter could exit the opposite end of the CENTRONICS plug. Take your time (test fit often), and don't make the hole too large.

Since my QL's components are nestled inside a salvaged PC clone case, my old style, MIRACLE parallel interface can now rest near the back of the case (on the inside) out of harm's way with the 25-pin connector firmly attached to the rear of the case using an available opening intended for this type of connection.

In addition, I can now employ ANY LENGTH of any "standard"

Parallel Printer Interface Plug Adaptation

Al Feng

If you ever had mechanical problems with the old Miracle Centronics Printer Interface, then you'll find THE solution in this article!

Many of you already know that MIRACLE SYSTEMS has redesigned their parallel printer interface so that its circuitry now fits safely within the confines of what would appear - from a distance - to be a standard, metal CENTRONICS plug.

This is good news because if your experience with their old style parallel interface has been similar to mine, you may have had the misfortune of breaking the blue plastic, CENTRONICS plug away from both the black box and circuit board!

Okay, the misfortune means one of a few things:

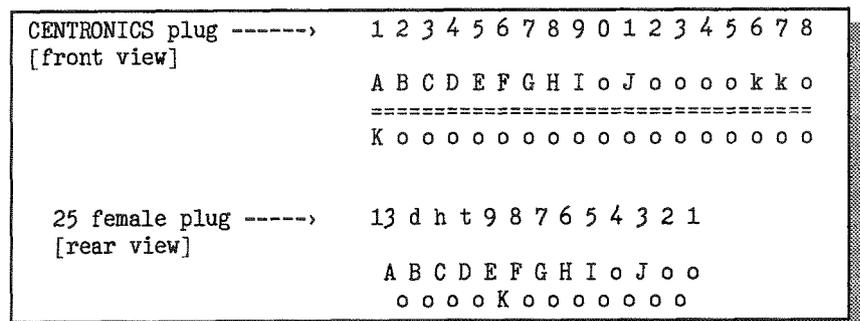
- 1) I am clumsy; AND,
- 2) my printer is NOT well situated; and/or,
- 3) the design is unintentionally fragile.

I finally decided - after "breaking" my parallel interface for the umpteenth time - that it was time to find my ohm-meter, warm up my 15 watt soldering iron, and make a short adapter cable that would allow me to employ a "standard" IBM parallel printer cable.

After opening the interface box and looking at the short jumpers I had previously employed to RE-connect the CENTRONICS plug to the circuitry, it occurred to me that I might as well make a hard wire connection directly between the circuit board and a 25-pin D-sub (female) plug [NOTE: If you are not confident of your soldering skills in tight areas then should consider making an adapter cable].

You only need to make eleven (11) connections to the D-25 plug since there are only that many signals being used by the MIRACLE parallel interface.

The main thing to note is that the presentation of the CENTRONICS plug is the "front view" and D-25 is the "rear view" of the plug.



Im stillen Winkel 12 • 47169 Duisburg • Germany
 ☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)
<http://www.j-m-s.com/smsq/> email: smsq@j-m-s.com

Software, Games, Applications & QL Spares

QL Games

BlackKnight Chess	DM 119,90
Pipes	DM 29,90
BrainSmasher	DM 39,90
Arcanoid	DM 39,90
Firebirds	DM 39,90
QShang	DM 39,90
Diamonds	DM 39,90
The Oracle	DM 39,90
MineField	DM 39,90
Double Block	DM 39,90
The Lonely Joker 2	DM 59,00
SuperGamesPack	DM 90,00

QL Spares

ZX8301	DM 19,90
ZX8302	DM 9,90
Keyboard membrane	DM 25,00

An ideal time to get things NOW - the £ is worth DM 3,- (!!!) at the time I write this!

QL Applications

QD Editor	[V9.15]	DM 125,00
QD Upgrade from V8		DM 24,90
FiFi II File Finder	[V4.16]	DM 49,90
FiFi II Upgrade from previous Version		DM 19,90
QMAKE	[V4.21]	DM 44,90
QLiberator SuperBASIC Compiler		DM 139,00
QLoad-Ref		DM 49,90
QLQ	[V1.13]	DM 69,90
QMAC Macro Assembler	[V1.01]	DM 69,00
QMENU	[V7.04]	DM 41,90
QMENU Upgrade		DM 16,90
QPAC 1	[V1.06]	DM 61,50
QPAC 2	[V1.39]	DM 119,00
QTYP 2 Spell-Checker	[V2.17]	DM 82,50
QPTR Pointer Toolkit	[V0.30]	DM 89,90
QSpread Spreadsheet	[V1.43]	DM 169,00
QSpread Update from V1.34 or before		DM 16,00
QSpread Update from V1.35-1.41		free
QSUP	[V3.09]	DM 79,90
EPROM Manager	[V3.02]	DM 61,50
WINED	[V1.21]	DM 49,90
I/O 2 Toolkit	[V2.16]	DM 99,00
BASIC Linker	[V1.12]	DM 49,90

Applications

LDUMP	[V1.05]	DM 65,00
DISA Interactive Disass.	[V3.02]	DM 95,00
DISA V3 - Upgrade from V1 or V2		DM 35,00
EasyPTR Part 1 + 2		DM 99,00
EasyPTR Part 3		DM 49,00
Stylus-Driver for text87 and text91		DM 69,00
HyperHelp for BASIC		DM 44,90
DiskMate 5		DM 69,00
CueShell		DM 95,00
SERMouse Driver		DM 29,00
QDOS/SMS Reference Manual		DM 84,90
Update sheets from March 1997		DM 13,00

ProWesS + Applications

(all ProWesS Applications require ProWesS which is not included!)

ProWesS WindowManager+HTML Reader	DM 129,00
DataDesign Database	DM 79,00
LineDesign Vektor/DesktopPublishing	DM 79,00
fsearch	DM 49,00
Pflist	DM 49,00
fontutils	DM 79,00
PWfile	DM 64,00

Ink for EPSON-Printers

Alternative supplier, tested "extremely good"!

