IQLR

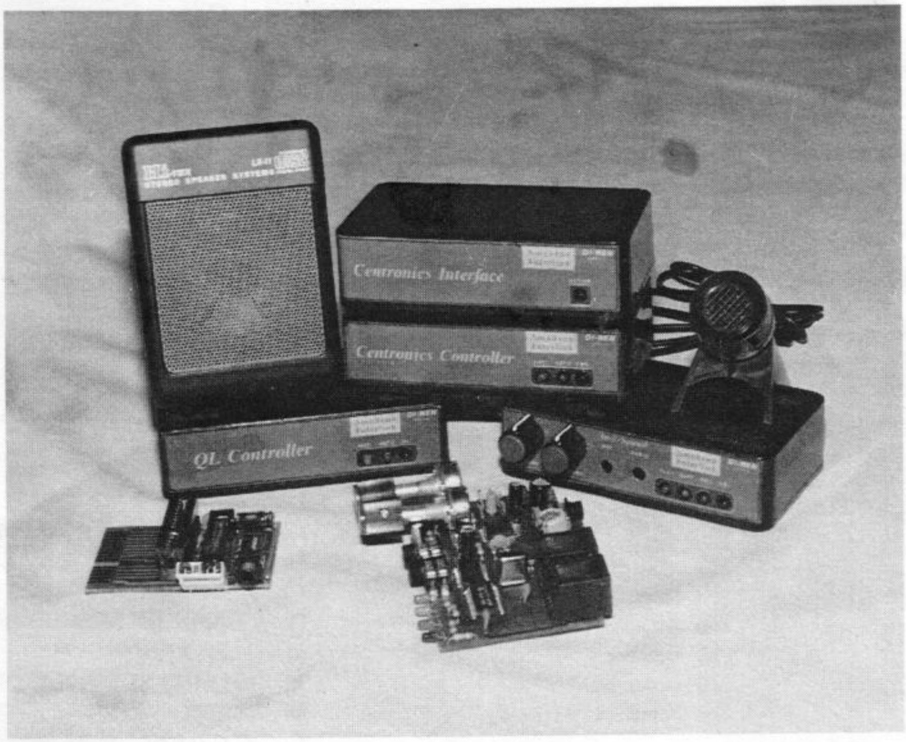
International QL Report

ISSN 1078-5787

Volume 5 Issue 1
May/June
1995

AMADEUS INTERLINK

has arrived!



by

DI-REN

IQLR.......

International QL Report is published by:

IQLR
P. O. Box 3991
Newport, RI 02840-0987
USA

IQLR
23 Ben Culey Drive
Thetford, Norfolk IP24 1QJ
GREAT BRITAIN

Tel/Fax: +1 401 849 3805

PUBLISHER: Robert Dyl, Sr. (ISSN 1078-5787)

IQLR is published bi-monthly, our volume year begins on 1 May and runs through 30 April. Subscriptions begin with the current issue at the time of sign up. Subscription rates are as follows:

USA	$24.00 per year
British Isles & Europe	£25.00 per year
Canada	$27.00 (US Funds)
Central/South America	$34.00 (US Funds)
Rest of World	$40.00 (US Funds)

UK and European readers may send their subscriptions to our European office listed above. Postal, Euro, Bank and Personal Cheques in Pounds Sterling, drawn on a UK bank should be made payable to IQLR.

Payment in US $ can be made by either a Postal, Bank or Personal cheque (drawn on a US BANK) or bank notes (£ or DM equivalent to the US $ amount) should be sent to our North American office.

We welcome your comments, suggestions and articles. YOU make IQLR possible. We are constantly changing and adjusting to meet your needs and requirements. Articles submitted for publication should be on a 3.5" disk in Quill or Text87 format. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hard copy of all screens to be included, don't forget to specify where in the text you would like the screens placed.

Article and Advertising DEADLINES are as follows:

Issue 1	10 April
Issue 2	10 June
Issue 3	10 August
Issue 4	10 October
Issue 5	10 December
Issue 6	10 February

IQLR reserves the right to publish or not publish any material submitted. Under no circumstances will IQLR 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 IQLR. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

Contents...........

Advertisers.......

The 3rd North American QL Show

An Open Invitation to All QDOS/SMSQ Users

Come, join with us, as we celebrate the QL's eleventh year and Jochen Merz Software's tenth year at the 3rd annual North American QL show. The show will be held on Saturday the 10th of June 1995 in OAK RIDGE, TENNESSEE, USA. Those who have attended in the past, will tell you that the show is really secondary to the real event, the GREAT OLD-FASHIONED GOOD TIME to be had by all. Many of us arrive two days in advance and stay one or two days afterwards, others stay a day or two while still others arrive the day of the show. No matter how long your with us, you will have a great time.

Traders expected to attend include: Stuart Honeyball of Miracle Systems (UK), Tony Firshman of TF Services (UK), Bill Richardson of W N Richardson and Co (UK), Jochen Merz of Jochen Merz Software (GERMANY), Frank Davis of Mechanical Affinity (USA), Carol and Frank Davis of Update Magazine (USA), Bill Cable of Wood and Wind Computing (USA) and while not traders, John Impellizzeri and Don Walterman of QBOX-USA (USA) will be demonstrating their QL Bulletin Board.

New Products and Old are expected to abound, including: the Masterpiece Enhanced Graphics Card - Super Gold Card - QXLs from Miracle Systems, Super Hermes - Minerva - I2C interfaces from TF Services, and Mechanical Affinity will have Qubide hard disk interfaces and just about anything else you might want.

Jochen Merz will be demonstrating SMSQ/E plus a number of his PE compatible programs including some new products. Wood and Wind Computing will be demonstrating their state of the art financial package QLerk along with their other products and Mechanical Affinity if they are true to form, will have just about every program currently available for the QL plus plenty of spare parts. For the first time we'll have a table set up for Trade and Sale, so bring those excess items you have with you, people are always looking for good buys.

The Show Registration Fee will be $3 (US) per person in advance and $5 (US) at the door. To register for the show (not the motel) contact IQLR at our North American office. As has been our practice, a Dutch Treat Dinner (you pay for your own meal) will be held following the show.

Base of Operations will be the SUPER 8 MOTEL 1590 Oak Ridge Turnpike Oak Ridge, TN, USA. The room rates are $37 for a single and $41 for a double plus local taxes. Room rental includes the use of their outdoor swimming pool, 25" color television (in each room) with multi-channel Cable, and a FREE Continental Breakfast. For RESERVATIONS Telephone: US 615 483 1200 or Fax: 615 615 482 9834

Please Note: The room rates listed above ***"Are Special Show Rates"*** and will only be honored for reservations made by 9 May 1995. After that date rooms will be on a first come basis at rates 16% higher. ***YOU MUST*** state that you will be attending the ***"3rd North American QL Show"***.

The Venue of the show (with plenty of free parking) will be just down the road from the motel at : FAITH LUTHERN CHURCH, 1300 Oak Ridge Turnpike, Oak Ridge, TN. Show hours will be: 10 am (EST) to 4pm (EST).

International Travellers should book their flights into NASHVILLE, TN if at all possible, there is an additional three hour Coach ride to Oak Ridge. An alternative route would be to land in Atlanta, Georgia and take a 5 hour Coach ride to Oak Ridge. Internal flights should land in Knoxville, TN there is a short 20 minute ride to Oak Ridge.

For those of you driving to Oak Ridge we have included (on the next page) a Location Map of Oak Ridge and helpful directions provided by The Oak Ridge Convention and Visitors Bureau. For additional information please contact:

Mel LaVerne
103 Endicott Lane
Oak Ridge, Tennessee 37830-4117 USA
Telephone: +1 615 483 4153

To the Show!!

Oak Ridge, Tennessee, USA - Mel LaVerne

Or, How to Get Here from There:

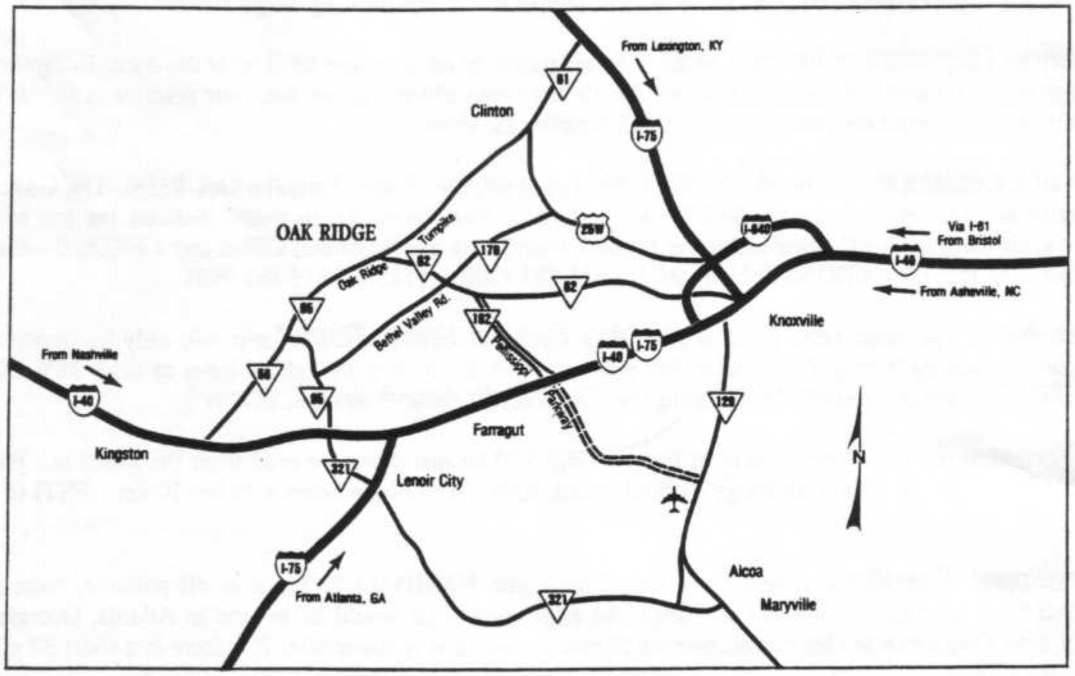
From Nashville: On I40, eastbound. Exit 356 from I40 onto TN58. Tn58 to intersection with TN95. TN95 to intersection with TN62. Super 8 Motel will be on your left.

From Bristol, VA-TN (via I81) or Asheville, NC: On I40, westbound. Exit 376A onto TN162 (Pellissippi Parkway). TN162 to TN62. Bear left onto TN62. TN62 to intersection with TN95. Super 8 Motel on your right.

From Lexington, KY: On I75, southbound. Exit 122 onto TN61. TN61 to and through Clinton, TN. Stay with TN61 until it makes a right turn just outside Oak Ridge city limits. Continue straight ahead into Oak Ridge. You are now on TN95 (Oak Ridge Turnpike). TN95 to intersection with TN62. Motel on your right.

From Atlanta, GA: On I75, northbound. At intersection with I40, take I40 westbound. Exit 364 from I40 onto TN95. Stay with TN95 for a scenic drive through the countryside. At the intersection with TN62, we have (guess what !) The Motel on your left.

Miscellaneous notes: Tennessee Interstate exits are numbered by the nearest mile marker; very convenient when you want the distance to a given exit. The motel parking lot entrance is from the O.R. Turnpike, just east of the Cancun restaurant. Traffic signals on the O.R. Turnpike are numbered (well, most of them). Thus, to find the Show, proceed east on the ORTP to light #9 and turn left. Turn left again into the Church parking lot.



Air Travellers: International visitors can make connecting flights from all major airports to Knoxville, TN (McGhee-Tyson Airport) if your preference is to fly rather than take a coach. Shuttle service is available at McGhee-Tyson to Oak Ridge at reasonable rates from ABC Airport Limo Service (they have a counter on the lower level of the Airport Terminal for your convenance).

SMSQ Language Dependent Modules

Le Grand Pressigny, FRANCE - Tony Tebby

In order to be able to distribute versions of SMSQ which provide messages in more than one language and to support multiple keyboard layouts, SMSQ uses a uniform "language dependent module" structure.

At the moment, SMSQ supports four types of language dependent module:
1. language preference tables,
2. message tables,
3. keyboard tables,
4. printer translate tables.

The principle underlying the language dependent module structure is that each module is identified by a code: for the main languages supported this is the international dialling code (usually 1, 33, 44 or 49). New modules can be added, but these do not (at present) replace existing modules. For this reason, if you live in the south of France and wish to add messages in Occitan, it is strongly recommended that you identify the messages with a code other than 33 (for example 3300).

Language Module Headers - SMSQ creates a table of language modules. This is just a table of pointers to the individual modules. Clearly this does not impose any particular constraints on the size of a module, but it does require that a module has a header in a known form (defining the type of module and its language code). In addition, each module header includes a link pointer to another module header so that many modules may be added to the system in one call. Finally, to provide the maximum flexibility, the module header is not attached directly to the module, but the last long word of the module header is a relative pointer to the module itself.

The code to link in a list of language modules is quite simple.

```
    moveq       #sms.lldm,d0                key to link in
    lea         lang_mod,a1                 pointer to modules to link
    trap        #1                          do SMS call
    rts

lang_mod                                    modules start here
```

This code can be written to a file using WPUT (LPUT can be used for greater efficiency, if you wish).

```
                    WPUT #fch, $7030, $43FA, $0006, $4E41, $4E75
```

Alternatively, the code can be written directly to the computer's memory (useful for testing).

```
                    POKE_W base, $7030, $43FA, $0006, $4E41, $4E75
```

Language Preference Tables - The language preference tables are the most important and the simplest. A language preference table is simply a language name (usually the international car registration letters) followed by a table of acceptable language numbers in order of decreasing acceptability.

This allows the creation of a new language variant without the need to define all of the tables. Thus for Occitan, the second preferred language would probably be French, and, since there is a complete set of French tables, further preferences would not be needed.

```
occitan_pref
    dc.w    0                       it is a preference table
    dc.w    0                       always zero
    dc.w    3300                    Occitan language number
    dc.w    next-*                  relative pointer to next or zero
    dc.l    occ_pref-*              pointer to preference table
```

```
occ_pref
   dc.l      'FOC '                          Occitan is a language of France
   dc.w      3300                            Occitan is the most acceptable
   dc.w      33                              French is next most acceptable
   dc.w      0                               . . . and that is all
```

If the preference table is to be written to a file using SBASIC, WPUT and BPUT are the most appropriate routines to use.

```
   WPUT #fch, 0, 0, 3300, 0, 0, 4            the header - no next, table follows
   BPUT #fch, 'FOC '                         the name: left justified, space filled to 4 chars
   WPUT #fch, 3300, 33, 0                    the preferred languages
```

If this seems a bit heavy, it is. The system is designed to cope with many more languages than you are ever likely to need and to allow dialects or personal variations to be added without inhibiting access to the standard languages. Once you have linked in this new preference table, you can check whether it is there by printing LANGUAGE$:

```
   PRINT LANGUAGE$ (3300)                    should print          FOC
```

Message Tables - SMSQ/E uses four message tables itself. The messages are in four groups. It is possible to add new message tables for these four groups for new languages. It is also possible to add new groups of messages for the existing languages and for new languages. Software developers are requested not to treat this too lightly and create new groups frivolously.

At present, the groups are numbered in 4s (message group 0 is the set of old QL standard messages, message group 4 is the set of SBASIC syntax and execution error messages etc.). The error or message code used to access a message is

-(error message number + 32 x error message group number)

The messages in a group may be listed using REPORT

```
FOR mess = 1 to 50: REPORT #n, -(mess + 32 * group)
```

A message table has the standard language dependent module header. The table itself has pointers to the messages which are relative to the start of the table. As there can be no message 0, the zeroth pointer is replaced by the language number. We can add a new message table for group 12 (table of months and days of week) which (fortunately for this example) only has two entries.

```
occitan_ms12
    dc.w     3                               it is a message table
    dc.w     12                              group 12
    dc.w     3300                            Occitan language number
    dc.w     next-*                          relative pointer to next or zero
    dc.l     occ_ms12-*                      pointer to preference table

occ_ms12
    dc.w     3300                            Occitan
    dc.w     occ_mnth-occ_ms12               pointer to first message
    dc.w     occ_dow-occ_ms12                pointer to second message

occ_mnth
    dc.w     36,'IchNi SanGo Ro ShiHa Ku Ju JuiJun'

occ_dow
    dc.w     21,'Ni Ge Ka SuiMo KinDo '
```

Note that the first message must follow the last pointer: the first pointer is, therefore, twice the number of messages in the table, plus 2. It is possible to write this message table to a file using a simple BASIC program as follows, but for the more complex tables with variable length messages a more complex program would be required.

```
WPUT #fch, 3, 12, 3300, 0, 0,      4          the header - no next, table follows
WPUT #fch, 3300, 6, 6+2+36                    the number and two pointers
PUT #fch, 'IchNi SanGo Ro ShiHa Ku Ju JuiJun'
PUT #fch, 'Ni Ge Ka SuiMo KinDo '             the two strings
BPUT #fch, 0                                  a pad byte because the string was odd length
```

Once you have linked in this new message table, you can use it by typing the LANG_USE command:

```
LANG_USE 3300          or
LANG_USE FOC
```

Keyboard Tables - The language dependent module that is next most likely to be added is a keyboard table. A keyboard table has the standard language dependent module header. For historical reasons, the module pointer in the header points to an intermediate structure. This intermediate structure has the language code followed by a pair of relative pointers the first of which points to the "normal" keyboard table, the second points to a table of "non-spacing characters". These are keys which when pressed do nothing but modify the next character typed. These keys are usually an accent key which is used to add an accent to the next letter. (Some keyboard drivers may not support non-spacing characters.)

```
occitan_kbd
   dc.w     1                         it is a keyboard table
   dc.w     0                         no group
   dc.w     3300                      Occitan language number
   dc.w     next-*                    relative pointer to next or zero
   dc.l     occ_kbd-*                 pointer to preference table

occ_kbd
   dc.w     3300                      Occitan
   dc.w     occ_ktab-*                pointer to keyboard table
   dc.w     occ_nsid-*                pointer to non-spacing character table
```

If the keyboard table follows immediately after the header and the non-spacing table immediately after that, the header may be written by a simple SBASIC program

```
WPUT #fch, 1, 0, 3300, 0, 0, 4        the header - no next, table follows
WPUT #fch, 3300, 4, 2+512             the number and two pointers (all but QL kbd)
or WPUT #fch, 1, 0, 3300, 0, 0, 4     the header - no next, table follows
WPUT #fch, 3300, 4, 2+256             the number and two pointers (QL kbd)
```

The size of the keyboard table depends on the keyboard itself. The table is divided into four blocks: normal keystrokes, control keystrokes, shifted keystrokes and shifted control keystrokes. The tables are the characters produced for each of the possible keyboard codes. For the main keyboards, these keyboard codes are as follows.

QL keyboard - 64 entries in each of 4 blocks

```
F1-F5       57    59     60   56   61
Top row     51    27      9   25   62     58    10     63   8      16    13  21    37   45   53
             19      11     17   12    20    14    22    15     18    23   29   32   40
              33      28     35   30    36    38    26    31     34    24   39    47   48
               (0)        41    3    43    4     44  6       46      7   42   5      (0)
               (1) 49        52                      54                       50   55   (2)
```

Thus, for a normal QL keyboard layout, the codes for the digit keys are 27, 9, 25 etc. The 27th entry in the keyboard table should be the character '1', the 9th entry '2' etc. (The table starts with the zero'th entry.) *The key codes in brackets are trapped by the driver (Shift, Control and Alt) and the corresponding values in the keyboard tables should be zero.*

Atari ST TT keyboard - 128 entries in each of 4 blocks

F1-F10	59	60	61	62	63	64	65	66	67	68					
Top row	1	2	3	4	5	6	7	8	9	10	11	12	13	41	14
	15	16	17	18	19	20	21	22	23	24	25	26	27	83	
	(29)	30	31	32	33	34	35	36	37	38	39	40	28	43	
	(42)	96	44	45	46	47	48	49	50	51	52	53	(54)		
	(56)	57	58												

Cursor pad	98		97	Numeric pad	99	100	101	102
	82	72	71		103	104	105	74
	75	80	77		106	107	108	78
					109	110	111	
					112		113	114

Thus, for a normal ST keyboard layout, the codes for the digit keys are 2, 3, 4 etc. The 2nd entry in the keyboard table should be the character '1', the 3rd entry '2' etc. (The table starts with the zero'th entry.) *The key codes in brackets are trapped by the driver (Shift, Control and Alt) and the corresponding values in the keyboard tables should be zero.*

QXL AT keyboard - 128 entries in each of 4 blocks - The PC models AT and later incorporate an "intelligent" keyboard controller which has three main functions:

1. converting the easy to handle, explicit, AT 102 key keyboard codes into garbled sequences of PC XT keyboard codes (up to 10 keycodes for each keystroke!);

2. losing keystrokes;

3. getting shift keys "stuck down".

The keyboard tables are, therefore, based on the PC XT key codes.

Esc-F10	1	59	60	61	62	63	64	65	66	67	68				
Top row	41	2	3	4	5	6	7	8	9	10	11	12	13	43	14
	15	16	17	18	19	20	21	22	23	24	25	26	27		
	58	30	31	32	33	34	35	36	37	38	39	40	28		
	(42)	86	44	45	46	47	48	49	50	51	52	53	(54)		
	(29)	(56)	57	(56)	(29)										

Odd pad	55	70	(69)					
Cursor pad	114	103	105	Numeric pad	(69)	133	55	74
	115	111	113		71	72	73	
					75	76	77	78
		104			79	80	81	
	107	112	109		82		83	124

Numeric pad Without Num Lock	(69)	133	55	106
	103	104	105	
	107	108	109	110
	111	112	113	
	114		115	124

Thus, for a normal PC keyboard layout, the codes for the digit keys are 2, 3, 4 etc. The 2nd entry in the keyboard table should be the character '1', the 3rd entry '2' etc. (The table starts with the zero'th entry.) *The key codes in brackets are trapped by the driver (Shift, Control and Alt) and the corresponding values in the keyboard tables should be zero.*

Sample Keyboard Table - It is easiest to see the format of the keyboard table if the PC AT 102 key keyboard layout is taken as an example. The first block of 128 characters is for unshifted characters.

```
occ_ktab                    ; unshifted keys for UK kbd (in groups of 16)
    dc.b    0,$1b,'1','2','3','4','5','6','7','8','9','0','-','=',$c2,$09
    dc.b    'q','w','e','r','t','y','u','i','o','p',$5b,$5d,$0a,0,'a','s'
    dc.b    'd','f','g','h','j','k','l',';',$27,$9f,0,'#','z','x','c','v'
    dc.b    'b','n','m',',','.','/',0,'*',0,' ',$e0,$e8,$ec,$f0,$f4,$f8
    dc.b    $ea,$ee,$f2,$f6,$fa,0,$f9,'7','8','9','-','4','5','6','+','1'
    dc.b    '2','3','0','.',0,0,'\',0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,$d5,$d0,$d4,0,$c0,0,$c8,0,$dd
    dc.b    $d8,$dc,$eb,$ca,0,'/',0,0,0,0,0,0,$0a,0,0,0
```

This block is followed immediately by the block of 128 shifted characters, then the 128 characters which are produced when the control key is held down and finally the 128 characters which are produced when both shift and control keys are held down. As there are only 256 different values that can be stored in a byte and there are 512 total keyboard table entries, there will naturally be a large number of zeros in the tables as well as a certain number of duplicate codes.

Non-Spacing Characters - The non-spacing character table is a little bit odd. It is a 256 byte table which is (nearly) filled with zeros. For any character which can be used as a non-spacing character, the corresponding entry in the table is non-zero. Thus, if the ' is used as a non-spacing character to produce accented characters, the 39th entry in the table is non-zero (' is ASCII code 39). The non-zero value is the offset from the end of the table to the list of modified characters for this non-spacing character.

The table is immediately followed by a variable size table of modifiable and modified characters. This table has entries which are one longer than the number of modifiable characters (each entry is terminated by a zero). The first entry lists the modifiable characters. This is followed by an entries giving the corresponding modified characters for each of the non-spacing characters.

```
occ_nsid
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,8,0,0,0,0,0,0,0,0        apostrophe for acute accent
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,16       open quote for grave accent
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

    dc.b    'a','e','i','o','u','E',' ',0          modifiable characters (including space)
    dc.b    $8C,$83,$93,$96,$99,$A3,$27,0          acute accented characters (including ')
    dc.b    $8D,$90,$94,$97,$9A,'E',$9F,0          grave accented characters (including £)
```

SMSQ Language Dependent Modules - (cont'd)

This table can be written to a file using BPUT commands in an SBASIC program.

```
nsid = FPOS (#fch)                    : REMark - remember where it starts
BPUT #fch, FILL$(CHR$(0),256)         : REMark - fill table with zeros
BPUT #fch\nsid+39, 8                  : REMark - fill in apostrophe
BPUT #fch\nsid+159, 16                : REMark - fill in open quote
BPUT #fch\nsid+256                    : REMark - back to end of table
BPUT #fch, "aeiouE ",0                : REMark - unmodified characters at end
BPUT #fch, "áéíóúÉ'",0                : REMark - then acute accents
BPUT #fch, "àèìòùE`",0                : REMark - and grave accents
```

Once you have linked in this new keyboard table, you can use it by typing the KBD_TABLE command:

```
KBD_TABLE 3300        or
KBD_TABLE FOC
```

Printer Translate Tables - The printer translate tables are directly compatible with the old, not very useful, QL printer translate tables used by the TRA command.

These "language dependent" tables came into existence because someone at Sinclair had the rather strange notion that, in some way, the character ä (for example) should be printed differently depending on the country. I think that an ä is an ä wherever you are.

SMSQ has a standard printer translate that works on any PC compatible printer set to the USA character set (the use of a non-USA character sets tends to make it impossible to print certain characters).

For peculiar printers, however, you can set up your own tables.

A printer translate table has the standard language dependent module header. For historical reasons, the module pointer in the header points to an intermediate structure. This intermediate structure has the language code followed by a pair of pointers relative to the language code. The first of these points to the "byte to byte" translate table, the second points to a table of "byte to three byte" translates".

The byte to byte table has, naturally, 256 single byte entries. The first entry is zero (null stays as null) all other entries are either the translated character or zero. If the entry is zero, the character is translated using the three byte table.

The three byte table is preceded by a zero byte (historic) and starts with the number of three byte sequences (in a byte). This is followed by groups of 4 bytes, the first of which is the QL character, the next three are the characters to be sent to the printer.

