

# QL Today

Volume 14  
Issue 1  
Sept. - Nov.  
2009

ISSN 1432-5454

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

Over 11 pages  
in this issue  
- and there's  
more to come!

# QL

# Firmware

# Bugs

# Myths

All the details about  
what happened 25 years ago at Sinclair  
- revealed by Tony Tebby!  
Extremely interesting! Hair-raising!  
Frustrating! Amazing!

Well, a bit of everything!  
Once you start reading, you can't stop!

[www.QLToday.com](http://www.QLToday.com)

# Contents

- 3 Editorial
- 4 News
- 7 Font Browser *Ian Burkinshaw*
- 9 Programming in Assembler - Part 24  
*Norman Dunbar*

## FEATURED STORY

- 12 QL Firmware Bugs Myths - Part 1  
*Tony Tebby*
- 23 Some non-Software Updates  
*Jochen Merz*
- 24 Fun with Fonts - Part 4  
*Dilwyn Jones*
- 30 Simple GPS receiver *Ian Burkinshaw*

# Advertisers

in alphabetical order

Jochen Merz Software	31
Quanta	29
Quo Vadis Design	15

**The deadline for  
the next issue  
is the 15th of  
November 2009  
- but please send  
material a.s.a.p. so  
that we can have  
the issue ready for  
you before  
Christmas!!**

# QL Today

ISSN 1432-5454

## German office & Publisher:

Jochen Merz Software      Tel. +49 203 502011  
Kaiser-Wilhelm-Str. 302      Fax +49 203 502012  
47169 Duisburg              email: smsq@j-m-s.com  
Germany                      email: QLToday@j-m-s.com

## Editor:

Geoff Wicks                      Tel. +44 1332 271366  
Flat 5b                              email: gtwicks@btinternet.com  
Wordsworth Avenue              email: QLToday@j-m-s.com  
Derby DE24 9HQ  
United Kingdom

## Co-Editor & UK Office:

Bruce Nicholls                      Tel +44 20 71930539  
38 Derham Gardens              Fax +44 870 0568755  
Upminster                          email: qltoday@q-v-d.demon.co.uk  
Essex RM14 3HA                  email: QLToday@j-m-s.com  
United Kingdom

**QL Today** is published four times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage [www.QLTODAY.com](http://www.QLTODAY.com).

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

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

This magazine and all material within is Copyright 2009 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Dilwyn Jones' web site where you find more information about lots of interesting QDOS software and INFOZIP at <http://www.dilwyn.me.uk/arch/index.html>

We are starting volume 14 with a news story on QL games. This may raise a few eyebrows because Sinclair wanted the QL to be a business and not a gaming machine. In practice, as a recent article in Retro Gamer demonstrates, this is one of the myths concerning the QL's development.

The early days of the QL are shrouded in similar myths, misunderstandings and half truths.

Throughout volume 14 we shall be looking at these early days in some detail as seen through the eyes of Tony Tebby. It is a detailed account with some surprises that could change the way we think about the QL's development.

Your first reaction to this issue of QL Today is probably a different sort of surprise. It is much thinner and lighter than previously. As we explained at the end of last year the economics of current magazine production faced us with a choice of either putting up the price or of producing a smaller magazine at a lower price. We opted for the latter.

In the last issue I reported how the cost of producing the Quanta Magazine had risen by over 34% per reader in three years. Similar economics apply to QL Today and in our case there is an additional headache of international postage costs. The size of the present magazine is determined by the need to keep its weight below 100g and thus in the cheaper postal tariffs.

A further consideration was future editorial viability. Although the magazine is still healthy editorially, I cannot guarantee how long it will remain so. Bluntly there are not many chickens left in the QL community. Indeed, I was reminded forcibly of my own vulnerability earlier this year when I reached the age of 67. My father died at 68 and my grandfathers at 65 and 69. (I do not intend to follow their example.)

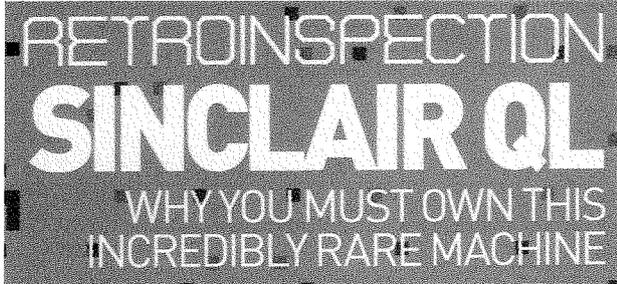
We are taking steps to ensure as much editorial content as possible, and are working on the assumption that most readers want the emphasis to be on articles by our regular and occasional writers. We have slightly reduced the font size to enable us to get more information on each page. The news pages will have special attention in an attempt to keep the news stories crisp and concise. In fact we anticipate that there will be fewer news stories and fewer shows to report, which will free editorial pages for other content. Finally long program listings will no longer appear in the magazine. We shall still welcome these, but they will be posted on the internet for you to download.

The series of articles by Tony Tebby will be taking up a lot of space in volume 14 - almost 11 pages in this issue - but when you read the content I am sure you will agree that the series is more than worthwhile. It is an inside story of the QL that you will not read elsewhere. Obviously we still welcome contributions from other writers, but it may take a little longer than previously for your work to appear in the magazine. We hope we shall have your patience and understanding that this is for a good reason.

## QL GAMES HIGH SCORE

"SINCLAIR QL- why you must own this incredibly rare machine."

The latest tribute to the QL from the present day computer press came from the unexpected source of a gaming magazine.



With this teaser on its front cover "Retro Gamer" had a 6 page feature on the QL.

After repeating the well known reasons for the QL's commercial failure, the article made some interesting comments on the QL and computer games:

cost gaming software available, while on the other, insider rumblings suggested that gaming was seen to be beneath the dignity of the company. David (Karlin) disagrees with reports that gaming was a dirty word within Sinclair.

'There were lots of people in the company who understood the gaming market very well and put plenty of work into ensuring that the Spectrum thrived in it, and the QL did its best. If anything, I swam against the tide within the company by focusing the QL resolutely on business.' David's initial intention was to develop a pure business machine designed to be hooked up to a monitor, but his hand was forced as the project progressed. 'Part way through development, I was told firmly not to alienate it so far from the Spectrum's market, at which point things like the TV interface and joystick ports were added. Retaining a tape port à la the Spectrum was discussed but discarded - the microdrives were supposed to be good

enough.'

Retro Gamer suggests the QL had the raw power to carry out the complicated calculations that games require, but that its main handicap was the display that occupied 32K of ram memory compared with the Spectrum's 7K. This made fast full screen scrolling difficult and thus QL games were largely static screen affairs. David Karlin suggested that a raw gaming machine would have had no mode 4 and no co-processor.

However Retro Gamer notes that the two most fa-

## COMMUNITY THE BEST QL RESOURCES ON THE WEB

### RWAP Software

[www.rwapsoftware.co.uk](http://www.rwapsoftware.co.uk)

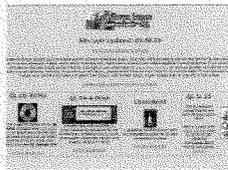
Rich Mellor of RWAP Software has been supporting the QL since 1987 and his site should be your first port of call for information about QL software and hardware. Rich also has lots of items for sale and offers a repair service for faulty machines.



### Dilwyn Jones Sinclair QL Pages

[www.dilwyn.co.uk](http://www.dilwyn.co.uk)

Dilwyn's site is perhaps the most regularly updated QL resource and is definitely worth bookmarking. There's a wealth of software available to freely download, including games, plus a vast library of QL-related documentation and information.



### QUANTA

[www.quanta.org.uk](http://www.quanta.org.uk)

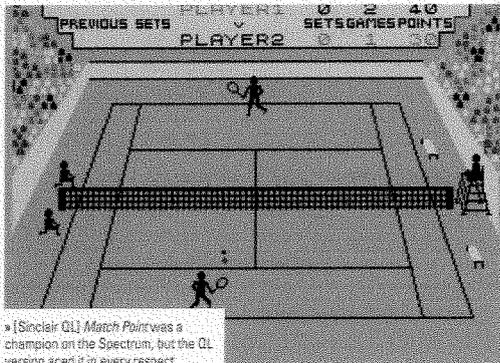
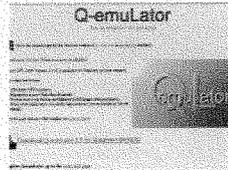
The independent QL user group QUANTA (The QL Users and Tinkerers Association) was formed prior to the machine's launch and is still going strong today with regular meetings and workshops organised around the UK. The group also publishes a bi-monthly magazine for its members.



### Q-emuLator Homepage

[terdina.net/q-emuLator.html](http://terdina.net/q-emuLator.html)

There are several QL emulators available, but if you're looking to run QL games on a PC or Mac then you should opt for Q-emuLator. The full version is shareware, but the free trial includes everything you need to run the vast majority of games. Visit the site for downloads and support.



» Sinclair QL Match Point was a champion on the Spectrum, but the QL version aced it in every respect.

**BJ Land 3D** NEW ONLY £49.95

Amazing! It is off again, this time in a 3D mode that will make a challenge to all other 70,000 titles of 3 dimensional action. It is further, against Robots, energy, monsters, zombies and even giant Chess pieces. Unbelievable!

Quinn has also written a sequel to this game. Great reviews! Full version available for £49.95. Full version available for £49.95. Full version available for £49.95.

ONLY £49.95

ONLY £49.95

ONLY £49.95

» Eidersoft published some decent QL software, including Damon Chaplin's first game, Spook, and the second and third BJ games.

'Sinclair seemed to have a love/hate relationship with games. On one hand, the success of the Spectrum was fuelled by the abundance of low-

cost gaming software available, while on the other, insider rumblings suggested that gaming was seen to be beneath the dignity of the company. Most other software houses preferred more

colours to more detail. The magazine makes a special mention of Pyramide's Wanderer saying that it illustrates the QL's strength of displaying vector style graphics - "...the 68008 could eat 8 bit processors for breakfast".

As top ten games Retro Gamer names the following:

- 1: Match Point
- 2: Karate
- 3: Jungle Eddi
- 4: BJ in 3D Land
- 5: QL Quboids
- 6: Deathstrike
- 7: Speedfreaks
- 8: Pudge
- 9: Mortville Manor
- 10: QLPawn

The feature includes a screen shot from each of these games.

Prominently displayed in the article are links to four QL websites - RWAP, Dilwyn Jones, Quanta and Q-Emulator. There was no favouritism for the inclusion of RWAP - Rich Mellor had assisted in the preparation of the feature.

QL Today still has a free copy of Dilwyn Jones freeware games CD for any reader willing to write a review of it.

## CONTINENTAL CELEBRATION

Urs König who has done much to publicise the quarter centenary of the QL is organising a celebratory show and dinner in Switzerland. To QL Today's knowledge this will be the only continental event to commemorate 25 QL years. The event will be held in Lucerne on the weekend of 31st October/1st November and will also celebrate the 25th anniversary of the Mac. The conference is being financed and supported by members of the former "Sinclair User Club Schweiz" with COWO as partner. Urs would welcome the support of other partners.

Urs writes:

"It will be hosted in a very special place, the brand new Conference Center, in the famous Verkehrshaus (Swiss Transport Museum).

For first details please visit the web-page:

[http://www.qivs Jaguar.homepage.bluewin.ch/QL\\_and\\_Mac\\_are\\_25\\_international\\_event.html](http://www.qivs Jaguar.homepage.bluewin.ch/QL_and_Mac_are_25_international_event.html)

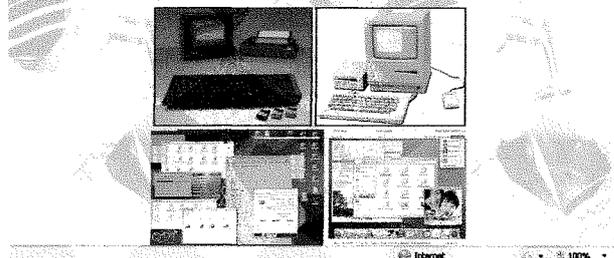
Even if those historical and recent QL and ICT themes - the event will be much more than ancient/retro - are not enough motivation for some to attend think about a few days "Indian summer" holiday in the in the beautiful heart of Switzerland."

Urs would welcome feedback about the show and offers from speakers willing to do a presentation.