Ink for Canon-Printers

Stylus 800	black	DM 13,90
Stylus Color	black	DM 16,90
Stylus Color	colour	DM 27,90
Stylus Color II, IIs, 820	black	DM 16,90
Stylus Color II, IIs	colour	DM 27,90
Stylus Color 500,600	black	DM 17,90
Stylus Color 500	colour	DM 27,90
Stylus Color 800, 1520	black	DM 23,90
Stylus Color 600, 800, 1520	colour	DM 34,90
Stylus Color 1500	black	DM 27,90
Stylus Photo	colour	DM 36,90

BJC 6...	black	DM 9,90
BJC 6...	Cyan or Yellow or Magenta	each DM 9,90
BJ 3...	black	DM 11,90
BJC 4...	black	DM 11,90
BJC 4...	colour	DM 17,90
BJC 8...	black	DM 14,90
BJC 8...	Cyan or Yellow or Magenta	each DM 14,90
BJ 30	black, Pack of 3	DM 29,90
BJ 70	black, Pack of 3	DM 22,90
BJ 70	colour, Pack of 3	DM 27,90

Minimum order: 3 items (can be mixed)!

Minimum order: 3 items (can be mixed)!

TERMS OF PAYMENT

Postage and package [Germany] DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). [Europe] DM 14,50 (if total value of goods is up to DM 50,- then only DM 9,50). [Overseas] between DM 14,50 (1 item) and DM 35,- (maximum). All



prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.



IBM printer interface cable to complete the connection of my QL to my printer (long IBM-type parallel printer cables tend to be much cheaper than a CENTRONICS-to-CENTRONICS cables of equal length; and certainly, easier to find

in a store).

Of course, I now enjoy the option of either plugging directly into a parallel printer or employing an IBM-type printer cable as an extension.

[Editor's note: I can tell from

experience that you should not use cheap parallel printer cables longer than 5 metres, otherwise you may loose data - with high-quality cables 8 metres were possible]

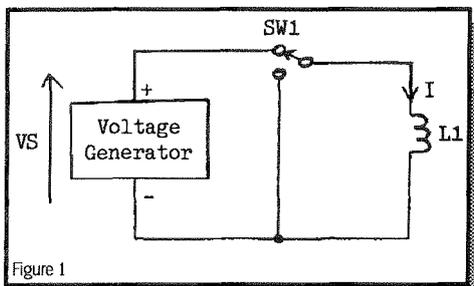
Electronics for QL Peripherals

Part 2 - Reactive components

Stuart Honeyball

The first article in this series dealt with energy transfer from a current or voltage source to a resistive load where the electrical energy was turned into a different form and dissipated from the system. There are 2 types of reactive components - inductors and capacitors - which store energy.

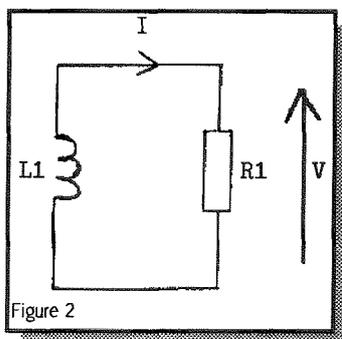
The diagram shows a voltage source feeding an inductor. (The symbol for an inductor, also called a coil, reflects the physical implementation which is simply a coil of wire often wound around a magnetic core.)



In Figure 1 the current I will increase steadily with time at a rate of V/L Amps per second. In calculus terms the current is the integral of V/L with respect to time. Now imagine that switch $SW1$ is flicked so that it short circuits the inductor. There will be $0V$ across the inductor and so I will now remain constant flowing around the circuit loop made by $SW1$ and $L1$. The energy stored in $L1$ is $(1/2) \cdot I^2 \cdot L1$ Joules. What would happen if $SW1$ had a third position where $L1$ was neither connected to VS nor short circuited? $L1$

would not have a complete circuit and so would not be able to pass any current. This means the current would have instantly gone from a non-zero value to zero which implies the voltage across it must be infinite! In practice this can not, of course, happen because real world inductors are not perfect - they have losses. The voltage would however increase to a value substantially above V_S and this effect is widely used to derive high voltages from low voltage sources.

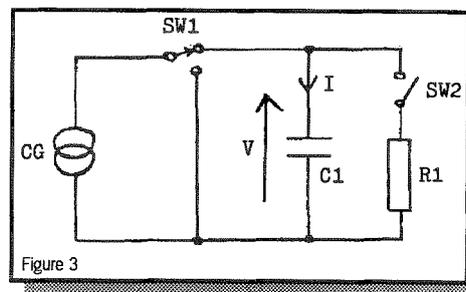
In Figure 2 assume $L1$ has a current I flowing through it to begin with.



The voltage across $R1$ (and $L1$) is V and is simply given by Ohm's Law: $V=I \cdot R1$. The energy comes out of the inductor $L1$ and heats up the resistor $R1$. The voltage will decrease at a rate of $I/L1$ volts per second. If V decreases then I also decreases and so both decrease exponentially over time decaying towards the asymptotic limit of $0V$.

For the equivalent treatment given to a capacitor (the symbol

shows the 2 conductive plates separated by an insulator) consider Figure 3:



V increases at a rate of $I/C1$ volts per second and the energy stored in $C1$ is $(1/2) \cdot V^2 \cdot C1$. Now imagine the current source CS is switched off by flicking $SW1$. It may seem odd to switch off an energy source like this current generator by short circuiting it. Consider what would happen if it were set open circuit: the voltage would rise to try to bring the current up to the rated level which would involve the generator producing its maximum voltage which could well be greater than it can safely deliver. Shorting it causes zero voltage to be produced and thus no energy is taken from it. Now back to the main storey, $C1$ is open circuit and not connected to anything so $I=0$ and so V remains steady. The energy can be extracted from $C1$ by closing switch $SW2$. The current I is given by $-V/R1$ (negative because it flows out of $C1$ the opposite way to which it went in) which causes V to decrease and so, like the inductor, both V and I decrease exponentially tending to 0 and the energy is transferred from the capacitor to the resistor and dissipated. Also note that short circuiting a charged capacitor can

cause high currents to flow through it which can cause damage.

The maths gets even more difficult when an AC (alternating current = oscillating) rather than a DC (direct current = steady) current or voltage is applied to these reactive components. For this reason I will simply state formulae and give the effective conclusions leaving out the derivations. The ambitious reader is welcome to work through the maths himself. Also, apologies for referring to AC voltages. It is clear that voltages are not currents but this terminology is in common use.

The simplest AC current or voltage is one where its instantaneous level at any time t is given by $A \cdot \sin(2\pi f t + \Phi)$ where A is the peak amplitude, f the frequency in Hz (cycles per second) and Φ is its phase relative to another signal. It is usual to use ω (lower case omega) to substitute for $2\pi f$ giving the frequency in radians per second.