The following is a copy of the IBM printer translate table which may be used as a basis for other printers.

```
occitan_tra
        dc.w    2                       it is a printer translate table
        dc.w    0
        dc.w    3300                    Occitan language number
        dc.w    next-*                  relative pointer to next or zero
        dc.l    occ_tra-*               pointer to preference table

occ_tra
        dc.w    3300                    Occitan
        dc.w    occ_byte-occ_tra        pointer to byte to byte table
        dc.w    occ_3byte-occ_tra       pointer to three byte table
```

```
occ_byte
        dc.b        $00,$01,$02,$03,$04,$05,$06,$07
        dc.b        $08,$09,$0A,$0B,$0C,$0D,$0E,$0F
        dc.b        $10,$11,$12,$13,$14,$15,$16,$17
        dc.b        $18,$19,$1A,$1B,$1C,$1D,$1E,$1F
        dc.b        $20,$21,$22,$23,$24,$25,$26,$27
        dc.b        $28,$29,$2A,$2B,$2C,$2D,$2E,$2F
        dc.b        $30,$31,$32,$33,$34,$35,$36,$37
        dc.b        $38,$39,$3A,$3B,$3C,$3D,$3E,$3F

        dc.b        $40,$41,$42,$43,$44,$45,$46,$47
        dc.b        $48,$49,$4A,$4B,$4C,$4D,$4E,$4F
        dc.b        $50,$51,$52,$53,$54,$55,$56,$57
        dc.b        $58,$59,$5A,$5B,$5C,$5D,$5E,$5F
        dc.b        $9C,$61,$62,$63,$64,$65,$66,$67
        dc.b        $68,$69,$6A,$6B,$6C,$6D,$6E,$6F
        dc.b        $70,$71,$72,$73,$74,$75,$76,$77
        dc.b        $78,$79,$7A,$7B,$7C,$7D,$7E,$00
        dc.b        $84,$00,$86,$82,$94,$00,$00,$81
        dc.b        $87,$A4,$91,$00,$A0,$85,$83,$89
        dc.b        $8A,$88,$8B,$A1,$8D,$8C,$A2,$95
        dc.b        $93,$A3,$97,$96,$E1,$9B,$9D,$60
        dc.b        $8E,$00,$8F,$90,$99,$00,$00,$9A
        dc.b        $80,$A5,$92,$00,$E0,$EB,$E9,$00
        dc.b        $E6,$E3,$ED,$AD,$A8,$3F,$EC,$00
        dc.b        $AE,$AF,$F8,$F6,$00,$00,$00,$00

        dc.b        $C0,$C1,$C2,$C3,$C4,$C5,$C6,$C7
        dc.b        $C8,$C9,$CA,$CB,$CC,$CD,$CE,$CF
        dc.b        $D0,$D1,$D2,$D3,$D4,$D5,$D6,$D7
        dc.b        $D8,$D9,$DA,$DB,$DC,$DD,$DE,$DF
        dc.b        $B0,$B1,$B2,$B3,$B4,$B5,$B6,$B7
        dc.b        $B8,$B9,$BA,$BB,$BC,$BD,$BE,$BF
        dc.b        $F0,$F1,$F2,$F3,$F4,$F5,$F6,$F7
        dc.b        $F8,$F9,$FA,$FB,$FC,$FD,$FE,$FF

        dc.b        0                   ; pad

  occ_3byte
        dc.b        15                  ; 15 replaces
        dc.l        $A54F087E           ; O bs tilde
        dc.l        $AF5C082E           ; \ bs .
        dc.l        $7F63084F           ; c bs O
        dc.l        $8161087E           ; a bs tilde
        dc.l        $856F087E           ; o bs tilde
        dc.l        $866F082F           ; o bs /
        dc.l        $8B6F6500           ; o e
        dc.l        $A141087E           ; A bs tilde
        dc.l        $A64F082F           ; O bs /
        dc.l        $AB4F4500           ; O E
        dc.l        $B76F0878           ; o bs x
        dc.l        $BC3C082D           ; < bs -
        dc.l        $BD3E082D           ; > bs -
        dc.l        $BE5E0821           ; ^ bs !
        dc.l        $BF760821           ; v bs !
```

Once you have linked in this new printer translate table, you can use it by typing the TRA command:

```
TRA 1, 3300      or
TRA 1, FOC
```

A Complete Language Dependent Extension - This SBASIC program creates a complete language dependent extension with preference, keyboard and message tables. SBASIC procedures are used to set the relative pointers: you can try to decipher them if you wish.

The keyboard table is the standard UK PC (QXL) keyboard with one difference: the F6 to F12 keys are used as non-spacing characters. To avoid conflicts with existing key codes, the ALT cursor key codes are borrowed (the ALT key is handled within the driver and so the only ALT key codes which appear in the table are those for HOME (=ALT SHIFT UP) and END (=ALT SHIFT DOWN)).

The message tables are very slightly modified versions of the English tables: All four standard groups are included to make it easier for you to create your own message tables.

Running a Gold Card From 5 Volts

Thetford, Norfolk, GREAT BRITAIN - W. Geraint Jones

The QL normally runs as we all know from 9 volts, which if you have transfered your QL to a PC case (with a switch mode power supply), causes a few problems. The solution to this problem is to run the whole system on 5 volts, but the backup clock on Gold Card will not function without a 9 volt supply.

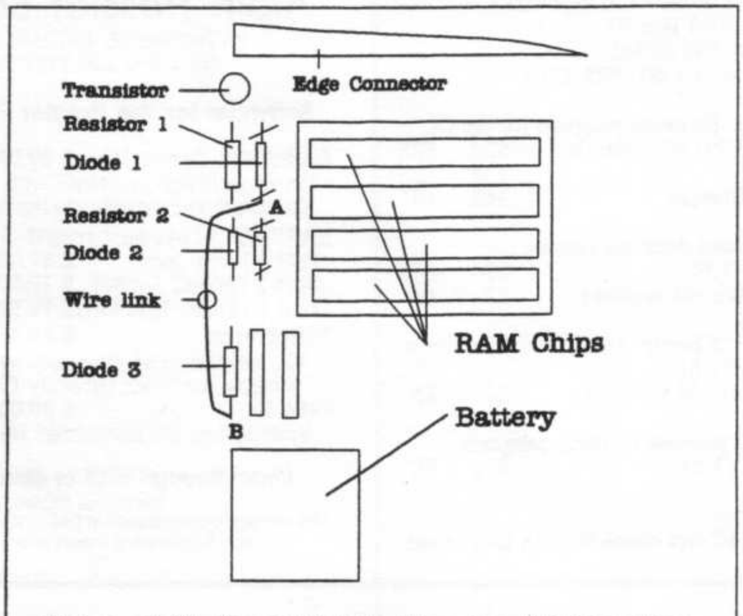


Fig. 1. Gold Card Modification for 5V Operation

This problem can be over come by making the following modifications to the Gold Card ***(DO NOT DO THIS TO A SUPER GOLD CARD).***

1) Locate and remove Diode 1 and Resistor 2 (see fig. 1 above).
2) Link points A and B with a short length of insulated wire.
3) Bridge out the 5 volt regulator.
4) Connect a 5 volt supply to the expansion port power rails and ensure NO 9 volt supply is connected to any parts of the QL.

5) If you have any other cards connected - QUBIDE IDE interface etc., ensure the regulator is out of circuit - by following the suppliers instructions in the appropriate manual for 5 volt operation.

A Note From Digital Precision

London, ENGLAND - Freedy Vachha

We have found a benign, harmless anomaly in the QL COLLECTION's menu options: Users of early versions (pre Version 6) of the menu program should note that in order to expand PC Conqueror Gold SE or its documentation you should select the Standard version. Choose the Gold SE version for the Standard version or its documentation. Users are reminded that the Special Edition will only work on systems with over 1 Mb of RAM. All others should use the Standard version.

In addition, users of very early versions of the Collection (pre-version 5) may benifit by upgrading. The latest version has new material including the whole text of the Maastricht treaty and other programs. To upgrade send the disk marked "QL 1" together with £10 by UK cheque, Eurocheque, VISA, MASTERCARD, Postal Order or Cash. Please note our advert and address elsewhere in this issue.

Independent Producer's *Independent Producer's*

Mini Mart of Values

"The painless way to shop world wide "

Wood and Wind Computing

RR3 Box 92
Cornish, NH 03745 USA
Telephone: +1 603 675 2218

QLerk - A complete financial program for the QL

QLerk software (V3.21) with tutorial	$34	£23
QLerk manual	$38	£25
QLerk Software & Manual	$62	£41

DBEasy - A menu based database system

DBEasy software (V1.6)	$24	£16
DBEasy upgrade (give old version)	$7	£5

DBProgs - A toolkit of handy ARCHIVE procedures

DBProgs software (V1.8)	$18	£12
DBProgs update (give old version)	$7	£5

DBTutor - A general purpose learning program

DBTutor software (V1.5)	$12	£8

SEE our Full Page advert elsewhere in this issue

Albin Hessler Software

Im Zeilfeld 25 D-72631 Aichtal
Tel + Fax +49 7127 56280

Software for the Pointer Environment

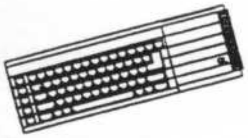
Cueshell (Ver 1.16)	**£ 39.00**	**$ 63.00**	DM 98
Desktop Program. Now with screen saver **CueDark** especially designed for SMSQ			
EASYPTR 3 Development System			
Part 1 Basis Jobs	**£ 39.00**	**$ 63.00**	DM 98
Part 2 SBASIC Toolkit	**£ 19.50**	**$ 31.50**	DM 49
Part 3 Library Routines	**£ 19.50**	**$ 31.50**	DM 49
SERMouse	**£ 29.00**	**$ 47.00**	**DM 73**
QL serial mouse (works with SGC and SMSQ/E) Ready for SER2 (specify BT or SUB-D9)			
DISA 2	**£ 39.00**	**$ 63.00**	DM 98
Interactive Disassembler. Many new features			

Order through IQLR or directly from us

We speak English.
We accept eurocheques in DM or personal cheques in any EU currency drawn to any EU bank

Your Software/Hardware

Product(s) Could Be Advertised Here



Contact IQLR For Details

UPDATE! MAGAZINE

P.O. BOX 17
MEXICO, IN 46958 USA

The LARGEST magazine in the world supporting the Sinclair QL, Spectrum,TS2068, ZX81, and Cambridge Z88 Computers.

A quarterly magazine with issue year running from OCTOBER to JULY. We are NOW in our 8th year of publication.

To subscribe send a check made payable to UPDATE Magazine for the amount that is applicable to you.

USA & CANADA	**$20**
Rest of World	**£18 or DM40**

Cash, Checks, or Money Orders are acceptible.

How To Order

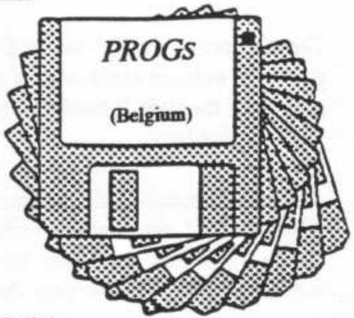
You may order direct from the supplier (prices listed include VAT where applicable and airmail post anywhere in the world. For your convience you may place your order through IQLR and take advantage of our ability to accept Personal or Bank cheques drawn on a UK bank or Euro cheques drawn in Pounds Sterling. You may also make payment using Postal (drawn in either US$ or Pounds Sterling), Personal or Bank cheques in US$ drawn on a US bank.

When ordering through IQLR, PLEASE make all cheques payable to IQLR and post your order to either our UK or North American office. We will process you order and the individual supplier will post it directly to you.

PLEASE NOTE: All ENQUIRES are to be made directly to the supplier NOT IQLR.

PFdata & the DATAdesign APL

Portslade, Sussex, GREAT BRITAIN - Roy Wood



DATAdesign was the first program I bought when I made the transition to Pointer Environment programs and I have been a PROGS fan ever since. Archive, for me at least, was a seriously annoying program to deal with but I think that is because I tend to use databases in a different way to a lot of other people. I have three main databases: an address book, a list of videotapes and a list of venues for shows which I use as a card index more than anything else.

Using them in this way with Archive is very slow unless you copy the file to a RAM disk and then having to type in the commands in full slowed it down even more. I did use a short routine to nominate single letters as commands for the most often used actions but when I got DATAdesign everything moved at a cracking pace and, even if the computer crashed mid-edit, the file was still safe because all actions are performed on a copy of the file in RAM and saved at the end of the session. Being a Pointer Program you can just sit back in the chair and let the mouse do the walking to browse through the database if you so wish.

There were however, a couple of points at which I would have liked to a have a bit more control over the database and that is where these two packages come in. When I used Archive for my data I had another short routine that I used to produce a printout, with the bare DATAdesign program you get a simple option to print a record or the whole file but no way of positioning or sizing the text. Into the spotlight steps PFdata.

PFdata - this is aprogram to produce printouts from any DATAdesign database and is an added advantage over the kind of printout you can get from Archive because you can use a variety of fonts, sizes and if you use LINEdesign you can include frames and pictures in the same printout.

The program is supplied on two 3.5" disks and comes complete with a comprehensive A4 manual. It contains all you need to print your database (although it is obviously useless if you don't use DATAdesign) and the cost is a modest 1000 Belgian francs (or around £20/$29 depending on the exchange rate) including postage.

You will need at least the minimum memory expansion (256K) to run this because in order to operate the program requires both the PROforma and DATAdesign engines to be running. The disks contain the following files:

Disk 1

Boot	a sample boot file
Config	the standard Pointer Config program
CPUdetect	a program to detect which operating system you use and act accordingly
drivers	a short program to give the driver numbers in the fontmap (more on this later)
HOT_rext	the standard Pointer extensions file
PFConfig	a program to configure the fontmap to your choice
PROforma	the PROGS screen and printer driver manager

plus 13 different printer drivers and the QVME screen driver.

Disk 2

PFdata	the main program

plus 17 fonts and various example files, LINEdesign pictures and programs.

If you already use LINEdesign you can add these fonts to your fontlist from that program (they are different) and then you have a larger range of fonts to print with.

PFdata & the DATAdesign APL - (cont'd)

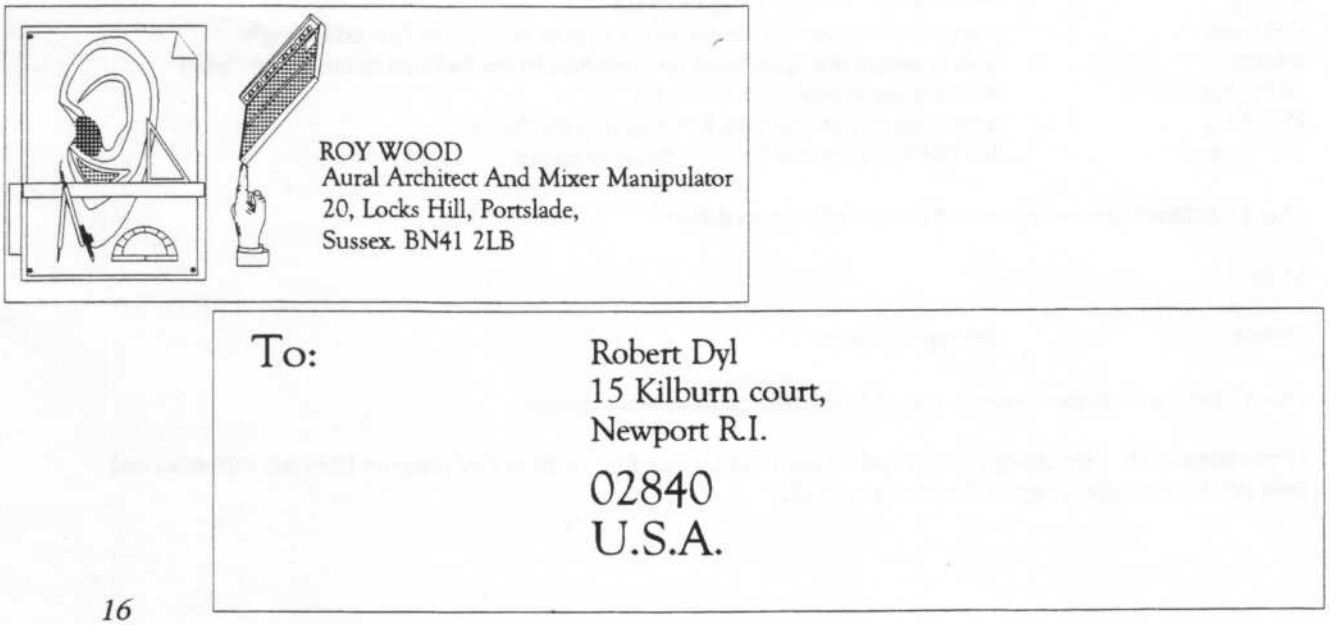
The program itself is one of those transparent processes that only makes itself apparent when it goes wrong or you have entered some wrong parameters but, in order to make it function you need to prepare a plain text file to give it all the data it needs to operate. PROGS recommend QD from Jochen Merz and I prepared mine using QD with no problems.

Although I bought this program to print extracts from my 'venues' file for the various music groups that I work with I needed something shorter for this review and decided to use it to print the addresses on envelopes. Prior to this I had kept a separate file list of addresses in Text87 for this purpose but the end result with DATA design was so good that I have now dumped that in favour of this system. The text file I use is:

```
\DATA adbk WIN1_DATA_Dbases_           : this line gives the database to use
\FONT Grand Old Style                  : this line gives the font
\Size 14                               : and the fontsize
\BACK WIN1_data_Envelope_ldp           : this is the LINEdesign drawing
@ 150 170                              : this is the  print position
name                                   : and this the field name to print
@ 150 185
address
@ 150 200
address_2
@ 150 215
town
@ 150 230
county
\size 20                                       :here we change the fontsize to give a clearer postcode.
@ 150 245
postcode
```

(the items behind ':' do not appear in the text file and have only been added as an explanation for this article)

Once you have written and saved this text file all that remains is to execute the program. If your boot file has already loaded PROforma and the DATAdesign engine you can execute PFdata directly from the command line but if not you do need to use the kind of boot file supplied on the first disk. Since I use both DATAdesign and LINEdesign a lot and have the Super Gold Card (and therefore lots of free memory) I run this directly from the command line thus: EX PFdata;'<WIN1_Envelprint_pfd n' (where 'n' is the record id of the file to print and WIN1_Envelprint_pfd is the text file above).



ROY WOOD
Aural Architect And Mixer Manipulator
20, Locks Hill, Portslade,
Sussex. BN41 2LB

To:

Robert Dyl
15 Kilburn court,
Newport R.I.
02840
U.S.A.

This will print the LINEdesign picture, text and frames and then place the appropriate address in the lower frame but PFdata has a lot more flexibility than this. By adding parameters after the text file you can specify the number of copies and use a different printer driver and output device to the default.You can also give a sequence of record ids for to identify the records to be printed. In practice this means that you can, for instance, re-specify one of the printer drivers using the 'config' program to give a different page size or print density and then tell PFdata to use that driver. Of course that means that you will need to know the driver id from the fontmap and this is the reason for the program 'drivers' on the first disk. Once you have modified your driver and added it to the list using the PFconfig program all you then have to do is to 'EX drivers' and you get a numbered list of the printer drivers and their ids (do not forget that PROforma loads the drivers and fonts when it is first run, so you will need to reset and reboot if you want to use a driver you have just modified).

The default print orientation is Portrait style but, because these are vector fonts, one line in the text program will print the whole thing in landscape format and the result is just as good.

Conclusions

The program is very easy to use and the text file is straightforward to produce but there are one or two points that could be improved. The first thing I noticed was that this program lacked a 'preview mode'. Once you have prepared the whole file for printing the only way to find out if you have the co-ordinates correct is to print it. This can be a bit paper and ink consuming if you have made a mistake. Even without mistakes it took several adjustments to get a printout that I was happy with. A small program that takes the text file and puts it on the screen would be a great bonus. Another missing feature is that the print is inflexible in its printing of the fields. I would have liked to be able to tell it to miss out those fields which were blank but there is no way to do this (although I can see that this would not be easy). One last thing that I think it lacks is a routine to call up the database that is to be printed and to select the record ids from it. The present method expects you to know those ids and put them into the command line but that means that I have to go to the database, find the files, and then 'Hit' the '?' icon which will then give me the information about the record. I then have to write these down and type them into the command line. This is not much work, I grant you, but a program to do that for you and save the numbers as a text file for later insertion would make the process smoother. Apart from this, however, the results that you can get with this program are remarkably better than can be obtained using Archive and the price is so minimal that it is well worth the purchase.

DATAdesign API - The comments above about abstracting the record ids for use in PFdata led me to purchase the DATAdesign API in the hope that I could write just such a program. I must say (and I have said it before) I am no programmer although I have come a long way from my earlier position of understanding nothing of programming.

I had no intention, at this stage, of attempting the assembler or C interface with DATAdesign and I cannot comment on the implementation of these two methods so I will confine myself here to the Superbasic side of the API. The bulk of this is in the manual and the disk provided is mostly examples of programs using the three types of interface and a couple of files for these examples to work on. The disk also contains the conversion programs to convert old DATAdesign files to the current type, produce an export file and convert Flashback files to DATAdesign ones.

The manual is the usual A4 bound PROGS documentation and explains the terminology of the database at some length before launching itself into a description of the commands available. Most of these commands and extensions are installed when you run the DATAdesign engine so there is actually no program to run with the API only PROFORMA_rext. The main thing that the potential user has to do is to study the manual - and this is really where it all comes down to how much you actually already know.

When I was an Archive user I wrote a few routines to use with that program, none were very elegant or clever but they worked and did what I wanted them to do. I was able to do this because that part of the User Guide was very well set out and built up to big programs by providing short routines to do small tasks. There were also several PD archive programs and I was able to plunder them for the routines I needed. Unfortunately the API manual is of the Easyptr style and will probably make a lot of sense to an experienced programmer but to an amateur like me the whole thing is confusing. It does not help that all three languages are lumped together under the command heading and that no printed examples are given. There is a Superbasic program supplied on disk which uses most of the commands available from the API but this takes up 7 A4 pages when printed and is very sophisticated in its operation. I have managed to extract a few routines from it but, so far, I have not managed to write a program.

After I had written a few procedures in the Archive language I found it easier to construct others in the Psion OPL (this is the language used on the Psion organiser range and developed directly from Archive). The Psion manual was not too helpful either but because the Archive part of the User guide took things step by step and built up a collection of usable routines I found it quite easy to learn. The API manual, on the other hand, races through a description of the routine and their parameters without even a line of code to illustrate the use of the command. When you come to look at the example program on the disk you find a few sections that are not only not referred to in the manual but also have very misleading REMarks. Instead of providing a program which opened one file and performed a few operations on it they provide one which opens three files and uses them to interact with each other. This means that tracing a path through the program is more difficult and extracting useful sections harder. To give an example: I spent ages leafing through the manual looking for a mention of what "Initialising the field ids" did until I realised that they used a variable name for each field and gave it a number because the API related to search fields by the fieldid number and not by name. It would have been more obvious had it only been one file I was working on and not three.

This said, I can see that the programming language is very comprehensive in its application and the ability to combine and use frames and logos from LINEdesign in the programming is a great advantage. I am still working on writing some programs using these extensions (if anyone else has any short routine or useful comments to add to this I would be very pleased to hear from them) and I am sure that, given enough time, I will get somewhere but this could have been a lot easier.

I wanted to give PROGS to chance to comment on this so I wrote to them. Their reply was "We have to agree that the DATAdesign API could be served by some better examples. However, we have not currently got the time to do something about this." Since they are a small operation and I believe they are working on a new program at the moment I can appreciate the problem. If you are an experienced programmer and you use DATAdesign (and I have found this a far better and faster program than Archive) then I am sure that you can find a good use for this if you are a novice - then, believe me, you need a lot of time and patience.

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
☎ and Fax: 0203-501274 • Mailbox: 0203-591706

JMS

10 years Jochen Merz Software! I am supporting the QDOS scene now for 10 years, and that's a good reason to celebrate - 10 years of fun which resulted in a very close relationship to my customers, in many cases friendship (club meetings are tending to become a sort of "family" meeting. I would like to thank all users for their support.
Therefore, a special "birthday offer", valid until 1st of May 1995. Any new program or game purchase of any product found in the right column of this ad (no updates/upgrades) can be discounted:
4% if you buy 2 programs at the same time, 7% on 3 programs, 10% if you buy 4 or more programs at the same time!!!
Alternatively, any new program bought together with SMSQ/E will be discounted by 7%!!!

QL Hardware & Spares

FLP/RAM Level 2 device drivers for SuperQBoard or TrumpCard			DM 56,–
SER Mouse software driver			DM 40,–
SER Mouse Package (mouse, adaptor & driver)			DM 87,–
ZX8301	DM 24,90	ZX8302	DM 22,90
QL Keyboard membrane	DM 28,–	2 membranes	DM 50,–
SuperGoldCard			DM 699,–

QL-Emulators for ATARI

QVME – High-Res QL-Emulator for ATARI Mega STE and TT (up to 1024x800 pixels and more. PRICE CUT!!	DM 499,–
EXTENDED4-QL-Emulator for ATARI 260ST, 520ST/STF/STFM, 1040ST, Mega ST (but not STE!!!)	DM 289,–
Hi-Res Mouse (microswitches) + Mousepad	DM 29,90

QXL (QL-Emulator for PC)

QXL 4MB	DM 699,–	QXL 8MB	DM 999,–

Have you already got **QMON/JMON**? Ideal for **SMSQ/E**, works on 68000, 68020, 30 & 40 (essential for QXL, SuperGoldCard and Hi-Speed ATARIs).
Remember: most products are not only improved but also upgraded to **Config Level 2** - updating your software was never as easy as it is now: you configure your program once, and from then on **MenuConfig** does it for you whenever you get a new version!!!
The latest **QDOS/SMS Reference Manual** Updates are on the way: they describe the new features of SMSQ and **SMSQ/E**, all the new traps, language handling etc. If you have not booked for 4 updates, then it's time to do it now!!!

SMSQ/E V1.52

More news on **SMSQ/E**: updates are free, just send master disk(s) & return postage (or use the free mailbox update service!
Global (system) character set can be re-defined. This means, you can use YOUR preferred scren font in almost any program now, and you can add characters for other languages like Greek, Polish etc.
More new commands: HPUT, HGET, WGET, WPUT, LGET, LPUT, DISP_UPDATE (for QXL).
_ is accepted in DOS filenames now, the . for DOS/TOS filenames in SBASIC (no quotes needed anymore). Configurable initial screen size for ATARI.
CTRL ALT SHIFT TAB gives a quick reset - many customers demanded for it. QL EPROM banners are listed now. DEV can be renamed. And much more to come ...

	Feature	ATARI ST(E)/TT		(Super)GoldCard		QXL	
This exists NOW!	New Operating System	NEW	DM 199,- (if you own the QJUMP-drivers Level C, D or E already) else DM 249,-	NEW	DM 199,-	already ex.	DM 199,-
	Multiple, fast BASICs	NEW		NEW		already ex.	
	Flexible Level 3 Drivers	NEW		NEW		already ex.	
	HD Disk-drive support (STE/TT)	NEW		already ex.		already ex.	
	TT Fast RAM support	NEW		impossible		impossible	
	Monochrome Screen-driver	NEW	+ DM 50,-	impossible		impossible	
Later..	New Screen-driver	NEW	+ DM 50,-	NEW	+ DM 50,-	NEW	+ DM 50,-
	"background" Disk/Harddisk	NEW	+ DM 50,-	impossible		NEW	+ DM 50,-
	BASIC-Development-Environment	NEW	+ DM 50,-	NEW	+ DM 50,-	NEW	+ DM 50,-
	Total price (when all is available)	with rebate,	DM 349,-	DM 299,-		DM 349,-	

As a special bonus (we know that many users own more than one system), we offer a full package which contains versions for all three systems for only 33% more. This also applies to the upgrades, e.g. DM 66,66 instead of DM 50,-, which, we think is very fair (it just covers extra disks and manuals).