## "QL & Mac are 25" international event - Oct 31-Nov 1, 2009

Added: July 1st 2009

Lucerne, July 1st 2009 - 2009 is the year of the 25th anniversary of breaking innovations in personal computing. Early 1984 two milestones were unfolded on both sides of the Atlantic Ocean: Europe was first with the launch of the 32bit multiwindowing Sinclair QL. 10 days later America saw the launch of the 32bit GUI multiwindowing Apple Macintosh. They both entered the field to compete IBM's PC.



## INTERNATIONAL QL SCENE

Quanta claims on the home page of its website, "We have close links with international QL groups".

In practice, as far as QL Today can ascertain, apart from Quanta there are only two active QL groups remaining - Sin\_QL\_Air in the Netherlands and Sinclair QL Spanish Resources in Spain. Sin\_QL\_Air was hoping to run 3 meetings this year, but health problems have prevented the organiser from doing this. Sinclair QL Spanish Resources made the news earlier this year when it scanned and placed a large quantity of QL documentation online.

Quanta can claim good links with the Spaniards - Javier Guerra designed the logo that Quanta is using for its QL is 25 celebrations - but there has been no contact with Sin\_QL\_Air for many years. When the Dutch user group hosted the QL2004 international show, Quanta failed to reply to any of the emails sent to them.

Last year former members of the Italian user group met again for a one-off meeting and this year former members of the Swiss user group are organising a major celebration of the QL's quarter centenary. The last North American show was held three years ago.

Quanta has recently commissioned Dilwyn Jones to investigate the situation and he writes:

"Quanta currently lists contact details for several QL sub-groups in Britain.

We would like to include contact details for QL user groups in other countries too.

If you would like to have contact details for your group included in Quanta magazine, please could you send the details to me as soon as possible, to the email address:

[news@quanta.org.uk](mailto:news@quanta.org.uk)

What we are trying to do is to establish which countries still have user groups and how active they are. It doesn't matter if it is a formal user group or just an informal group of individuals meeting occasionally.

Alternatively, if you are a QL-using individual in a country where you are not aware of a user group, I would consider publishing a "contact request" if you would like to invite other QL users in your country to contact you."

Within the UK there could be a similar situation with up to half of the advertised Quanta subgroups being non-active.

## MEL LAVERNE

At the end of June the sad news came from North America of the death of Mel Laverne.

His son Doug informed the QL-users email group that Mel had suffered a stroke on January 15th and had slowly declined eventually passing away on 24th June. Doug wrote:

"He had retained his faculties but his body would not co-operate.

Mel acquired his first QL, the ubiquitous "black box", in the 80s although I don't know the year. He moved up to floppies, HDDs and Aurora. He acquired his first desktop (admittedly, a non-QL machine) through QL-ers. He had extensive collections of QL magazines, manuals, and even 5 1/4 floppies.

He wrote articles for the defunct IQLR; I would have to do research to know whether he did or did not for QLT or even Quanta. He attended QL NAs including Rhode Island 1994 and one hosted by NESQLUG and hosted one in Oak Ridge, TN, USA (1995 I believe). The last few years, as his wife Eleanor's health declined, his participation in many things decreased. At the time of his death I, Doug, his son, was encouraging him to keep his mind stimulated in the skilled nursing facility by making certain enhancements to Mr. Kennedy's "Tower of Hanoi" game program from an old Quanta issue."

Several European QL-ers who had met Mel at North American shows paid warm tributes to him.

## DILWYN JONES ADDRESS CHANGE

Please note the new address for Dilwyn Jones as of 14th August 2009:

Dilwyn Jones  
22 Erw Las  
Coetmor New Road  
Bethesda  
Gwynedd LL57 3NN  
United Kingdom

Email:

[dilwyn@evans1511.fsnet.co.uk](mailto:dilwyn@evans1511.fsnet.co.uk)

or [jones@dilwyn.me.uk](mailto:jones@dilwyn.me.uk)

(Old email addresses will all remain active for now at least)

## NEW GD2 SPRITE EDITOR RELEASED

Dilwyn Jones announces:

"Duncan Neithercut has sent me a new GD2 sprite editor to make available on my website.

It is an icon/sprite editor to create sprites in mode 64 or mode 4 or to save them as .bmp for use in Wolfgang Lernerz program. Existing sprites in a variety of modes (including mode 32 QXL/QPC and mode 33 Q40/Q60) can be loaded and edited. Or sprites can be created from scratch. The editor has a number of novel features including an independently editable alpha channel, undo function and a simple merge to combine 2 sprites of the same size, and other features such as home directory and colour theme awareness.

It is in an alpha/beta status but is fully usable, but there may still be bugs due to the complexity of the program, that require additional users to identify. In other words, try it and let Duncan know of any problems you find!"

The program (92Kb) can be downloaded from the Sprites page on Dilwyn's website:

<http://www.dilwyn.me.uk/sprites/index.html>

## GEORGE GWILT UPDATES

George Gwilt has announced seven updates to his programs, about half of which concern the use of IO\_EDLIN to edit a line by adding a test of buffer overflow:

Program	Update
GWASL	EDLIN
GWASS	EDLIN and a corrected assembly of MOVEC NET_PEEK
EDLIN	(In the files GWPK)
GWDISS	EDLIN (In the files GWPK)
DISP	EDLIN
EasyPEasy	Slightly better method of producing window working definitions
SETW	Allows much more overlap of information windows (due to remarks by Bob Spelten)
TurboPTR	A fault in TPTR_BAS corrected plus several small improvements

The programs can be downloaded from:

<http://web.ukonline.co.uk/george.gwilt/>

## FLYING THE SALTIRE

As any Belgian will tell you copper wire was invented when two Dutchmen saw a small coin lying on the ground at the same time. In the UK the English regard the Scots in the same way. In return the Scots, with some justification, accuse the English of stealing their oil. Or in the more modern version covering their hills with our turbines to steal their wind for our electricity.

Quanta's Scottish subgroup has now raised the Scottish flag or Saltire in a challenge to Quanta's Sassenach committee even though one member of the committee is a fellow Celt. When Dilwyn Jones recently appealed on the QL-users email group for news for inclusion in the Quanta Magazine John Sadler replied that he should perhaps pay for Scottish news:

"You may like to receive the SQLUG magazine. It will tell you what is happening in SQLUG and there is part of an article each month. £4.00 Subscription will get you all this year's issues."

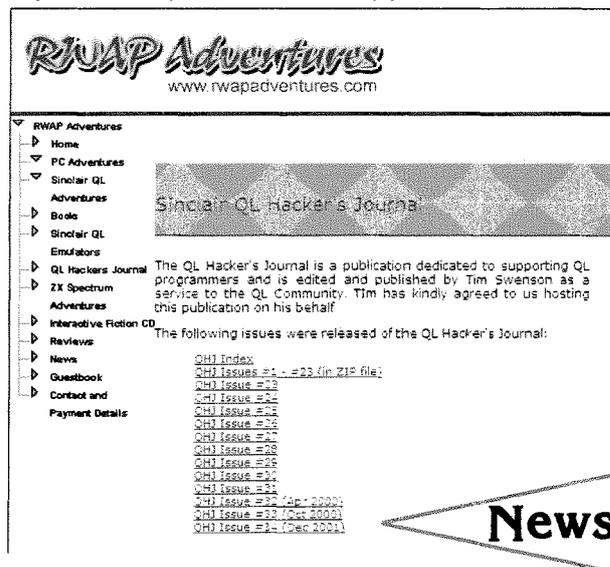
Don't be a meanie Dilwyn! Cough up! Quanta still has thousands in the bank.

## QL HACKER'S JOURNAL

Rich Mellor writes:

"With geocities due to close later this year, I have, with the agreement of Tim Swenson, now added the QL Hacker's Journal to my RWAAP Adventures website."

<http://www.rwapadventures.com/qhj.html>



**RWAAP Adventures**  
www.rwapadventures.com

▼ RWAAP Adventures  
▶ Home  
▼ PC Adventures  
▼ Sinclair QL  
▶ Adventures  
▶ Books  
▶ Sinclair QL  
▶ Emulators  
▶ QL Hackers Journal The QL Hacker's Journal is a publication dedicated to supporting QL programmers and is edited and published by Tim Swenson as a service to the QL Community. Tim has kindly agreed to us hosting this publication on his behalf.  
▶ ZX Spectrum  
▶ Adventures  
▶ Interactive Fiction CD The following issues were released of the QL Hacker's Journal:  
▶ Reviews  
▶ News  
▶ Guestbook  
▶ Contact and  
▶ Payment Details

Sinclair QL Hacker's Journal

QHL Index  
QHL Issues #1 - #22 (in ZIP file)  
QHL Issue #23  
QHL Issue #24  
QHL Issue #25  
QHL Issue #26  
QHL Issue #27  
QHL Issue #28  
QHL Issue #29  
QHL Issue #30  
QHL Issue #31  
QHL Issue #32 (Apr 2000)  
QHL Issue #33 (Oct 2000)  
QHL Issue #34 (Dec 2000)

**News**

## Font Browser

by Ian Burkinshaw

I did enjoy Dilwyn Jones' article on fonts, "Fun With Fonts - Part 1" in Volume 13 Issue 2 of QL Today. This set me thinking, since I use different fonts in some of my own programs. One of the issues I find is sorting out which font I wish to use, also if a particular font will work at a given character size, not all do. I don't recall a font browser before being published, but I could be wrong. You will see that I have used Dilwyn's short program from his original article for loading and displaying a font as a basis for my program. This short program will display each font set one by one, and display each font in all the variations of character size (CSIZE). You can browse all the fonts that you have in a given directory. So first I combined all my font files into one directory, which could be on your hard drive or on a floppy, even a microdrive. These font files came from the downloads from Dilwyn Jones web site,

<http://www.dilwyn.uk6.net/fonts/index.html>

and also from my copy of Lightning from Digital Precision which is difficult to find now. Unless you can find someone who is willing to sell you their copy. It is, I think still copyright, so should not be copied. But the fonts available from Dilwyn Jones web site should cover most peoples requirements, legally.

Enter the program below, you will need to set the 'dev\$' in line 1010 to the directory name you are using. Or you could add an INPUT line here if you wish to enter the directory names as you use the program. If you use the name 'Founts' as the directory name that hold all your fonts, as in my case, then you need make no other changes.

However if the name you use has a different length, that is longer or shorter than 6 characters plus the underscore making 7 so the first character you want to use is the eighth, then the remainder of the string is used to select the required font, is from the eighth character to the end of the string. Then you need to change the numbers, currently 8, in lines 1190, 1200, 1210 and 1220. file\${n[8] to length). Change n to the length, number of characters, of your directory name. I did this so only the name of the fount and it's extension, \_fat, \_font, \_fnt and \_chr are displayed. Just makes it a little easier to read. So if you do have a INPUT line to select the directory you want to use then you need to take this into account.