Apply an AC current to an inductor (Figure 4):

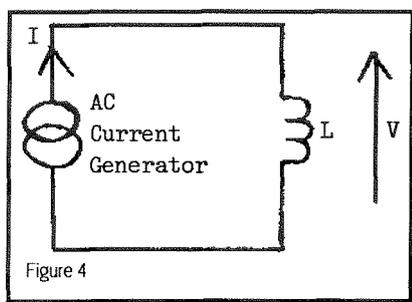


Figure 4

What is V ? As stated earlier I is the integral of $V dt/L$ so V is the derivative of I so $V=L \cdot (dI/dt)$.

$$\text{If } I=A \cdot \sin(\omega t + \Phi)$$

$$\text{then } V=A \cdot L \cdot \omega \cdot \cos(\omega t + \Phi)$$

$$=A \cdot L \cdot \omega \cdot \sin(\omega t + \Phi + \pi/2).$$

In other words the voltage waveform has the same sort of shape as the current with the differences being that it is ω times bigger and leads the current by $\pi/2$ radians (a quarter of a period). Analysis shows that for half the time the inductor is taking energy

from the source and for the other half it is actually giving the energy back. Also when the current is in that part of the cycle when it passes through zero then the voltage is at its maximum and vice versa. Note that the circuit has a frequency related filtering characteristic. The voltage is proportional to frequency. If this were part of a hi-fi then it would give the effect of increasing the treble and reducing bass. Voltage or current generators applied to inductive or capacitive loads exhibit both a \pm quarter period phase change and an increasing or decreasing amplitude versus frequency.

Make a circuit from an inductor and a capacitor (Figure 5):

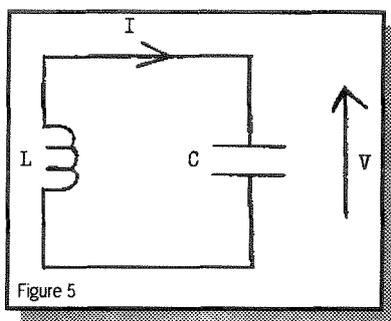


Figure 5

Start with the inductor charged up to a current of I_{max} and the capacitor fully discharged at $0V$.

The current will flow out of the inductor and charge up the capacitor. As it does this the current will fall. When the current has fallen to $0A$ all the energy will be in the capacitor and the current will reverse and start flowing back into the inductor causing the voltage to fall. When the voltage reaches zero all the energy will be back in the inductor and the current will have reached its maximum negative value, $-I_{max}$. The current then falls and charges up the capacitor to its maximum negative voltage then the capacitor loses its energy to the inductor and it's back to where it started so the cycle then repeats ad infinitum. Both the current and voltage give sinusoidal waveforms. It is interesting to note that for each complete cycle the current cycles once, the voltage cycles once but the energy is transferred backwards and forwards between the 2 components twice. The V and I resonant frequency in radians/s is $1/\sqrt{L \cdot C}$. To see whether you've been paying attention or not work out the peak voltage magnitude in terms of I_{max} , L and C .

Q40 - More Hardware Details

Peter Graf

SMSQ/E for the Q40 is still under construction, so this article continues with the description of some hardware functions.

First an addition to the article about graphics hardware in the last issue. There were questions why the Q40 has an own graphics chipset, especially developed for the Q40, instead of a graphics card for Intel-PCs, for example from S3. This has a number of reasons:

1. This is the only way to make the QL screen modes directly available, without software emulation, speed losses or possible distortion of the aspect ratio. Writing directly to QL screen memory at

address 131072 only works with a compatible hardware.

2. The Q40 needs no new screen drivers for the QL screen modes. Writing drivers for the highcolour modes is simplified by easy initialization and addressing.

3. Graphic cards for Intel-PCs come and go in very short intervals nowadays and show large incompatibilities among themselves, even if they come from the same manufacturer.

4. Performance is optimized by directly coupling the graphics

hardware to the bus of the 68040 CPU, without detour to an Intel-PC bus system.

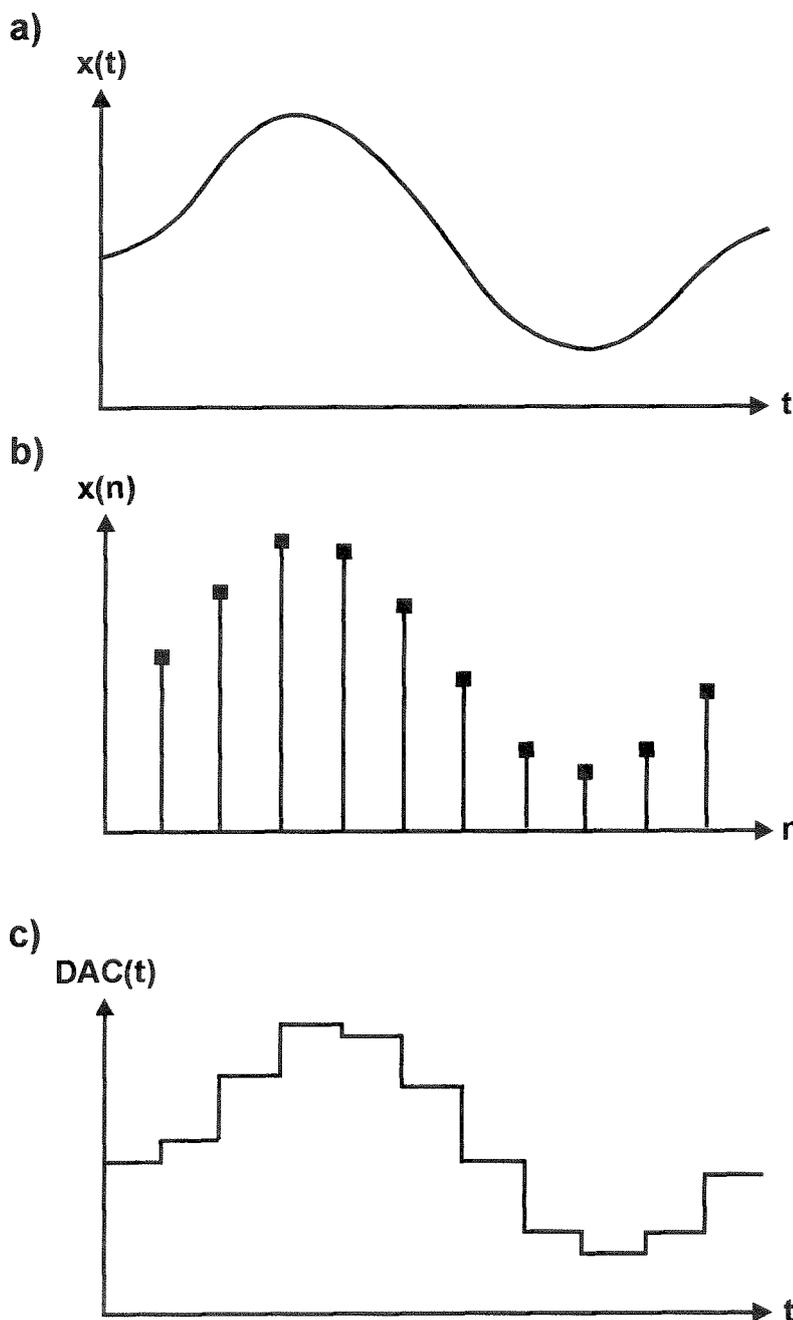
5. The overall system becomes smaller.