Applications

QD	New Version	[V7.11]	DM 125,–
QBASIC		[V3.00]	DM 49,90
QDOS/SMS Reference Manual			DM 84,90
4 * Update sheets for Ref.Manual (incl. p&p)			DM 35,00
FiFi		[V3.00]	DM 49,90
QPTR		[V0.25]	DM 92,–
QSUP	New Version	[V3.04]	DM 79,90
QLQ		[V1.13]	DM 69,90
EPROM Manager		[V3.00]	DM 61,50
EasyPTR Part 1 DM 89,–	Part 2 DM 49,-	Part 3	DM 49,–
LineDesign Version 2			DM 249,–
FontPack for LineDesign (100 Fonts)			DM 199,–
HyperHelp for SuperBASIC		[V2.01]	DM 49,–
DISA		[V2]	DM 95,–
QMAKE		[V3.11]	DM 44,90
QMENU	New Version!	[V6.10]	DM 39,90
typeset 93-ESC/P2 (Stylus driver for text87)			DM 69,90
QLiberator	Price Cut	[V3.36]	DM 139,–
WINED	New Product!	[V1.08]	DM 49,90
DiskMate	New Version!	[V5]	DM 69,–
QSPREAD		[V1.28]	DM 169,–
QMON/JMON		[V2.11]	DM 94,90
CueShell			DM 95,–
DataDesign V3			DM 149,–

Upgrades

HyperHelp Upgrade from V1	DM 6,–
FiFi Upgrade from V2	DM 6,–
FiFi Upgrade from V1	DM 16,–
QMAKE Upgrade from V2	DM 6,–
QBASIC Upgrade from V1	DM 6,–
QMenu Upgrade from any Version	DM 16,–
QSUP Upgrade from any Version	DM 16,–
EPROM Manager Upgrade from any Vers.	DM 16,–
QD Upgrade from V6	DM 16,–
LineDesign Upgr. from V1	DM 129,–
DISA V2 Upgrade from V1	DM 35,–
QMON/JMON Upgrade from V2.xx	DM 16,–
QMON/JMON Upgrade from QMON only	DM 32,90
QSpread Update	DM 16,–

Games

Diamonds	(Action)	DM 35,90
BrainSmasher	(Strategy)	DM 45,90
Arcanoid	(Action)	DM 35,90
Firebirds	(Action!)	DM 35,90
SuperGamesPack	(5 various games)	DM 90,–
QShang	(Strategy)	DM 45,90
MineField	(Strategy)	DM 39,90
The Oracle	(Strategy)	DM 49,90
BlackKnight	(Chess)	DM 119,90
DoubleBlock	(Tetris)	DM 42,90
Lonely Joker V2	(6 Card games)	DM 59,–
Lonely Joker V2 Upgrade from V1		DM 29,–
Pipes	New Game!	DM 29,-

If you require more information about any product, then please write for a free catalogue!

TERMS OF PAYMENT

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

EUROCARD. VISA Access MasterCard Cards

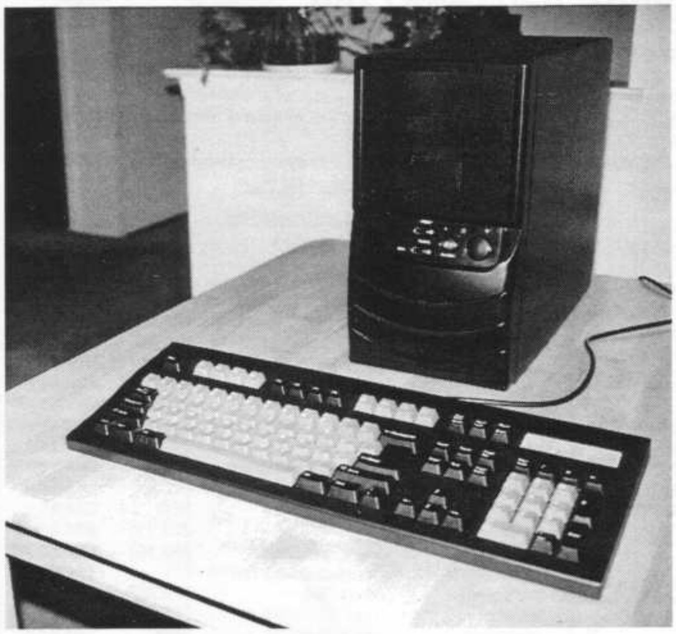
UPDATES

Updates of our software are usually free. The exception: major changes on a program (a new Version number before the '.'). Always send the master disk(s) to us, together with 4 international reply coupons for up to 5 discs or 8 IRC's for more. If you send updates together with a software order, then the return postage is covered by the wholesale postage. If a disk is faulty, add 1 IRC for a replacement. As the software changes from time to time, you may order a new manual together with the disc update. With upgrades, you automatically get a new manual.

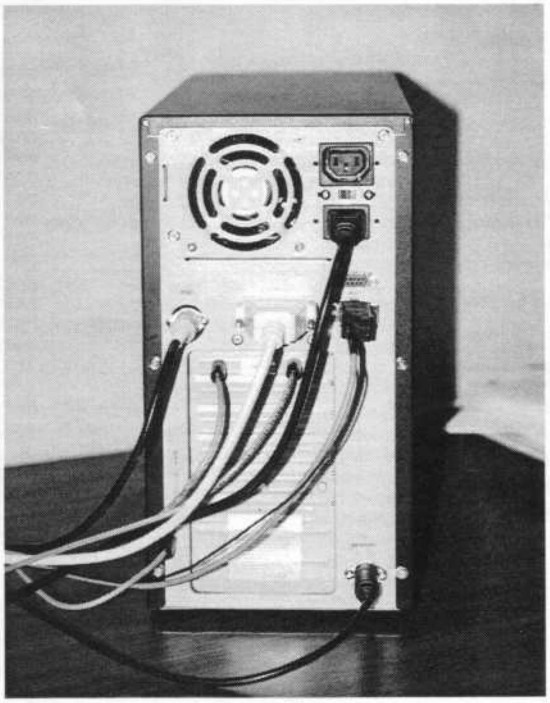
"BLACK BEAUTY"

Shelby Township, Michigan, USA - John J Impellizzeri

Bob named it. After helping Don Walterman build and review the Falkenberg DIY kit to put a QL into a PC type mini tower case and seeing the system that Bob had built for his own QL that was all in black, it inspired me to want to do the same.



fully assembled Tower cased QL



rear view, showing controlled arrangement of all the QL's cables

Bob had wanted his tower-cased QL in black with a black keyboard and mouse. When teamed with an original QL Vision monitor (in black obviously), it makes for a unique looking computer that retains some of the QL's original character while avoiding looking like just another PC clone in their 'putty' colored cases.

Finding the required hardware in that color was hard to do at first and when you did find it, it was expensive. I held off for awhile. Once my QUBIDE arrived and I got it hooked up to a hard drive in an external case, along with the external floppy case, I took a look at the maze of cables and devices on my desk and decided that now was the time to finally put it all in one case. I really wanted to do it all in black but the hardware wasn't much easier to find or cheaper. I had almost settled for the usual PC color when I stumbled across a dealer at a local computer trade show who had mini tower cases and keyboards in black available for very reasonable prices. I bought one of each on the spot and began gathering the parts needed to put it all together.

It was well worth the time spent, expense and down time to build my QL into the case. I am glad I went with the all-black look. It has generated many comments especially since I use it with a black mouse and mouse pad and have a QL Vision monitor. When the Miracle Graphics Card becomes available requiring a switch to a new monitor, I will search for a new monitor in black to keep the look. I have seen a few available and I am starting to see PC systems already assembled in black. Most seem to be marketed using the 'unique looking' angle.

One other thing I did when I set the case up was to set the digital speed display to read either the Super Gold Card's clock speed of 24 MHZ or a seven segment rendition of 'qL'. A push of the otherwise unused Turbo button on the PC case changes the display to read either. When it is showing qL, it makes most people do a double take trying to determine what numbers it's trying to show. I simply say it says QL and that's all that needs to be said!

The Greek New Testament (on the QL)

Sutton Coldfield, West Midlands, UK - George T. Morris

1. Those of us who have an intense desire to understand the reasons and purpose of life, search, study and discuss what we can of the contents of many different records of the past, including the Sumarian Scripts, the Egyptian Cuneform writings and the Mayan Records. Many of these writings are to-day being studied by scholars in the hope of resolving the enigmas concerning the origin and purpose of man himself. We know that many areas of man's observation and past experiences are to-day being analysed, not as myths but as a record of reality. Beside the inner world of his mind and its power there is the life and influence of the Sun and Stars, not to mention the apparent unnatural condition and surface features of some of the planets, such as Mars and Jupiter, the anomalies which stand against the conventional interpretation of the Moon's origin and position, movement and surface features. Then there is the question of true origin and organisation of life, and not least, the enigma of the possible existance of A God, His identity and purpose, His/Her character and apparent blatant indifference to our need.

'Old hat' some of you say, 'I worked out my answers years ago, no need for further challenges now I am on the wrong side of 40. I am out to enjoy life and let the 'truth' look after itself!'. You will not be interested then, as we who are young in heart, in resolving such problems as:

1. Is the earth flat and supported on four very large elephants ? (There is still a Flat earth Society, you know!) Would I tip over the edge at the North and South Pole ?
2. What is death ? Am I still concious when I die ?
3. Has the speed of light always been constant ? Was it and is it infinite under certain conditions ?
4. Who built the city on Mars ? After whom was it named ?
5. How does the 'soul' differ from the body ? What is spirit ?
6. Has Evolution ever been proved, or is it rightly made up of misrepresentation, willful disbelief, hoaxes and lies?
7. Why was The Migration of Species so easy.
8. Does Sin exist ? If it has no reality why do we suffer disease, kill one another and war for gain. Did a man die to free us from all this horror ?
9. Are we the only living creatures in the universe ? Does E.T. exist, is he a German/Russian/American in disguise, could he be a bionic robot living in inner earth or is he interdimensional?
10. Can man time travel ? Or are we limited to only viewing and receiving second hand reports ? Etc., etc., etc.

'You cannot be serious' you say. Oh! yes, but I am...... Very serious. Just as the scholars rely upon those ancient writings as being fact, There are many that rely upon the Hebrew writings and the Christian writings as fact. Modern critics search these writings for errors, but those who know them well, know them to be the source of absolute TRUTH. I am one such person and it is for this reason that I wished to make them available to all other QL users. Why should they not have the opportunity of the uplifting joy of new found truths? One can always hope that others will find in them real answers to such questions as those above.....or does this sound too much like hard work and pie in the sky, eh ?

2. That was the motivation, what then, was to be the course of action ? The first and most straight forward objective was to provide users of the QL with the English version of the Christian Scriptures............The well known Standard King James Bible. A compressed PC version was obtained from The University of California at Irvine and willingly transposed to QDOS by Dr.Bill Fuggle. You can purchase this on seven DS/DD discs for a nominal fee from Dilwyn Jones Computing. Below is an extract from this K.J. version of the English Bible, the first five verses of the Gospel of John. This is the translation of the Greek examples that follow:

JOH 1:1 In the beginning was the Word, and the Word was with God, and the Word was God.
JOH 1:2 The same was in the beginning with God.
JOH 1:3 All things were made by him; and without him was not any thing made that was made.
JOH 1:4 In him was life; and the life was the light of men.
JOH 1:5 And the light shineth in darkness; and the darkness comprehended it not.

The Greek New Testament (on the QL) - (cont'd)

3. As excellent and as freely available as was the K.J. version, from long experience I knew that there was deeper understanding to be gained from the original Received Greek Texts, i.e. The Koine Greek, particularly using the quick reference facilities of a computer. So I took a deep breath and plunged in by purchasing the English Keyboard Code of The Koine Greek from the same University in California. Again, it was provided in compressed form suitable for PCs. Bill again kindly did the basic work and provided me with the code in ASCII format which, at that time, I imported into the XCHANGE version of QUILL.

4. The special code for each of the 27 files for the books of the N.T. was made up of Capital Letters and various Keyboard Signs. These indicate the letters of the greek alphabet with their capitals, accents, breathings, iota subscript and puntuation marks. Every line had its verse number:

```
John 1:1  *)EN A)RXH=: H)=N O( LO/GOS, KAI\ O( LO/GOS H)=N PRO\S TO\N
John 1:1 QEO/N, KAI\ QEO\S H)=N O( LO/GOS.
John 1:2 OU(=TOS H)=N E)N A)RXH=: PRO\S TO\N QEO/N.
John 1:3 PA/NTA DI' AU)TOU= E)GE/NETO, KAI\ XWRI\S AU)TOU= E)GE/NETO
John 1:3 OU)DE\ E(/N. O(\ GE/GONEN
John 1:4 E)N AU)TW=: ZWH\ H)=N, KAI\ H( ZWH\ H)=N TO\ FW=S TW=N
John 1:4 A)NQRW/PWN:
John 1:5 KAI\ TO\ FW=S E)N TH=: SKOTI/A: FAI/NEI, KAI\ H( SKOTI/A AU)TO\
John 1:5 OU) KATE/LABEN.
```

(HERMENEUTIKA CODE FOR THE GREEK N.T. e.g. JOHN'S GOSPEL ch.1 verses 1 to 5)

5. Firstly, I wanted to make the greek N.T. available in QUILL because of its simplicity and general availability. At first sight, the task of changing this code to greek for use in QUILL seems to be formidable. So it would have been, but for a simple and 'easy to use' programme called EPSCOMFOUNT designed by Howard Clase specifically for use with QUILL. I could continue in my usual elementary way taming The Magnificent Beast by using this programme. It not only allows you to design any new fount, both for the screen and EPSOM(compat) printer, but also enables you to patch it into QUILL in place of the Standard SuperScript fount. It prints in draft quality only(This can be improved. See below). Unfortunately the programme cannot patch QUILL with the typeface design. But this is only a slight disadvantage, overcome by having the typeface file present on the QUILL disc. When loading QUILL, the GREEK_eps file is down loaded to the printer by a command in the boot file, provided the printer is 'on line' at the time. This EPSCOMFOUNT programme is freely obtainable by members of QUANTA from their large free library and includes a Russian and a number of alternative English screen founts and printer typefaces. It is also available with other Howard Clase programmes on a PD disc.

6. A quick look at the Character Set list in the QL Manual shows that the number of greek letters is very limited, none of course being on codes suitable for our purpose. Firstly we must determine how many....

[1] different numbers, signs, lower case letters, capital letters and letters with accents, breathings and/or subscript of the New Testament Greek characters could be used considering...

[2] the limited number of signs, numbers, lower case and capital letter keys on the keyboard(94) and...

[3] the limited number of 'standard' ASCII codes available(33 to 126 = 94). Because of this limitation of 94 codes, the accented letters could not be used. These letters give help in standardising the pronunciation of the text, but fortunately they do not affect the technical accuracy.

7. From the QL Manual, the ASCII code list for the codes 33 to 126 inclusive was copied out together with their keyboard letters and signs. The numbers 1 to 0 will be needed for the verse numbers, so these can correspond to the standard keys. The required Greek letters and signs were applied to the remaining codes, in the nearest way to correspond with English i.e. a,alpha/b,beta/d,delta etc. This is not a direct application because some Greek letters are the equivalent of combined english letters: th, theta (θ) / ph, phi (ϕ) / ch, chi (x) / ps, psi (ψ), duplicated in the capitals. Examples of this list of codes and letters looks like this:

SCREEN
Fount:
GREEK_chs

PRINTER
Typeface:
GREEK_eps

ASCII code	KEYING	DISPLAY LETTER or	GREEK SIGN
33	shift 1	!	ᾳ
34	shift '	"	(spare)
35	shift 3	#	ᾁ
34	shift 4	$	(spare)
37	shift 5	%	ἐ
38	shift 6	^	ἑ
56	8	8	8
57	9	9	9
58	shift ;	:	·
88	shift x	X	χ
89	shift y	Y	ψ
90	shift z	Z	ς
91	[	[	[
92	\	\	ῳ
93	]	]	]
94	shift 6	^	ἑ
95	shift –	_	ἱ
96	£	£	ᾀ
97	shift a	A	α
98	shift b	B	β
etc.		etc.	etc.

(EXAMPLES OF ITEMS ON PREPARED LIST OF CODES, LETTERS & SIGNS)

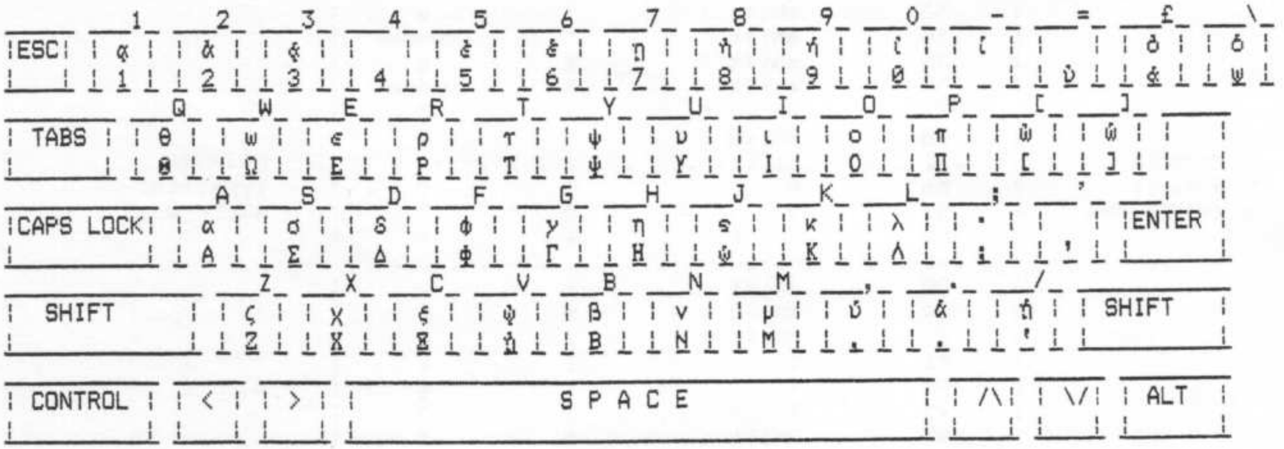
8. The greek letters with the breathings, and/or iota subscripts were grouped on the keyboard in a way which could be easily referable. This can get complicated when trying to arrange those greek letters having both breathings and subscripts. There follows a list of such marked letters:

Definition of Greek letters with
iota subscripts and breathings.

Name of Letter	with	Greek lower case letter only	Key
alpha	iota sub.	ᾳ	!
alpha	rough breath.	ἁ	>
alpha	smooth breath.	ἀ	@
alpha	rough b. iota sub.	ᾁ	#
alpha	smooth b. iota sub.	ᾀ	£
epsilon	rough breath.	ἑ	^
epsilon	smooth breath.	ἐ	%
iota	rough breath.	ἱ	_
iota	smooth breath.	ἰ	)
omicron	rough breath.	ὁ	!
omicron	smooth breath.	ὀ	~
upsilon	rough breath.	ὑ	<
upsilon	smooth breath.	ὐ	=
eta	iota sub.	ῃ	&
eta	rough breath.	ἡ	(
eta	smooth breath.	ἠ	*
eta	rough b. iota sub.	ᾑ	?
eta	smooth b. iota sub.	ᾐ	v
omega	iota sub.	ῳ	\
omega	rough breath.	ὡ	}
omega	smooth breath.	ὠ	}
omega	rough b. iota sub.	ᾡ	j
omega	smooth b. iota sub.	ᾠ	V

(COMPLETE LIST OF GREEK LETTERS WITH BREATHINGS AND IOTA SUBSCRIPTS)

9. At this stage I found it was a great help to have a pictorial view of the Keyboard showing both the upper and lower case greek letters and signs with, for reference purposes, the english letter/number/sign:



1 2 3 4 5 6 7 8 9 0 - = £ \
ESC ᾳ ἀ ᾷ ἐ ἑ ῃ ἠ ἡ ἰ ἱ ὀ ὁ
1 2 3 4 5 6 7 8 9 0 _ ὐ ᾧ ῳ
Q W E R T Y U I O P [ ]
TABS θ ω ε ρ τ ψ υ ι ο π ὠ ὡ
Θ Ω Ε Ρ Τ Ψ Υ Ι Ο Π [ ]
A S D F G H J K L ; '
CAPS LOCK α σ δ φ γ η ς κ λ · ENTER
Α Σ Δ Φ Γ Η ῷ Κ Λ ; '
Z X C V B N M , . /
SHIFT ζ χ ξ ῷ β ν μ ὑ ἅ ἧ SHIFT
Ζ Χ Ξ ἧ Β Ν Μ , . '
CONTROL < > SPACE /\ \/ ALT

(QL KEYBOARD PLAN FOR THE KEYS USING SCREEN FOUNT GREEK_chs)

```
32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54
[]  !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /   0   1   2   3   4   5   6

55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78
7   8   9   :   ;   <   =   >   ?   @   A   B   C   D   E   F   G   H   I   J   K   L   M   N

79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,100,101,102
O   P   Q   R   S   T   U   V   W   X   Y   Z   [   \   ]   ^   _   £   a   b   c   d   e   f

103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126
 g   h   i   j   k   l   m   n   o   p   q   r   s   t   u   v   w   x   y   z   {   ¦   }   ~
```

(LIST of ASCII CODES with THEIR CORRESPONDING LETTERS, NUMBERS and SIGNS)

```
ᾳἀᾷ ἐἑῃἠἡἰἱ ὀὁ   θωερτψυιοπὠὡ   ασδφγηςκλ·    ζχξῷβνμὑἅἧ lower case greek
!@#$%^&*()_+~¦   QWERTYUIOP{}   ASDFGHJKL:"   ZXCVBNM<>? on upper case eng
------------------------------------------------------------------------
1234567890 ὐᾧῳ   ΘΩΕΡΤΨΥΙΟΠ[]   ΑΣΔΦΓΗῷΚΛ;'   ΖΧΞἧΒΝΜ,.' upper case greek
1234567890-=£\   qwertyuiop[]   asdfghjkl;'   zxcvbnm,./ on lower case eng
```

(GREEK / ENGLISH EQUIVALENT LETTERS and SIGNS)

10. In order to design the greek fount for the screen/printer I needed a good printed copy of the complete greek alphabet. This can be found in most Greek Grammar books. I used The Greek Language Notes taken from 'The Elements of New Testament Greek' by J.W.Wenham. The fount designer has a grid of nine high by five wide pixels on which the characture of each greek letter and its capital must be represented. An interesting, but time consuming task. The printer typeface fount can be based upon the screen fount design, but with the EPSCOMFOUNT designer it has the added advantage of a grid width of five wide with the use of intermediate points giving nine positions. This can give a more accurate representation of the printed greek letter. This printer typeface however is limited to eight pixels high.

The following examples of the lower case letters 'B' (beta) and ûx0ûE 'v' (nu) illustrate these differences:

```
    1 2 3 4           1 2 3.4 5        1 2 3 4 5     1.2.3.4.5

      O O               O O      :
2_  O     O           O     O    :
    O     O           O    X     :   _ O       O     O       O
    O O O             O O O O    :       O     O      X     X
    O     O           O       O  :       O   O         O   O
    O     O           O       O  :       O O            X X
7_  O O O             O O O O    :   _   O               O
    O              8_ O          :
9_  O                            :
 SCREEN (_chs)      PRINTER (_eps)      SCREEN (_chs)  PRINTER (_eps)
```

'X' indicates the extra pixel point. Sometimes for the sake of clarity and reasonable proportion, it is not necessary to use all the five horizontal positions of the grid.

(GREEK LETTER COMPARISON between SCREEN FOUNT and PRINTER TYPEFACE)

11. We now come to the task of transcribing the English Code, refered to in paragraph 4 above, using the greek characters. Not being capable of designing a suitable programme, I started by using TRUMPCARD with the Search and Replace facility of XCHANGE QUILL. When GOLDCARD became available I switched to using the same facility in TEXT87. The New Testament was divided into about six large documents and a planned sequence of search and replace commands was determined using the Gospel of John as the test run.

GREEK:	α	β	ν	Α	Β	Ν	5	ἐ	ῃ	ὡ	ς	σ	Σ	Ὁ	Ψ
CODE:	A	B	N	*A	*B	*N	5	E)	H|	W(	S	S	*S	*(O	*Y

(SAMPLE List of HERMENEUTIKA CODES with the CORRESPONDING GREEK LETTERS)

The speech accents: \ / = + were cancelled as a first step.

12. The greek sigma(our letter 's') is in two forms. As the last letter in a word it differs in shape from that used in the word itself. The Search and Replace command changes all the codes, in both places, to the latter form. To then use the same command to change the form of sigma at the end of a word, we must search for 'sigma,space'. This worked perfectly except for words with an 'end of word' sigma at the end of a line. The code, as supplied, has a 'line return' immediately following the last letter. 'Sigma,space' is therefore not recognised and consequently all words ending in sigma at line ends cannot be changed except by means of a very tedious eye search. Fred Tussi of Software87 fame suggested a simple conversion programme for each N.T. file of code to place a space at the line end before the return. He provided this simple programme which modified each file in seconds. This resolved the the last transcribing problem.

```
Ιωαννης 1·1    ʼΕν ἀρχῃ ἠν ὁ λογος, και ὁ λογος ἠν προς τον
               θεον, και θεος ἠν ὁ λογος.
Ιωαννης 1·2    οὑτος ἠν ἐν ἀρχῃ προς τον θεον.
Ιωαννης 1·3    παντα δι' αὐτου ἐγενετο, και χωρις αὐτου ἐγενετο
               οὐδε ἐν. ὁ γεγονεν
Ιωαννης 1·4    ἐν αὐτῳ ζωη ἠν, και ἡ ζωη ἠν το φως των
               ἀνθρωπων·
Ιωαννης 1·5    και το φως ἐν τῃ σκοτιᾳ φαινει, και ἡ σκοτια αὐτο
               οὐ κατελαβεν.
```

(FULL GREEK FORM with capitals, iota subscripts hard and soft breathings. Without accents)

13. Having completed this work with the Gospel of John, I realised that the use of all the capital letters and iota subscripts in the text could hinder its fast computation when using it for reference purposes only. Therefore a simplified fount and typeface was designed, excluding these features, and again applied to the code of the Gospel of John.

```
ΙΩΑΝΝΗΣ 1·1   ἐν ἀρχη ἠν ὁ λογοσ, και ὁ λογοσ ἠν προσ τον
              θεον, και θεοσ ἠν ὁ λογοσ.
ΙΩΑΝΝΗΣ 1·2   οὐτοσ ἠν ἐν ἀρχη προσ τον θεον.
ΙΩΑΝΝΗΣ 1·3   παντα δι' αὐτου ἐγενετο, και χωρισ αὐτου ἐγενετο
              οὐδε ἐν. ὁ γεγονεν
ΙΩΑΝΝΗΣ 1·4   ἐν αὐτω ζωη ἠν, και ἡ ζωη ἠν το φωσ των
              ἀνθρωπων·
ΙΩΑΝΝΗΣ 1·5   και το φωσ ἐν τη σκοτια φαινει, και ἡ σκοτια αὐτο
              οὐ κατελαβεν.
```

(SIMPLE GREEK FORM) For ease of computer reference, all lower case sigmas are shown as the letter 'S'. The breathings are shown but there are no capitals, subscripts or accents. For comparison, this example is printed in draft quality only.