When you run the program it loads the selected directory into an array, d\$( ). Only file names which end with \_fat, \_font, \_fnt and \_chr loaded are loaded into the d\$( ) array. Up to 300 font file

names can be loaded, this should be more than enough, but can be changed in line 1020. The name of the font being display is shown at the top of the screen. Using the 'U(up) key will move

you up the list of fonts and the 'D(down) key will move you down the list of fonts in the chosen directory. The 'S' key will quit the program.

```

20 init
30 dir_fonts
40 read_file_names
50 browser
60 CLOSE#3:FORMAT ram1_0:STOP:REMark close data channel, clears ram and stops program
1000 DEFINE PROCEDURE init
1010 dev$='win1_Founts_':REMark Change this line for the drive and directory name were the fonts
    are saved in you system
1020 DIM d$(300,32):REMark change the first number if you need more or less space as required,
    needs to more than the number of font you have in this directory
1030 FORMAT ram1_80:REMark space to save the directory data temporarily
1040 WINDOW#1;SCR_XLIM,SCR_YLIM,0,0:REMark Window that will display the resultant fonts
1050 PAPER#1;0:INK#1;7:CLS#1
1060 END DEFINE init
1070 DEFINE PROCEDURE dir_fonts
1080 DIR \ram1_tmp,dev$
1090 OPEN_IN#3;ram1_tmp
1100 INPUT#3,first$:REMark Move file pointer on.
1110 INPUT#3,second$:REMark Move file pointer on to the real font file names
1120 END DEFINE dir_font
1130 DEFINE PROCEDURE read_file_names
1140 LET counter=1:REMark Sets counter to read file names from d$() array
1150 REPEAT e
1160 IF EOF(#3) THEN file_end=counter-1: EXIT e:REMark file_end is the total number of valid
    files in this directory
1170 INPUT#3;file$
1180 lenght=LEN(file$)
1190 IF file$((lenght-3)TO lenght)=="_fnt" THEN d$(counter)=file$(8 TO lenght):counter=counter+1
1200 IF file$((lenght-3)TO lenght)=="_fat" THEN d$(counter)=file$(8 TO lenght):counter=counter+1
1210 IF file$((lenght-3)TO lenght)=="_chs" THEN d$(counter)=file$(8 TO lenght):counter=counter+1
1220 IF file$((lenght-4)TO lenght)=="_font" THEN d$(counter)=file$(8 TO lenght):counter=counter+1
1230 END REPEAT
1240 CLOSE#3
1250 END DEFINE read_file_names
1260 DEFINE PROCEDURE browser
1270 CLS
1280 counter=1
1290 display:REMark displays the first valid font file entry in d$() array
1300 REPEAT loop
1310 key$=INKEY$
1320 IF key$=="u" THEN counter=counter-1:REMark Moves counter file point up; ie up the font
    directory list
1330 IF key$=="d" THEN counter=counter+1:REMark Moves counter file point down; ie down the font
    directory list
1340 IF key$=="s" THEN EXIT loop:REMark exit loop and stop program
1350 IF counter<1 THEN counter=1:REMark Stops under run of the file array d$()
1360 IF counter>file_end THEN counter=file_end:REMark Stops over run of the usable files in the
    file array d$()
1370 IF key$=="u" OR key$=="d" THEN key$="":display
1380 END REPEAT loop
1390 END DEFINE browser
1400 DEFINE PROCEDURE display
1410 CLS
1420 CHAR_USE #1,0,0:REMark resets character set to normal for headings
1430 IF counter>file_end THEN PRINT "No More Founts":GO TO 1460
1440 PRINT#1;d$(counter)
1450 display_font
1460 PRINT#1:PRINT#1
1470 END DEFINE display
1480 DEFINE PROCEDURE display_font
1490 LET font_size=FLEN(\dev$&d$(counter))
1500 base=ALCHP(font_size)
1510 LBYTES dev$&d$(counter),base
1520 FOR wi=0 TO 3
1530 FOR hi=0 TO 1

```

```

1540 CSIZE#1;wi,hi:REMark Set character size
1550 CHAR_USE#1;0,0:REMark Sets character to normal for 'CSIZE' heading
1560 PRINT#1;"CSIZE ";wi,hi
1570 CHAR_USE #1,base,0:REMark Sets character to current loaded font
1580 FOR a=32 TO 191:PRINT#1,CHR$(a);
1590 PRINT#1:PRINT#1
1600 NEXT hi
1610 NEXT wi
1620 END DEFine display_font
32000 DEFine PROCedure update
32010 SAVE win1_Founts_Fountbrowser_bas
32020 PRINT "Update Complete"
32030 END DEFine update

```

# Programming in Assembler – Part 24

by Norman Dunbar

## Creating Your Own Windows With SETW

### Introduction

In this episode of the series, I shall be taking a small diversion into one of George Gwilt's utility programs. This one, SETW, allows you to interactively create windows for your applications. SETW then goes away and does all the hard work of setting everything up.

### Downloading SETW

SETW, and other useful utilities, is available from George's web site

<http://web.ukonline.co.uk/george.gwilt>

and from there I advise you to download the following three utilities:

SETW - setwp05.zip

EasyPEasy - peassp02.zip

GWASL - gwaslp07.zip

The latest version of GWASL is required to enable you to assemble PE programs created using SETW and using the EasyPEasy library files in peassp02.zip. As you will require these for the remainder of the tutorial then you should download them all now to save time later.

There are other files with similarly names but with a 'p' replaced by an 's' - these are the sources for the utilities and while educational, you don't need them.

The files are zipped up using the QDOS version of zip, so copy them from wherever you downloaded them to into your QL system (QPC etc) and unzip them using the QDOS version of unzip. There is one supplied with the C68 system and that works fine.

### Running SETW

In order to create correctly written assembly source for GWASL, we need to pass a single parameter to SETW when we execute it. The parameter is "-abin" with no spaces. This tells SETW that the code produced will be used to build a binary file rather than a relocatable one which will be subsequently linked with other relocatable files to produce the final binary.

We GWASL users don't have a linker so all our programs need to be self contained, or may include pre-assembled modules and libraries using the LIB command.

```
EX SETW ; '-abin'
```

The command above is all we need. If you do not have Toolkit 2, then the EXECUTE command from Turbo Toolkit can be used instead.

We will use SETW to create a file that we will use later on. It will be a very simple window with a single information window near the top and a single text object within the information window. Feel free to follow along on your own QL system as we go.

The program starts by opening a window as big as it can on your screen, it displays a few bits of information and prompts for the root name of the various files to be created.

For our example, we simply set the name to 'hello' - without the quotes. Type it in and press ENTER.

SETW will create three files when we are done. They will be created on ram1\_ (in my case) or wherever you have configured SETW to put them by default. The three files created will be:

**Ram1\_hello\_wda** - a file for use by George's TurboPTR utility. It is of no use to us and can be safely deleted when finished.

**Ram1\_hello\_asm** - a file for use by an assembler, in our case, GWASL, this is the file we will need.

**Ram1\_hello\_z** - a file for use with another of George's utilities, CPTR, a program to help C68 users write PE programs. Again, we don't need this file and it can safely be deleted.

## Entering Text Objects

The next screen that appears is titled 'ALTER TEXT' and is where we enter every text object to be used in our finished utility. We must be very careful here and not forget any because SETW creates code for what we enter and we cannot go back and add another if we forget one. (Well possibly we can in the generated assembler file, but I have not confirmed this yet.)

To enter your text objects, press the 'N' key to create a new text object and simply type in the required text. For our example window all we need is one single object containing the text 'Hello World' (without quotes) - for the main reason that this is how everyone starts to learn a new language! Press ENTER when you have entered the text.

In slightly more complicated programs, there would be a lot more text objects to enter, but for now, press the ESC key to exit from the ALTER TEXT screen.

## Entering Sprites, Blobs & Patterns

We don't need any sprites, blobs or patterns in this example, so simply press the ESC key when prompted for each of these.

## The Main Window

The next prompt is to tell SETW about the main window, how many windows are needed and so

on. In many cases the default is correct and all we need do is press ENTER at each prompt - however, make sure you read the prompt and think before pressing ENTER - once you have done so, there's no going back! (Ask me how I know!)

When asked for the number of main windows, accept the default of 1 by pressing ENTER.

When asked for the number of loose items, accept the default of zero by pressing ENTER.

When asked for the number of Information Windows, we will need one, so press the '1' key and press ENTER.

We are now asked to enter the number of information objects in each information window. We require one information object in our one single information window. Type '1' and press ENTER.

When asked how many Applications Windows you want, accept the default of zero.

Next we are asked to select a shadow size. I find a size of 2 to be adequate so, for now, type '2' and press ENTER.

For the border size choose a width of 1.

For all the prompts asking us to select a colour, select option 1 each time. Use the arrow keys to highlight the desired option and press ENTER to select it. We want "1. Default" for our colours. (I will explain the others later on in the series.)

Next we get to choose the sprite to be used as a pointer in the main window. I much prefer the standard arrow, so select it as above, and press ENTER. If you wish, you can choose another sprite.

## Information Windows & Objects

Now that all the details for the main window have been entered, or default chosen, we get to enter the requirements for each (or in our case, one!) Information Window.

First of all we need to enter the border width, I use a width of one pixel for all my programs. Type '1' and press ENTER.

Next we need the border colour, as before, select "1. Default" and press ENTER.

Select the default again for the paper colour.

We are now asked to select a type for our information objects for this information Window.

As we only entered a single information object way back at the beginning and that was a text object, we should select "text" and press ENTER. Other object types would be available if we had entered any sprites, blobs or patterns.

Next we see a window appear with the list of (one!) text objects. As there is only one, it has been highlighted for us, so simply press ENTER to select it.

Select the default colour again.  
When asked for the character sizes for X and Y for this text object, select zero for both.

## Interactively Sizing The Window & Contents

Now the fun begins! A window appears that allows us to interactively resize the main window and the information window we have created. Once done, we can position these items almost at will.

Looking at the window currently being displayed, the lower right corner shows the currently defined dimensions for the main window itself. At the top left is an outline of the noted dimensions. We can use the arrow keys to change the dimensions - up makes the window less tall, down makes it taller, left makes the window narrower and right makes it wider.

Pressing the ALT key makes the change in size bigger. This saves wearing out your keyboard getting the window to the size you would like!

For this demonstration program, we require a window size of 200 wide by 100 deep. So use the ALT and arrow keys to make the dimensions 200 wide and 100 deep. When the desired dimensions have been achieved, press ENTER.

We are now asked if a variable window is to be created, this will be covered in a future tutorial so for now, type 'N'. (There is no need to press ENTER.)

Next we need to set the origin of the window. Again the arrow keys move things around and the ALT key makes the movements bigger. For the demonstration, set the origin to 50, 50. You will get a rough idea of where the origin will be as a small dot moves around the screen under the control of your arrow keys. Press ENTER when the origin is where you would like it to be.

Next up, we get to size our information windows, or window in our case! I have decided to make it slightly wider than the space required for the text object. That itself is 12 characters of 6 pixels wide or 72 pixels in total. So, I like to have a bit of leading and trailing space in my information windows, so using the arrow keys and ALT as before, resize information window number one to be 74 wide by 12 deep. You may choose a different dimension if you like, but it will need to be a minimum of 72 pixels wide to hold all the text.

The program starts off in position mode rather than in size mode. You may need to press F2 to toggle between the two modes. Check the prompt on screen for advice about which mode you are currently in.

Once you have the desired size, press F2 and move the window to a position of 62 across by 2 down. If the information window size plus the position causes it to extend off the edge(s) of the main window, you will not be allowed to position it where you want to. In this case, toggle between size and position with the F2 key until you have it correctly sized and positioned.

Press ENTER when done. This takes you now to the sizing and positioning of the information window object (where the actual text object will be placed). If you remember the text object is 12 characters or 72 pixels wide, so we need an object big enough to take that plus a little space at the beginning and end. As I like a couple of pixels either end of my objects, set the information object to be at position 4 across and 1 down. Press ENTER when satisfied.

That's it for our little test window. SETW now displays some information about the files it created and after a pause, or when you press a key, it will cycle through all the main windows we created - one in our case - and display them on screen as they have been defined.

At this point, there's not much we can do if it all went horribly wrong. We simply have to start again - or get down and dirty in the generated assembly file! Press ENTER to exit from SETW.

On ram1\_, in my case, we now have the files that SETW generated for us. We are only interested in the hello\_asm file and can happily delete the others. Feel free to examine the generated file in an editor and compare what has been created with the previous articles where I explain what the individual bits of a WMAN window definition are.

The assembly source file generated is not able to be assembled as it is and then run, it has no code in it to make it a correctly functioning QDOS job. That comes later.

Until next time, feel free to generate more windows of your own and get to know George's SETW utility - we will be using it in future articles in this series.

## Coming Soon...

Next time, we will take the file we created with SETW and feed it into another of George's utilities, EasyPEasy, which tries to make coding for the PE much easier. Until next time, happy windowing.

# QL Firmware Bugs Myths - Part 1

by Tony Tebby

There is a recurrent and repeated myth about the bugs in the QL firmware.

The most comprehensive expression of this myth that I can find is:

*"The QL was launched, for delivery within 28 days, on the 12th January 1984, but 28 days later, the firmware was still incomplete and shipping did not start until April 1984. The first QLs were plagued by a number of problems, particularly bugs in the QDOS operating system and SuperBASIC which led to multiple releases of the firmware. Early production QLs were shipped with an external 16 kb ROM cartridge (infamously known as the 'kludge' or 'dongle') containing part of the firmware until the QL was redesigned to accommodate the necessary 48 KB of ROM internally, instead of the 32 KB initially specified. The first stable firmware version, JM, was released in Autumn 1984, six months after the first machines were shipped, although most of the bugs were not fixed until April 1985 with the release of version JS."*

Like the best myths, there are grains of truth in this story, but the reality is covered by a thick accretion of unfounded rumours, distortions and downright lies.

## The background

I said I would never do it, but apparently, never is about 25 years. The story of the QL is the story of the ZX83, the SuperSpectrum and the LC3. I must, therefore, place the foundation stones by setting out the background to the QL development.