Now something about sound generation. On the original QL the output was restricted to the artificial sound of the coprocessor, and not much additional sound hardware was around. The Q40 mainboard has the capability to output sampled stereo sound by a digital-to-analog converter. This gives the user the possibility to hear speech or music in good quality, by connecting a Hifi system, active loudspeakers or a headphone to the Q40.

To explain how the sound output of the Q40 works, an example of a waveform is shown in figure 1a. This could be a short section of a music or speech signal. The analog signal $x(t)$ varies continuously over the time t . For storage or processing on a computer the signal must exist in its digital form. The digital representation $x(n)$ of the signal is a sequence of numbers, which contains the amplitudes of the analog signal at evenly spaced points of time, as shown in figure 1b. $x(n)$ are samples taken from $x(t)$. The more samples are taken within a period of time, the higher is the maximum

audio frequency and the more room for storage is needed. The Q40 uses selectable sample rates of 10 kHz or 20 kHz, so 10000 or 20000 numbers give one second of sound. Instead of a single sequence of samples, the Q40 uses two sequences, to independently represent the left and right channel of the stereo signal. (Of course it is also possible to output mono signals.) There are different possibilities to use digital sound. Simply playing music, filtering and mixing, effects like distortion or echo, sound for games, electronic generation of music, speech synthesis, and so on. How far the possibilities will be used depends on the application software.

To output sound on the Q40 a digital-to-analog converter (DAC) is used. It converts the sequence of data into an analog signal as shown in figure 1c. As you can see, the waveform still contains steps like a staircase. For this reason an analog filter follows behind the DAC to smooth the signal and give it its original waveform from figure 1a. Double DACs and filters exist on the Q40 for the stereo signal. The DACs have a resolution of 8 bits, so they can output 256 different voltages. The Q40 provides the output signals in two different voltage levels, as a headphone output or as a LINE output for Hifi systems and active loudspeakers. Of course monitors with integrated loudspeakers can also be used. The audio DACs of the Q40 can be directly accessed at the addresses \$FF008000 and \$FF008004, but normally the sound drivers should be used for output. The Q40 generates hardware interrupts for sound output. To make programming as easy as possible, the Q40 does not include specialized sound chips like Soundblaster even though they are available cheaply.



The keyboard interface of the Q40 is not constructed to be directly connected to a matrix keyboard. So a QL keyboard can not be used, the outline of the mainboard does not allow mounting in a QL case anyway. There is a connector for MF-II keyboards instead, which are current PC keyboards like those also used with some QL keyboard interfaces. The keyboard informations are transmitted by a serial line, synchronous to a clock line. Data and clock signals, together with the power supply are on a 5-pin DIN connector. The Q40 converts the serial keyboard data into an 8 bit parallel signal. By the way, the data from the keyboard does not directly represent the letters, but contains so-called Make- and Break- codes. They signal the events of pressing and releasing single keys, ignoring their language-specific meaning. For example the key Z on a German keyboard has the Make-Code 53 and the Break-Code sequence 240, 53. On English keyboards these codes stand for the letter Y. The MF-II keycodes are not relevant to the user, because they are interpreted on the lowest level by the keyboard driver of the operating system.

The series production of the Q40 has not yet started, because the adaption of SMSQ/E still goes on. But single prototypes without operating system are already available for firmware developers.

For those who wait for the 68060 mainboard, it should be said that a prototype with 50 MHz clock frequency is already working in test operation. But it is doubtful that this version will ever be produced, because Motorola has already released ordering information about the 66 MHz and 75 MHz CPU's and hopefully will start series production of these new chips soon.

Letter-Box



H.P.Huyg writes:
How not to format a hard disk.

In a previous article I have been moaning about chicken and feeders. Upon reflection it should not have been chicken, but geese. For the innocents among us a keyword could be: pate de foie gras. Why? Look at the logic to format hard disks in the QXL/QPC environment. In the last couple of years there have been some 'improvements' to that function, which are stuffed down our throats, whether we like it or not.

1. You can no longer format a hard disk within a (BASIC) program. Now you have to give a 'WIN_FORMAT' command first. Fair enough, why not 20 commands? But then you have to enter some random letters to proceed with the 'FORMAT' operation. I suspect that some time ago some fool initialized his or her hard disk by mistake. This can happen to every one. Now, what is the difference with a disk crash? One should have made provisions for backup and restore facilities anyway. Do we have to enter a random number to prevent a crash? You might wonder why on earth one would

Sinclair Z88

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

MOBILE: 0850 597650

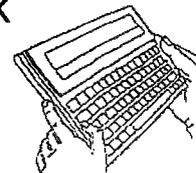
E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL & PC

THE CAMBRIDGE Z88 A4 NOTEBOOK

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



NEW USERGROUP NOW FORMED

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

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	£120	\$180
CAMBRIDGE Z88 COMPUTER	£99	\$130
★ RECONDITIONED Z88	£60	\$90
32k RAMPACK	£18	\$25
128k RAMPACK	£38	\$50
32k EPROM PACK	£18	\$25
128k EPROM PACK	£24	\$33
256k EPROM PACK	£55	\$75
EPROM ERASER	£32	\$43
PARALLEL PRINTER LEAD	£39	\$50
SERIAL PRINTER LEAD	£12	\$16
* MAINS ADAPTOR (230vac: 6v,500ma)	£10	
TOPPER (PROTECTIVE COVER)	£12	\$16
CARRYING CASE (PLASTIC)	£10	\$12
FILE TRANSFER TO OTHER COMPUTERS		
SPECIAL Z88-QL SERIAL LEAD	£10	\$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	\$3
PCLINK KIT (For PCs)	£25	\$35
Z88 TO MAC KIT	£25	\$35
Z88 TO BBC KIT	£25	\$35
} ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!		