14. The shape of the Greek letters of the original text as penned approximately 1920 years ago are not only differently shaped but have no capital letters, no breathings, no iota subscripts and no gap between words. Therefore another screen font and printer typeface were designed and applied to the code of the Gospel of John.

ΕΝΑΡΧΗΗΝΟΛΟΓΟϹΚΑΙΟΛΟΓΟϹΗΝΠΡΟϹΤΟΝΘΕΟΝΚΑΙΘΕΟϹΗΝΟΛΟΓΟϹΟΥΤ
ΟϹΗΝΕΝΑΡΧΗΠΡΟϹΤΟΝΘΕΟΝΠΑΝΤΑΔΙΑΥΤΟΥΕΓΕΝΕΤΟΚΑΙΧωΡΙϹΑΥΤΟΥΕ
ΓΕΝΕΤΟΟΥΔΕΕΝΟΓΕΓΟΝΕΝΕΝΑΥΤωΖωΗΗΝΚΑΙΗΖωΗΗΝΤΟΦωϹΤωΝΑΝΘΡωΠ
ωΝΚΑΙΤΟΦωϹΕΝΤΗϹΚΟΤΙΑΦΑΙΝΕΙΚΑΙΗϹΚΟΤΙΑΑΥΤΟΟΥΚΑΤΕΛΑΒΕΝ

(ANCIENT GREEK SCRIPT) Each letter is based upon a five by five pixel grid which limits the representation somewhat.

15. There is yet another version in general use which does not require greek founts or typefaces. Each greek letter is represented by the equivalent english letter(s). This gives a rough approximation but is often adequate for reference purposes. The code of the Gospel of John was used to sample this version also.

```
JoEn  1:1  en archE En ho logos, kai ho logos En pros ton
           theon, kai theos En ho logos.
JoEn  1:2  houtos En en archE pros ton theon.
JoEn  1:3  panta di' autou egeneto, kai chOris autou egeneto
           oude hen. ho gegonen
JoEn  1:4  en autO dzOE En, kai hE dzOE En to phOs tOn
           anthrOpOn:
JoEn  1:5  kai to phOs en tE skotia phainei, kai hE skotia auto
           ou katelaben.
```

(ENGLISH FORM) No capitals, iota subscripts, accents or soft breathings. Hard breathings are indicated by the addition of a letter 'h'. A capital 'E' (Ε) indicates a long 'e' as in 'feet'. A capital 'O' (Ω) indicates a long 'o' as in 'tone'.

16. So that I could test the opinion and needs of potential users of the greek text, Dilwyn Jones agreed to let me write to those customers who had purchased the English K.J.Bible discs. I sent them printed samples of the four Greek text versions given above. The replies received asked for the full greek text version shown in paragraph 12.

17. All that now remained was to transcribe the other five New Testament files. Having established the system and using TEXT87 with Gold Card, this did not take long. With use, I was able to see ways of refining the design of both the screen fount and the typeface. I found that the typeface was improved by using text embedded printer commands for Epson(compat) Expanded Print. A READoME_doc was added showing how to use and pronounce the KOINE GREEK text, together with helps for understanding it.

18. My final task transfered the files to TEXT87plus4 and provide a corresponding fount file...GREEK_F87 and READoME_T91 file. Those wishing to improve the printed greek letters can use the Software87 programme FOUNTED89, although the GREEK_F87 file should provide an adequate screen fount and basic typeface.

19. All these greek N.T. files of FULL GREEK FORM are now available from Dilwyn Jones at a nominal cost. Both the QUILL and TEXT87 versions are on two DS/DD discs with the screen fount and typeface files. The QUILL version has a patched QUILL file. Dilwyn added two useful new fount files for 24pin printers.....GREEK_24pin_chs and GREEK_24pin_NLQ_eps. He also added a corresponding explanation to the READoME_doc. Either of the patched QUILLs can of course be used for Greek documents, study or correspondence, as can TEXT87 using the Greek_F87 fount.

20. For my own letterwriting purposes I have now designed and patched QUILL with a combined simple Greek/Hebrew screen fount and typeface. This is particularly useful for comparing and illustrating the Divine Interchange of the corresponding biblical meaning of Greek and Hebrew words. It is also very useful for general correspondence.

If you have remained interested and continued with me this far, there is just the possibility that you may now be asking: 'But what of the future?'......Yes! You've guessed it, I am now attempting to transcribe the O.T. Hebrew code in the same way. As you might imagine this code is much more complex and does not easily lend itself to the straight forward use of the search and replace command. This causes me to progress slowly because the hebrew text code includes numbers identical with verse and chapter numbers up to 95. But, my motivation to discover and recover Truth is as strong as ever, so as I said before, if scholars can take the ancient Sumarian/Egyptian/Mayan scripts as 'gospel', Why should I not benefit therefore, by taking the Hebrew/Christian scripts as Gospel? I know that many have found the results of doing so, to be lifesaving.

21. Finally, let me say that I did not choose the questions at the start of this article without believing that I knew the answers.....well at least, some of the answers. So I give below references which will point you in the direction of the understanding I have gained. These answers are not lightly given, so if at first they seem to be irrelevant, do not give up, take a long harder look.

1. Job 26:7
2. Genesis 2:7 Ecclesiastes 12:7 Romans 5:12 John 1:29

3. The Generations of the O.T. add up to approximately 4,200yrs. It is an erroneous assumption to believe that the decay of the Carbon atom has been constant over time. Genesis 5:1-32, 10:1, 11:1, 11,27 Genesis 25:19, 37:2 Matthew 1:17.

4. Genesis 2:4 Genesis 4:14,16,17
5. Genesis 2:7 1Corinthians 15:45 Ezekiel 18:4 Matthew 10:28
6. Genesis 1:11 Genesis 1:21
7. Genesis 1:9, 10:25
8. Romans 3:23,5:12 1John 1:10 James 4:1 John 3:16
9. Genesis 6:4 Numbers 14:28,33 Philippians 2:10 2Timothy 3:13
10. Daniel 10:14 Daniel 2:45 Revelation 1:1, 1:10

Like so many QLers, there are many things that I would not have done or learned had it not been for the Magnificent Beast. So if anybody feels that they would like to help or criticise this work, or this article come to that, please do let me know.

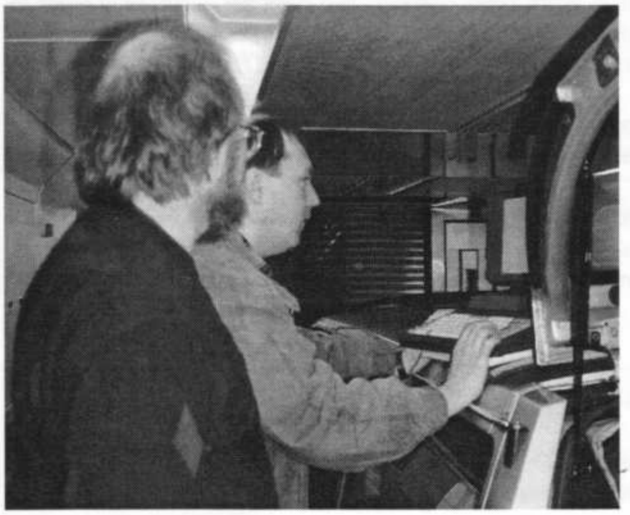
George T Morris
67 Wood Lane
Sutton Coldfield, West Midlands B74 3LS
UK Tel: 0121 353 8571

Going Dutch for the Weekend

Portslade, Sussex, GREAT BRITAIN - Roy Wood

Ah, the best laid plans of mice and QLers! With the best of altruistic motives I offered to transport three members of Quanta in my car to the Eindhoven meeting on the 4th March but my car had other ideas. A couple of days before the meeting it's radiator decided to commit Hari Kari and I found myself looking for a lift to the meeting. Luckily Tony Firshman of TF services had a spare seat in his car so I did a swift bodge job on my radiator (enough to get it from my home in Brighton to the London Orbital Motorway services) and, at 11pm on Friday evening I transferred my QL, monitor and bag of assorted leads into the back of Tony's Volvo and we set off for Holland.

The Journey Begins: For those of you who have never made the journey from our little island to the wide continental mainland I will now make a short digression from matters QL. Until last year, if you wished to take your car to the continent you were faced with a collection of Ferries, Sea-Cats, Hovercraft and Hydrofoils. Ferries were, on the whole, the worst of these options being slow, boring and pretty uncomfortable but now, with much fanfare, we have the Channel Tunnel. It would be pretty difficult to produce a mode of transport more uncomfortable and inefficient than the standard cross channel ferry but the Channel Tunnel has succeeded admirably. When we arrived at the high tech terminal we found it almost completely closed.The only thing open at 12.30 am was the duty free shop and the only refreshment available was from a coffee machine. There were only about twelve vehicles travelling on this train and the travellers were outnumbered by the Channel Tunnel staff who demonstrated the wonder of bureaucratic confusion. (One QL team were unable to board the almost empty 'blue' train because they had run out of blue windscreen stickers and had to wait for the next 'red' train !)



Phil Borman & Tony Firshman conducting the first QL Show in the Channel Tunnel

Tony's car was well laden with five people (Phil Borman, John Gregory, Laurence Reeves, Tony Firshman and myself) and piles of QLs, monitors, Minervas, Hermes, books, tools and a prototype Super Hermes and was too low to go through the carriage without hitting the safety clamps on the floor so we had to drive into the last carriage and back out at the other end. As soon as we were sealed into this metal tube Phil got his QXL powered laptop onto the roof of the car and began showing us what it would do - this has to be the first QL show in the Channel Tunnel!

After three and a half hours on the roads of France, Belgium and Holland we arrived at the motel and, with typical Dutch hospitality were allowed to check into our rooms at least six hours early. After a long and much much needed breakfast at which QL faces popped every few minutes we set off for the show.

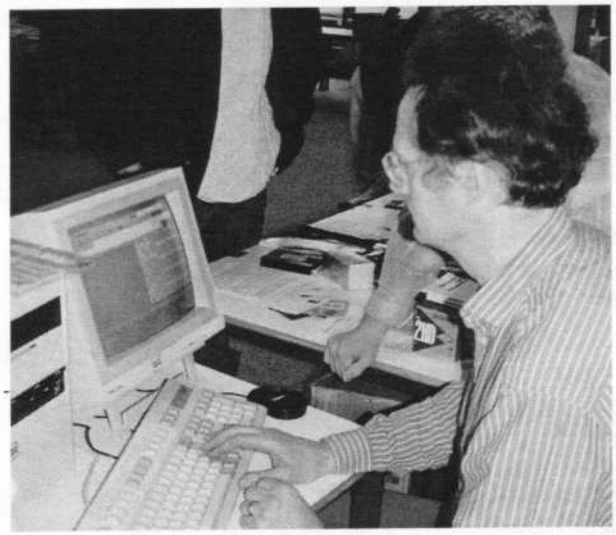
The Show: We were among the first to arrive Jochen Merz had been there for an hour already and was already set up with his Atari TT ready for action and a colourful demonstration flashing away on his Atari Jaguar and Pål Monstad had his PMdata stand set up selling Norback and Disk Mate - now at version 5!). Sin_QL_Air, the hosts of the show, were also set up with a variety of QL's running and occupied an area between the Quasar and the Quanta stands. Bill Richardson had one corner selling Falkenburg interfaces, disk drives and a few of his QL's still in their old silver boxes. T.F. Services had their stand adorned with Minerva and Hermes and seemed to be also doing a few few repairs to other QLs. Beside them was Zeljko Nastasic's Croatian team set up and demonstrated their Backplane and a few graphics disks.

Stuart Honeyball sold out of Gold Cards and Super Gold Cards quite soon and was doing a brisk trade in QXL's and the roars of laughter and jokes that emanated from the stall beside him came came the 'PROGS Brothers' Joachim and Nathan Van der Auwer who also seemed to be in great demand.

Ron Dunnett's Qubesoft stand was selling Qubide interfaces and PD disks (and a box of microdrive cartridges marked 'Lucky Dip 50pence each) and Albin Hessler had a stand selling Cueshell and Easyptr disks.

The QItaly user club were also there in force with the Ergon people (certainly among the smartest dressed of the QLers present) and Freddie Vacha of Digital Precision fame had a stand there too. QItaly have a nice little sticker for tower case QLs that says "QXL Inside", "Cut out this nice logo and stick it on your PC's case in place of that annoying little 'Intel Inside'". Apart from all these traders and clubs there was a large number of users setting up there own systems and then heading off to find someone to sell them this or explain that and, by midday the whole room was a hubbub of voices.

One of the people who was most in demand was Tony Tebby who seemed to be at everyones' stand at the same time flitting from table to table and talking at the speed of light. He came over to my computer and installed a new version of SMSQ/E, rewrote part of my boot file and in passing helped PMdata debug a DM5 problem (which turned out to be my fault and not Pål's - oops) At every turn someone wanted to get him to sit at their computer and explain why something was not working or doing something strange. At one point he wrote a SMSQ/E v2.49 and a bit for someone whose printer kept doing strange things.



Tony Tebby hard at work on MY computer

There were three lectures: Tony Firshman talked about the new Super Hermes and both Albin Hessler and Freddie Vacha talked about their products but I was so involved with the other people at the show that I did not manage to see any of them and I cannot, therefore, give any details. Since I travelled over with Tony and Laurence I got quite a few details of the forthcoming Super Hermes and this looks to be quite a product. Serial mouse input, extra (and completely separate) serial input/output and keyboard interface are all added to the standard Hermes - this has got to be on every QL owner's 'must buy' list.

On the whole all those who attended seemed very happy about the way it went and the atmosphere was very friendly. When they announced the end it seemed that a lot of people would clearly have liked a few more hours to talk.

The Aftermath: So, having repacked the car, we drove back to the hotel. Our little team had not slept at all the night before but we found enough energy to go to Bill Richardson's room for an aftershow party and then to the vast hotel restaurant for a meal. All sorts of plans and ideas were discussed there and Bill, Tony , Laurence, Phil, the Quanta crew and myself sat in the hotel bar at midnight for a final drink to end a good day. When I got back to the room that I was sharing with Drazen Glojnaric, one of Zeljko's team, he was still up, QL humming and copying newly purchased programs onto his hard disk- he was still at it as I drifted off to sleep.

We finally got back to London at 5.30pm on the next day and there was one last little hurdle to be overcome. My car was on the Eastbound side of the motorway services and we were on the Westbound side. There was, of course, no connecting footbridge but we solved this problem carrying all my stuff across the service bridge (having limbo'd under a couple of barriers marked "Staff Personnel Only"). Laurence had spent most of the journey from the tunnel trying to put his name on the high scores of the games on my Psion Series 3 and he held onto it, still playing, even as we walked across the bridge.

I hope this has convinced more of you to go to these events. I learned a lot from the people there (I would have learned even more if I had been able to keep up with Tony Tebby's 100mph delivery) and had a thoroughly good time. If Zeljko and his friends can drive all the way from Croatia then I am sure that the rest of us have no excuse. Support your friendly QL now!

Hardware geniuses at play, Stuart Honeyball of Miracle fame and Zeljko (Nasta) Nastasic creator of Qubide IDE interface.

Nathan and Joachim Van der Auwer (PROGS) seen in a jovial mood (must have had something to do with Joachim's trousers).

The Merry Cavaliers carry MY gear across the service bridge. Where am I ? Taking the photo of course.

Jochen Merz demonstrating one of his PE programs to John Gregory and John Hall.

Publisher's Note: The QL shows at Eindhoven (sponsored by Sin-QL-Air) are quickly becoming the International showcase for QDOS/SMSQ users and traders. This recent show was attended by over 100 users from counties all over Europe. It truly deserves the support of us all.

For the first time, since the birth of IQLR, I'll be visiting with QLer's outside North America, attending the Quanta AGM in Bristol, UK at the end of April (please note "The Town Crier" for details). On my return I'll get ready to attend our own show in Tennesse 10 June (see "An Open Invitation to QDOS/SMSQ Users") The two trips should empty our travel account for this year.

We sincerely hope that in the next calendar year, we'll be able to attend a show at Eindhoven and possibly even one in Italy (I love Italian food... I may even challange Stuart Honeyball's consumption record.) All said and done, we hope to meet many of you in the coming months.

AMADEUS INTERLINK

has arrived!

The ultimate expansion for the QL

Now easily connect up to 255 I/O interfaces to the QL. The Amadeus Interlink system is a local area network capable of linking up to 255 devices such as computers *(PC's & QL's at the moment, others to follow)*, Centronics interfaces (bi-directional parallel printer ports), Sound interfaces, RS232 ports and other useful I/O interfaces.

Gone is the jungle of QL expansion problems! Amadeus system interfaces are housed in small, smart, black enclosures that are easily stacked on their sides or tops. Approx. measurements 4.3"x2.3"x1.2" (112x62x30 mm).

Now it is possible to access *any number* of Centronics interfaces. These are used for fast data transfer to parallel printers or, Lap-Link style bi-directional communication. *Unlike some,* these interfaces are implemented to the - full Centronics Standard, i.e., all error, control, and data lines are connected. All are accessible from software. Like other network interfaces, network printers can shared by all linked computers. Using Amadeus, the QL is quite capable of printing to more than one printer at the same time! The cost for these network interfaces?, just £35.00.

Now another first for the QL, Record and Play back sounds via you computer. Our brand new product, Ama-Sound, is capable of recording and playing back sounds via any networked computer. Recorded files may be edited, stored and replayed. Complete with Microphone, Speaker and software, this interface at just £49.50, represents exceptional value for money.

Now Transfer data between connected computers at high speed. Over seventeen and a half thousand bytes a second are transferred between a Gold card QL and 486 PC!, *impressed?*, well try it with a Super Gold Card! The speed of transfer on a Trump Card QL roughly equates to 3.5" disc drive transfers.

TRIALS TABLE	Bits/second	Bytes/Second	RS232 equivalent
Trump Card QL - 386 PC	40000	5000	55000bps
Gold Card QL - 486 PC	142000	17750	195250bps
Trials carried out with 125,000 size file being transferred from QL RAM drive to PC RAM drive			

Soon Amadeus Protocol advanced inter-computer data transfer and file handling software will provide Amadeus with a highly advanced file serving and data transfer capability. All computers on the system will be able to; access each others disc drives and devices, send messages, handle remote text screens, etc., regardless of machine type.

Soon Fast RS232 interfaces capable of up to 115kbps will be available.

Soon An interface, especially designed for other manufactures or DIY enthusiasts, will be available to provide links for adding your components to the system.

Projects: Fileservers, memory buffers and many other useful interfaces are planned.

Amadeus Interlink has significant implications for the QL. Not only is it possible to link QL's to any number of I/O interfaces, but linking to other types of computer has also become a viable reality. Currently, QL's and PC's are able to access the system. Over a period of time, many other popular types of computer will be linked. Amadeus provides the ability to maximise resources, easily, and most importantly, *at low cost.*

Potential users will obviously be interested in the network capability. Amadeus is capable of shifting data around the network at about 2.5 Mbps ***(unlike some networks there is no usage degrade)***. This may not sound very fast, but when you consider that 2.5 Mbps stands for 2.5 million bits per second, it will become apparent just how powerful the system is, (compare it to your 9600 bps RS232 baud rate).

Low Cost networking, from Di-Ren

Fleet Tactical Command II, QL & PC Versions
Users, please note this programme will be updated in the *near* future to allow usage over the Amadeus Network.

Contact Di-Ren for products catalogue

CARDS ACCEPTED - ACCESS/MASTERCARD/VISA Etc.

Di-Ren - Working for QL's since 1987

Telephone 01922 33580
Monday - Friday 0900-1700 (UK Time)
Answer phone sometimes enabled

⇦⇦For Complete details contact⇨⇨

Di-Ren
59 William St
Walsall, West
Midlands
WS4 2AX, England

WHAT WILL THE QL COLLECTION COST ME?

Just **£179 in total.** There is nothing to add, no hidden taxes, and P&P to anywhere on earth is included (add £5 for Airmail). You save over £2,100. DP recognises that you probably have some titles already (though perhaps not the latest releases), and some may not be of interest to you yet (likely to change when you see them!) - the price reflects this! **THE QL COLLECTION** is worth it even if you only want to update all your existing DP products: however, DP will continue to accept orders for *individual* DP programs at the prices quoted if you really insist.

WHAT EXACTLY IS INCLUDED IN THE QL COLLECTION?

You get the fullest, very latest, most up-to-date releases of all - **every single one** - of the **66** QL programs listed. The software is the finest, and the QL's very best. The titles would cost you over £2,300 (plus applicable P&P) to buy individually - you can check this by summing the prices overleaf, or those quoted in earlier ads. The only titles omitted are Mega Dictionary (only for 2Mb RAM systems - add £15 for it), MS-DOS v6.22 upgrade (add £90) and any less capable variant of a title that is itself included in **THE QL COLLECTION** (e.g., since the top-of-the-range PROFESSIONAL PUBLISHER is included, DESKTOP PUBLISHER is obviously excluded). You get both versions of PC CONQUEROR (as the list shows) so as to cater for all hardware variations. You may never ever need to buy another QL program again. The range of software you will get is truly staggering. It is too good to be true. But it is true - while the offer lasts....

WHAT ABOUT THE PROGRAM DOCUMENTATION?

All the latest applicable documentation (lots and lots of it) is included on disk, and can be read and printed using Perfection Special Edition or Editor Special Edition, which are also both included, and which can - of course - be used to search, browse, analyse or "edit" manuals at your leisure. Printed copies may be bought later if wanted - full details are sent with the order.

WHY CAN'T I FIND THE CATCH IN ALL OF THIS?

Because there isn't any. DP, whose QL commitment continues, makes this super offer to celebrate the birth of a marvellous baby daughter Michelle, now through all her early problems, to Julie and Freddy. **THE QL COLLECTION** is licensed for use by the purchaser alone, who by buying it agrees not to resell or otherwise pass on any part of it, or of any DP software already possessed. Technical support is negotiable: full details are supplied with the order for you to take up should you want to do so. DP *reserves the right* to withdraw **THE QL COLLECTION** offer at any time later than 14 days after your receiving this magazine, so please do hurry.

WHAT HARDWARE WILL I NEED TO RUN THINGS?

You will need a twin disk drive (DD, HD or ED, 3.5" or 5.25"), lots (123?) of blank disks and over 1.5Mb RAM (Gold Card, Super Gold Card, QXL, ST/QL and equivalents) to fully use **all** the software. The vast majority of titles will, however, work on much smaller systems: earlier DP ads indicate with precision the minimum hardware needed to run each program. If no disk size is specified when ordering, DP will assume 3.5" DD. If you do not yet have a powerful enough QL system, you may wish to contact a hardware dealer and buy, say, a second-hand Gold Card and/or twin disk drive (preferably 3.5" DD or HD - avoid Mitsubishi, and if HD or ED, ensure 100% compatibility with all Gold Cards is *fully guaranteed* by the supplier) as needed.

HOW CAN I GET MY COPY OF THE QL COLLECTION?

THE QL COLLECTION can only be obtained directly, by posting your order (including payment of £179 by cheque drawn on a UK bank / building society, Eurocheque or postal order, or quoting a VISA or MASTERCARD credit card no: and card expiry date) as soon as possible to:

DIGITAL PRECISION LTD, 222 THE AVENUE, LONDON E4 9SE

THE QL COLLECTION

When It All Started

Santa Clara, California, USA - James D Hunkins

How does the QL really stack up, then and now?

A co-worker dropped a copy of an old PC World on my desk a few weeks ago, asking me if its date was about when the Sinclair QL was first introduced. The year on the magazine, 1984. Thumbing through the magazine, I found it interesting to compare where computer technology stood when the QL was first introduced versus where it is today. As I am prone to do, I will share some of the highlights.

An Overview of the Competition: 1984 was a turbulent year in the home/small business computer market. Back then there were many players. The lower end included of course Sinclair (and Timex) with the Spectrum (TS2068) and the ZX81, the Commodore 64 and Vic 20, TI-99/4A, ColecoVision, and Intellivision. At the higher end of the market stood the IBM PC XT and the Apple II.

Two of the major computer introductions that year were the IBM PC Jr and the Sinclair QL. IBM was trying to move the PC into the home market and Sinclair was trying to move into the business sector.

Computer	***QL***	***PC Jr***	***PC XT (& compatible)***
Processor	68008	8088	8088/8086
Graphics			
b&w			80C 25L
color	42C 25L [4] 84C 25L [8]	40C 25L [4/16] (80C 25L [4/6])	(40C 25L [16])
Sound	limited	3 voice	limited
Memory	128K (1 Meg)	64K (128K)	128K (640K)
Storage			
tape	2-microdrives 100K	cassette	---
cartridge	1-ROM type 2K-16K	2-ROM type 8K-64K	---
disk	---	5 1/4" option 320K	1(2) 5 1/4" 320K
I/O	2 serial 2 joystick television BW/color mon	1 serial 1 parallel 2 joystick television color monitor	1 serial 1 parallel BW/color mon
Other	expansion port	internal modem (optional)	internal expan. card slot
Software	4 business apps SuperBASIC QDOS	BASIC MS-DOS 2.1	

Table 1: Basic System Configurations [# of colors] (maximum or optional) C=columns L=lines

In the interest of brevity, the scope of this article will be constrained to a comparison between IBM desktops and the Sinclair QL. Table 1 compares the hardware features of the Sinclair QL, the IBM PC Jr and the IBM XT (and compatibles) as they were originally shipped.

Hardware Comparisons

Casing/Keyboard: the casing/keyboard combination was probably the most distinctive feature of the QL's physical design. By incorporating a type of membrane keyboard into the actual computer case and using a tight layout with custom chips, costs were held down and the computer was nearly small enough to be a 'portable'. The PC desktops used a larger case, primarily allowing for internal expansion cards (versus the external QL card concept) and disk drive(s). The PC also used an external keyboard.

Now-> Some QL Users today use optional higher quality external keyboards and PC style cases to hold their collections of expansion devices.