But why do it when there are well researched histories such as the "must read" by Ian Adamson and Richard Kennedy ([http://www.nvg.ntnu.no/sinclair/computers/ql/ql\\_sst.htm](http://www.nvg.ntnu.no/sinclair/computers/ql/ql_sst.htm))? Partly because I need to condense the story a bit, partly because these stories skirt over some grey areas that I can fill in and partly because they were influenced by a long campaign of disinformation by Sinclair.

Quotes from the Ian Adamson and Richard Kennedy history are marked (A&K).

## The state of the firmware after launch

The firmware was ready for delivery long before the hardware.

1. The firmware was complete to the original specification well within 28 days of the launch and testing started as soon as prototype QLs were available (with exception of one major glitch when the firmware requirements were changed dramatically in December 1983, the firmware had been on standby for imminent launch since late 1983).
2. The extended firmware was tested and ready for ROM in March 1984 (version JM) well before the first QLs were shipped and this set the standard until the release of the MG ROMs for the foreign language versions in 1985.
3. All QLs had always had 48k ROM space reserved for firmware. From the first full prototypes onwards, the QL had two internal sockets decoded for 64k mask ROMs and could take one 64k ROM or any combination of 28 pin 8k, 16k or 32k ROMs. The dongle was purely for show. The journalists really should have noticed this.
4. JM version was shipped from late June 1984 although a number of QLs were shipped with the earlier test version AH after this!
5. JS was a development version that should never have been shipped, but, owing to internal problems at Sinclair, it appears that the TB source had gone missing. When some small patches for the US version had to be made, the only version available was JS which had a large number of untested extensions to SuperBASIC, a few minor bug fixes and a number of harmless fiddles to the Microdrive routines.

The little bit of truth is that, although JM was ready for ROM well before the first QL was shipped, all QLs before build D07 (seventh Dundee build), and a fair number after, were shipped with pre-test and test versions of the firmware, mostly on three 16k byte EPROMs.

When machines with dongles were returned for a "firmware upgrade", these "pre-production" machines (up to D05) were not upgraded, they were scrapped and replaced by "full production" QLs with a hardware build D06 or later and AH (test) or JM (release) versions of the firmware.

## The state of the hardware after launch

If the firmware was fit to be released soon after the launch in January 1984, what was the state of the electronics? Some journalists suspected that there was no prototype – as far as I am aware, they were right. As far as production QLs were concerned it was to be 9 months before build D14 (issue 6 PCB) QLs were shipped. This is a critical point in the delivery of QLs as the issue 6 PCB in the D14 build corrected a serious electronics design problem that had been identified in 1983 and had a workaround for a related electronics design problem discovered in February 1984. From D14, hardware modifications were made to improve production yields and reliability, not the basic design.

The original design for the ZX83 was a highly integrated "glue" and peripheral system using just two custom chips. A "quantum leap" in computer design by comparison with the rather old fashioned PC and its multiple standard chips. These two chips corresponded approximately to the custom chips in the ZX Spectrum and the Spectrum Interface 1. Studying an issue 6 PCB (or the circuit diagram) shows how far from this original design concept the QL really was. In place of the two custom chips, there are two ULAs, one HAL and one masked programmed 8049 microcontroller.

1. The ZX8301 "master chip" provided two functions.
  - A. The glue logic, interfacing to the MC68008 to provide handshaking, chip selects, RAM timing signals and the RAM bus arbitration between the display controller and the MC68008.

This chip had two errors, one of which was discovered when the first PCB layout was being prepared, the other during prototype testing after the launch.

The peripheral chip select (but not the ROM output enable) was derived from the RAM arbitration signal instead of the processor bus signals. This introduced timing jitter that seriously affected the Microdrives and network.

There was a timing fault in the RAM arbitration that caused 12 cycle delays (1.6CEs) when the chip warmed up.

These errors were mitigated, but not corrected, by bypassing some of the glue with a HAL from build D14 onwards.
  - B. The display generator was minimalist, even by Sinclair standards. The memory organisation of the display turned out to be sub-optimal for both hardware and software and the problems were compounded with a hasty patch. The output was designed for the built in flat screen display and was never adapted to standard monitors.
2. After the many shifts in specifications, the ZX8302 "peripheral chip" should have provided (in order of complexity):
  - A. The Spectrum compatible network I/O.

This was a simple one bit input and one bit output port. Within the ZX8302, it worked fine, but, because of the errors in the ZX8301, the bit timing was extremely erratic when operating at Spectrum speeds. The bit rate had to be changed to make the network work at all, making it incompatible with the Spectrum. When D14 QLs made it into production, it was too late to change the bit rate to match the Spectrum because it would not then work with earlier QLs.
  - B. The keyboard/joystick interface  

There were simply not enough pins available for a full keyboard/joystick interface in the ZX8302. The Spectrum had a very minimalist and electrically noisy solution that would have been possible if the ZX8302 had been in a 48 pin pack.

For a 40 pin ZX8302, a full solution could have been implemented together with the Spectrum compatible network using a standard PIA, on its own using two cheap MSI chips or even combined with a full games sound generator in an AY-3-8910 (see below).

Instead of which, it was adequately implemented in the 8049 IPC. A better implementation was later provided by the "Hermes" replacement chip.

C. The interrupt controller and status registers

Rather than each peripheral having its own interrupt and status registers, in the initial design they were combined into single registers for all sources. This was cheaper in hardware and more efficient in software. Unfortunately, with some parts of the peripherals being moved to the 8049, the status register bits for these moved as well and were no longer directly accessible. The result was functional but far from ideal.

D. The real time clock

This was the world's first battery powered real time clock with amnesia. The problem was that the reset circuitry did not take account of the MC68000 series processors' propensity for making arbitrary bus accesses during reset. This was a known "feature" of these processors that was, apparently, not well enough known. The real fix would have required greatly expanded reset conditioning circuitry in the 8302 and a quick fix would be to add a dedicated reset conditioning chip. The production solution was to forget the battery.

E. Two RS232 ports (originally 1 RS232 port + modem)

This was nothing like the original plan. The original plan was for a built in modem (one output pin for seize and one signal pin) and an output only serial printer port (two pins).

The problem with changing over to two RS232 ports with handshaking was that it required a total of eight pins rather than four and the ZX8302 was running out of pins. It would have been possible to squeeze them in by changing the addressing method to use "standard" peripheral chip techniques.

An asynchronous serial port receiver is not only a classic example of logic design, well understood and documented, it is also enormously simpler than the Microdrive interface that made it into the ZX8302. It is, therefore, difficult to understand the logic of moving the serial port receivers to an 8049 microcontroller (where they did their job so badly that one company made its (not very large) fortune selling external adapters to transform QL RS232 ports into "real" RS232 ports). The two transmit registers were left in the ZX8302 with a common clock which meant that, unless you were using two devices at the same baud rate, you could only use one RS232 port at a time.

F. The Microdrive interface.

The design was based closely on the Spectrum Interface 1. This appeared to work on the prototypes. Much later, when all the other, more obvious, Microdrive interface faults had been corrected in production, the design was found to be a poor match for the real signal timing (rather than the theoretical timing) of the QL Microdrives which had a data rate 20% faster than the Spectrum Microdrives.

G. The sound generator.

The sound generator was not an original part of the ZX83; it was an additional feature that never made it into the ZX8302. The emulation in the 8049 was more than adequate as competition for the PC's beeper, but for Sinclair core market it was definitely sub-standard.

*"The beeping noises it makes are more variable than the Spectrum's, but just as useless"* (A&K). It was no match at all for the AY-3-8910 series used by competing manufacturers (and built into the later Spectrum 128).

A pin count shows that all of this functionality could have been incorporated into the ZX8302 with at most three extra MSI chips or one standard peripheral chip, in either case providing significantly better performance at lower cost than the solution delivered with an 8049.

So, something had gone wrong: a major part of the "highly integrated glue and peripheral system" had been dumped onto a hopelessly underspecified 8049 and parts of the ULAs were dysfunctional. The level of integration (four custom chips) was lower than that of the Spectrum released 2 years earlier.

But what was the state of the hardware immediately after the launch? Sinclair is not a company to change a product in production for no good reason. Given that the fundamental problems fixed in D14 had been known from the first prototype build, it is reasonable to assume that the problems that were fixed in the first 12, nearly twice monthly, production build changes were more serious than those known faults. The state of the hardware when the first QLs were delivered was not, therefore, very good.

QUO VADIS  
DESIGN

Independent Information  
Technology Services

www.ql-qvd.com

QL/QDOS/SMSQ/E Software

QUO VADIS  
DESIGN Independent Information  
Technology Services

QL/QDOS/SMSQ/E Software

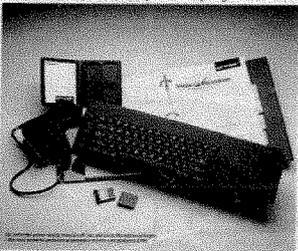
Home Products Support Company Contact

Welcome

Quo Vadis Design sells software for the Sinclair Quantum Leap computer (QL) and variants including a new OS called SMSQ/E.

The QL is a computer in its 25th year Anniversary.

The Sinclair QL - a quantum leap in personal computing.



Software emulations of the QL now exist which can run on a PC/Mac with Windows/Linux or Mac Operating systems.

News

- # QVD QL News Blog - keep up to date  
News Blog  
24/02/2009
- # Quo Vadis Design Website Launched  
01/02/2009

FEATURED PRODUCT



BUY NOW!

Copyright © 2009 Quo Vadis Design. All Rights Reserved.

[Home](#) | [Products](#) | [Support](#) | [Company](#) | [Contact](#) | [Privacy](#)

Bruce@ql-qvd.com

Quo Vadis Design  
38 Derham Gardens  
Upminster  
RM14 3HA  
UK

Tel: +44 (0)20 71930539  
Fax: +44 (0)870 0568755

Special Offers available from  
Jochen Merz Software for its  
25 years in QL Trading

Check the QL News Blog on  
our website for updates.  
[www.ql-qvd.com/blog](http://www.ql-qvd.com/blog)

**QL Today** The QL magazine  
for all QDOS, QL,  
SMSQ ... users!

Subscriptions taken online

David Karlin, in an interview in PCW celebrating the 25th anniversary summed it up very neatly: "Everything worked – sort of. Only 'sort of' was not good enough."

It would be simple to ascribe the problems to a clash of philosophies and the fanatical "not invented here" attitude of the "old guard". An AY-3-8910 would have provided far more capability, at lower cost, than the 8049. But the AY-3-8910 was a standard chip and I do not believe David Karlin would ever have managed to slip that past the old guard.

But there was far more than that.

## Flashback - Shifting targets translate into shifting specifications

The target for delivery was 1983 (hence the name) and this never shifted, it was simply missed. The other targets were for capability (what it should do) and market (who is going to use it). They shifted a lot.

David Karlin was quoted earlier in PCW as saying "there was a spec, but this was modified almost every day". Sir Clive put it slightly differently "The project started off in a totally different fashion, and then diverged from what I originally wanted". He blamed "the engineers".

There were target shifts and David Karlin was not the only one to blame these for the delay to, and ultimate failure of, the ZX83 project. But how many significant shifts were there, who was responsible for these shifts and how many of the real problems of getting the QL into production can genuinely be ascribed to these shifts?

On the software/firmware side, there were a number of shifts that had a definite impact on the software/firmware development time, size and performance. These shifts should also have had an impact on the electronics development time and performance but chronically bad project management turned what should have been manageable hardware design delays into catastrophe.

### Base camp 1 - The time line

The starting point is late 1982, when Sir Clive decided to build the next Sinclair computer around the MC68008. The outline concept of the product evolved over the next few months up until about March 1983 as more and more features were added, largely to distinguish it from the Spectrum whose sales showed no signs of flagging and for which add-on hardware was in the pipeline. The ZX83 was to be a new product in a new market. There was a target of delivery within 9 months (end of 1983) although Jim Westwood, Sir Clive's right hand man of many years standing, pointed out that Sinclair had never brought a computer to market in under a year and had never undertaken such an ambitious project. It is amazing how people laugh at the voice of experience.

### Base camp 2 - The product

The outline concept was a "portable, battery-powered machine with the famous flat-screen display, using twin Microdrives for storage, and incorporating a modem for communication via the telephone system" (A&K). This machine was targeted at the business market and ICL contributed to the hardware development costs to use the technology in their desktop OPD.