PAYMENT BY CREDIT CARD OR CHEQUE PLEASE

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

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

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

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

need to format a hard disk by program. So did I. But then I HAD to write a utility ('QCopier', now donated to QUANTA) to copy files from/to QXL and PC. And it appeared that SMSQ etc does not like somebody else tampering with its hard disk units. And then the automatic formatting seemed useful.

2. QPC now proudly presents its ability to format hard disks in no time. I grew immediately suspicious. No time means: nothing done. And so it is. If you think the contents of your files have been erased after formatting, you are wrong. The files are all there with all their little secrets. Basically, just the File Allocation Table (FAT) and the Main Directory are initialized, nothing else. Why am I not happy? A 'winx_-'unit on the QXL/QPC is just a file on the PC. If I assign say a 100MByte win-unit, that is a 100Mbyte file, although I have used effectively only 40 Mbyte. When I do my backup on my PC on a tape-cartridge (I have written two years ago about it in QL-Today) then the standard backup utility uses compression. With the above 'improvement' in formatting (done only ONCE!), EACH time my backup takes up to 2.5 times longer than in the past. This, because the old 'FORMAT' did write always the SAME information to each sector, and compressors react by writing only once and referring to it.

I was going to be very rude, referring to the introduction, but let me just finish with: 'Oh Father, please forgive them, for they know not what they do'.

P.S. To my opinion 'FORMAT' should not be part of the Operating System, it should be a separate Utility Program.

Marcel Kilgus replies:

1. The WIN_FORMAT is very useful and it was requested by a number of customers. You can't

please everybody, but it has saved many people from losing harddisk contents. Especially if you use removeable drives (this is why it was acutally implemented) you can imagine how easy it is to type FORMAT win1_ instead of win2_. 99% of the normal work is done on win1_, so it is quite easy to type win1_ if you want to format win2_. WIN_FORMAT protects you from doing this. I hope you are aware that you can add FOR x=1 to 8:WIN_FORMAT x

to your BOOT file and you're back to how it was before - where is the problem? ■

2. Again, one can't please everybody. Many people complained that formatting 1MB of ZIP took about 42 seconds - which was really slow when you formatted 100MB or so. The solution which exists now has pleased everybody except for you so far. If I remember correctly, there has been an article in QL Today which told you how to cut down QXLWIN files to the currently used size, as QPC will make it grow automatically if required (it may have been in a German issue).

Indenting BASIC Programs

Dilwyn Jones

SuperBASIC is a highly structured language. By avoiding the use of GOTO and GOSUB commands in your BASIC programs, you can use procedures, loops, functions, multi line IFs and SElects to write programs which are easy to follow, since courses of action and program branches can be clearly followed, as they have defined start and end points, rather than the arbitrary start and end points of program branches made with GOTOS. I am not against the use of GOTOS by consenting adults in private, of course, but where it is used, it is best used only for short jumps less than a screenful of lines apart from each other, so that you can see the origin and destination of any GOTOS in a single edit screen.

Another technique available to the SuperBASIC (and SBASIC) programmer is indenting programs. What this means is that spaces can be placed before certain lines in a program, allowing code contained within loops, for example, to be displayed further to the right than the bounds of the loop. Here is an example:

```
100 LET number = 1
110 REPeat loop
120 PRINT number
130 LET number = number + 1
140 IF number > 10 THEN
150 PRINT 'PROGRAM FINISHED.'
160 EXIT loop
170 END IF
180 END REPeat loop
```

This short example BASIC program just prints a few numbers by counting up within a loop. There is nothing wrong with it as it stands, but I hope that by repeating the listing with structures (loops and IF statements) indented with a few spaces, I hope you'll see that it will be a bit more readable than the above example.

```
100 LET number = 1
110 REPeat loop
120   PRINT number
130   LET number = number + 1
140   IF number > 10 THEN
150     PRINT 'PROGRAM FINISHED.'
160     EXIT loop
170   END IF
180 END REPeat loop
```

If you type a few spaces between the line numbers and the rest of the line, SuperBASIC remembers those and this technique is quite useful - I write all my programs indented like this. It's also common practice among professional programmers. Modern programming tools often perform

Professional & Graphical Software

ProWesS

ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.

The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands (including pictures) and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.

Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that, when you get a new program, all you have to do is insert the disk and indicate "start the program in floppy", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.

ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

PfList

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PfList can create your listings in two columns and in landscape (or both).

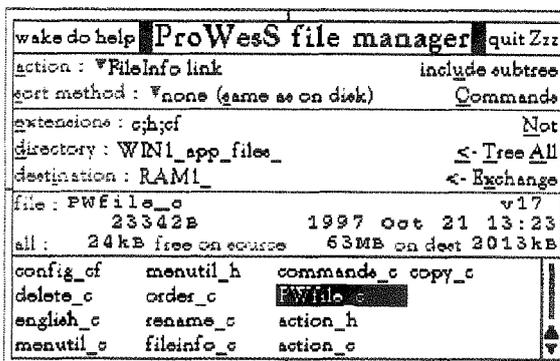
fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurrences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word delimited string.

fontutils

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

*New ProWesS application
a powerful and very user
friendly file manager*



LINEdesign

Create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can use graphics objects to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independent, meaning that the printout will be the same on any printer (e.g. same size and position).

LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts can be displayed at any size, rotation, etc. All the fonts which are available to ProWesS can be used in LINEdesign.

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it. You can choose which fields are displayed and also which records. You can have a hidden comment for each record, look at the file in tabulated form and transfer data to the scrap or hotkey buffer. Files can be memory based (for speed) or disk based (for safety).

new address !!

Dr. Fr. Hemerijckxlaan 13 /1
2650 Edegem
Belgium

tel : +32 (0)3/ 457 84 88
fax : +32 (0)3/ 458 62 07
email : joachim@club.innet.be

www : <http://www.club.innet.be/~year2827>

ProWesS - BEF 2400

DATAdesign - BEF 1200

PWfile - BEF 900

PfList - BEF 600

Payment terms :

LINEdesign - BEF 1200

fontutils - BEF 1200

fsearch - BEF 600

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutils and PfList work (even though DATAdesign uses wman).

All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra costs of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.

If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).