Mass Storage: At the time of the QL's design, floppy drives were expensive. To keep costs down, Sinclair took their microdrives and incorporated them into the QL for mass storage. Even with the price drops of floppy drives in 1984, IBM only shipped its more expensive PC Jr and the XT systems with disks, leaving the entry level PC Jr with only a cassette tape mass storage interface. To upgrade the PC Jr to a single drive (360K capacity) would cost over $300.

Now-> QL's and PC's can have up to four floppy 3 1/2" drives (1.4 to 2.8 Meg capacity each). Hard disk options exist for both (20 Meg and UP!!!)

Memory Expansion: Memory expansion for the QL was allowed through their plug in side card port, with a maximum capacity of 1 Meg of DRAM. The PC Jr was limited to the 128K that the QL originally shipped with. The PC XT at that time could only use 640K (normally shipped with 128K to 256K).

Now-> The original QL can now have up to 4 Meg (due to a replacement of the original processor and some of the control circuitry). QXL cards and emulators can be expanded even further. The PC XT systems were allowed limited expansions with special add-on cards and software extensions which were NOT commonly used by most programs.

Other: Other devices included joystick ports built into the QL and PC Jr (option on the XT), a new mouse option on the IBMs (only $195 from Microsoft), and cartridge program loading on the QL and the PC Jr.

Now-> the mouse is finally common on the QL systems through the use of the serial ports and software (and costs as low as $20). PCs no longer offer cartridge programs, while many QL owners still use cartridges. However, with memory expansion, many of these QL programs are now being loaded directly into expanded memory for speed improvement.

Considering the above, it is interesting to note that the 10 year old QL is still in use today, sometimes in its original form but much of the time with multiple hardware and software enhancements. The PC Jr, on the other hand, was short lived. The PC XT is now entering its 5th generation (Pentium - 586 based) where owners have had to scrap their original systems for brand new computers every few years just to keep up with the software performance requirements.

Software Comparisons

Software: The QL was originally shipped with four business programs in addition to its built in structured SuperBASIC. SuperBASIC expanded the language to include many properties of 'C' and Pascal but maintained the easy use of tradition BASIC.

The business programs were integrated (could share data with each other). They included a word processor, spreadsheet, data base, and business graphics. At the time these programs were considered to have fairly complete feature sets. For new users, their visible 'menu' prompts and on-line help were invaluable.

The PC Jr included a rudimentary version of BASIC within the on-board ROM memory. An optional cartridge based BASIC added access to its color graphics, sound capabilities and other features. All other software had to be purchased separately.

In an interesting side note, PC programs in those early days were still fairly compact. WordPerfect and Word (word processors) and Lotus 123 (a spreadsheet) were still available on single 360K disks. Today some PC programs have to be shipped on CD-ROMs or on up to 26 1.4 Meg disks. QL software normally came on single microdrive cartridges of 100K maximum. Today's biggest QL software package that I have found is LineDesign (a graphics program). It came on a single disk and included nine 720K extra disks containing clip-art and fonts. Granted, PC programs often are more complex (yes, people do use some of those extra bell and whistle functions, but only a relative few) and may come with hundreds of fonts. But stripping away the extras, they still take ridiculous amounts of disk space and memory.

Operating System: The PC Jr and XT were shipping with MS-DOS 2.x as the main operating system. Users also had the option on some XT compatibles to run CPM (a popular operating system just before MS-DOS). MS-DOS was a command line single-tasking operating system. In 1984 MS-DOS was limited to a maximum of 640K memory access. It could only run a single program with a few background drivers at the same time.

The QL on the other hand was shipped with QDOS, a fully multi-tasking operating system. It was also command line driven but came with SuperBASIC built-in and incorporated as part of the command line interface. This greatly extended its capabilities for system level operations. Its multi-tasking allowed it to run several compiled programs simultaneously along with a single SuperBASIC interpreted program. QDOS also had the unique ability to be easily expanded by adding or replacing keywords, drivers, and new types of devices. To fit in a limited ROM on the QL, QDOS opted many times for smaller code at the loss of some speed. Also, as memory was limited, windows were destructive (in other words, a window over another one would not save the original. It was up to the software to restore the buried one when it was switched to).

It has been fascinating to watch the two operating systems grow up over the years. MS-DOS is now at version 6.22 and has a software package that sits on top of it to give a limited multi-tasking capability. Other true multi-tasking systems are just now becoming more common on the higher end (and expensive) PC systems of today. These operating systems have the advantage of supporting better graphics and sound. They also have a larger selection of programs that can run under them. However, these other operating systems are slower running and much more complex to write for. Disk space and memory is consumed in huge amounts.

QDOS today has also grown, but not by being replaced with newer versions. It instead has had new commands, things, and drivers added to it to take advantage of the faster processors and/or expanded memory that have become available with the add-on cards, replacement chips, the QXL card and the emulators. QDOS today is still fully multi-tasking (about time that the PC world is finally experiencing this kind of power) and user friendly. With the pointer environment enhancements, it is now mouse driven and has non-destructive windows. Its only lacking is in the number of colors available (still only eight plus dithering patterns). But as was mentioned, it is easy to program for, the languages either come with the system or are free or low cost, and the programs are very memory and speed efficient.

One more note on QDOS, its first major rewrite is just now appearing as SMSQ/E, which incorporates all the above and more. In the tradition of QDOS, SMSQ/E also is designed to allow enhancements in the future without requiring a redesign. Comparing this to MS-DOS, QDOS has had just one rewrite 11 years after its introduction, while MS-DOS has gone through five major upgrades since 1984 and requires a totally different program on top of it to even start to approach the multi-tasking abilities of QDOS. And to really match QDOS in multi-tasking, it must be completely replaced with an entirely different operating system such as OS/2 (or Windows 95?).

Again, in comparison, QDOS was and is an advanced operating system that has yet to outgrow itself. It still maintains a lead in the ease of programming for it and in the compact size of it and programs designed to run under it.

In Summary

I believe the QL had a suggested list price (SLP) of ~$600, which included SuperBASIC and the four business programs. The PC Jr came in two configurations. The 40 column screen, 64K memory, MS-DOS 2.1 and cartridge BASIC configured PC Jr had a SLP ~$1000 (street price of ~$670). The 80 column, 128K memory with disk drive configuration had a SLP of ~$1600 (street price ~1270).

When It All Started - (cont'd)

Sinclair was a common name in Europe, while IBM was big in the USA, so both companies were at least somewhat solid. IBM of course was definitely larger with its world wide mainframe business.

Considering the hardware feature list, included software, and the price, the Sinclair QL definitely had a chance to do very well against its competition.

Eleven years after the QL's introduction, despite the original delivery and marketing problems, and despite the IBM juggernaut that has literally rolled the PC standard right over most competition, we see today a comparatively small but solid base of QL users daily using their favourite machines.

My personal selection of the technical reasons behind the unusual survival of the QL Community while so many others have folded are:

. the power and flexibility of the QDOS operating system
. the expansion capability of the original hardware (expansion ports, processor,and drivers)
. the ready availability of technical information
. the ease of programming

But, as we are all aware, technical reasons may be enough to get a system introduced. But they are never enough to ensure survival. The main reason that I feel the QL is alive today and will be alive tomorrow:

"The solid dedication of the QL Community of users and developers."

Professional & Graphical Software

Amelia Iowan Old Style **BREMEN BOLD** Kuenstler 480 ∞ΔΓαβ Goudy Handtooled
Caslon Oper [illegible] ion Bodoni
Revival 565 **SHOTGUN**
Shelley Alleg Staccato 555
Venetian 30: agull Bold
Calligraphi iel Gothic
Carmina Bold Italic Olmysteel **INFORMAL 011** Lydian **UMBRA** Greeknku 591 **Allegro**
Old Dreadful No.7 Goudy Old Style Freeform 721 **Dom Casual** Amerigo *Carmina Italic*

PROforma fontpack

A collection of 100 high quality Bitstream® fonts

All fonts have high quality outlines, a full charset, hinting and kerning !

NEWS
- There is now a Sbasic interface for PROforma
- LINEdesign now also has Qfax and LaserJet 4 driver

LINEdesign

This is the program which brought QL computing into the nineties. Finally you get access to a modern technology, "vector graphics." This means that your page will be stored in a mathematical form, and not as a collection of pixels.

With LINEdesign, you can create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can draw lines, curves, circles, ellipses, pies, squares, rectangles, rectangles with rounded corners, and any combination of these to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independant, meaning that the printout will be the same on any printer (e.g. same size and position).

Also LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. You can choose from a large variety of fonts (you get 130 with the program), and they can be displayed at any size, rotation, etc. If the fonts which are given with the program are not enough for you, there is a special program to convert Adobe Type 1 fonts for use by LINEdesign (pfb2pff).

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. To add a graphical touch, you get about 150 clipart pictures, including banners, borders and general purpose drawings. LINEdesign will reproduce everything at the highest possible quality !

LINEdesign is delivered with an extensive manual, which includes a full printout of all the fonts and the clipart given with the program.

PROforma

PROforma is a vector graphics library for C (and Assembler) programmers. It is very powerful, and can be used for any application which needs high quality output. PROforma is the graphics library which is used by LINEdesign, PFdata and PFlist to produce the output. PROforma supports black and white vector graphics and includes :

- clipping paths
- transformation matrixes
- thick lines, grayshades, lines and bezier curves
- filling using even odd and winding rule
- vector (outline) fonts, which can be used in any size. Hinting is used to make sure small fonts look good.
- true WYSIWIG. PROforma can generate output for screen and printer, and the output will be exactly the same on both (with any difference due to difference in resolution).
- bitmaps. Although PROforma is a vector graphics library, you can include classic bitmaps so you can still use your old graphics.

The PROforma package contains a dedicated C68 library to access the PROforma extension thing. It is supplied with a comprehensive manual and examples.

• PFlist : Very easy to use program to create listings on any printer (especially inkjet and laser printers). Can include a footer with filename and filedate. Always allows perforation of your pages. The font and fontsize can be chosen (PFlist uses PROforma). PFlist can create your listings in two columns, and in landscape (or both).

• DATAdesign : This is probably the most userfriendly database program in the world ! Never before has it been so easy to create, fill in and maintain your personal databases. If you want to start a new file, just type the names of the fields. If you want to add or delete a field, no problem, just do it. If you want to change the name of a field. just indicate it.

What's more you can choose to look at only those fields you want, and in any order you specify. And you can select which records you want to view, and which not.

DATAdesign allows you to have some hidden comments for each record, have a general look at the file (in tabulated form) or to transfer a record into the scrap or hotkey buffer, so you can easily import a record in your favorite word processor or editor !

Security is another strong point for DATAdesign. Usually files will be memory based, which offers you maximum speed. However, files can also be disk based, making sure that all changes to the file are immediatly stored on disk, so even in the event of a power failure, you can at most lose the changes to one record !

Naturally, DATAdesign is good at sorting and searching. And if you were using another database, you can convert Archive or Flashback files to DATAdesign.

Wait no longer, start using DATAdesign !!

• DATAdesign API : Unleash the real power of the DATAdesign engine. For programmers, DATAdesign turns into a relational database with a bonus, you don't even need a key field. The Application Programming Interface allows you to use database capabilities without learning a new language, you can just use the extensions to SuperBasic, C or Assembler. The API gives you a unique and powerful record at a time data manipulation language !

• PFdata : Very interesting program for all DATAdesign owners. This program can create hardcopy of your DATAdesign files using PROforma. This means that you can use a large selection of fonts (such as the ones included in LINEdesign), in any requested size. Also LINEdesign pictures can be included to add logo's, boxes, etc. Several records can be printed on each page,...

• prices : PROforma fontpack BEF 4000, PFdata BEF 1000, PFlist BEF 1000, pfb2pff BEF 3000, LINEdesign BEF 5000, PROforma BEF 5000, DATAdesign BEF 3000, DATAdesign API BEF 1000, postage and VAT included. Outside EEC : call! Send Eurocheque in BEF, or your VISA/ EuroCard/ MasterCard details.

for more information, just write, phone or fax !!

Haachtstraat 92, 3020 Veltem, Belgium, tel/fax (016) 48 89 52

Town Crier Announcements of Upcoming Events

To have your Show, Workshop or AGM listed by the Town Crier, send all relevant information to IQLR's North American address. Please note deadline dates for submissions listed on page two of this issue.

30 April 1995 ***(SUNDAY)***

QUANTA AGM

(see advert elsewhere in this issue)

Walton Park Hotel
Clevedon
Bristol
GREAT BRITAIN

7 May 1995 ***(SUNDAY)***

6th ITALIAN QL SHOW

Contact: Davide Santachiara
Tel: +39 522 300409

Reggio Emilia
ITALY

27 May 1995 ***(SATURDAY)***

QL SHOW

Contact: J. Tassett
Tel: +32 02 233 1222

Eurovolley Center
Beneluxlaan, Vilvoorde
Brussels
BELGIUM

10 June 1995 ***(SATURDAY)***

3rd NORTH AMERICAN QL SHOW

Contact: Bob Dyl
Tel: +1 401 849 3805

On Site Contact: Mel LaVerne
Tel: +1 615 483 4153

Faith Luthern Church
1300 Oak Ridge Turnpike
Oak Ridge, Tennessee
USA

!!YES WE'RE BACK!!
THE BRISTOL QL USER GROUP PROUDLY PRESENTS

****THE 1995 QUANTA AGM AND WORKSHOP****

ON SUNDAY 30TH APRIL 1995 AT 9.00AM TO 5.00PM
(AGM AT 4.00PM)

IN THE WALTON PARK HOTEL
WELLINGTON TERRACE, CLEVEDON, AVON
(JUNCTION 20, M5)

ALL TOP QL TRADERS
PRODUCT DEMONSTRATIONS
LECTURES / TEACH-INS
HELP AND ADVICE CLINIC
BRING 'N' BUY
BAR, FOOD AND REFRESHMENTS

A PROFESSIONAL SCHEDULE OF EVENTS ORGANISED
AS A SERIES OF DEDICATED SHOWCASES

- MEMBERS SET UP ROOM
- LARGE TRADERS HALL
- PRESENTATION ROOM

A MARVELLOUS DAY OUT FOR ALL QL'ERS AND THEIR FAMILIES

FOR FURTHER INFORMATION CONTACT MIKE ASHFORD (0117) 9629981

Tips/Tricks & Problem Solutions

Duisburg, GERMANY - Jochen Merz

If you have a problem with SMSQ/E... then it is usually not a problem with SMSQ, but rather a problem with other software. Here is a compilation of the most common difficulties reported when using SMSQ/E.

Installation: There should be no problem installing on a QXL - you replace the original SMSQ with SMSQ/E, that's it. The next step is to remove all the extensions (which are in-built into SMSQ/E) from your boot file that are no longer needed (they just waste memory space). Get rid of LRESPR PTR_GEN, WMAN, HOT_REXT, and DEV.

On an ATARI- replace the old emulator software with SMSQ/E. If you boot from floppy disk, just shuffle the SMSQ/E disk into the drive, if you boot from harddisk, remove the old software from your AUTO folder and put SMSQ/E in its place. Check your BOOT file for surplus extensions as above.

The GOLD and SUPER GOLD Card version is the most difficult of the three - you have to add one line to your BOOT file. The statement **TK2_EXT: LRESPR flp1_SMSQ_GOLD** as the first line of your BOOT, then copy SMSQ_GOLD onto that disk. Check for the extensions listed above that are not required and remove them. Reset your QL and BOOT as usual.

There is not a lot than can go wrong here, unless you use a line like

```
A=RESPR(180000):LBYTES flp1_SMSQ_GOLD,A:CALL A
```

as this will endlessly re-load SMSQ/E and will not carry on with your BOOT file. The LRESPR command is the only one which will detect that SMSQ/E is already installed and not load it again. If it still loads itself on and on, then the program header might be wrong. Try: **PRINT FDAT(\flp1_SMSQ-GOLD)** you should get 4. That's the way that LRESPR will check for SMSQ/E, if it is not 4, it thinks that SMSQ/E is not loaded and will re-load again and again.

Problems with the Miracle SER/PAR adapter: We have to apologise and say, that the serial ports of SMSQ/E are running much faster than normal. We thought you'd like this (and you like getting more date in the same amount of time), but unfortunately some adapters don't like this. A new command **SER_PAUSE** adds a user-definable PAUSE between each byte sent, generating longer stop bits. Not serious, as it will still run at the speed of the standard QL serial ports. But if you have a modem or directly driven serial printer connected, then you'll benefit from SMSQ/E's higher speed, especially when you print bitmap graphics (which usually take a while).

ALTKEYs and ALT ENTER do not work anymore: This is usually a problem for people who have not loaded the HOTKEY System before (anyone out there???). Both features have been put into the HOTKEY System II, and you therefore have to get the HOTKEY job going before these facilities work. Just type **HOT_GO** and your ALTKEYs will work as before (or better).

WINDOWs in general, the WINDOW command & opening a CON creates problems: This is probably only a problem for users who are not familiar with the concept of a well behaved window with an outline. The Pointer Environment requires that the first window which is opened by the job (called the Primary window) has its outline defined so that all other windows of that job (which are opened latter) will fall into that area. Therefore, if we look at the windows which are defined by the WMON (or initial F1 setting), we will find that the outline has to be 512x256, as all three channels #0, #1 and #2 are within this area. The outline has to be set to channel #0, as this is the first window opened. There might be strange effects in compiled programs, which have the "default" #0, #1 and #2 settings, which are opened by the compiler during run-time, then closed and re-opened again. You should not do this in SuperBASIC, as closing 0 is deadly. No point in doing it in compiled (or SBASIC), but people do. Just execute these programs with the "G" (Guardian) option and they should behave better.

Tips/Tricks & Problem Solutions - (cont'd)

WMON and WTV define an outline (which can be re-defined by the OUTLN procedure found in various toolkits, like QPTR). If you don't have this command, don't worry, then you don't need to define an outline. Note that on higher-resolution screens, only windows with an outline defined can be moved with its contents preserved. You can notice the difference quite easily, but you need a higher resolution than 512x256. Try **SBASIC** to execute a new copy of SBASIC, then type in **WMON , 50, 20** and you will see parts of the first window position left lying around the screen. This is because the window had no outline. But, it now has one implicitly. Type **WMON** and it will jump back to its original position, this time leaving no garbage on the screen.

Problems start if you open or define windows which fall partially or completely outside the defined outline. Move the window again **WMON , 50, 20** Then try **WINDOW #2, 10, 10,10,10** and you get an "out of range", as the window falls outside the range of the outline. **OPEN #3, CON_10X 10A 10X 10** will open a window, but will take the whole of the defined outline (of the primary window), e.g. occuping the whole area of all three windows. Easy to check this, **CLS #3** and an area which is definitely larger than 10 x 10 pixels is cleared.

If you have a little problem with QPAC2: Then it's probably very easy to solve! The most common problem seems to be: when I try to print a file from the FILES menu, then QPAC2 suggests an output device which I don't want, e.g. PRT. How can I change this? Quite easy, just set the destination printer device (do it with **SPL_USE**, not with **DEST_USE**, otherwise an underscore will be added) BEFORE you start a FILES program or a BT_SLEEP button which will wake into a FILES menu. Every time, when you start a new FILES menu, it will take the current setting set by SPL_USE or DEST_USE.

If you have a problem with CONFIGuration of a program: Then you probably haven't read the instructions of the program to configure or ignored the on-screen texts more or less, as there really isn't a lot you can do wrong.

The most common fault is that the changes you've made don't become active. The solution: re-load the software which you have configured. In case it is a resident extension (like MENU_rext or QPAC2), you have to LRESPR the configured version again (or just re-boot). In case of an executable file (like QD or QSPREAD) you have to execute it again or HOT_CHP (HOT_RES etc.) it again in case you have put it onto a HOTKEY. If it still does not work as configured, then you've probably not configured the right file. Check for various occurrences of the same program/extension in various subdirectories and make sure that the one which you configure is the one you really load.

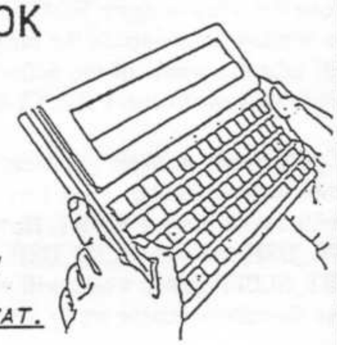
If CONFIG (or MenuConfig) refuses to configure a file by telling you, that the config level is too low, then you should look at a new version of MenuConfig on the disk - more and more programs are getting Config Level 2 information (which makes updating files a lot easier), but CONFIG (and older versions of MenuConfig) cannot deal with Level 2. This is not a problem, as every program supplied by me (Jochen Merz Software) comes with Config Level 2 information embedded with the latest version of MenuConfig.

PRIZES!! PRIZES!!

As an added attraction to the upcoming 3rd North American QL show, we will have a series of drawings for door prizes that have been supplied by the following traders:

Miracle Systems Ltd	Masterpiece Graphics Card
Qubbesoft P/D	Qubide IDE Hard Disk Interface
Progs	DATAdesign 3 PE Database Software
Jochen Merz Software	QD 7 PE Editor Software
Mechanical Affinity	10 Disk set of Clip-Art ported over to the QL
Wood & Wind Computing	QLerk Financial Software Package

All users attending the show will be eligible for the prizes listed (you must be present to WIN!), and we expect the list to grow. See you in Oak Ridge on 10 June 1995 !!



BlackKnight Version 1.4

Oak Ridge, Tennessee, USA - Doug LaVerne

INTRODUCTION

General: For those who didn't see the review of BlackKnight v1.01, BlackKnight is a chess-playing program by Francois Lanciault for QDOS and compatibles. V1.01 had 10 levels of play with time to respond to a move averaging from 5 seconds at Level 1 to 60 minutes at Level 10. To these 10 levels v1.4 adds a Level 11, "Infinite." BlackKnight has an opening library of 5000 moves and will play nearly any chess opening.

I do mean nearly any--from Bird's Opening to the Nimzovich Defence, or the Caro-Kann Defence. In fact, to get it to play something other than an off-the-beaten-path opening, you may have to set up the first few moves!

BlackKnight has other properties I should mention: the ability to set up positions, load and save games, take back moves, play itself (demo mode) or allow both sides to be human players, keep players' elapsed times on clocks, put the chess game to sleep, and move the main window. The Pointer Environment (which is supplied) is required. BlackKnight will multitask.

A useful feature 1.4 added is default devices; now you don't have to type "flp2_" in that small requestor when saving games or loading saved games.

I assume someone thinking of buying BlackKnight would want an idea of how he or she might fare against the program. Some prospective buyers may simply want to play chess and win. Others may want to have the machine play well and instructively, whether they win or lose.

Those chessplayers who consider themselves "woodpushers" but like to win will be heartened to know that on the lower-numbered levels BlackKnight can be beaten, even while it plays respectably. Those who want serious play can also be satisfied (especially if you don't mind, say, the average 30-minute wait for each response at Level 9). In one Level 8 game, BlackKnight played the ugliest-looking move... which I eventually realized fingered every strategem the position dictated for me to try to implement!

This review is mainly examples and summaries of BlackKnight's play in openings, middlegames, endgames, and some difficult set-up positions from various championship and grandmaster games. I have included some comparison and contrast between v1.4 and v1.3. I've done what I can to give BlackKnight a good "wringing out", despite my modest qualifications. Much of what follows is normal chess commentary, such as I used to do in "Tennessee Chess News" in my tournament-playing days.

The technically curious may be interested in this tidbit related to the difficulty of adequately testing a chess package: there are an estimated 25x10**115 distinct 40-move games, while the estimated number of electrons in the universe is 10**79--see Oxford 92, under "openings, number of possible". That also shows that M. Lanciault's accomplishment in creating a chess package is considerable.

The Hardware & Other Parameters, and A Modest Digression: These games were on a QL with a Gold Card, Hessler Serial Mouse with the M.E. LaVerne adapter, QL-90 Keyboard Interface & PC-type keyboard, 2 3.5" HD/DD disk drives of indeterminate make courtesy of Don Walterman, and Acorn Computer color monitor.

Some brief testing on a monochrome monitor from Wood & Wind Computing demonstrated to my satisfaction that one could play on a monochrome system. Black pieces on dark squares were the most difficult to see.

Hermes was installed, and the 68008 removed, prior to my reviewing v1.4. I started putting "EXEC BlackKnight_exe;'350'" in the boot program so as to reserve memory to pull in Quill and write letters, or comments on the game, as the game progressed in the background. BlackKnight would conveniently beep to let me know to come out of Quill and get back to the game for a bit.

BlackKnight - (cont'd)

I have, in fact, had Psion Chess and BlackKnight going simultaneously while writing letters & commentary in Quill. I regret to report that Psion won this first encounter, played with both programs at 1 minute average response time.

BLACKKNIGHT'S PLAY: When playing at 5 minutes' average response or less (Level 8 or lower), v1.3 had shown tendencies to attempt, or to fall prey to, what some would call "coffeehouse" attacks. V1.4, to the degree I was able to test it, did better or at least as well on these lower levels. Since one game at 30-minute or 60-minute response could take several evenings, I don't have any experience to report on at those levels. Perhaps a future issue can carry a (brief) update on that.

Diagram 1 is a position from one of my games with v1.3, which I also gave to v1.4 to see any difference in play. V1.3 did not take care to castle its king into safety and launched a premature attack, which created the exciting (looking) position of Diagram 2. V1.3's attack ran out of steam quickly and Black checkmated about a dozen moves later. However, for a 5-second average response time, this was very creditable play.

```
Diagram No. 1
LaVerne - Black
r..qk..r
.pp.n.pp
.p.ppn..
.P..p...
........
P...PN..
.BPP.PPP
R..QK..R

White -
BlackKnight v1.3
& v1.4 (Level 1)
```

```
Diagram No. 2
LaVerne - Black
r....rk.
.ppq..p.
.p..pNQp
.P..Bn..
Pn......
........
..PP.pPP
R..K...R

White -
BlackKnight v1.3
(Level 1)
After 23.Ne4-f6+
(N-B6ch)
```

```
Diagram No. 3
LaVerne - Black
.....rk.
.pp...p.
.p...n.p
.P.pr...
PRP.p..P
........
..P..PPK
...R..N.

White -
Blacknight v1.4
(Level 1)
After 31.Pc3-c4
(31.P-QB4)
```

V1.4 played quite differently starting from Diagram 1. 1.4 made sure to castle and fight continually for the center of the board as one should. After we reached the position in Diagram 3, I was content to end play and assume BlackKnight could probably draw the game. Again, for a 5 second average response time, this play was surprisingly good. Another game that I tried on both v1.3 and v1.4 used the Ruy Lopez Opening, Exchange Variation. The Exchange Variation is a clearcut, strategic-level struggle, and I wanted to see what BlackKnight would do with the White pieces against a radical approach by Black which I had used once in a tournament game.

```
Diagram No. 4
LaVerne - Black
r..qkbnr
.pp..ppp
p.p.....
....p...
....P.b.
.....N..
PPPP.PPP
RNBQ.RK.

White -
BlackKnight v1.3
& v1.4 (Level 8)
After 5.0-0 Bc8-
g4 (5.0-0 B-KN5)
```

```
Diagram No. 5
LaVerne - Black
..k.....
..p..p.p
p.p..nr.
.p..p.r.
.P..P..Q
N.P.R..P
P..q.P..
.....R.K

White -
BlackKnight v1.4
(Level 8). After
20.Ra1-f1 Rg8-g6
(20.R-KB1 R/1-N3)
```

```
Diagram No. 6
LaVerne - Black
r...r.k.
b.p..ppp
p.p.qn..
.p..p.B.
....P...
.QNP...P
PPP..PP.
....RRK.

White -
Blacknight v1.3
(Level 8)
After 14...Qd6-e6
(14...Q-K3)
```

Starting from Diagram 4, I gave BlackKnight the task of finding a way to execute the traditional Exchange Variation strategies when Black heretically gives up his Bishop pair. V1.4 on Level 8 (5 minutes' average response) played creditably. However, it paid a little too much attention to pawn-grabbing with its Queen. Consequently it ended up allowing a fairly straightforward attack against its King's position. The game arrived at Diagram 5, with Black winning shortly after. The traditional strategies in the Exchange Variation are for White to create and take advantage of a four-to-three Kingside pawn majority in the endgame (since Black's Queenside four-to- three pawn majority is crippled by the Queen Bishop pawns' being doubled), and for Black to seek counterplay through his having two Bishops against only Bishop and Knight. V1.4 did show some awareness of what the position called for at the abstract level, but didn't completely follow through. It would be interesting to see it playing this position on 30 minute average response time.