Both operating system development and electronics design started on this basis.

The electronics design started with the MC68008 microcomputer and 64k of RAM divided equally between a bit mapped display and the programs. Peripherals were, at this stage, a bit peripheral to the concept.

On the software side, I added a bit of flesh to the outline specifications for the operating system (a UNIX that works) and set about finding a company who would be able to produce such a system in 6 months (GST 68KOS). That settled, I started working on a "backup" operating system (6 months later, the first versions of 68KOS and Domesdos were tested head to head and Domesdos was selected – I came in for a certain amount of harsh criticism for spending so much of Sinclair's money on software that was not actually used. I would, however, not have been able to take the technological risks that I did if I had not been confident that, however badly I screwed up, GST would produce the worlds best multitasking workstation operating system on time). As the intended

operating system was being developed elsewhere, to start off with I was under no pressure, I could afford to take the time to investigate side avenues such as interesting experiments with sheer black lingerie and black latex (but that is another story).

Meanwhile, someone had managed to hook up a flat screen display, complete with anamorphic magnifying glass, to a spectrum. Although the spectrum only had a 32 x 24 character display (well below the established VT52/VT100 standard of 80 x 24 characters for business use) the characters were barely readable. I set about trying to do something about this. The way out seemed to be to use proportionally spaced fonts. The characters should not only be more readable but also you could pack more into a line, maybe up to 60 ens per line.

### **Shift 1 - The SuperSpectrum**

The decision was taken to leverage the custom chips in design for the ZX83 to build a Super-Spectrum: a machine in the Spectrum line with an MC68008 processor and a more advanced BASIC. The development costs would be very low as the electronics would be taken from the ZX83 effectively unmodified. I recruited Jan Jones to bring the Spectrum BASIC "up to date" to compete with the BBC computer, the Spectrum's BASIC having been blamed for the loss of the BBC contract to Acorn.

This did imply changing the ZX8301 design slightly as the first cut of the display controller had 512 x 256 pixels in four shades of grey. A second "Spectrum compatible" display mode of 256 x 256 pixels in 8 colours (or 8 greys) was added. This might have been a minor change in hardware, but it was a major problem for any software trying to handle the display.

The ZX83 first cut display processor had a very quirky display mode. This was a real pig to program, but time was short so we were stuck with it.

However, when the 4 bit, 8 colour Spectrum compatible mode was added, it was all too obvious that a 4 bit quirky mode would be far too expensive in hardware, so a hybrid packed quirky pixel mode was patched in. This certainly saved a bit of hardware development time, but it turned the pig into wild boar with a hangover.

The display change was not the only effect, however. The SuperSpectrum would also require better sound generation: a whole new function in the yet to be designed ZX8302. More hardware design work, and no extension of the schedule.

### **Shift 2 - ZX83 goes desktop**

The portability was lost. The portability was based on the same battery pack as the Sinclair Microvision 2700 (TV80). Unfortunately, back of envelope calculations had revealed that the ZX83 would have a (non-rechargeable) battery life of about 30 minutes. More detailed calculations showed that this was likely to be closer to 10 minutes with a Microdrive running.

The battery was dropped.

This had little incidence on the electronics or software/firmware but it dramatically changed the target market so the sheer black lingerie was thrown out at the same time.

### **Shift 3 - An office suite is bundled**

The Psion office suite (in development) was bundled with the ZX83. This apparently was Nigel Searle's decision.

The first effect was that the ZX83 had to be shipped in a version with more than 64k RAM. This implied adding a second /CAS output to the "glue logic". This was probably a good idea.

There was no immediate effect on the software/firmware.

### **Shift 4 - Built in display is dropped**

The flat screen display was lost due to production problems. This was greeted with relief by all concerned with the ZX83. There never had been any possibility of the Psion suite being usable on the flat screen – one (Sir Clive's pet) or the other (Nigel Searle's pet) had to go.

This should have had an effect on the electronics as the display timing had been designed for the flat screen and the number of lines and refresh rate were unsuitable for any standard computer monitor at the time and it would over-scan on a UK television standard monitor. There was no slack

in the design schedule, so no changes were made. The effect of this shift was that the ZX83 now had a video output unsuitable for any potential display device.

#### **Shift 5 - CGA compatible display required**

Another shift was made possible by the loss of the flat screen. The design of the Psion business suite was based on the PC basic display mode (CGA). This required a 80 x 25 cell display, with 8 x 8 pixel cells (640 x 200 = 128,000 pixels). At this stage a critical decision was required for the electronics design: go with the Psion requirements and adopt the "industry standard monitor" or stick with the existing "TV compatible" 512 x 256 (131,072 pixels) and fix the over-scan. The decision was to do nothing so there was no impact on the electronics.

There was, however, a significant impact on the software/firmware. The proportional font handling was dropped and a fixed cell size implemented to provide a near CGA compatibility of 85 x 25 with 6 x 10 cells.

#### **Glitch 1 - The preferred ASIC supplier goes bust**

The optimistic schedule for completion of the hardware depended very strongly on a single ASIC foundry. This foundry provided not only an advanced technology, but the promise of 2/3 week turn-around on prototypes. The foundry went bust. This meant that the design up to this point might need to be reconsidered and the prototype turn-around was extended from two weeks to two months.

At this stage, the project schedule was no longer even vaguely realistic. The project should have been cancelled, rescheduled or reoriented. There was a categorical refusal to consider any of these options.

#### **Shift 7 - The modem falls off**

The modem had been announced for the ZX83 and it was a critical feature of the OPD and, therefore, was part of the contract with ICL. A ULA, however, is not an ideal technology for it and there was simply no time for the design work. The feature just disappeared.

This shift was not due to specification changes, it was forced by external factors and project management failure.

#### **Shift 6 - The sound generation disappears as well**

The modulator part of a modem is just a fancy real-time sound generator. The Spectrum beeping noise had always been a bit embarrassing. The ZX83 / SuperSpectrum should have been able to do much better with an integrated sound generator based on the modem circuit. But the modem was lost and with it the possibility of complex sound generation for the SuperSpectrum.

#### **NIH 1 - The keyboard man cometh**

We received a representative from the "world's largest" keyboard manufacturer with his wares. One was a bare keyboard (no electronics) with an excellent feel and a very competitive price. This was passed to production who claimed that, with 2 million Spectrums built, Sinclair was the "world's largest" keyboard manufacturer and that they could make it much better and cheaper. In any case, the keycaps were the "wrong shape".

6 months and £100,000 of mould tooling costs later, Sinclair had a sticky, unnatural keyboard that cost 50% more than the opening offer from the other world's largest manufacturer. "Not Invented Here" reigns supreme.

#### **Shift 8 - The SuperSpectrum is dropped**

The ZX83 hardware design was not really advancing.

The serial ports, the sound generator and the keyboard interface were still missing from the ZX8302 design. It was late summer, a new estimated lead time of 11 weeks for prototype chips meant that by the time all of these functions had been designed, the prototype chips would not be ready for a launch in time for Christmas.

The decision was taken to drop these functions from the ZX8302 and implement them in a micro-controller. In the QL Technical Manual, David Karlin makes it clear that this was not a decision taken on technical grounds: *"IPC communications is a very slow process and excessive use of the IPC ...*

will cause very high processor overheads". This design change was forced by a refusal to reschedule the project in response to repeated shifts in direction and external factors. New recruit Aaron Turner was given the unenviable task of trying to make this work and interfacing it to Domesdos. I think he did quite a good job. To quote Johnson it "is like a dog dancing on its hinder legs. It is not done well; but you are surprised to find that it is done at all".

The loss of critical functions from the ZX8302 was compounded with the lack of a display mode that could display a full Spectrum screen on a television. Any idea of basing the SuperSpectrum on the ZX83 chip set was now quietly abandoned.

## **NIH 2 - The floppy disk man cometh**

We received a representative from a leading Japanese electronics manufacturer with his wares. One was a 3.5" floppy disk drive. It would be nice to think that his offer was rejected on reasonable grounds. There were reasonable grounds. A Microdrive cartridge was smaller than a 3.5" floppy disk. A Microdrive was cheaper than a floppy disk drive, but this floppy disk drive was cheaper than two Microdrives.

So was it a considered decision? No. It was pure "Not Invented Here". At this stage, Sinclair had been developing the Microdrives for two years, and was not going to give up on an opportunity to ship them. Instead, the Microdrives would be "overclocked" to improve the performance.

## **Glitch 2 - The PCB don't fit no more**

There was an intriguing statement in PCW April 1984 "the most difficult problem the team encountered was how to assemble the case". Assembling the case was not a problem. Designing the PCB to fit the case was. Tooling for large plastic mouldings is on a long lead time, so the case dimensions and connector positions were fixed long before the hardware design was finished, based on an initial guess. Unfortunately, the form factor (long and thin) was not ideal for noise or reliability. Adding another 40 pin pack for the 8049 not only made it longer and more susceptible, but also added another noise source (the crystal oscillator).

## **Diversion 1 - The Low Cost Colour Computer**

The LC3 was not a SuperSpectrum, it was more "Martin Brennan fights back". With the SuperSpectrum (a sort of anti Acorn / BBC machine) no longer even on the distant horizon, a demonstration/development unit for an alternative evolution of the Spectrum line was designed and built.

*"This cheap and powerful machine, with superior display handling to that of the QL, was one of the topics discussed at a planning meeting in November 1983 ... The LC3 project was chopped ... Further development of the LC3 would be costly, and the view was being sustained at this time that the QL was almost ready for production."* (A&K)

Who was sustaining the view that the QL was almost ready for production? The LC3 only existed as an emulation using standard SSI and MSI chips. But the ZX83 was not significantly further on: prototype custom chips were in the pipeline but there was still nothing that you could "hang your hat on".

Of course, the LC3 was much simpler than the ZX83 (ROM and NVRAM cartridges instead of Microdrives), but that is the very reason why it might have been possible to get it into production sooner than the ZX83. Of course, the LC3 could not have been used to attack the business market, but by that stage, I could no longer imagine that the ZX83 would ever be able to do that either unless there was a fundamental re-thinking of what it was about – and I was not the only one.

ICL had put up a ridiculous sum for access to the ZX83 technology. I can only assume that the cost of getting out of the OPD contract was so high that it totally distorted any decision making at the top of Sinclair.

So, where there should have been a radical shift, it was just "Carry on Regardless". (Gerald Thomas and Peter Rogers, 1961) but not as funny.

With the loss of the SuperSpectrum and the rejection of the LC3, Sinclair no longer had any planned upgrade for its core hobby / games market.

## Shift 9 - The ZX83 becomes a slightly Spectrum compatible

The loss of the upgrade path for the Spectrum, led to a push towards making the ZX83 more of a games machine: a complete U-turn on the decision to abandon the SuperSpectrum. Standard PAL encoder/modulator circuitry (well tried and tested on the Spectrum) was added, and some of the keyboard lines were mapped onto two joystick ports requiring changes to the PCB layout, but no changes to the basic design. *"Adding a television outlet was as much a response to this [games machine mindset] as the fact that Sinclair Research hadn't produced a monitor to go with the QL or arranged an OEM deal with someone who manufactured a monitor"* (A&K).

It sounds easy, doesn't it. Just a few changes to the PCB. That adds another crystal oscillator for the colour signal, generating more noise, while the new circuits and connectors make the PCB longer. The existing connectors have to be moved to line up with the openings so longer tracks are required, creating more noise and crosstalk, taking up more board space and making the board even longer, which makes the tracks even longer .... and the only place you can fit the modulator is right next door to the Microdrives - ouch!

## Shift 10 - The ZX83 becomes the SuperSpectrum

The loss of the upgrade path for the Spectrum, combined with the fact that Jan Jones was now working on software for a machine that was now dead, led to the almost inevitable, almost irresistible, "put SuperBASIC on the ZX83" wave that rolled over the project. This became irresistible when marketing announced that if the QL did not have a built in BASIC, then they would have to reduce their sales forecasts by a factor of five.

*"The late decision to hedge the bets yet again and include a BASIC was not only a failure of nerve in the concept, but productive of more problems"* (A&K). The paragraph following this the A&K article is one of the few parts with factual errors (at least I think they are errors).

This change was traumatic for the software/firmware, but not for the reasons A&K gave.

\* Yes, adding the full SuperBASIC in firmware would inevitably take the take the firmware over 16k, but this should have been simply a question of procurement cost: the electronics design allowed for ROMs up to 64k byte. More ROM, more cost: a simple equation.