Payment can be done by EuroCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

copies	priority mail			ordinary mail		
	Belgium	Europe	World	Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

```

100 REMark INDENTING BASIC PROGRAMS
110 MODE 4
120 WINDOW 448,202,32,15:WINDOW#2,448,200,32,16
130 WINDOW#0,448,40,32,216:CLS#0:BORDER 1,255:CLS
140 CSIZE 0,0:PRINT FILL$('-',74)
150 CSIZE 2,1:PRINT'  INDENTED BASIC PROGRAM LISTINGS'
160 CSIZE 0,0:PRINT FILL$('-',74)
170 PRINT\TO 21;'Copyright (C) Dilwyn Jones, 1990'
180 REPeat get_filenames
190 INPUT\ 'Indent which program?';inp$
200 IF inp$='' THEN EXIT get_filenames
210 INPUT'Print to where (e.g. ser1)?';out$
220 IF out$='' THEN EXIT get_filenames
230 IF NOT(inp$=out$) THEN EXIT get_filenames
240 BEEP 5000,70:PRINT'Names must be different!'
250 END REPeat get_filenames
260 IF inp$='' OR out$='' THEN AT#0,1,0:PRINT#0,'PROGRAM FINISHED.':WINDOW 448,200,32,16:STOP
270 REPeat get_filenames
280 INPUT\ 'Indent by how many spaces per step (0 to quit)?';spaces
290 IF spaces>=0 THEN EXIT get_filenames
300 BEEP 5000,70:PRINT'Invalid entry!'
310 END REPeat get_filenames
320 IF spaces=0 THEN AT#0,1,0:PRINT#0,'PROGRAM FINISHED.':WINDOW 448,200,32,16:STOP
330 PRINT\ 'Indenting, please wait...'\
340 OPEN_IN#3,inp$
350 OPEN_NEW#4,out$
360 indent=0
370 REPeat get_lines
380 IF EOF(#3) THEN EXIT get_lines
390 IF INKEY$=CHR$(27) THEN PRINT'\ESC.':EXIT get_lines
400 INPUT#3,in_line$
410 backwards=0
420 space=' ' INSTR in_line$
430 out_start=space
440 REPeat unspace
450 IF in_line$(out_start)<>' ' THEN EXIT unspace
460 out_start=out_start+1
470 END REPeat unspace
480 out_line$=in_line$(out_start TO )
490 IF ('DEfine ' INSTR out_line$)=1 THEN indent=indent+spaces:backwards=spaces
500 REMark checking for the colons in the following lines allows suppression of indents for short forms
510 IF ('REPeat ' INSTR out_line$)=1 AND (':' INSTR out_line$)=0 THEN indent=indent+spaces:backwards=spaces
520 IF ('FOR ' INSTR out_line$)=1 AND (':' INSTR out_line$)=0 THEN indent=indent+spaces:backwards=spaces
530 IF ('SElect ' INSTR out_line$)=1 AND (':' INSTR out_line$)=0 THEN indent=indent+spaces:backwards=spaces
540 IF ('WHEN ' INSTR out_line$)=1 AND (':' INSTR out_line$)=0 THEN indent=indent+spaces:backwards=spaces
550 REMark I use the next line for the Turbo pseudo-keyword WHEN_ERROR
560 IF ('WHEN_ERROR ' INSTR out_line$)=1 AND (':' INSTR out_line$)=0 THEN indent=indent+spaces:backwards=spaces
570 REMark in the next line, an unconditional NEXT is most probably the end of a FOR loop rather than part of a
REPeat loop
580 IF ('END ' INSTR out_line$)=1 OR ('NEXT ' INSTR out_line$)=1 THEN indent=indent-spaces
590 IF ('ELSE ' INSTR out_line$)=1 THEN backwards=spaces:REMark multi line IF statement
600 IF LEN(out_line$)>5 THEN
610 REMark look for multi line IF - will fail if THEN omitted
620 IF ('IF ' INSTR out_line$)=1 AND out_line$(LEN(out_line$)-4 TO)='THEN ' THEN
630 indent=indent+spaces:backwards=spaces:REMark simple single line IF...THEN
640 ELSE
650 REMark try for single line IF... without THEN - it normally won't have a colon
660 IF ('IF ' INSTR out_line$)=1 AND ('THEN ' INSTR out_line$)=0 AND (':' INSTR out_line$)=0 THEN
670 indent=indent+spaces:backwards=spaces
680 END IF
690 END IF
700 ELSE
710 IF ('IF ' INSTR out_line$)=1 THEN indent=indent+spaces:backwards=spaces:REMark VERY short form of IF!
720 END IF
730 total=indent-backwards:IF total<0 THEN total=0
740 done$=FILL$(' ',6-space)&in_line$(1 TO space)&FILL$(' ',total)&out_line$
750 PRINT done$:PRINT#4,done$
760 END REPeat get_lines
770 CLOSE#3:REMark input file
780 CLOSE#4:REMark output file
790 AT#0,1,0:PRINT#0,'PROGRAM FINISHED.'
800 WINDOW 448,200,32,16:REMark restore default window
810 STOP

```

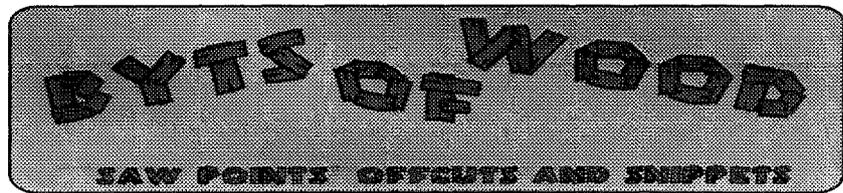
this indenting automatically. Its advantage is that properly done, indenting allows you to see some types of structural errors at a glance. If you indent anything within loops or IFs, the corresponding END IF or END REPEAT will have the same level of indenting as the start of the structure, if not, you'll be able to work through it to find the point at which things have gone wrong, e.g. you may have one less or more END IF than IF statements. This causes errors when your program runs. An IF statement which fails in this way can mean your program can just stop without explanation, or give a misleading error.

As ordinary SuperBASIC programs are stored as simple plain text files, it makes it quite easy for this program to read it in, analyse it and write it out with added spaces to indent the listing. Run this program, enter the filename of the source BASIC program, where to write it out to (this can be a printer port name such as SER1 or PAR if you just want an indented listing, or a filename if you wish to create an indented copy of your BASIC program), and finally enter a number indicating how many spaces to add for each level of indenting. I normally use one or two. The higher the number you enter, the better the indenting, but deeply nested loops will shoot off to the right and be harder to read, so high values become counter productive.

Try the program on itself - if you typed in the listing correctly, it will correctly indent itself to show you what it can do! We set this from an original listing, which we know worked, so hopefully there will be no errors in this listing!