My radical treatment was to give up the Bishop pair, accept the doubled pawns, and try to keep White from using his endgame advantage. V1.3 had actually fingered all of Black's intentions by playing its Queen to b3 (QN3)--the previously mentioned "ugly move". There the White Queen covered all the white squares Black needed for his purposes (which I won't go into in detail). Diagram 6 shows a critical position from v1.3's response to my Ruy Lopez Exchange Variation. A continuation of this v1.3 game actually wound up on my friend Gordon's Chessmaster 2100 on his Gateway 386SX in Minneapolis. The two sides drew.

BlackKnight in Endgames: I tried BlackKnight against several endgames, the basic sort you have to know how to win after achieving a pawn or piece advantage in the middlegame. In each case, I started the side that had the winning advantage with its pieces as out of play as possible. Play was on Level 8. With a king and queen against a lone king, BlackKnight mated quickly. A king and rook against a lone king also was a quick checkmate. A king and two bishops mated a lone king, despite taking a long time to get started on the job. V1.4 confidently announced "Checkmate in four moves!" And made it. Interestingly enough v1.3 had only drawn the king and two bishops ending, falling into chess's rule that declares a draw after the third repetition of a position.

I gave BlackKnight a standard winnable king and pawn vs. king endgame (White Pawn e2, White King e4, Black King e6) but it only drew. BlackKnight did not show a clear idea of maintaining "the opposition" (keeping an odd number of squares between the kings) or of tempoing with the pawn to do so. Hans Berliner, former World Postal Chess Champion and Carnegie-Mellon computer scientist involved in computerized chess and backgammon, has said the endgame is the hardest part of chess to capture in computerization.

Finding Difficult and Forced Play: BlackKnight had considerable success in the set-up tests I gave it from tournament and grandmaster play. I grilled it with positions from the famous "Gold-Pieces" game and other grandmaster play, from the finals of the 1972 Tennessee Postal Championship, and from a tournament game of mine.

1000 Short gives the "gold pieces" story (possibly apocryphal-- spectators showering the board with coins either in awe or in paying off bets) and calls the brilliancy Frank Marshall created "the most beautiful move ever played", as does Treasury 61. See Diagram 7 for the starting position.

```
Diagram No. 7
Marshall-Black
. . . . . r k .
p p . . . . p p
. . . . p . . .
. . R . . . Q .
. . . n . . . .
. . q . . . . r
P . P . . P P P
. . . . . R K .

Levitzky-White
```

[The 1912 International Tournament at Breslau. White has just played 23.E5-C5 (23.R-QB5) Yes, both Black's Queen and Rook are dangling].

BlackKnight - (cont'd)

I gave BlackKnight the Black pieces in the above position on Level "Infinite." This was about 10pm one night. Next morning as I readied for work it had not moved, so I hit Shift-Esc to force it to give me its "best so far." It found "the move"! Marshall's brilliancy was 23...Qc3-g3!! (...Q-KN6!! One "!" means "good move", two or more mean "brilliancy"). BlackKnight 1.4 had started out considering ...Nd4-e2+ (...N-K7ch).

The curious but baffled should see p. 500 of 1000 Short to find out why, though the Black Queen seems to have committed suicide, it actually cannot be taken... and cannot be left untaken. Levitzky resigned in the above position. V1.3's best, after tries on Levels 8, 10, and "Infinite" was 23...Nd4-e2+ 24.Kg1-h1 Ne2-g3+ 25.Kh1-g1 Rh3xh2?! (23...N-K7ch 24.K-R1 N-N6ch 25.K-N1 RxRP?!), threatening checkmate. White cannot take the loose Rook, but 26.Qg5xg3 (26.QxN) got White out of immediate difficulties, unlike the original game.

In Diagram 8, from the final round of the 1972 Tennessee Postal Championship, things look bad for White. BlackKnight got the mission of finding the save. BlackKnight v1.3 and v1.4 always started out correctly with 25.Ne6xf8+ (25.NxRdis ch) and I always countered with 25...Kg8-h8?! (see Diagram 9), trying to fool BlackKnight by having Black maintain all his threats against White's King, Queen, and Rook.

```
Diagram No. 8          Diagram No. 9
LaVerne - Black        LaVerne - Black
r....rk.               r....N.k
p.....bp               p.....bp
.qb.N.p.               .qb...p.
....p...               ....p...
..B..P..               ..B..P..
.P..n...               .P..n...
PB....PP               PB....PP
..RQ.R.K               ..RQ.R.K

White -                White -
BlackKnight v1.3       BlackKnight v1.3
& v1.4                 & v1.4
```

[Originally Doug LaVerne as White against Glenn Horton as Black. In Diagram 8, Black has just played 24...Nd5-e3 (24...N-K6)].

V1.3 on Level 8, in Diagram 9, stumbled with 26.Nf8xg6+? (26.NxPch?) and lost; however, it did find "the move" on Level "Infinite." V1.4 on Level 10 found the correct 26.Nf8-d7! (26.N-Q7!). As my old notes say, "Faced with a bishop check which not only picks up a pawn but forks K and R while forcing the K onto a square where the Q can be taken by the N with discovered check (don't forget Black's N-fork on Q and R)--White walks right into it!" And BlackKnight found that save.

White keeps a winning advantage in material after 26.Nf8-d7, which is one reason Black played 25...Ne3xc4 (25...NxB) in the actual game. There is a famous game from the 1904 Cambridge Springs tournament between Emmanuel Lasker, World Champion around the (previous!) turn of the century, and Grandmaster William Napier, British Champion. Epic 52 makes the comments "...two masters, both with considerable justification, play to out-combine one another in the same combination.... One of the most beautiful, most profound, most exciting, and most difficult games in the whole literature of chess."

After a certain point in the game, the play on both sides is nearly forced for some time. I gave BlackKnight the burden of playing both sides through this stretch of the game, on Level 10. BlackKnight showed a sense of some of the ideas of the position, but it was too much to ask it to come up with such a game move for move.

I did something similar with a tournament game of mine from 1973. In the actual game, a series of 15 forced moves lead from a seemingly hard-to-win Q+R+R+B middlegame to a readily won R+pawns endgame. Given the winning side in the middlegame position, v1.4 at Level 10 did pare down to a won Rook and Pawns endgame. V1.3 found the first couple correct moves, but repeatedly strayed from the forced win thereafter. It was on Levels 8 and 9.

SUMMARY: The differences in play between v1.4 and v1.3 from the same situation point to a definite change in the programs, algorithms and heuristics. V1.4 is an improvement.

Even on the lower levels BlackKnight will not roll over and play dead. It will come after you. However, it is beatable by an average player on the lower levels. More advanced players can still get a good game out of it on higher levels. BlackKnight does show a sense of tactics and strategy.

It is not entirely evident from what little I've presented here, but BlackKnight places priority on aggressiveness and on getting pieces and pawns on the fifth rank. V1.4 seems more sensible about aggressiveness than 1.3 was.

Incidentally, I have two games going between BlackKnight (playing White) and friend Gordon's Chessmaster 2100 in Minneapolis; also two games between Psion Chess and Chessmaster 2100. In each case, one game is at one minute average response, and the other is at four or five minutes average response.

Rather, I had four games going until this week when 2100 checkmated BlackKnight in one game.

I do enjoy occasionally setting BlackKnight on a minute or five minutes response and having a quick go.

REFERENCES

1000 Short. Chernev, Irving. The 1000 Best Short Games of Chess. Simon and Schuster, New York, NY, 1955. LOC 55-5949.

Epic 52. Coles, R.N. Epic Battles of the Chessboard. Edited and revised by Reinfeld, Fred and Horowitz, I.A. David McKay Company, Inc., New York, NY, 1952.

Oxford 92. Hooper, David, and Whyld, Kenneth. The Oxford Companion to Chess, 2d ed. Oxford University Press, Oxford, Great Britain, 1992. ISBN 0-19-866164-9, LOC GV1445.H616, 83-23733 (1984); British Library GV314.K.

Treasury 61. Horowitz, Al et al. The Golden Treasury of Chess. Compiled by Al Horowitz and the Editors of "Chess Review." Barnes & Noble, Inc., New York, NY, 1961. Everyday Handbook series.

A Note From DI-REN

Walsall, West Midlands, ENGLAND - Robin J A Barker

QL-PC Fileserver: has enjoyed considerable popularity since its release in 1992 but is now showing serious signs of age. The programme has some inherent bugs that to most users are not significant, however, some find them a nuisance. This, and the the more crucial problem of it proving to be incompatible with Super Gold Card, has resulted in our withdrawing this item from general sale. A replacement version, QL-PC Fileserver II, is currently under development and will be available in the near future. This NEW version will include additional features, improved performance, and compatibility with the Super Gold Card. Links to the PC (currently restricted to RS232C), are to include Amadeus Interlink network options and other link formats utilising standard DOS and QDOS device driver software.

The new programme will be available as a low cost upgrade to existing bona-fide QL-PC Fileserver owners. As we do not have the records of QL-PC Fileserver customers, users wishing to receive upgrade information should write us, (not telephone), stating that upgrade information is required, remembering to include your full name and address in BLOCK CAPITALS. (Please note our address in our advert elsewhere in this issue.)

More on Timing

Oak Ridge, Tennessee, USA - Mel La Verne

Eros Forenzi's article on operating speeds (IQLR 4-6) got me to thinking about timings other than those shown. For instance, only three of the four cache and second screen combinations are shown and SMSQ/E is not covered. The following constitutes my feeble efforts to date.

Not having access to the Archive 2.38 and QSI combination recommended, I headed in a different direction, timing several different operations as, at least, a starter. The following Basic skeleton illustrates the approach.

```
                . . .
190      time = DATE
200             FOR i = 1 TO imax
210                    FOR j = 1 TO jmax
220                           (Some operation)
230                    END FOR j
240             END FOR i
250      time = DATE - time
                . . .
                . . .
```

A few initializing lines and some processing and reporting lines, as desired, complete the program. The nested FOR loops are needed in order to accommodate the integer loop variables available in SMSQ/E. The number of passes required ranged from about 320000 to 8E6, far in excess of integer variable capability.

The operations used were three different assignments: a numeric constant, x = 1; an algebraic function, x = SQRT(2); a trigonometric function, x = SIN(1).

The machine setup was Super Gold Card 2.49, Minerva 1.97. For some of the timing, SMSQ/E 2.51 was used.

imax and jmax were selected to give a running time of about 200 seconds for the fastest combination in order to maintain accuracy in the timings.

SMSQ/E declares each of "SCR2EN" and "SCR2DIS" (the second screen enabling and disabling procedures) to be an "unknown procedure". Effectively, this gives us a permanently disabled second screen under SMSQ/E. Consequently, I have lumped the SMSQ/E results with the "Screen2 Disabled" figures.

The numbers in the table bodies were obtained by dividing the total times through the loops (imax*jmax) by the total loop time, giving a quantity proportional to operating speed. This allows comparison of running speeds for the given assignments . Thus, for the case of Cache On and Screen2 Disabled, the simple numeric assignment shows 4416 operations/second for Minerva, 39801 for SMSQ/E with floating point loop variables, and 66650 for SMSQ/E with integer loop variables.

The headings on the following table indicate the type of assignment timed, e.g., "1" indicates the numeric assignment previously defined. Since SMSQ/E ignores the second screen, the "Enabled" lines are empty.

		Minerva			SMSQ/E			SMSQ/E		
		Floating Point Loop			Floating Point Loop			Integer Loop		
Cache	Screen2	1	SQRT	SIN	1	SQRT	SIN	1	SQRT	SIN
Off	Enabled	4093	1808	1315						
On	Enabled	4293	1918	1381						
Off	Disabled	4203	2227	1516	36866	7635	2947	61429	8317	3048
On	Disabled	4416	2338	1606	39801	7711	3065	66650	8442	3216

A final note: If I change line 220 of the above fragment from, say, x = 1 to x = 1: y = 2, the running time will not double. The assignment time will double but the loop bookkeeping time will remain the same. A proper evaluation would account for this overhead; I have not done this.

MIRACLE SYSTEMS LTD

20 Mow Barton, Yate, Bristol, BS17 5NF, UK

NOW you can obtain HIGH QUALITY Pointer Environment software directly from Miracle Systems.

POINTER ENVIRONMENT ESSENTIALS

QPAC2 The definitive front end for the PE	***£40.00***
QPAC1 Offering additional facilities	***£20.00***

TELEPHONE/FAX: (01454) 883 602

PROGS APPLICATIONS

(See their advert for details)

LINEdesign 2 Graphical design package	***£100.00***
PROforma Fontpack 100 additional high quality fonts	***£ 80.00***
PFlist Text listing utility	***£ 20.00***
PFdata Data listing utility	***£ 20.00***
DATAdesign 3 PE alternative to Archive	***£ 60.00***

PAYMENT - Make Cheques or Eurocheques (in Pounds Sterling, drawn on a UK branch of a Bank or Building Society) made payable to **"MIRACLE SYSTEMS LTD"**. Payment by MASTERCARD, VISA or SWITCH is welcomed. Please state your Credit Card number and expiry date (SWITCH card holders, please quote issue number as well).

DEV device driver & DEV Manager

Shelby Township, Michigan, USA - John J. Impellizzeri

DEV Manager is a software utility for setting up and controlling the 'DEV' device driver. While you can do these same things direct from SuperBASIC, DEV Manager makes it easier by using the Pointer Environment to make it a point and click operation.

Before I go too far reviewing DEV Manager, I should explain just what the DEV device is since it is a fairly recent arrival to the QDOS scene. As far as I know it first appeared on the Gold Card. The Super Gold Card also has it as does the QXL.

As the Miracle manual states, the DEV device is a fudge for existing or old software that will not tolerate subdirectories. If you have moved up to ED disks or a hard drive and want to set up your old software to take advantage of the extra storage space you may find that some old titles won't cope too well. Fortunately most new software is written with the new storage devices in mind and will allow you to configure it directly with a subdirectory name. But since there is still a lot of older but still useful software out there that was written before the days of hard drives and ED disks, the DEV device was introduced to allow transparent access without having to rewrite the program.

For instance, you may want to set up the original Psion quartet on your new drive with each program having its own subdirectory for its data files, help files, printer driver, etc. But when you go to reconfigure the software to tell it where to find these new subdirectories, you find that the configure program only allows you to type in 5 characters leaving you with choices like MDV1_, FLP2_, WIN1_, when you really want something like win1_quill_letters_, or flp2_archive_.

Enter the DEV device to the rescue. It will allow both you and Quill (or whatever) to have your own way. From SuperBASIC you would need to enter a line (could be placed in a boot program) such as:

```
100 DEV_USE 1, win1_quill_letters_
```

What this line will do is set up a new device called DEV1_. This device is like any other such as MDV or FLP. You can save or load programs to it, open files on it, etc. However, when you do a file operation to DEV1_, the device driver takes over and translates this to win1_quill_letters_. This allows Quill to be happy because you can configure it for DEV1_ which it will take and yet still have the files actually residing in a subdirectory called win1_quill_letters_. The Gold Card/Super Gold Card allow up to eight of these 'pseudo-drives'. So you could set up a second device as:

```
110 DEV_USE 2, win1_archive_files_
```

With this line, DEV2_ is now really win1_archive_files_ which Archive won't accept but it will take DEV2_ keeping everybody happy.

As you can see, having more than a couple and possibly up to eight of these devices in use could get confusing. You can get a list of the currently active devices and their translations from SuperBASIC by entering DEV_LIST. Given the above two examples and entering DEV_LIST would show a display like this:

```
DEV1_ win1_quill_letters_
DEV2_ win1_archive_files_
```

There are two other SuperBASIC functions which are used with the DEV device. DEV_USE$(x) will return the name of where DEVx_ has been directed. The second function, DEV_NEXT(x), will report the next DEV in a chain. A chain is a list of alternate devices to look on for a given file name if the file is not found on the first device. This is similar to other operating systems where you can set up a search path to other drives and/or subdirectories to look for a file before giving up and reporting an error. A chain is set up by issuing a DEV_USE command and adding a comma and the DEV number to try next after the real device/subdirectory name:

```
DEV_USE 1, win1_quill_letters_,2
```

Since a second device has been chained above, it must be defined:

```
DEV_USE 2,win1_quill_temp_
```

With the above two lines, trying to open a file on DEV1_ will cause a search on win1_quill_letters_ for that file. If the file is not found there, DEV2_ (which is actually win1_quill_temp_) will be searched next for the same filename. This second DEV could be chained further by simply adding a parameter as above.

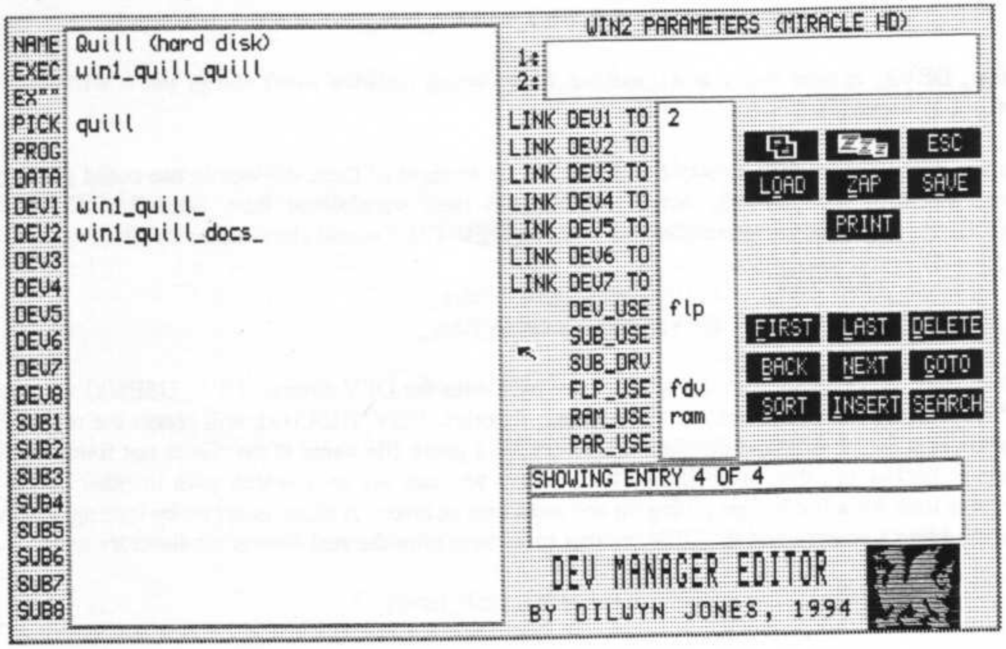
The DEV name itself can be changed to any other which should allow software that will only allow MDV and/or FLP to work with hard drives and subdirectories. If a program only knows about MDV1_ and MDV2_ for example, and won't even allow DEV1_ and DEV2_, then the DEV name can be changed to MDV fooling the program into thinking it is still using MDV1 and MDV2 but instead is using an alternate location:

```
DEV_USE 1, win1_basic_
DEV_USE 2, win1_data_
DEV_USE MDV
```

With this example, when the program calls for MDV1, it will actually get win1_basic_ and will get win1_data_ when it calls for MDV2. As you can see, the DEV device is very powerful and can be quite handy but with those benefits can come some confusion when trying to recall what DEV is translated where, what devices are chained together and what has been renamed to what. Sure you could switch to SuperBASIC and get some of the info, but there is an easier way.

Enter DEVManager, which makes setting up, changing and displaying info on various DEV's much easier. It is actually comprised of two programs, an editor and the manager itself. The editor is used to set up a list of your commonly used programs and the DEV settings that are required to run them. The manager portion is designed to be set up on a hotkey so that it can be called up at any time. It will load in a default data file containing all the information that was entered using the editor. From that point, you simply select the program you are about to use from the list and all the required settings are put into effect. The program itself can even be started right from the DEV Manager.

The hardest part will be determining what settings a particular program requires. There's not much that can be done to automate this so you will have to experiment to see what capabilities the software has or doesn't, as the case may be. Once the DEV settings are known, they can be entered into the list using the DEV editor program. As delivered, the editor contains examples for some common programs. These can be edited or deleted as necessary for your own setup.



```
NAME  Quill (hard disk)
EXEC  win1_quill_quill
EX""
PICK  quill
PROG
DATA
DEV1  win1_quill_
DEV2  win1_quill_docs_
DEV3
DEV4
DEV5
DEV6
DEV7
DEV8
SUB1
SUB2
SUB3
SUB4
SUB5
SUB6
SUB7
SUB8
```

```
WIN2 PARAMETERS (MIRACLE HD)
1:
2:
LINK DEV1 TO  2
LINK DEV2 TO
LINK DEV3 TO
LINK DEV4 TO
LINK DEV5 TO
LINK DEV6 TO
LINK DEV7 TO
     DEV_USE  flp
     SUB_USE
     SUB_DRV
     FLP_USE  fdv
     RAM_USE  ram
     PAR_USE
SHOWING ENTRY 4 OF 4
```

DEV MANAGER EDITOR
BY DILWYN JONES, 1994

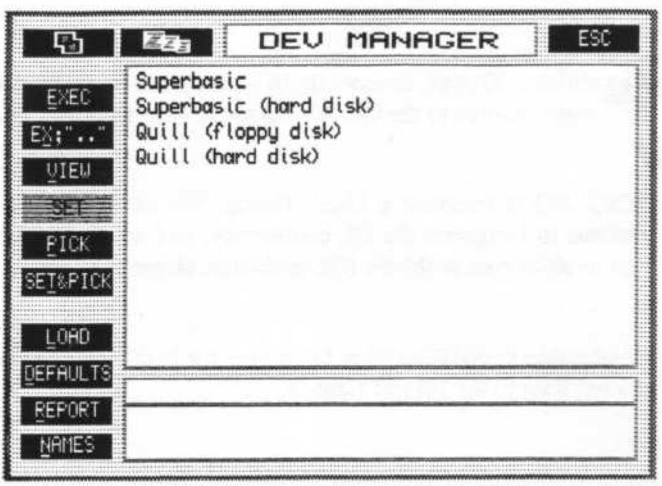
DEV device driver & DEV Manager - (cont'd)

The screen on the previous page, shows the DEV editor program displaying one of the examples which happens to be to set up Quill to run from a hard disk. Using this example, Quill and it's help file and printer_dat file would reside in the win1_quill_ directory while your _doc files would reside in win1_quill_docs_. The NAME at the top left is simply any name you want to use for this setup so that you know what it is for. EXEC and EX"" are used in DEV Manager to start the program. Since some programs allow you to pass them parameters on the command line, this option is provided for. These are simply what you would have typed after entering EXEC or EX to start the program. PICK is used so that DEV Manager can be used to pick, by name, a particular job out of all the running jobs to be the current, or 'on top' job.

PROG, DATA and all the DEVx fields are used to set up the defaults that you want set for this program when it runs. SUBx fields are used to set up the SUB device, written by Phil Borman and supplied with the QUBIDE hard disk interface. SUB is similar to DEV in many ways but different enough to be included here so that it's default settings can also be set by DEV Manager. The WIN2 parameter fields are used to set up the WIN2 device on a Miracle Hard Disk. QUBIDE and Falkenberg interfaces do not use this, so they can be left blank if you use one of these. Below this is where chains from one device to another can be specified. It is also where the various devices can renamed to accomodate the needs of your programs.

Using the example shown, Quill could be run from a subdirectory on win1_ with _doc files in another, while setting up a chain from one subdirectory to the other to look for files. Note that the DEV device has been renamed to FLP. This means that a copy of Quill originally configured to use FLP1 and FLP2 could be transferred to hard disk and run from there using the subdirectories specified without even having to reconfigure it! Also note that the FLP device has been renamed to FDV. This still allows access to the floppy drives in case you need to load a file from there.

The remaining items are used to move back and forth thru the entries in the list, insert new ones, sort it, delete, etc. You can also load in another list or get a printout of the settings in the list in addition to the standard pointer environment items, sleep, window move and escape.



Superbasic
Superbasic (hard disk)
Quill (floppy disk)
Quill (hard disk)

The screen to the left is DEV Manager itself. In the large window there will be the list of programs you set up using the editor program. Next to it are the various options you have. Only one of the top six can selected at a time. EXEC and EX"" are used to start a program. VIEW will present on screen a list of settings for the program you select. SET will cause the settings for the program you select to be put into effect. PICK can be used to bring to the top a program which is already running. SET & PICK combines the two options.

At the lower left are four items that allow you to load in another list (LOAD), or to reset all settings to a known condition (DEFAULTS - specified when you initially configure DEV Manager). You can also get a report on screen of the current settings (REPORT). Selecting NAMES brings up another menu allowing you to change device names on the fly (MDV to FLP, DEV to WIN, etc).

I reviewed DEV Manager using a QL with a Super Gold Card and a QUBIDE hard disk interface. It is said to also be compatible with the Miracle and Falkenberg hard disk interfaces and the Gold Card and QXL. Floppy disks and at least 384K of RAM are required. The pointer environment files which are required are included. An excellent manual comes with it which gives a background on the purpose of the program, how to configure it, set it up and use it. Examples of its use are included as is a short tutorial on the pointer environment. DEV Manager was written by Dilwyn Jones.

I found it to be a handy utility which so far has allowed me to get a few older programs easily running from my hard drive which never did before.

NESQLUG

"The NEW ENGLAND SINCLAIR QL USER GROUP"

NESQLUG is the only active, regularly publishing User Group in the US that supports the Sinclair QL exclusively. We are now entering our sixth year.

NESQLUG began, in the early 1980's, as a 'Timex-Sinclair User' subgroup, under the auspices of the BOSTON COMPUTER SOCIETY. In 1989, we decided to leave the BCS which was, by that time, too involved with Apple and IBM to be very concerned with Sinclair, to establish an independent User Group focused solely on the QL.