\* Yes, I handed in my notice, but this was 5 days after the launch when I discovered that 28 day delivery was being offered. This had nothing to do with SuperBASIC. The decision to "downgrade to SuperSpectrum" was made more than a month before the launch.

Unfortunately there was a nasty case of hedging bets. The current version of the prototype PCB design had only one ROM socket (OS only) and was in the process of being revised to accommodate the television modulator and joysticks. Given the state of development it would seem reasonable to accept the cost of a 64k byte ROM and just get on with finishing the machine off. But it was decided that a 64k byte ROM would be too expensive so the QL would be shipped with only the core SuperBASIC functions in a 32k ROM and the rest would be supplied either as a plug in ROM or on Microdrive.

But, just in case, a second socket would be added to the PCB so that the whole of SuperBASIC could be delivered internally using a 32k byte ROM and a 16k byte ROM. It was also suggested that the second socket could take a 32k byte ROM with a compressed copy of the Psion programs! This extra ROM socket took more space on the PCB, making it even longer, making the tracks even longer ...

Incorporating SuperBASIC was, however, a far more serious proposition than just needing a lot more ROM.

This was the ZX83 and the ZX83 was a computer destined to be released in 1983. For some time, I had been on "four week standby". Prototype ZX83s, ready to go into production, were expected "any time right now".. As soon they appeared, I would have about four weeks before pre-production units rolled off the line. One week to finish off any changes in progress, one week to debug the system and two weeks lead time on small quantities of ROMs.

***As I was writing this, it occurred to me that this might seem strange to some people. David Karlin produced a functional specification of the hardware and then he developed the electronics and I developed the operating system and drivers using this specification. Neither of us thought that it***

**was odd to expect the operating system and drivers to be fully functional two weeks after the first meeting of hardware and software. 25 years later, with all the sophisticated software development tools now available, would anyone be rash enough to expect the system to be working in 6 months? How the world has changed!**

At this stage, SuperBASIC was several months from completion. In particular, Chris Scheybeler at GST was developing the graphics routines but these would need interfacing. Spectrum BASIC wrote graphics directly to the screen, but in a multitasking environment, graphics operations have to be managed by the operating system to prevent conflicts. The graphics had to be transferred to the console driver: I off-loaded that job to Aaron Turner (again).

Meanwhile Jan Jones and I tackled the major problem.

The internal structure of SuperBASIC was modelled on the "I own the world" Spectrum BASIC philosophy, which was fundamentally incompatible with sharing the machine with other tasks. In normal times, the solution would have been simple: just go through SuperBASIC and identify every sequence of instructions that allocated space, released space or made assumptions about contiguity and then rewrite all those bits. It would take no more than a few weeks to adapt the interpreter and then maybe another month or so to complete it if a few corners were cut.

I made a terrible mistake. With the expectation of a launch before Christmas, less than 4 weeks away, I went for the quicker approach of patching SuperBASIC and adding a special job category to Domesdos to emulate a special Spectrum compatible environment for just one job.

OK, we had SuperBASIC running under Domesdos in about a week, but the cost was high. The integrity of Domesdos was compromised and the interface was never really clean. That only 3 or 4 bugs ever showed up in the interface is more due to luck than anything else.

I should have said "Fine, we will get on with re-writing SuperBASIC, it should take about a month, and then another two to three months to add all the Spectrum BASIC functions" and then made hasty exit. It is easy to be wise after the event.

#### **A divergence**

*I am quoted as having said "Communications were deliberately distorted. If I talked to marketing, they would describe to me a product I'd never heard of. They said, 'Well, give us the finished product in a couple of weeks' time and we'll review our position.' I said, 'But it's not going to be working for six months!' They say, 'But we're starting the ad campaign in two weeks' time, placing the ads.'"*

I am not sure that this was exactly what I said (I tended to deliberately avoid the press and if I did get cornered, I was usually put in the position of confirming or denying hearsay) or whether that is exactly what happened – I think it rolls a number of occasions into one.

The occasion two weeks before the launch was when I had no more development work planned and David Karlin was waiting for the next delivery of ULAs. We went to see the marketing director. As I remember it, I might be completely wrong, it was us who told the marketing director what the ZX83 would be like, and it was the marketing director who was horrified. The advertising had been placed for two weeks ahead for a completely mythical machine.

At that time, I think that David Karlin and myself were both in agreement that committing to delivery in less than 6 months would be unwise.

#### **No more shifts - The stake in the ground**

*"At some point in a project that has been going on for 18 months, you have to put a stake in the ground and say you are launching the product on such and such a date. If you wait for the guys who are working on the product to tell you when it will be finished, you will wait for ever."*

This quotation, in *International Management*, from Nigel Searle has been rolled out a number of times. If the delays to the ZX83 had been caused by "the guys who are working on the product" fiddling around adding bells and whistles, then he would have been entirely justified. Given the real state of affairs, with the bells and whistles falling off like autumn leaves and the hardware fundamentals as sound as a sub-prime mortgage, it only shows his inability to grasp the state of the project.

A more realist attitude would have been "At some point in a project that has been going on for 18 months, you have to put a stake in the ground and say that if you cannot show a working prototype that marketing thinks is saleable, the project is dead".

In reality, the the project had only really started 9 months before and the target shifts had not caused any significant delays to the electronics design simply because the schedule had not revised to allow for the shifts. This resulted in an accumulation of design failures resulting from attempts to "cut corners", to meet increasingly unrealistic project milestones, effectively delaying the project even more. The shifts also had an effect on the software. In particular, there was the decision to build in, at very short notice, a major package (SuperBASIC) that was not only incompatible with the operating system, but also scheduled for completion months later.

The shifts had also caused what I considered to be terminal damage to the product.

*"So the downgraded ZX83 project lurched along in what one source called the 'disorganised shambles' that was Sinclair Research at the time. The absence of a project leader, a board acting divisively and throwing up conflicting views masquerading as decisions, and the lack of co-ordination all compounded each other and combined with the absence of [Sir Clive] Sinclair from the R & D scene to produce a fiasco" (A&K). I could not have expressed it better myself.*

The ZX83 had turned into a product with no clear market and no clear function (and no working hardware in sight).

Just before the launch, David Karlin and myself received a draft of the press release. This was a typical "vapourware" announcement with no detailed information about the product and an estimated delivery date of 3-4 months – optimistic, but not impossible.

At the launch, the press release had been modified to promise 28 day delivery.

#### **Yet another shift - The extended SuperBASIC procedures are built in**

At the beginning of December 1983, the plan was to have a minimum BASIC built in to a 32k ROM and provide an extension to a full SuperBASIC, either on a ROM cartridge or on Microdrive. The SuperBASIC initialisation was, therefore, modified to allow additional procedures to be linked in. This was probably a good idea anyway, but David Karlin designed in a second ROM socket (either as a precaution or with considerable foresight) so there was no real barrier to shipping full SuperBASIC as the base configuration. Was there any sense in delivering SuperBASIC in two bits? There was a good reason: the base facilities could have finalised and tested by the end of January or mid February at the latest and the rest delivered as an add-on later, but the project was upstaged by marketing once more.

*"The QL manual handed out at the launch was a stop-gap construct leaning heavily on the Psion package's documentation, since at least the user's actions and their consequences could be described with some accuracy, even if they were not yet converted so that they actually worked on the QL. The SuperBASIC section of the manual was a confabulation of existing [Spectrum] facilities, hoped-for additions and some straightforwardly inventive writing" (A&K).*

So rather than following a project plan, the development of SuperBASIC became a matter of trying to fulfil expectations raised by a totally irresponsible launch. Not only was there a promise of 28 days delivery of something that did not even exist, the promised SuperBASIC as described in the manual was a product unknown to the developers!

To make life more difficult, the contents of the "manual" were kept secret from the software developers "to avoid distracting them". With every successive release of test versions with more and more facilities, a new wish list was sent back, without any indication of whether these wished-for facilities had been announced to the press or whether they were just someone's "good idea".

#### **And yet another shift**

The system initialisation procedures had a switch for 64k bytes and 128k bytes (or more) of RAM (32k and 96k program space respectively). The target since summer 1983 had been for an entry level machine with 64k bytes for £249/£299 and a real machine at £399. Unfortunately, when the Psion programs arrived, running a single application in 32k was simply out of the question. As either

the Psion contract required the Psion programs to be bundled with every QL or Sinclair had made a firm public commitment to bundle them, Sinclair was obliged to drop the 64k QL.

This shift did not actually affect the hardware or software but it made a nonsense of the whole development. Soon after the price had been agreed for the MC68008 in December 1982, Motorola informed Sinclair that they would be able to deliver full MC68000s at a lower price.

Using an MC68000 would have improved the performance by a factor of 3, that is three years of processor speed improvement at a stroke. The MC68000 was not used as it would have required two banks of RAM, two ROMs and a separate "glue" chip, which would have pushed the entry cost up. But when the QL was delivered it did have two banks of RAM and two ROMs and it soon had the separate glue chip as well, so it had all the entry cost of a full MC68000 with only one third of the performance.

Clearly, the right decision when the Psion applications were adopted would have been to cancel the MC68008 and go for an MC68000, accepting the slightly larger case and two month delay that would have been incurred.

So with all the shifting specifications, we end up with a machine that no-one can be proud of: the performance was compromised by the premature selection of the processor, the electronics design was compromised by the hasty addition of a microcontroller, the operating system was compromised by a bodged-in BASIC interpreter and the hardware was compromised by overclocked Microdrives, the ludicrously expensive, low quality keyboard and the serious lack of compatibility with standard peripherals.

So far, so good. The story of how and why Sinclair created the faulty firmware myth will be told in the next issue.

## Some non-Software Updates

by Jochen Merz

After having written so much in the past few issues, I thought I would keep you updated about the situation, but I will keep it short.

At the time I write this (end of August), the situation is as follows:

Roy has not paid me what I requested in May. I still delivered QL Today to all readers to ensure there was no disappointment.

Some renewers added a comment, that I can charge for the single issue in case Roy does not pay me. I will, of course, not do this - but I would like to thank you for the offer and the fairness.

I have written two further (real) letters to Roy ... no reply, as he seems to have forgotten his public promise to pay. No money has arrived in June, July or August so far! A quarter of a year. This is exactly what I experienced over all the years.

I chatted to Tony Firshman about this, and he asked Roy about three weeks ago. He forwarded me the reply, which said that he was too busy in the past few months and would do it after the weekend. Too busy for months to pay the debt. Wow! How many hours does this take, one wonders. And, of course, no money after this or any following weekend.

You can read my frustration out of these lines, can't you?

I am not familiar with this kind of problem, and I don't know much about English law.

Any help or suggestions would be very much appreciated. Please email privately to

**JMerz@j-m-s.com**

Now on to positive subjects: I would like to thank everybody for the congratulations regarding J-M-S's 25th. Especially Jon and Elisabeth from Switzerland for the card with the uplifting wishes - THAT was really appreciated and reminded me immediately why I like being part of the QL community.

I can't promise another 25 years, but I hope to be part of it as long as possible, and as long as there's interest.

But let's look forward to the next year first.

I think I have proved again that I try to be as reliable as possible, even though the QBranch problems were not caused by me.

Thanks to all the renewers from QBranch for putting their trust into us, the QL Today team.

We work hard and ensure not to disappoint you!

# Fun with Fonts – Part 4

by Dilwyn Jones

## Angled Text

The standard QL printing routines only do horizontal left to right text printing. Here is a useful little routine which can print text at an angle.

HELLO

Figure 8 shows a sample output from the routine. The demonstration rotates a string around a notional axis or origin.