I wrote the following program to indent SuperBASIC programs as described above. As it's only a short SuperBASIC program itself, it won't cope with all possible situations and will occasionally get it wrong, leaving you to correct the indenting by hand. Properly structured programs should be treated correctly.

■



First things first. I would like to extend my thanks to Frank Davis and all the other people who organised the U.S. show in May. This was another good show and I enjoyed it thoroughly. A special mention must go to Kevin Mackeverly who purchased Aurora / Qplane / SuperHermes etc and proceeded to build his tower case at the show. Not only did he get it all assembled but he had it running by the time we packed up. Another special mention must go to Jim Hunkins who organised transport for Jochen, Tony Frishman and myself and also booked the hotel for our three day break in New York. Thanks Jim!

A New Paragraph

One of the things that we were shown at the US show was François Lanciault's new word processor, 'Paragraph'. This is written to use the ProWesS system and it is clearly going to be a very good package. It is currently in Beta test version and there are some things that do not work yet but the things that do work are very good indeed. The program will not be available for release for at least three months but, in its current form, it can use all of the ProWesS fonts, print in colour if the colour printer drivers are used and use graphics in _pic, _ldp and _gif formats. This is going to be a goodie.

The Floppy on my Laptop!

Continuing on from the mentions of the 'other system' in last issue's column I have recount something which happened to me. I have been running a series of beta

copies of Windoze 98. Many people think this is suicide for a system but, since I refuse to use Windoze for anything but internet access I really don't care if it crashes (what do I mean 'if?'). One thing that hardly gets used on a modern PC is the disk drive because PC programs and files are usually so bloated that you can't fit them on to a standard HD disk, so it was not until I tried to sort out a mouse driver for a friend of mine that I found out that the disk drive on my laptop did not work. It worked fine from QPC and from the native DOS but, as soon as I ran Windoze, it refused. After a week of searching through dlls, re-installing Windoze, and running diagnostic programs which all said 'This device is working properly' I finally asked a friend of Steve Hall's who knows a lot about this kind of thing. He asked if I had a virus checker. 'Of course I do', I said. 'Turn it off then', he replied. It all seemed illogical to me but I tried it and the disk drive sprung back into life. It seems that the virus checker is designed for Windoze 95 and thinks that Windoze 98 is a virus. It all made sense! Windoze spreads from computer to computer when you don't really want it, it makes it crash a lot and you lose data! Tell the world to beware, Windoze is not an operating system - it is a virus.

Don't Try This At Home

In recent weeks I have found two foolproof ways to destroy QPC so I thought I should pass these on to save other people from having similar problems. The first came from a customer of mine who had a hard disk crash on him

and lost his QPC installation. He decided to use the second installation and, having had a problem reported when he tried to read it, he thought he would check the disk first to make sure it was OK. He ran Scandisk from the PC which told him there was an error on the disk and asked if he would like it fixed. 'OK' --- and there goes the disk. When he ran the installation program he got 'Illegal Copy'. This is because scandisk altered the master disk and then QPC thinks it is an illegal copy so this is one thing you should never do. Another good way to destroy your QPC installation is to convert the master disk from 16 bit to 32 bit without first removing QPC. Converting the File Allocation Table is something you should consider carefully if you run a lot of older or DOS based programs but if, like me, you only run a few modern PC programs and use it mostly as a QPC platform it could be worthwhile. The speed increase and the amount of disk space freed by this action make it very worthwhile. FAT 32 is only available on later versions of Win-doze 95.

Iconography

I recently received a letter from Mr W.J. Henry renewing his subscription to this magazine. He said 'Keep up the good work but with less emphasis on trying to make QLs compatible with the PC, complete with Windoze, Itchcons (Incomprehensible), infestation of meeces and pointless pointers.' (his spellings). I wanted to reply to this here because it is a subject which I have touched on before and fundamental to the development of our system. On the whole QL programs which use the Pointer Environment do not rely on the use of a mouse - they just enable it. For many people, and I am definitely one of these, a mouse is

very useful and aids my productivity by there others for whom it is an annoyance. On the PC many programs will not work without a mouse or become very hard to control because the mouse is not controlled by the cursor keys as it is on our system. As far as I know 99% of pointer driven programs on the QL are able to be controlled without ever touching a mouse because all of the commands have keypresses which activate them or, at worse, function keys which will pull down menus that are controllable by the cursor keys. Some people find the mere presence of icons off putting and I am pleased that many programmers are using removable toolbars so that these are not so intrusive. With regard to compatibility with the PC this is something which should be encouraged. Many people have left the QL because they cannot access files from their friends or workmates and anything that we can do to enable this form of compatibility is fine by me.

Wherever you may ROM

There may be several people out there who have had problems with their RomDisqs and they will be pleased to hear that the cause of the problems has now been found and corrected. It seems that there was a de-bugging routine that was used while developing the driver for the RomDisq and part of this was inadvertently not removed in the released version. Every now and this code would pop up and then make a call to part of the system which was not there - result: lock up.

This code was removed in v2.11 but then, when Tony confidently sent me the new version I got the dreaded 'howling sysmon syndrome'. A quick fax to Tony Tebby solved this and resulted in v2.12 and the cryptic fax saying 'Ah this is a bug in a fix of a fix of a fix of a

problem that occurred on Minerva. I have now fixed the fix of a fix of a fix and it should be OK'. I am pleased to report that it is.

The new version has the ability to load in ROM images and LRESPR files which are called ROM1, ROM2, ROM3, ROM4. I wanted this feature so that I could put the SuperHermes code onto it and have the keyboard instantly available in order to be able to test programs directly from their master disk with nothing else loaded. It is also good to be able to run The Knight Safe that way to maximise the available memory.

It was only when I was asked to set up a RomDisq for a customer with QPAC2 etc on it that I realised how good this little device was. I plugged it into a standard QL with a Super Gold Card and a set of DD drives and set up the boot file for SMSQ/E, QPAC 2, Text 87 etc. When I rebooted to check the files I was amazed at the short time it took to boot up. If you have no hard disk check this out!

Reasons to be Cheerfull

One little ray of sunlight that has entered into things this week occurred during a disaster which befell my wife's laptop. After her hard disk failed recently I attacked the thing with a screwdriver and, after hunting up a new hard disk, installed it into the heart of the machine. Of course nothing is simple in 'PCLand'. You first have to tell the BIOS what disk it has, after poking and prodding the BIOS for some hours I finally convinced it to look at the disk and to get the thing formatted and ready to receive the new backup.