NESQLUG holds six general meetings each year from 11 AM to 5 PM, usually on the first Saturday of odd-numbered months, and meetings on special interest topics, from 1 PM to 5 PM, usually on the first Sunday of each even number- ed month. We encourage participation by all members, and time is set aside at each meeting to take calls from members that are unable to be present.

Our meetings are usually held in the greater Boston area, but have also been held in Rhode Island, Cape Cod, and in Canada. A Fall meeting has been held, for the past 3 years, in rural New Hampshire, as a QL weekend retreat at the home of WOOD AND WIND'S, Bill Cable. Bill's name should be familiar to most IQLR and UPDATE subscribers for the many fine programs and articles he has written in and on ARCHIVE. (eg, QLerk, DBEasy, DBTutor and DBProgs)

We publish a 20 page, bi-monthly (6 issues/year) Newsletter that features articles from our members on virtually every topic relative to the QL. We currently have members in 16 US States, Canada and Europe.

NESQLUG is foremost a User's Group. We share a like mind and similar goals to IQLR. Our desire is to continue to invigorate the QL community, and we cordially invite you to join with us and share your interests, your ideas or your problems (QL problems, please).

Membership in NESQLUG is $12 a year for North America and $20 US (Airmail) to Europe. We also offer a $6 Student Rate in the US and Canada.

For a 'trial' copy of our latest Newsletter, (we're very non-profit) please send: $1.50 in US$, check, MO or stamps ... FOR US AND CANADIAN ADDRESSES ONLY, PLEASE. If you subsequently decide to become a member of NESQLUG, you may deduct this amount from your membership dues, eg, $10.50.

We apologize, but due to cost, we aren't able to make this 'trial offer' available outside of North America.

We are eager to exchange Newsletters with other groups having a similar interest in the QL, whether in North America or in Europe.

Edwin L. Kingsley
2 PM - midnight EST

Tel (617) 233-3671
16 Highland Avenue

Editor -- NESQLUG
Saugus MA 01906 USA

QXL In Command

MIRACLE
QXL
SYSTEMS

Bedford, Massachesetts/ Plyesville, Maryland, USA
Al Boehm Tom Robbins

QXL PC Interaction:A program can be started in the QXL and it will continue computing when control is passed back to the PC by pressing CTRL SCROLL LOCK. This feature is handy when using a long number cruncher type of program. Of course, the program will pause when it tries to print to the screen, access a floppy or the hard disk, or is waiting for an input. Furthermore, the QXL program will continue even when the PC is rebooted with a soft reboot!

PC Formatted Disks:The QXL can directly read and write on PC format disks even though the PC uses a "." instead of a "_" to attach extensions to the file name. For example, COPY FLP1_DOS.TXT TO RAM1_DOS.TXT works fine, although a few control characters not used by the QXL will also be included in the document. Quill (in Xchange) will not work with PC formatted disks; neither load, save, print to file, or export works. It is easy enough to work around this limitation by first copying to RAM1.

The PC uses "\" to separate directories, but the QXL must use "_" for directories. If for example, the PC disk had a file in a sub directory, A:\QXLSTUFF\CONFIG.DAT, then the QXL could copy it with, COPY FLP1_QXL_CONFIG.DAT TO RAM1_Comfig.dat. Also the "." could be a "_" in the PC file name or the QXL file name and the copy would work.

There is one glitch. If a PC format disk is used, then a second PC format disk is put in the drive, SOMETIMES the file header information is NOT UPDATED. Stupefying it is when you put in a new disk, do DIR, the drive lights up, but you get the directory from the old disk! To update the file header, put in a QL disk, do a DIR, then put back in PC disk and do a DIR. Voila! There is the proper list of files.

Thus, PC formatted disks can be pretty much used interchangeably with QL disks. However, it does present a problem when I try to use one of them on the QL. To help keep disks sorted out, I put a black mark on the side of disks that are QL (1440 sectors) formatted.

CAPTURING SCREENS: The number of bytes for a QXL screen is:

$$X * Y * Bp / BB$$

where X is the number of pixels across, Y the number of rows of pixels, Bp the number of bits per pixel, and BB the number of bits per byte (8). For mode 4, two bits are needed for each pixel so that Bp/BB=2/8=1/4 so the formula is X * Y / 4:

Screen	X	Y	bytes for whole screen
QL	512	256	32768
EGA	640	350	56000
VGA	640	480	76800
SVGA	800	600	120000

For MODE 8, the requirement that each pixel needs twice as many bits is canceled out by the fact that there are only half as many pixels across. So the same number of bytes is required for MODE 4 or MODE 8.

In the QL mode, the screen bytes start at address 131072 and the screen can be stored in a RAM1 file named store by:

```
SBYTES ram1_store,131072,32768
```

and later brought back for viewing with:

```
LBYTES ram1_store,131072
```

The higher resolution screens are not stored starting at address 131072, instead they start at 303776. At least in SMSQ version HBA 2.47. Since this address could change in future versions, it is hoped that a systems PEEK or possibly even a KEYWORD will be made available to always get the proper address. So with the version 2.47, the screen can be saved with:

```
SBYTES ram1_store, 303776, 56000    for EGA,
SBYTES ram1_store, 303776, 76800    for VGA,
SBYTES ram1_store, 303776, 120000    for SVGA.
```

All three higher resolution screens can reloaded with:

```
LBYTES ram1_store, 303776
```

WINDOWS Screen Capture: It is possible to run the QXL from Windows by selecting (in Windows) the MS DOS Prompt icon and starting the QXL as usual. However, I/O is slow and I've had a few unexplained crashes. However, you can capture the QXL screen by pressing ALT and PRINT SCREEN together. My computer blinks when I do it. Then when you exit the QXL as usual with CTRL Scroll Lock, the QXL screen will be available for pasting into Windows programs such as Clipboard Viewer and other programs as a graphic by selecting the EDIT drop menu and the PASTE command.

FREEDOM: I (AL) have been looking for one for a long time. Some people said they didn't even exist. I am talking about a portable PC that has an ISA slot that the QXL could fit in. Last month I bought a used Epson Equity LT-286e with 1M ram, 40M harddisk, 3.5 floppy drive, and a 10.5 inch monochrome screen for $350. All the QXL functions - floppy, harddisk, printer, etc. appear to work OK except only in black, white or shades of gray and no SVGA mode. The other modes work fine. The QL fonts are easily read even with my tired old eyes.

Yes, I took it with me on a business trip and made slides on the fly. And I programmed right there during the boring parts of the meeting. But the best part is at home sitting in my favorite recliner early in the morning with my QXL on my lap. How sweet it is!

"DiskTidy" A Quanta Library Gem

St. Cloud, Florida, USA - March R Renick Jr.

One of the most interesting and useful utility programs which I have been using for some time is called DISKTIDY. This program allows you to catalog all of your many disks in an orderly manner and brings a definite means of maintaining useful data files. It was written by Alan Pemberton of the Scottish QL Users Group. I came across it in the QUANTA library on disk UG 07 as version v1.12. It has since been updated as v1.20 which I received from, Alan recently. I would like to review this utility and describe how to use it:

01.0 - This program is predicated on the need to assign a sequential number for each disk and you should write this number on each disk label for reference.

02.0 - You choose the first disk that you wish to start with and having done this you update the program and it replaces a simulated listing that is part of the start up routine.

03.0 - The program presents on the screen a menu from which you make a choice of the many options available. It is necessary for you to make a choice for each disk to be cataloged in an ascending numerical sequence starting with number 1. The program reads the file- name of each disk, along with its directory. It presents on the screen, the menu, as well as specific data on the number of free sectors still available, the number of files on the disk, the title of the disk, each file name in sequence along with the number of sectors used for this file and the date that it was placed on the disk.

04.0 - Once you have chosen a quantity of disks and placed their records on file you have many options to work with. The menu presented to you on screen looks like this:

```
DISKTIDY MENU

VIEW DIRECTORY
FIRST DISK
LAST DISK
PREVIOUS DISK
CHOOSE DISK
ADD DISK
UPDATE DISK
SEARCH FOR FILE
REVIEW SEARCH
DUMP OPTIONS
EDIT TITLE
QUIT/RESTART

↑ ↓ & SPACE TO SELECT
```

Using the up or down cursor keys you highlight each option until you find the one you want and then you press the space key to have the option presented in a window where you can bring up what you wish to have done.

05.0 If you wish to have a hard copy of all information on a specific disk, then you would choose the DUMP OPTION which will give you a printout of all data on file as mentioned previously.

06.0 If later on you find that a specific disk has been changed with respect to the number of files on it or possibly the names of the files have been changed, place this disk in flp1_ and call for the UPDATE OPTION which will now make all of the necessary changes for this disk in the programs master file, which is on the disk in flp2_.

07.0 If you wish to look at the files of a specific disk you would use the CHOOSE DISK option and you will be asked for its number and when it is presented on the screen you then select the VIEW DIRECTORY option and it will be presented to you on screen. If there are more than 14 files on this disk you may use the up or down cursor to scroll through the listing regardless of how many files there are.

08.0 If you remember the name of a particular file, but not where it is located you can choose the SEARCH FOR FILE option and it will search through every disk file that it has cataloged and present a listing of all files that are similar as well as the specific file you are looking for. You can scroll through all of these titles along with an indication of what disk each file is on.

09.0 If you wish to change the title of a specific disk on file (which usually is the title placed on the disk when it is formatted) you would call up the EDIT TITLE option and you will be asked to type in the new title, which can be up to 40 characters in length if you desire. You must have chosen the specific disk before you make a change.

10.0 All other options not described are straight-forward and do not need further description.

11.0 One last comment concerns the length of each file name, it can only be 18 characters in length. Any name that is longer will be truncated at the 18th character as a (*).

See you all at the Quanta AGM

ISSN 1351-1343

QReview

£2.50

News, reviews and articles for the QL and compatibles

Volume 2 Issue 2

December 1994

--- NEWS ---

QReview (ISSN 1351-1343) is a magazine devoted to the QL and compatibles. It is produced entirely on the QL by a dedicated team of QL enthusiasts. The magazine regularly contains articles suitable for the beginner, tinkerer and expert. There are articles on programming in SuperBasic and machine code, updates of programs released in the public domain and of course the latest news from around the world. Independent reviews of QL software also feature prominently. A new regular feature is the Help Desk, in each issue readers hardware or software problems / questions are answered.

QReview is published, at present, four times a year and costs £2.50 (UK), £3.00 (Europe) and £4.25 (any other country) per issue including postage and packaging. A four issue subscription to QReview costs £10.00 (UK), £12.00 (Europe) and £17.00 (any other country). Each volume contains four issues. Back issues are available at the single issue price.

QReview is available from Quo Vadis Design, 57 Shaftesbury Road, Romford, Essex, RM1 2QJ, UK. Please make payments payable to 'Quo Vadis Design'. Payment in UK Pounds Sterling currency only. Cheques (drawn on a UK branch of a bank or building society), Postal Orders, International Postal Orders and Eurocheques are all accepted. Subscriptions may also be placed with IQLR in North America. A four issue subscription rate is $30. Payment in US $ can be made by either a Postal, Bank or Personal cheque (drawn on a US BANK) or bank notes. Please make payments payable to 'IQLR' and send to IQLR, P.O. Box 3991, Newport, RI 02840-0987, USA.

Magazines Available Now

QReview Volume 1
Issues 1 - 4

QReview Volume 2
Issues 1 - 3

For more information telephone/fax the Editor, Bruce Nicholls, on (01708) 755759.

Seeing is Believing

Zapresic, CROATIA - Zeljko Nastasic

(From bits in memory to pixels on the screen...)

You sit down, switch on your QL and patiently wait for the start-up prompt to appear on the screen... how easily we take for granted that a computer can display pictures an a monitor. Would you ever think that a complex process is performed by about 30% of the hardware in your computer just to make it possible to see anything on the screen of your monitor, and no less than 50 times a second at that!

Have you ever wondered how it all works? No? And have you ever wished someone explained to you why you can't have 256 colours on the screen instead of 8? Well, to put it bluntly, to know why, you'll have to get back to the basics, and ask the 'how does it work?' question...

The screen: Being visually oriented, it is quite logical that people first thought of making a computer display symbols and pictures as a result of it's operation. I guess that the reason they stumbled across the idea to use a cathode ray tube (yes, that's the technical name for the glass bit you're looking at when you say you're looking at the screen!) to display symbols was due to the fact they kept getting tangled up into the paper coming from the printers and teletype machines. What was needed was some 'virtual' paper - one that could be simply erased and used as many times as necessary. They found the device needed in a piece of equipment commonly known as an oscilloscope, which was at the time really common for any computer installation, as a test and repair tool (computers used to be far less reliable then!).

An oscilloscope's main part is an electron tube in which a tight beam of electrons is forced to collide with the glass front of the tube. The important things here are that the front is coated with a special compound which lights up when struck by the beam, and there is a system of electrodes which can deflect the electron beam, thus moving the point of collision on the glass front - which is actually the screen. The stream of electrons can also be turned on and off. The whole thing has all the necessary controlling electronics and power supplies - together with the tube itself, this makes up a simple oscilloscope.

Much like a pencil and a paper, turning on the beam and deflecting it moves the dot the beam produces around the screen. Unlike a pencil, the trace does not stay there, but instead fades out very quickly. Fortunately, if you move the dot fast enough, the eye will provide an effect of a continuos line, or figure. If you can switch the beam on and off fast enough, you can also produce multiple figures, because you can move the point without it actually leaving a trace, when the beam is switched off. The whole thing is very similar to illuminating a wall in the dark, by a flashlight beam. If you move the flashlight fast enough, you can 'draw' simple figures.

To get the computer to display things on an oscilloscope, it was connected to a few bits and pieces of electronics which made it possible to move and switch the electron beam using binary data from the computer.

This was probably known as the worlds first 'random scan' display. The 'random' prefix is there because the electron beam can be moved around the screen in any pattern, and not only in a pre-defined one. At that time it was probably simply called 'display' as there were no other display types. The distinction came later, as new types of display were invented to fix problems with the original display type.

One of the most serious problems was the need to continually re-draw the contents of the screen (which were, by the way, extremely simple by today's standards - 8 rows of 16 characters were considered superb!). This had to be done simply because the special coating would stay lit only for fractions of a second after excited by the electron beam. Needles to say, at first many coating types were developed to offer longer and longer illumination sustaining times - the time a coating could sustain illumination was called 'persistence' (sounds familiar?) and it still is, because it's still a very important parameter. This continued on and on until the discovery of the 'storage tube'.

A storage tube has several modifications which enable the coating on it's screen to stay lit once the electron beam passes over it, until an 'erasing' pulse is supplied to a special electrode inside the tube.

This technology was perfected by a famous oscilloscope and test equipment manufacturer, by the name of Tektronix. The people there soon realised it's potential - the computer did not need to use up it's time redrawing the picture - the screen 'remembered' the picture for it. In stead, the computer could use the time to display much more complex things - drawings. This was the beginning of computer graphics as we know it today, and it started with this new display screen, known as the 'vector display'. Until some 15 years ago this technology was a must for what would today be considered high resolution graphics, and Tektronix was a world leader in it, making very large sums of money.

Unlike today's screens, a vector screen is not limited by resolution (or rather the limit is very high - 4096 by 4096 was common), but by the number of lines ('vectors') displayed on it. The lines would be drawn on the screen by special circuitry, which was fed by line start and end points from the 'display memory' (also known as the display buffer, or the display store). Consequently, the memory size limited the picture complexity. All pictures had to be broken down into line segments by the computer, and the start and end points loaded into the display memory. Many improvements were done in due time so that even filled shapes could be represented using very few bytes of memory.

On the low end of the market, random scan displays were slowly pushed out by raster scan designs. In contrast to random scan, which literally draws out the picture, raster scan screens always draw the screen in one pattern - a series of tightly spaced horizontal lines - a line raster. This is why they got their name. Then, by switching the electron beam on and off at desired points in the raster, only parts of the lines are actually drawn, forming 'rasterized' pictures. The smallest length of a raster line (sometimes also called a 'scan line') that can be illuminated was then named a 'pixel' - well, I trust it's all starting to sound much more familiar now!

Indeed, what we think of as a display today is invariably of the 'raster scan' type - the random scan screens have long since perished - and one of the last places they were used were the old coin-operated 'meteor storm' game machines (does anyone remember those?!). But before I go into the details of a raster scan display - there is something to be said about the demise of the vector display.

Vector screens became more and more sophisticated, with advances in screen technology, and especially with the continually higher amounts of memory bought at the same price, as memory technology became better. One of the great advantages of vector displays was the relatively small amount of data needed to describe even complex shapes - 2 points for a line, 2 points and a radius for a circle, etc. This is why they were always considered a graphical output device. In the text output arena this advantage was lost - both the vector and raster systems generated character images from a character code. A special circuit would look up the actual shape of the character from another memory, which was of a fixed size no matter how many characters the screen contained. The idea behind this is very similar to the one behind a plotter (of which the vector display is a good analogue) and a printer (of which the analogue is a raster display). When the memory available became large enough to contain enough characters to fill a TV raster, all alphanumeric screens soon became raster based, because the circuitry and display parts needed to make a raster display were cheap - they were little more than simplified TVs. By the time millions of TVs were sold and the technology was already there, reliable and cheap. At this point the vector screen became confined to its own niche - graphics. To display raster graphics (made of pixels instead of lines) at any resolution even remotely comparable to a vector screen was unthinkable, because of the memory involved.

A vector display is limited by it's memory only by the amount of lines it can hold in it. The resolution, or line endpoint positioning accuracy of the display is only slightly affected by memory size. In contract, a raster display system has to have enough memory to contain every pixel it can ever display, and with more resolution the amount of memory needed grows very fast - at the time a 4096 x 4096 monochrome display which was common for vector technology was practically impossible to make using raster technology. If it was raster based it would have needed 4096x4096 bits of memory, which is 2Mb - that probably looked as all the memory in the world at the time. Because of this, many quirks of the vector screen were tolerated, such as the need to re-draw the whole image if any single element in it was deleted, changed or moved - because the actual electron tube which served the purpose of the raster memory could be erased only as a whole.

However, as memory became cheaper and faster, the first raster graphics screens started to appear. Their big advantage was that they could easily display even animated pictures (because the screen was not of the storage type), which was always the problem with vector displays (sometimes to the point of being impossible). To extend the life of vector technology, a solution was even found to the 'erasing problem', and the best vector displays could even use partial random scan techniques to animate certain objects on the screen - which was very important as they were mostly used in CAD applications, where the mouse and graphics tablet have just been discovered, offering moving cursors, menus, object dragging on the screen, and similar things not easy to do using vector technology. But even this was not the vector display's nemesis.

The end of vector technology came with the introduction of colour. During the years many attempts were made to at least simulate a few green levels (as a rule, vector displays had green screens, because the special 'memory' luminescent coating could only produce that colour). All have been only marginally successful. Finally, the continuing memory density rise made it possible for raster scan displays to use higher resolutions, and to add a completely new, and very desirable property - colour. This new property so overshadowed the vector screen's superior resolution that it was no longer a major advantage - things which needed precise outlines in one colour, could now be distinguished by being drawn in different colours. A colour solution for vector displays could not be found in time, and so the vector display became obsolete. Today, the only place where it is used is in certain oscilloscopes and in radar displays, and it is quickly being replaced by raster screens even there. Such an unfitting end for a technology which was developed from a storage tube used as memory in one of the very first real computers, and which was ironically organised as a raster of dots!!!

The monitor: Today, what we think of as a monitor, is invariably the 'oscilloscope' part of a display system, specifically designed for raster operation.

One peculiar thing about this design is that the computer does not furnish the monitor with any signals to directly control the path of the beam of electrons which 'draws' on the screen. Instead, a system which owes very much to TV was adopted - simply to cut on the signals needed.

The trait of raster displays is that the beam always moves in the same way, with the point hitting the screen describing a raster. This raster resembles very much the way a printer head moves to print on paper - the beam starts at the top left-hand corner, 'drawing' a line to the top right hand corner, then returns to the left edge and draws the next line in the same manner right under the previous line, and so on until the bottom right-hand corner of the screen. Unlike a printer, which would at this point go on to another page, the beam returns back to the top left-hand corner, and starts all over again. Please note that the beam is actually switched off where the screen is black, but if it were switched on, this is the path it would go. Curious readers can see that by making their QL display a fully white screen (preferably on a monochrome monitor) and look at it through a magnifying glass - the horizontal raster lines will be nicely visible.

One full pass of the beam over the screen is commonly called a 'frame' or a 'field'. The reason the beam has to redraw the field over and over again, and do it quite fast, is that just as the beam finishes the frame the parts it has drawn first begin to fade. By redrawing the raster, the image displayed in it is thus refreshed - so the frequency (= 'rate of occurrence') with which this happens is appropriately called the 'screen refresh frequency'. This is what all the 72Hz markings mean in the PC magazine commercials - the screen is refreshed 72 times a second. In practice, 50Hz is enough provided you have a display with a long enough persistence (remember from above?). However, on colour monitors the blue primary colour is very hard to make with a long enough persistence, so the user certainly benefits from a higher refresh rate, because of reduced flicker, but more of that later.

As can be seen the computer does not really have to generate the signals for beam movement - in effect it generates a signal which can be interpreted as a sort-of carriage return or form feed on a printer. Sometimes the 'carriage return' and 'form feed' signals are also separately supplied. These signals are called 'synchronisation' signals, and in the case of separate signals there is a 'horizontal' and 'vertical' synchronisation signal. When these are combined into a single signal, it's called 'composite' synchronisation, and they are again separated inside the monitor circuitry. Now, you are probably wondering where any kind of synchronisation comes into the story - well, all the circuitry that moves the beam in a pattern of the raster is actually contained in the monitor itself. In order for it to know when it should actually start drawing a line, and from which point of the screen, it has to synchronise to the beam switching signal which describes the picture, using the synchronisation signals - otherwise the pixels would get drawn in wrong parts of the line or even in wrong lines.

Seeing is Believing - (cont'd)

The horizontal synchronisation signal is very similar to a carriage-return / line feed signal on a printer. Where the later positions the print head at the start of the next row, the former positions the beam to the start of the next line of the raster. The difference is that the line is automatically drawn - there is no 'start drawing' signal, the synchronisation signal serves that purpose as well.

The vertical synchronisation signal is in turn very similar to a form feed on a printer. The later would go to the next page, and the former simply moves the beam to the top of the screen. A peculiar feature is that this will not automatically move the beam to the left-hand corner - so the synchronisation signals even have to be synchronised between themselves to have the desired effect. believe it or not, this was done so on purpose, to make interlaced displays possible, and those are explained later.

It is clear that the maximum vertical number of dots that can be displayed is equal to the number of lines in the raster. To describe a whole field of raster lines the horizontal synchronisation signal has to occur once each line, and this means this happens for every refresh. Thus, the frequency of the horizontal synchronisation signal is equal to the number of lines multiplied by the refresh rate. I'm explaining this because the horizontal synchronisation frequency (also called the 'line frequency') is a very important parameter for a monitor - a monitor always has a pre-determined range of frequencies it can be used with.

Knowing this, we can safely say that the maximum vertical resolution a monitor can display is dictated by the highest horizontal frequency it can use, and the lowest vertical frequency it can use (whilst avoiding image flicker!):

$$\text{Max. vertical resolution} = \frac{\text{max. horizontal synchronisation frequency}}{\text{min. vertical synchronisation frequency}}$$

In actuality this figure is somewhat lower because of reasons I will describe later.

Since the vertical resolution we can display is equal to the number of lines we can generate in a frame, similarly the horizontal resolution is equal to the number of pixels we can generate in a line, and this is actually the maximum speed with which we can switch the beam on and off in order to make pixels light up or stay blank. The worst case condition is of course when subsequent pixels in a line are arranged in a lighted - blank - lighted - blank etc. pattern, requiring the beam to switch on, off, on, off again, etc. Theoretically, the frequency (again that word!) with which the beam can be switched is limited only by the driving circuits for the beam itself. This is commonly given as a 'bandwidth' expressed in Hz. Because this is a very large number it's commonly given in MHz (millions of Hz). In practice, this is the maximum frequency with which the beam can switch itself on and then off again - this are in actuality two pixels, one lighted and one blank. So, to get the maximum horizontal resolution we need to see how many times this can happen in one raster line. You can trust me on this, the figure is:

$$\text{Max. horizontal resolution} = \frac{2 \text{ x Bandwidth}}{\text{Horizontal synchronisation frequency}}$$

Again, in practice there are more restrictions, so the figure will be lower. One of them is that, figuratively speaking, you cannot draw a point smaller than the tip of the pencil, which means that although you can switch the beam on and off fast enough, you will not see a clear pixel if the beam was switched on and off before it had a chance to move the distance equal to it's own width! However, usually this is not a limiting factor, as the monitor will be designed to have the bandwidth matched to the beam 'size' - simply because it would be a waste of money having a high bandwidth and no real way to use it because 'the pencil is too big'.

The first reason why the actual resolutions are lower in both directions is that it takes time for the beam to return either to the start of a line or to the start of a frame once a synchronisation signal has been received. Because of this, a certain amount of time (which can be expressed as a number of lines) will have to be unused by switching off the beam, to allow it to return to the top of the screen. In the same way, a certain amount of time (which can be expressed in pixels) in a line will have to be left unused for the beam to return to the start of the line.

These periods are called retrace periods. The vertical retrace period takes up only a relatively small percentage of the lines in a frame, usually less than 10 percent, most commonly about 5 percent. In contrast the horizontal retrace period takes up quite a few pixels, usually 20 percent or more.

Sometimes even more of the line or frame interval will be left unused, for instance to have a desired ratio of the usable screen height and width at the resolution used (this is called the 'aspect ratio') by including a 'border' around the screen image. For instance, to have an aspect ratio of 1 (meaning the actual pixels are square) the horizontal to vertical resolution ratio has to be 4 to 3, because this is the ratio of a standard monitor's screen height and width. Obviously the QL's screen does not have an aspect ratio of 1 because the ratio of horizontal and vertical resolution (2 : 1) does not correspond to the ratio of the screen width and height (4 : 3).

How does all this math work out for the QL? Well, one of the requirements of the QL as it was created was to be able to use a TV set for a display. This means it had to conform to the standards of frame and line frequency, as used in TV broadcasting - frame frequency has to be 50 Hz in Europe and 60 Hz in the US, and line frequency has to be 15625 Hz. Both frequencies can vary only very little, especially the horizontal frequency, which must be accurate within 250 Hz, which is 1.6%! For the European QL frequencies were chosen which generate a 312 line raster with 640 pixels in each line, and for the US QL the raster should have had a 262 line raster (I'm not aware of any US specific QL ever made - this would involve a different ZX8301 chip which generates all the screen related signals). Obviously, the QL does not have a 640 x 312 resolution - this is so because from the 640 pixels in a line only 512 can theoretically be used on a TV, the rest is the horizontal retrace. Realistically, though, this can only be achieved on dedicated monitors, with great many TVs not being able to display more than 480 pixels in a line - hence the TV and Monitor modes on the QL. As an aside, the usage of parts of the screen which cannot normally be seen, but are still within the standard norms as far as retrace periods are concerned, is called 'overscan'. Hence, it can be said that the QL's screen operates with 'horizontal' overscan.