Figure 8 - sample angled text string

```
100 REMark angled text printing
110 WINDOW 512,256,0,0
120 PAPER 0
130 CLS
140 col%=1
150 :
160 REPEAT loop
170   IF INKEY$ = CHR$(27) THEN EXIT loop
180   FOR size = 1 TO 4
190     FOR a = 0 TO 359 STEP 5
200       OVER -1 : Angle_Print #1,'HELLO',size,RAD(a),col%,0,256,128 : OVER 0
210       PAUSE 2
220       OVER -1 : Angle_Print #1,'HELLO',size,RAD(a),col%,0,256,128 : OVER 0
230     END FOR a
240     col% = col% + 1 : IF col% > 7 THEN col% = 1
250   END FOR size
260 END REPEAT loop
270 STOP
280 :
290 DEFine PROCedure Angle_Print (channel, str$, size, angle, colour, fat_font, x, y)
300   REMark older ROM versions can only handle 9 local parameters!
310   LOCAL w%,xoffset,yoffset,char,addr,xx,yy,down,byte,bit%,across
320   :
330   REMark fat_font=1 means 8 pixel wide (CSIZE 1,0) font
340   IF fat_font = 0 THEN w% = 6 : ELSE w% = 8
350   :
360   REMark separation between pixels at this angle
370   LET xoffset = size * SIN(angle)
380   LET yoffset = size * COS(angle)
390   FOR char = 1 TO LEN(str$)
400     LET addr = CHAN_L(#channel,42)+10+(CODE(str$(char))-32)*9
410     xx = x + ((char-1) * yoffset * w%)
420     yy = y - ((char-1) * xoffset * w%)
430     REMark all 9 rows down
440     FOR down = 0 TO 8
450       LET byte = PEEK(addr + down + 1)
460       REMark all 8 pixels (fat fonts) or all 5 pixels (standard fonts) across
470       bit% = 128
480       FOR across = 0 TO w%-1
490         IF byte && bit% THEN
500           BLOCK #channel,size,size,xx+(across*yoffset)+(down*xoffset),yy+(down*yoffset)
510             -(across*xoffset),colour
520           bit% = bit% DIV 2
530         END FOR across
540       END FOR down
550     END FOR char
560 END DEFine Angle_Print
```

Listing 10 - Angled text printing

The core routine is the procedure called Angle\_Print which takes 8 parameters:

channel            - SuperBASIC channel number  
str\$               - the text string to be printed

**size**                    - 1 is csize 0,0 2 is csize 2,1 and so on  
**angle**                   - in radians from the horizontal axis  
**colour**                   - INK colour  
**fat\_font**                - 0=standard font, 1=8 pixel wide font  
**x,y**                      - pixel co-ordinate of top left origin of character

Note that the routine has more than 9 local parameters, which older ROM versions may not allow.

The routine operates by working out where the top left origin of the character lies and locating the address of the character definition for each character of the string. To keep the routine brief, it only works for the lower font, although it should be a minor job to check which of the two system fonts the character codes belongs in and use CHAN\_L(#channel,46) instead of CHAN\_L(#channel,42) if the character code is in the upper font range.

Lines 380 and 390 work out the origin of each character by using SIN and COS of the rotation angle to work out where the origin of the character is rotated to.

The fat\_font parameter is 0 (for a standard 5 or 6 pixel wide character font) or 1 (for an 8 pixel wide font). This is used to set the variable w% which determines how many pixels across each font we need to check. The BLOCK command in line 470 plots each pixel of the character as long as the pixel is set (1). If unset (0) nothing is drawn, so in effect this is like printing using OVER 1. You will need an ELSE..BLOCK in paper colour to fill in paper pixels, though in this kind of drawing I find it less common to need paper.

Each pixel is represented by a horizontal block of the size indicated, e.g. size=3 uses a 3x3 horizontal block per pixel. This can produce some strange stepping effects at some angles because the block is not drawn at the same angle as the text, but it is generally not too bad. If OVER -1 is used to XOR the character against itself to erase it when used for animation, you get some strange effects when pixels overlap due to rounding off overlaps, but the routine is good enough to demonstrate the principle.

Using the routine is just a matter of calling the Angle\_Print routine with the appropriate parameters. Lines 150 to 260 make text rotate around an origin by repeatedly drawing and erasing the string in 5 degree steps with a small PAUSE in between to reduce flicker. The actual way of calling Angle\_Print is shown in line 190 - this is all you need without the OVER commands) if you wish to use the routine to annotate a graph or just label something at an angle.

## Rotating Text

Listing 11 shows a very simple way of rotating text about the horizontal axis. It is not particularly effective as there is no perspective effect (text changing size and/or colour as it goes into the distance or to the foreground), but shows a very basic way to use SIN and COS to rotate text. It achieves vertical perspective, but not horizontal - you can experiment a little with the code to improve the width of each row of the text as it goes further back or further forward than the origin.

```

100 REMark spin text around horizontal axis
110 WINDOW 512,256,0,0 : PAPER 0 : CLS
120 REPEAT loop
130 FOR angl = 0 TO 359 STEP 5
140   OVER -1 : Enlarge #1,6,4,0,100,128,7,0,'HELLO',angl : OVER 0
150   PAUSE 3
160   OVER -1 : Enlarge #1,6,4,0,100,128,7,0,'HELLO',angl : OVER 0
170 END FOR angl
180 END REPEAT loop
190 STOP
200 :
210 DEFine PROCedure Enlarge (channel, wide, high, spaced, x, y, ink_colour, paper_colour, str$, angle)
220   LOCAL base1,base2,cde1,cde2,nc1,nc2,char,byte
230   base1 = CHAN_L(#channel,42): REMark address of first font
240   base2 = CHAN_L(#channel,46): REMark address of second font
250   cde1 = PEEK(base1)           : REMark lowest valid character 1st font
  
```

```

260 cde2 = PEEK(base2)      : REMark lowest valid character 2nd font
270 nc1  = PEEK(base1+1)   : REMark number of characters-1 1st font
280 nc2  = PEEK(base2+1)   : REMark number of characters-1 2nd font
290 FOR char = 1 TO LEN(str$)
300   cde = CODE(str$(char))
310   SElect ON cde
320     =cde1 TO cde1+nc1: addr = base1+2+(9*(cde-cde1)): REMark font 1
330     =cde2 TO cde2+nc2: addr = base2+2+(9*(cde-cde2)): REMark font 2
340     =REMAINDER      : addr = base2+2 : REMark default character
350   END SElect
360   FOR byte = 0 TO 8
370     row_value = PEEK(addr+byte)
380     across = x + ((6+spaced+spaced) * wide * (char-1))
390     IF spaced = 0 THEN
400       REMark ordinary fonts (bits 7 to 2)
410       FOR bit = 128,64,32,16,8,4 : Plot_Pixel
420     ELSE
430       REMark fonts spaced more widely bits 7 to 0, e.g. "fat" fonts
440       FOR bit = 128,64,32,16,8,4,2,1 : Plot_Pixel
450     END IF
460   END FOR byte
470 END FOR char
480 END DEFine Enlarge
490 :
500 DEFine PROCedure Plot_Pixel
510   IF row_value && bit THEN
520     REMark INK pixels
530     BLOCK #channel,wide,ABS(high*COS(RAD(angle))),across,y+(byte*high*COS(RAD(angle))),ink_colour
540   ELSE
550     over_state% = CHAN_B%(#channel,66) && 12 : REMark OVER details
560     IF (over_state% && 4) = 0 THEN
570       REMark only plot PAPER pixels if OVER 0
580       OVER #channel,0 : REMark cancel OVER temporarily
590       BLOCK #channel,wide,ABS(high*COS(RAD(angle))),across,y+(byte*high*COS(RAD(angle))),
        paper_colour : REMark paper
600       REMark restore OVER state for this channel
610       SElect ON over_state% : =4 : OVER #channel,1 : =8 : OVER #channel,-1
620     END IF
630   END IF
640   across = across + wide
650 END DEFine Plot_Pixel

```

#### Listing 11 - rotating text around horizontal axis

It is simply a variation on the text enlarging routine presented earlier, with a few SIN and COS statements thrown in to introduce a little bit of rotation effects.

The next routine is a variation which rotates a text string around the vertical axis instead. See listing 12.

```

100 REMark spin text around vertical axis
110 WINDOW 512,256,0,0 : PAPER 0 : CLS
120 REPeat loop
130   FOR angl = 0 TO 359 STEP 5
140     OVER -1 : Enlarge #1,2,4,0,256,128,7,0,'HELLO',angl : OVER 0
150     PAUSE 2
160     OVER -1 : Enlarge #1,2,4,0,256,128,7,0,'HELLO',angl : OVER 0
170   END FOR angl
180 END REPeat loop
190 STOP
200 :
210 DEFine PROCedure Enlarge (channel, wide, high, spaced, x, y, ink_colour, paper_colour, str$,
angle)
220   LOCAL base1,base2,cde1,cde2,nc1,nc2,char,byte
230   base1 = CHAN_L(#channel,42): REMark address of first font
240   base2 = CHAN_L(#channel,46): REMark address of second font
250   cde1 = PEEK(base1)      : REMark lowest valid character 1st font
260   cde2 = PEEK(base2)      : REMark lowest valid character 2nd font
270   nc1  = PEEK(base1+1)   : REMark number of characters-1 1st font
280   nc2  = PEEK(base2+1)   : REMark number of characters-1 2nd font

```

```

290 FOR char = 1 TO LEN(str$)
300   cde = CODE(str$(char))
310   SElect ON cde
320     =cde1 TO cde1+nc1: addr = base1+2+(9*(cde-cde1)): REMark font 1
330     =cde2 TO cde2+nc2: addr = base2+2+(9*(cde-cde2)): REMark font 2
340     =REMAINDER      : addr = base2+2 : REMark default character
350   END SElect
360   FOR byte = 0 TO 8
370     row_value = PEEK(addr+byte)
380     across = x + ((6+spaced+spaced) * wide * (char-1) * SIN(RAD(angle)))
390     IF spaced = 0 THEN
400       REMark ordinary fonts (bits 7 to 2)
410       FOR bit = 128,64,32,16,8,4 : Plot_Pixel
420     ELSE
430       REMark fonts spaced more widely bits 7 to 0, e.g. "fat" fonts
440       FOR bit = 128,64,32,16,8,4,2,1 : Plot_Pixel
450     END IF
460   END FOR byte
470 END FOR char
480 END DEFine Enlarge
490 :
500 DEFine PROCedure Plot_Pixel
510 IF row_value && bit THEN
520   REMark INK pixels
530   BLOCK #channel,wide,high,across,y+(byte*high),ink_colour
540 ELSE
550   over_state% = CHAN_B%(#channel,66) && 12 : REMark OVER details
560   IF (over_state% && 4) = 0 THEN
570     REMark only plot PAPER pixels if OVER 0
580     OVER #channel,0 : REMark cancel OVER temporarily
590     BLOCK #channel,wide,high,across,y+(byte*high),paper_colour : REMark paper
600     REMark restore OVER state for this channel
610     SElect ON over_state% : =4 : OVER #channel,1 : =8 : OVER #channel,-1
620   END IF
630 END IF
640   across = across + wide
650 END DEFine Plot_Pixel

```

Listing 12 - rotating text around vertical axis

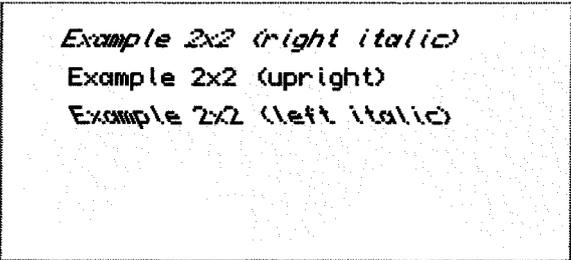
## Italics

By tweaking the Enlarger routine a little we can generate italic text, by adding one parameter to the Enlarge procedure. The extra parameter "italic" can take 3 values:

- 1 = print text sloping to right
- 0 = print text upright (normal)
- +1 = print text sloping to left

Figure 9 shows an example printout from the routine.