Here in the real world of QDOS / SMSQE Phil Borman's clever little Qubide just takes a quick look and says 'ah its one of those, OK' and you are off.

The ql-users section of the majordomo list has been active

again this month and there are reports the Phil is working on ATAPI support for Qubide. Don't get too excited over this because it will not mean automatic support for all ATAPI drivers (although the LS-120, a 120Meg disk that thinks it is a floppy disk and can read HD and DD disks as well, would be very welcome) but we are moving in the right direction here.

Major Jump

By the way I should apologise for misleading a few people about the way to log on to the ql-users list. When this list was first originated I was automatically enrolled and I had no idea how to do it. I misread some of the instructions and I am indebted to 'Spike' of Loughborough University for putting matters straight. The correct way to log onto this list if you have email access is to send a message to 'majordomo@vng.ntnu.no' and put the words 'subscribe ql-users' into the message box. that will put you onto the best ql mail list in existence. Equally 'unsubscribe ql-users' will allow you to exit.

I apologise unreservedly to all those who I have previously misled.

No Chips With Anything

You may remember that we have a few Super Gold Cards and QXL lying around here waiting to be used and that they have all lacking one component - namely the ALTERA EP1810LC-25T chip that is used as the Super Gold Card INGOT and the QXL GLUE.

In the last issue I mentioned the problems that we had been having with these items and fortunately the manufactures agreed to take the chips back and I was then back to square one.

Just recently we again thought that we had solved the problem when one of my suppliers said

that there were some available from the US. I eagerly awaited their arrival and I had hoped to be able to tell you that the last SGCs and QXL would be available soon. Well the chips arrived and we found that they were already programmed - but for some other device! Since these are one-time programmable that was that. Another false trail. We are still working on this problem and there will be a solution. If any of you are involved in the electronics industry and think that your company may have a few of these devices available do call us.

offer three choices. The first choice is to update a file and this will offer a files menu to choose the file to be updated. When you select a file with a 'DO' or 'ENTER' it will load menuconfig, updated the file from the 'menuconfig_inf' file and then quit all by itself.

The second choice offers to start menuconfig with a file already loaded for configuration by a similar method to that above.

The third choice will start menuconfig alone.

If you have menuconfig on a button in your system and you replace that with this program you

```
100 REMark Program to select a file and start
Menuconfig with it loaded or update it
110 DIM A$(2,25)
120 A$(0)='Update File'
130 A$(1)='Configure File'
140 A$(2)='Start Menuconfig'
150 A=LIST_SELECT('Action',A$,,,,50,400,-1)
160 SELECT ON A 170 --1 : stop
180 =0 : getfile : EX win1_progs_Menuconfig;"\U
"&file&" \Q"
190 =1 : getfile : EX win1_progs_Menuconfig;File$
200 =2 : EX win1_PROGS_MENUCONFIG
210 END SELECT
220 DEFine PROCedure Getfile
230 file$=FILE_SELECT$('Menuconfig
Select',,WIN1_PROGS_,,,,1,-1)
240 END DEFine
```

A Little Code

Just as an example of the kind of thing QL Today is looking for in the Hot Tips and Odd Procedures section I am including a bit of BASIC that I put together because I wanted to update programs with Level 2 config blocks. This relies on menu_rext being LRESPR'd and menuconfig being available and you should change the following lines to suit your system: lines 180/190 should show the location of menuconfig line 230 should give a target directory. See the listing above.

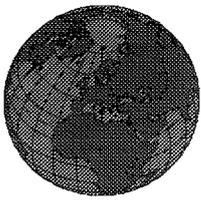
When this program is run it will

will have a more flexible way of using this facility.

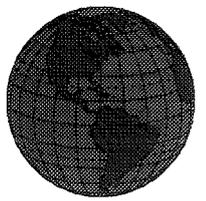
The Book

I have just spent a weekend with Rich Mellor putting the finishing touches to the manuscript of the SBASIC / SuperBasic Reference Manual. By the time you read this we should be able to quote a price and offer the manual for sale. Since this is such a large work it will probably be better to order it and then pick up a copy from the nearest QL Show.

■



The QL Show Agenda



Sunday 16th August 1998

9th Great Eastern Radio and Computer Rally

Doors open 10.00 am. The venue is Wallington Hall.

Located between King's Lynn & Downham Market, Norfolk. Approx 5 miles South of King's Lynn on the A 10.

Quanta, TF Services, Qubbesoft, Q Branch, W.N. Richardson attending!

Saturday, 5th of Sept. 1998: Eindhoven - regular meeting

As usual, between 10am and 4pm, in the St. Joris College in Eindhoven.

Sunday, 4th of October 1998: The BYFLEET WORKSHOP

The show will be on between 9:30 and 5, in BYFLEET VILLAGE HALL.

BYFLEET is just inside the M25, on A245. From M25 Jn 10 take A3 towards London then A245 towards Woking, and turn left (south) into Byfleet at the first or second roundabout past the A318 junction.

From M25 Jn 11 take A317 towards Weybridge then A318 to Brooklands/Byfleet. NB A318 now goes through Brooklands racetrack. Turn right (west) at A245.

Non-M25 is via A245 from west or east (or A3).

The hall is on the (old) High Road, western end, just south of A245. It is about 12 mins walk from Byfleet & New Haw station, on the main Waterloo - Woking line. Queries? Ken Bain (SQSG Sec) 01932 347 432, pre midnight, email kenb@bcs.org.uk

Sat. & Sunday, 10th and 11th of October

International QL Show in Heidenreichstein, Austria (near the Czech border).

The venue is the Gasthof-Restaurant Nöbauer, A-3860 Heidenreichstein, Schremser Straße 28 (Tel. 0043(0) 2862/52237 or 52746). A room the size of the one in Salzburg (after moving) is available. The Gasthof provides 22 rooms with 22 beds. You can also find other accommodation in the same town.

As usual, there is also a "tourist" program for the visitors: a 1-2 hour ride on an old-fashion train (paid for by the Vienna QL user group), followed by a dinner with local specialities. On Sunday, it is possible to visit the moor and one of the best-kept water-castles. A visit to a local glass-manufacturer should be no problem.

Saturday, 31st of October 1998: Eindhoven - International meeting

Again, as usual, between 10 am and 4pm in the St. Joris College. We expect all dealers to come, and of course lots of international guests will be invited.

Probably THE date on the continent for Autumn.

Sunday, 8th of November 1998: Portishead (Bristol) Workshop.

There is still plenty of time, therefore details will follow in the next issue.