Figure 9 - Sample italics output



```

Example 2x2 (right italic)
Example 2x2 (upright)
Example 2x2 (left italic)

```

Listing 13 shows how to create enlarged italics text. The extra parameter is used to modify the value of "across" in line 350, so creating an offset of one pixel for each row of pixels of the character.

```

100 REMark Enlarger, with Italics facility
110 WINDOW 448,202,32,12 : PAPER 7 : INK 0 : BORDER 1,255 : CLS
120 Enlarge #1,2,2,0,50,20,0,7,'Example 2x2 (right italic)',-1
130 Enlarge #1,2,2,0,50,50,0,7,'Example 2x2 (upright)',0
140 Enlarge #1,2,2,0,50,80,0,7,'Example 2x2 (left italic)',1
160 STOP
170 :
180 DEFine PROCedure Enlarge (channel,wide,high,spaced,x,y,ink_colour,paper_colour,str$,italic)
185   REMark italic=-1 means leftward, +1=rightward
190   LOCAL base1,base2,cde1,cde2,nc1,nc2,char,byte
200   base1 = CHAN_L(#channel,42): REMark address of first font

```

```

210 base2 = CHAN_L(#channel,46): REMark address of second font
220 cde1 = PEEK(base1) : REMark lowest valid character 1st font
230 cde2 = PEEK(base2) : REMark lowest valid character 2nd font
240 nc1 = PEEK(base1+1) : REMark number of characters-1 1st font
250 nc2 = PEEK(base2+1) : REMark number of characters-1 2nd font
260 FOR char = 1 TO LEN(str$)
270 cde = CODE(str$(char))
280 SElect ON cde
290 =cde1 TO cde1+nc1: addr = base1+2+(9*(cde-cde1)): REMark font 1
300 =cde2 TO cde2+nc2: addr = base2+2+(9*(cde-cde2)): REMark font 2
310 =REMAINDER : addr = base2+2 : REMark default character
320 END SElect
330 FOR byte = 0 TO 8
340 row_value = PEEK(addr+byte)
350 across = x + ((6+spaced+spaced) * wide * (char-1)) + (byte*italic)
360 IF spaced = 0 THEN
370 REMark ordinary fonts (bits 7 to 2)
380 FOR bit = 128,64,32,16,8,4 : Plot_Pixel
390 ELSE
400 REMark fonts spaced more widely bits 7 to 0, e.g. "fat" fonts
410 FOR bit = 128,64,32,16,8,4,2,1 : Plot_Pixel
420 END IF
430 END FOR byte
440 END FOR char
450 END DEfine Enlarge
460 :
470 DEfine PROCedure Plot_Pixel
480 IF row_value && bit THEN
490 REMark INK pixels
500 BLOCK #channel,wide,high,across,y+(byte*high),ink_colour
510 ELSE
520 over_state% = CHAN_B%(#channel,66) && 12 : REMark OVER details
530 IF (over_state% && 4) = 0 THEN
540 REMark only plot PAPER pixels if OVER 0
550 OVER #channel,0 : REMark cancel OVER temporarily
560 BLOCK #channel,wide,high,across,y+byte*high,paper_colour : REMark paper
570 REMark restore OVER state for this channel
580 SElect ON over_state% : =4 : OVER #channel,1 : =8 : OVER #channel,-1
590 END IF
600 END IF
610 across = across + wide
620 END DEfine Plot_Pixel

```

Listing 13 - *Italics*

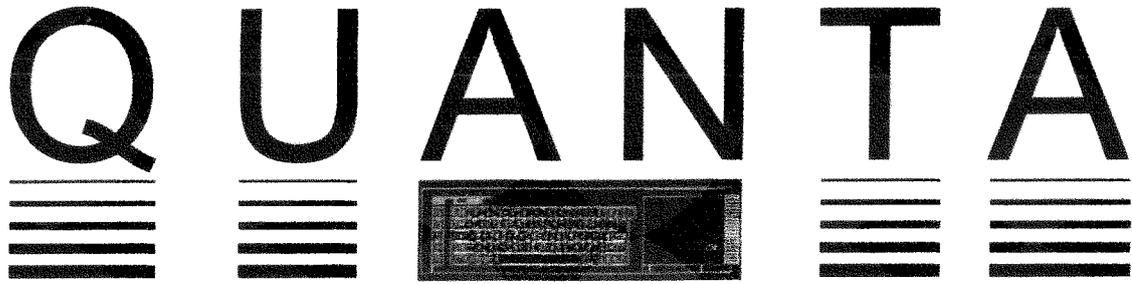
## Other Ideas

The basis of all these routines is knowing the font format and how to manipulate the data contained in fonts. Once we have that starting point, we can apply a little ingenuity to dream up new ideas for text drawing.

One area I haven't ventured into is high colour graphics. One example may be graduated text, where we tell the routine to start with one colour on the left, and work towards another colour at the end. Each pixel's colour is calculated by a gradient of colour change as we work across the string. Some information on this is contained in an earlier article of mine on graduated colour fills, in *QL Today* Volume 8 Issue 1.

I'm sure you, the talented reader, will be able to think up plenty of fun little routines and ways of manipulating text. You can download plenty of font files to play with from my website at: [www.dilwyn.uk6.net/fonts/index.html](http://www.dilwyn.uk6.net/fonts/index.html)

Have fun with fonts!



# Independent QL Users Group

**World-wide Membership is by subscription only,**

Offering the following benefits:

Bimonthly Magazine - up to 52 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 Full Membership

PayPal (see QUANTA Web Site),

Cash, Cheques and Postal Orders Accepted

**\*Now in our Twenty Sixth Year\***

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane,  
Stretford, Manchester, M32 9EH (UK).**

**Tel. +44 (0) 161 865 2872**

**Email: [membership@quanta.org.uk](mailto:membership@quanta.org.uk)**

**Visit the QUANTA Web Site**

***<http://www.quanta.org.uk>***

# Simple GPS receiver

by Ian Burkinshaw

In my previous article on GPS I have used fairly complex and/or expensive solutions for the receiver. Such as the EPE Camera Watch Mk2 project, which Hugh Room also used for his project, or the RF Solutions GPS evaluation card. In both cases they most likely provide more functionality than you may want. Also they are fairly large projects both from a construction point of view and physically large as well. This article looks at an alternative, which is simpler, smaller and cheaper.

This project is published in the October 2008 issue of *elektor* magazine. You can download the original article from the Elektor web site for a small charge. You do not have to purchase the

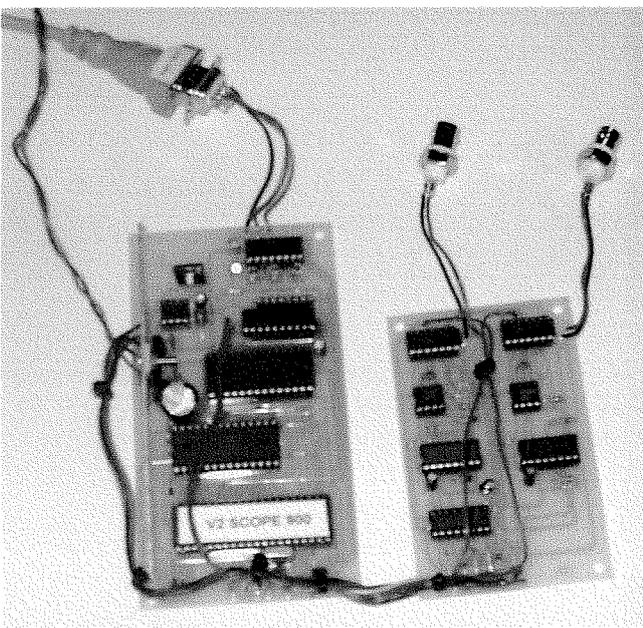
entire magazine. So I will not be going into the finer details of this project here. I wanted just to show people an alternative receiver solution.

This project does require soldering, and a fine tip iron for the GPS module connector is required. But otherwise is as simple as you can get. The PCB is silk screened with all the components shown and their orientation. There is very little wiring involved, just the battery connector. In all other respects it is all self contained. To connect to your QL/PC you will need either a RS232 (9 pin) cable or if your PC does not have RS232 connectors then a USB to RS232 adapter. I use an adapter from PC World with my ASUS Eee PC. See picture.

## Parts List, Where to obtain and price guide

Prices correct at time of writing but with the exchange rates as they are, this may well change.

- 1 x PCB, 14.22 Euro plus 6 Euro Post & packing, The PCB Shop
- 1 x EM-406 GPS module 62.71 Euro, Lextronic
- 1 x Connector for GPS module 2.01 Euro Lextronic
- For both the module and connector you need to add 15.05 Euro post and packing.
- 1 x MAX242 IC, RS Components 151-802, £3.20
- 1 x 78L05 Voltage regulator, RS Components, 189-1295, 53p
- 1 x 470uF, 25V, Maplin, VH47B, 39p
- 1 x 220nF, Maplin, JL02C, 11p
- 1 x 10uF, 25V, Maplin, VH22Y, 11p
- 6 x 100nF, Maplin, BX03D, 11p each, 66p total
- 1 x 100K, Maplin, M100K, 11p
- 1 x 9 pin D-Type female PCB connector, RS Components, 259-3582, £4.40
- 1 x 1N4004 diode, Maplin, QL76H, 15p



The total cost for this project will be around £100 depending where you purchase the components, the list above is only to give you a guide. It also does not include a case to house the project either.

I have tested this receiver with my version of Hugh Room's software that was published in Vol 13 Issue 1 of *QL Today*. I would like to thank the editorial team of *QL Today* for publishing the listing, since it is very long. I did not expect it to be published. I hope people have found it interesting. I use this receiver with my Eee PC since it makes a very good portable set up. The receiver having its own battery also means I get maximum life out of a charge of the PC. Unlike others who have commented on the Asus Eee PC I do get over 2 hours use per charge which I think is quite

good. I only get 1 hour out of my work Sony Viao laptop! One tip to getting good battery life out of the Eee PC is to keep the screen brightness to a minimum. With the screen saver on when the software is just recording data, you get even more life. You do not need to have the screen on while it is doing this. But it does depend on what you want to do with the unit and software.

## References

Elektor - Multi-purpose GPS  
October 2008 issue, PDF download available at a cost at [www.elektor.com](http://www.elektor.com)  
GPS Module : Lextronic, [www.lextronic.fr](http://www.lextronic.fr)  
The PCB : [www.thepcbshop.com](http://www.thepcbshop.com)  
RS : <http://rswww.com>  
Maplin : [www.maplin.co.uk](http://www.maplin.co.uk)

**JOCKEY MERZ SOFTWARE**

**Kaiser-Wilh.-Str. 302 D-47169 Duisburg**  
**<http://SMSQ.J-M-S.com> [SMSQ@J-M-S.com](mailto:SMSQ@J-M-S.com)**

**I can report that I also booked the weekend in Luzern for the 25 year QL event (see reverse side of this page).**

**However, I don't plan to be there as a "trader" - I'd like to take the other view this time, and be a visitor.**

**I can't carry many goods to Switzerland (non-EU) anyway.**

**If anybody would like me to bring something, I will do so, of course - but please let me know in advance what you would like me to bring.**

**I may also bring the latest selection of QL Today magazines ... but again, if you look for specific back-issues, please let me know in advance so that I will have them for you.**

**I expect to be there Saturday late afternoon/evening for the dinner, and the major part of Sunday.**

**Once again - if you need something, please mail me in advance and I will try to bring it!**

**Hope to see many of you there - I haven't managed to come to the UK, unfortunately, this and last year, but I look forward to seeing many of you in Switzerland again!**

**(And please don't get me wrong, I don't plan to give up the QL ... I'm happy to come to future shows as "trader", especially Eindhoven - if they are going to happen again next year, but Switzerland has a real border, as it is not part of the EU, so I can't carry goods anyway).**

# The QL Show Agenda

## **"QL & Mac are 25" international event**

**Oct 31st - Nov 1st 2009**

**Lucerne, Switzerland**

This event will be hosted in the brand new Conference Center, in the famous Verkehrshaus (Swiss Transport Museum).

All details can be found on the following web-page:

[http://www.qlvsjaguar.homepage.bluewin.ch/QL\\_and\\_Mac\\_are\\_25\\_international\\_event.html](http://www.qlvsjaguar.homepage.bluewin.ch/QL_and_Mac_are_25_international_event.html)

Urs König, well known in the QL Scene, is hosting this website and is sponsoring this event. Therefore, he is heavily involved in turning this event into a success! Let's all help!

### **Event overview**

Date: Sat/Sun Oct 31-Nov 1, 2009

Location: Verkehrshaus Lucerne, Switzerland

Type: Conference (sessions, talks), Exhibition, Traders area

### **Program**

Conference with sessions and talks on "QL & Mac early days", "innovation & design then and now", "RAD then and now", "current Mac, iPhone, Linux, Windows and QL developments", ... and an introducing key note session by a VIP

Retro-computing exhibition featuring QL, MAC and many more computers from the 80s

Traders area in the foyer

Official 25th anniversary Dinner on Sat Oct 31 2009

Please use the pre-registration link on the website to give feedback and help to outline and plan best possible.

Lucerne is also a famous touristic attraction, so why not combine this event with sightseeing in and around Lucerne?

# The Next Issue

We plan to have the next issue ready for you towards the middle of December.

As always, it depends on how quickly we get reviews, articles etc.

The more material we get and the sooner we get it, the quicker the next issue will be in your hands, and the better it will be.

Have a nice autumn, your QL Today Team!