As for the number of lines, 256 were chosen, so 56 lines are unused, and constitute the vertical retrace and border area. Were the QL to operating in 'vertical' overscan as well, it would probably have about 300 vertical dots. The reason why only 256 were used lies in the amount of memory needed to contain the picture - again more on that later.

We have not yet used the second 'formula' I mentioned, and that is the one using the bandwidth to see how many pixels there can be in a line. the TV norms state that the bandwidth of the beam control signal should be up to 5 MHz, which promptly gets us to the 640 pixel figure the QL uses. Obviously Sinclair stretched it to the limit as far as horizontal resolution is concerned - an herein lies the reason why the TV or even monitor picture can look somewhat blurry - it's the 'too thick pencil' problem, but this will be explained later.

By now I believe most of you are wondering how a Masterpiece graphics card is supposed to display 512 lines on a standard QL monitor, which also conforms to the TV norm - well, it's actually using a trick to do this.

Tricky business: Perceptive readers with a knack at calculating will have already noticed that you get a very strange result for the number of lines in a raster using the TV standard frequencies supplied - 15625 divided by either 50 or 60 do not result in an integer number of lines. For the European standard this is 312.5 lines!

Well, the reason for this is that the common household TV set uses a technique called 'interlacing' to actually have a double number of lines than the figures would suggest. The same is used by the Masterpiece when the card is connected to a standard QL monitor, or even a TV set.

When TV was developed, it was long established that for the eye to correctly perceive movement, at least 20 full pictures had to be shown in a second. This figure was of course arrived to by experience from the film industry. In fact, in the film industry the standard is 24 pictures a second.

A TV screen, in contrast, uses 50 or 60 frames a second, which limits the number of lines in a frame rendering the picture less detailed (smaller vertical resolution!) and is clearly too much for movement perception - the eye is already completely fooled with 24 pictures a second, so why over-do it? The reason paradoxically lies in machinery, and not electronics, and the one guilty for it - believe it or not - was originally from my home land, Croatia (small world, huh?). His name was Nikola Tesla, and his game was AC power - but the many exploits of Mr. Tesla, who emigrated to the US after his studies, are a different story. It is sufficient to say that he was the man who calculated that AC current had to have a frequency between 45 and 65 Hz to be most effectively used by all parties involved. So, 50Hz AC current became standard in Europe, and 60Hz in the US. What does all this have to do with the frame frequency of a TV? Well, to be able to have a picture in the first place, you have to have a camera. And to be able to use a camera, you need lights - strong electric arc lights, in fact - using AC power. Those lights flicker at the same rate as the AC power does, using the same principle as film to fool the eye in not seeing it. The problem with using those with TV cameras is that the frequency of the flicker of the lights would interfere with the raster scanning process (for instance the light switches off right in the middle of the frame). To avoid that the frame frequency had to be the same as the AC power frequency. Then they could be synchronised so that the light switches off during the vertical retrace period, where no picture is generated (those same problems can sometimes be seen if you use a home video camera and take shots of a fluorescent light - a dark horizontal band will appear across the screen).

To overcome these problems, a TV uses two frames to display the whole picture. A picture is actually scanned in a 625 line raster in the European system (525 in the US). This is then transmitted as two frames, one with 312 and another with 313 lines (262 and 263 in the US). The first one contains all the even lines of the 625 line raster and the second one all the odd lines. By slightly altering the timing of the synchronisation pulses the two rasters are displayed so that the lines of one are displayed between the lines of the other, so the lines of the two rasters are actually 'interlaced'. This creates a 625 line interlaced raster picture, and solves 3 problems at the same time - the frame rate is synchronised with AC power, the picture rate is lower and nearer to film rates (remember - you have to show films on the TV as well!), and the vertical resolution is doubled. What is more, it solves another, not so apparent problem - the luminescent coating on a TV cannot remain illuminated long enough to successfully display pictures with a 25 Hz frame frequency, but 50 or 60 Hz is OK.

The Masterpiece uses the very same principle, which is quite universal in fact, and can be used for higher resolutions as well. But there are pitfalls - a computer picture is not the same as a TV one. Interlacing relies on the fact that the two frames, each containing one half of the actual picture, are not that different - and with real-life pictures they are not. However, on a computer displaying text they are - anyone can try this by displaying a text screen, then deleting all the even lines, and then doing the same but with the odd lines. In many cases the actual contents are so different that were they shown separately, you would never have guessed they belong to the same screen! This is where the trap lies - for different frames the actual 'real' picture refresh frequency which is only half that of the frame frequency becomes apparent, and I have already mentioned that the luminescent coating has insufficient persistence for 25 Hz frame rate. As a consequence the screen will flicker, sometimes quite annoyingly. It should be noted, though, that if the display really had a 25 Hz frame rate, the perceived flicker would be much higher.

At this point you might ask why the screen does not have a longer persistence, and the answer is that due to the way a luminescent coating works, a solution could not be found which at the same time reduces flicker, but has no ghost images - the ones which remain after the contents of the picture change, for instance when text is scrolled - because of the persistence characteristics. This is even more problematic on colour displays.

To get a better picture using a standard, non-interlaced screen, the answer clearly lies in a special monitor capable of using higher synchronisation frequencies, and thus displaying higher resolutions without the need of interlacing - however, at the time the QL was made those were expensive, far more than even the original QL price tag. Some users who also follow the PC world will note that certain graphics cards used to display a 1024 x 768 screen using interlacing - in that case it was with a much higher frame frequency - 82 Hz, but the results were unsatisfactory because most monitors would incorrectly interpret the synchronisation signals, and the picture would not be correctly interlaced. As a consequence, it either flickered or looked as if it had half the resolution.

From one to many - VGA, SVGA, Multisynch: In the continuing competition to deliver higher and higher resolutions to computers, it was only a matter of time until the TV standards became insufficient, even using interlacing. To overcome that, monitors were needed which could work with higher synchronisation frequencies. Unfortunately, this made the problem of higher resolution a much bigger one. First of all, you needed a new monitor, which is not cheap, especially because it could not be a downgraded TV. You could not trade in your old monitor, because the old equipment which you might have to keep will not work on the new monitor. This made upgrading much more costly, and a precarious job at that, because there were as many solutions (read: monitors) as there were manufacturers, and all tended to be different for this reason or that. At this point you might ask why no-one made a monitor which could cover several frequency standards, or better still a whole range of frequencies. A simple answer - because of the costs involved. A monitor of that type costs almost double compared to a fixed-frequency monitor, and no-one was willing to pay that, because it would in most probability be supporting a competitors product. It was likely that once a monitor is changed, the next upgrade would be made by a product from the same company, which naturally tended to stay with it's monitors - an instant way to generate profit.

However, there had to be an end to that - and who better than the calculating minds of the Japanese to take advantage of it. As this went on, there were soon so many monitor standards that a user was practically forced to have at least two monitors. If so, why not have one, with a slightly lower price than the two, which could display pictures using both standards? Enter the NEC Multisynch range of monitors, which could do just that.

Although the monitor was expensive, it was compatible with so many standards, that if you were to buy all the monitors you needed to have them, it would have cost you much more. Of course, it was an instant success.

The very first Multisynch was actually more precisely a tri-synch monitor, offering 3 horizontal and 3 vertical frequencies in any combination, covering a vast majority of the standards then in use. As other manufacturers followed with similar products (and similar names for them) the technology evolved to the point where the monitor supports a very wide range of frequencies instead of discrete values. This range is usually such that the highest frequency is double that of the lowest, both for horizontal and vertical synch frequencies.

At that time a new standard appeared in the PC world, called VGA - offering compatibility with many existing standards while using a single monitor with a fixed horizontal frequency and a variable vertical frequency (much cheaper than a 'real' multisynch), but the multisynch deed was already done - especially as the multisynch monitors were the first to be able to use the new standard. Nevertheless, in the following years VGA gained practically universal acceptance on the PC market, and was even extended (again requiring a slightly different monitor, again compatible with multisynch monitors). Because of this, the current monitor market has 3 distinct segments. The first are monitors compatible with TV norms, which are today mostly used in studios and for computers aiming at games. The second are VGA and SVGA monitors, offering a wide vertical synch frequency range, usually 50 to 90 Hz, and a single horizontal synch frequency of 31500 Hz for VGA or either a 31500 to 35500 Hz range or separate 31500 and 35500 Hz frequencies for SVGA (this explains why you cannot watch TV or display the standard QLs picture on a VGA monitor - the horizontal frequency available is double the required). The third range are the multisynch monitors, offering many types, covering many frequencies. In practice, however, the 1:2 ratio of the lowest and highest frequency in the range covered makes it problematic for a single multisynch model to cover all the frequencies currently in use. Because the PC market has already gone beyond VGA and SVGA in the upper extreme, the lower frequencies are not covered any more, so monitors which can for instance cover the TV, VGA and SVGA norms are very rare, and also expensive. Among the last which could do it were NEC Multisynch 3D and Sony CPD 1402, both by now outdated by at least two models.

Since VGA and SVGA monitors are the most profuse, they are also the least expensive, and it is natural for the Masterpiece graphics card to take advantage of this. The reason why the Masterpiece has doubled the vertical resolution is simple - a VGA monitor has doubled the TV standard horizontal frequency, and can use the same vertical frequency. Using the simple formulae I gave at the start, we soon arrive to the conclusion that the number of lines generated in a frame is also double - 624, to be precise. To maintain the same aspect ratio of the picture, the number of pixels in a line is also doubled to 1280. Using the same rules for horizontal and vertical retrace we arrive at 1024 pixels arranged into 512 lines. To display the 'old' QL resolutions on a VGA monitor, the masterpiece still uses the same basic resolution (because it is the requirement of the monitor that the synch frequencies remain unchanged), but internally translates one pixel into two adjacent pixels (reducing the horizontal number of pixels from 1024 to 512), and also displays the same data in even and odd rows (reducing the vertical number of pixels from 512 to 256). The last, namely displaying several lines with the same contents in order to reduce the vertical resolution without changing the synch frequencies is called 'double scan'. No doubt the ability to do this was one of the important factors deciding the resolution the Masterpiece card can display -

otherwise we would have needed two monitors or a multisynch monitor to use it. The Masterpiece is likely to generate less lines than the maximum in order to use a slightly higher frame frequency to reduce flicker - in any event, the user will se no other difference aside from a better picture!

And now, let us take a trip from the monitor into the computer, and see how the monitor actually gets all the signals which make it display a picture in the first place.

Monochrome, RGB, Composite, Analogue, Digital - What are they?

Now that we know (I hope!) how the resolution is defined, the only thing remaining is the controlling of the beam in the cathode ray tube - how to switch it on, off, or even set it's intensity, creating a corresponding pixel - dark, light, and anything between.

Again we have to go back to the TV set. Broadcasting technology started with black-and-white TVs. Because there are plenty of grey levels in-between, but only one 'colour' it is best to call this a monochrome display (monochrome meaning literally 'one colour'). Clearly the TV would not have come far if it could only be able to turn the beam in the screen tube on or off - in fact, it uses a voltage to control the beam intensity. The higher the voltage, the brighter the dot on the screen which 'writes' the raster.

Because the first raster graphics systems had a general shortage of memory, a single bit's state was associated with a pixel on the screen, so the signal that went to the monitor in such a system was purely digital - it could be either on or off. In contrast to this, a signal which can assume many levels between on and off, including fully on and fully off, is analogue. It's state would have to be described using many bits - as many as needed to represent all the levels it can assume, assigning each bit combination one level.

When the first systems capable of displaying more than two levels for each pixel became available, this is how the signal actually went to the monitor - each bit from the ones needed to describe the intensity of the pixel was digitally transmitted using it's own wire. They were then combined in the monitor and converted to an analogue level using a device called an analogue-to-digital converter (ADC). The reason this was done so was that it was much easier to get a digital signal from a computer through a wire, and without noise or interference - since the same method was used in the computer itself, it being digital throughout. Such a monitor had inputs designed specifically for digital signals, which could only recognise two states - appropriately called digital inputs. In any case, all signals used to generate the contents of the picture, as opposed to the signals used to generate the raster, are called 'video signals', very commonly all of them are referred as a singular video signal, because they all describe one information, and that is the content of the picture.

Occasionally, because of broadcasting technology compatibility, the digital signals were combined at the output of the display circuitry, to form an analogue video signal. The two synch signals were also combined into a composite synch signal, which is itself digital. Finally, because broadcasting technology uses a single cable for all the signals, the composite synch and the video signal are combined together, forming a 'composite video signal'. The reverse process of decoding the composite signals is always carried out in the monitor, because the internal circuitry needs the components and not the composite signal. This was all well and fine until the advent of colour displays.

A colourful world: The technology which led to colour computer displays is of course derived from colour TV. As a result of many experiments in human perceptions of colour, undertaken by the leading researchers developing colour TV, helped by previous research in colour film, it was discovered that 98% of all colours a human can perceive can be obtained by mixing three colours in various amounts. Those three colours were then called primary colours, and they are green, red and blue. Of course, there are many greens, reds and blues around, so in actuality the colours have to be from a strict set to be able to synthesise the required range. The same applies for anything which has to display colour, but the way this is achieved in the broadcast and computer industry is different.

In the broadcast industry colour is a mixed blessing, because the premise of a TV is to be able to send all the necessary information combined into one single signal - and to get colour, apparently we need three! The solution to this was found after even more research (which still goes on today) in coding the signals into a composite colour video signal. A major requirement was that this signal be compatible with the old monochrome norm, which was known as composite video, and was now redefined to be either monochrome or colour composite video. There are many ways in which this coding can be accomplished, but they all come down to dividing the available bandwidth of the beam control signal into monochrome intensity and colour hue information. Because of this the actual resolution that can be displayed is reduced, about 25% for a TV picture. However, because of the process involved, for the synthetically generated computer picture, this is a greater percentage.

The QL has in-built circuitry which performs this coding process (which is by the way fairly complex, and is contained in the MC1377 chip right beside the TV modulator), and the reason why the picture does not look good lies in the restrictions outlined above - remember that the QL pushes the available bandwidth to the limit, and this is reduced by the colour coding, resulting in a blurred screen. Arguably there is no real way to display a picture which has a need for pixels of different colour for resolutions higher than 360 pixels across, and to approach this figure expensive equipment is needed. This probably gives you a clue to the look of the QL TV picture, which could have been better, but not very much.

Because a computer generated colour picture is used for different things, the way colour has been introduced to computers is different than the broadcast standards. Instead of multiplexing the three colour signals, one for each primary colour, into one signal, they are separately fed to the monitor, each signal having a single letter marking, the first letter of the primary colour signal it carries - R, G and B for red, green and blue. The same distinction between analogue and digital applies even here, so the signals are either fed as one or more digital signals for each colour, or one analogue signal for each colour.

The number of colours a monitor with digital inputs can display depends on the number of signals, and is 2 to the power of (number of signals). From all digital input standards one deserves special note, and this one has 4 inputs, 3 for the primary colours, and one for intensity. This one is common for all colours, and is marked with the starting letter of the signal name, namely I. When this signal is on, all primary colours that are on have a higher intensity than when the I signal is off. The 4 signals R, G, B and I are frequently concatenated to form the name of this system - RGBI. As stated, the system can display 2^4 colours, which is 16. The QL uses a subset of this standard, omitting the I signal, to have only the digital signals R, G and B. Because of this, the QL can display 8 colours. Today digital input monitors are considered archaeological, with almost all current monitors having analogue inputs. Since an analogue input can assume any number of levels in a fixed range, theoretically a monitor with analogue inputs can display an unlimited number of colours. It is a different matter altogether how many levels a computer connected to the monitor can generate. It is important to say that the digital signal being the simplest subset of the analogue signal, an analogue monitor can be used with computers supplying digital signals, usually with a few inexpensive components to convert the digital signals into levels that fall into the permissible analogue range. The reverse, i.e. using digital input monitors with analogue signals is very seldomly possible, even if the analogue signals are made to assume only two levels. This is because the highest and lowest levels an analogue output can assume are both so low that they would both be interpreted as the same digital level - a logic 0.

As far as the monitor is concerned, this is not the end of the problems with colour. The remaining one is how to make one screen be controlled with 3 signals. With colour films the first systems had the film divided into three areas, one for each primary colour, and the light that passed through the film during filming or projection was actually of different colour. For projection, for instance, the film had three images in the areas, each containing the contents for a primary colour, and the three were projected onto the same screen by a special lens, in order to combine the primary colours into one full colour image. The same system is still used in the best colour cameras, both film and video. Taking this into the world of the colour monitor would mean 3 screens, one for each colour, and a system of mirrors to view them all as if they were one.

Certainly, this was not the way to do it. The answer was found in using three beams in one tube. Before hitting the screen, the beams pass through a metal graticule which acts as a special lens, to ensure that each beam can hit only it's own special area on the screen. For each beam this area is covered by a luminescent coating which produces light in one of the primary colours. In this way each beam is used to display one primary colour. The separate pictures are combined by having the graticule sufficiently dense so that the individual holes, and coloured dots are invisible from normal viewing distance, creating the illusion of a smooth picture. The distance between apertures on the graticule is commonly called the 'dot pitch'. There are also several graticule types which do not perform equally. Since real life pictures are by nature smooth, it works very well for them even with a fairly coarse graticule. Computer generated pictures which are un-naturally sharp, especially if there is a limited number of colours, are a different matter.

Because the graticule on a colour screen has a defined number of openings ('dots') the number of pixels we can display on such a screen, and even the number of lines in many cases, cannot be larger than the actual number of dots in the graticule, in both the horizontal and vertical directions. In fact, to be able to clearly discern every pixel, there should actually be at least twice as many dots as there are pixels. This is the so-called Shannon's limit, named after a famous scientist who first noticed this as an universal phenomenon, and on whose work practically all digital communication is based. In practice, this is rarely the case, except for low resolutions or very high quality (and price!) monitors - which is also the reason why the picture on a colour screen always looks blurred in comparison to a monochrome picture. This is not always a curse, in fact, sometimes it can be a blessing. Because there has to be a defined number of dots for a given resolution, and the minimum dot pitch is limited by graticule manufacturing technology (currently this is about 0.22 mm), to be able to view higher resolutions, we have to use a larger monitor. When we attempt to view a picture with more pixels than there are dots, we get either the 'too big pencil' problem or graticule interference (patterns resembling wood grains), depending on the graticule type.

One technology which is worth mentioning as the most immune to both phenomenons is the one used in the Trinitron tube (by Sony). Another trait of this tube is that it does not use dots in the graticule, but thin vertical lines spanning the whole height of the display. This means Shannon's limit does not apply for the vertical direction, only for the horizontal. It also means that the part of the graticule which is not a hole, and which normally masks part of the screen is smaller, allowing for brighter screens and superior colour reproduction. No doubt this is the reason for it being so renowned and sought after for the highest quality displays, especially for high-end workstations where it practically holds the monopoly.

The other extreme of having a blurred picture has found use in devices used to transfer a computer generated picture to film, where the blurring pleasantly smoothes the picture and makes the individual dots and lines of the raster invisible.

There is another side to the colour problem - have you noticed that frame frequencies continuously rise as the number of colour screens in use rises? There is a very simple explanation for this seemingly strange problem - when all screens were green, the persistence of the screen coating was high, requiring a lesser frame frequency for a flicker-free picture. With the introduction of colour displays a completely unforeseen problem surfaced - it seems that the most desirable persistence characteristics can only be obtained with coatings producing green light. This can also be achieved to an acceptable degree with red coatings, but the blue coating has to this very day stayed a hard problem to solve. This persistence problém is not seen on the TV because the nature of the TV picture is that it changes rapidly, so the viewer is concentrated to that - the flicker effect can only be observed on static pictures - and computer screens are mostly static, things usually happen only in a small area of the screen.

The worst part is that it is extremely difficult to match the persistence characteristics of all 3 primary colour coatings. Although the problem is known from monochrome monitors, because the coating for paperwhite monitors is a mixture of coatings for the three primary colours, it is a bigger problem on colour screens. Some monochrome monitors were notorious for this - when you scrolled text a green ghost image would be left behind because the green part of the compound had the longest persistence. On colour displays certain areas can all be of a 'problematic' colour as far as persistence is concerned, so the 'flickering' is even more pronounced by different colour areas around it, which do not flicker. To date, the only way to solve this problem is to use a higher frame frequency, which also reduces resolution all other parameters being the same, or requires all other frequencies to be higher, which has repercussions not only to the monitor, but also to the computer generating the image.

To the source: This brings us to the very place where all the signals are actually generated, and this is the computer itself, or more precisely it's display generating circuitry. As was already stated, the contents of the display are generated by interpreting it's description in a dedicated part of memory, called display memory. The interpreting part is in actuality very simple for graphics displays, which are now common. In the past this was not so, because the memory needed to contain the graphics image was large and thus expensive. As I said before, a raster graphics system's characteristic is that every pixel it can display must be described in it's memory, making the memory size directly dependable on the number of pixels it has to display - which is commonly referred to in terms of resolution.

For displays where each pixel can assume only two states (or colours), one bit is needed to represent every pixel - because one bit can also assume two states. The number of bits needed to describe the whole display is then equal to the number of pixels and this is the horizontal resolution multiplied by the vertical resolution.

Since we can only operate on bits inside the computer, it is senseless to use a system where each dot can assume three states - by the same token, we would need two bits to describe 3 states, as we cannot use 1 and a half bit for each pixel. But two bits can describe 4 states, and it's a simple matter to use them. This is why the number of colours or shades of grey that a computer can generate invariably comes out as a power of two. However, there are systems where it is literally possible for a pixel to assume 3 states, and even some stranger things, because certain bits are shared by more than one pixel. Systems such as that actually came to be from character displays.

In character display systems, the resolution of the screen is far greater than the memory used to hold the screen contents - this is because instead of holding pixels, the memory holds character codes. The codes are read one by one as required by the screen generating circuitry, and converted into pixels 'on-fly' by interpreting the codes into pixels which make the character, using a character font memory. This process is performed by dedicated hardware. In some cases the screen contents are even compressed, so that parts of the screen which are unused (filled with empty spaces) are not stored in memory. One computer which used this approach was the legendary ZX81.

Since character display systems are by their very nature limited (to characters), a way was needed to emphasise certain characters, for instance by displaying them in boldface, underlined or even in a different colour. This was achieved by adding an attribute code to each character code, which was used to hold information on special properties of the character. Since many graphics systems evolved from character display systems, when colour was introduced, more information and hence memory was needed to contain the information on the colour of each pixel, so it seemed like a good idea to describe areas of pixels corresponding to a character by a colour attribute, and thus save on memory.

Two very popular computers using this method in a completely different way were Apple II and the ZX Spectrum. However, this was such a pain for programmers that it was abandoned as soon as memory prices would allow. This was helped much by another invention, which is similar to the atribute system in some aspects.

It was first noted in the area of industrial process control that a limited number of states for each pixel is enough (Note that I am deliberately referring to states and not colours or intensities). For instance it was enough to display a layout of the elements of an installation in 4 colours - it would have been counter-productive to confuse the operator by having a too colourful display. However, it was highly desirable to change all pixels with a certain state to mean something different - in effect four colours were enough as long as you could define what they were. For instance it was highly desirable to have four main parts displayed by four colours, and change one of them to flashing red when something went wrong in that part of the installation! This is how the idea of the colour palette came to being.

A colour palette is actually another memory which interprets the state describing a pixel. It usually holds which bit combination represents which colour, and sometimes the colours can also be made to flash. This is actually very handy when the actual number of states a pixel can assume is limited because of limits to the display memory. For instance, if I wanted to work with monochrome pictures, it would be nice to re-define most of the states available as grey levels, and retain only a few actual colours, for cursors or similar. As an aside, 16 levels of grey are perfectly adequate for displaying a black-and-white picture, especially if any sort of dithering or stipples are used. On the other hand, if I wanted to design a printed circuit board, 16 colours are again more than enough, but then I would use each colour to represent a different layer of the board, and would probably use very different and distinctive colours.

On the other hand, I might have a system which can only display a certain resolution, and my requirements are of a lower resolution but with more colours. Then I would have to use dithering. This is a technique which combines several pixels to represent one larger pixel, and uses the pattern of the small pixels to create effects of colour or grey level.

Anyone wishing to see a masterwork in this field should try Carlo Delhez's UNGIF, a picture viewing program. A palette can help even here, because some systems allow a further level of pixel state interpretation by allowing separate palettes for groups of pixels. The pixels are for instance grouped into blocks 2 by 2 pixels in size, where each pixel in a block has a different palette. This means that the top left pixel in all blocks is described by one palette, the top right by another, and so on.

Since the original QL was conceived 10 years ago, the 32k of memory deemed available solely to hold the screen, was organised in a compromise between resolution and number of colours, giving either 512 by 256 pixels with 2 bits describing each pixel, or 256 by 256 pixels with 4 bits describing each pixel (remember the flash bit - although to me that is a sheer waste, it should have been 16 colours instead!).

The Masterpiece doubles that in both directions, requiring 128k of memory for screen storage. Taking all into account, with the limited number of pixel states available, I find the palette feature sadly lacking on the QL, but hopefuly that will change in the future.

The need for such tricks is slowly disappearing as memory prices continue to drop, and computer speeds continue to rise (allowing the more complex picture to be processed in a reasonable time). It is very unlikely that there will ever have to be systems with more than 16 million colours (which means that each pixel is described by 24 bits, 8 bits for 256 levels of each primary colour) because all the display technologies known today cannot accurately display that number of colours, and even if they could, the human eye can distinguish between a far lesser number of colours in the first place. However, resolutions are certainly going to continue rising. Then how come, the memory prices being what they are today, that 2000 by 2000 displays are not common?

It's always down to speed: Well, in all this we have forgotten one very important thing. If we want a certain screen size and number of colours, aside from a monitor able to display it, and the memory size needed to contain it, we need to be able to read the entire memory once for each frame in order to generate the display at all - but can we read the memory fast enough to satisfy the refresh frequency?

The problem as always comes down to speed - and the speed which we can read the memory is the most prominent of all problems - it is not so hard to make monitors with higher synch frequencies, or with higher bandwidth, or with a larger screen with a finer graticule - only more expensive. To paraphrase a famous computer scientist, the memory is getting faster slower than the number of pixels and colours is getting higher.
In order to satisfy the requirements a whole lot of memory architectures have been developed, some of which I have described in my previous article.

Another requirement which makes the speed problem doubly worse is that something has to put the picture data into the display memory, so it also needs access to the it. And it also needs to do that fast, which means it needs an access with no waiting implied, or at least the least possible. Since the speed which the memory access can be performed is fixed by the technology available, the only way to get more data per single access is to make a single access fetch more bits at once. This is the same reasoning behind the evolution of microprocessors from 4 to 32 bits and beyond. However, for graphics we need vastly more right now, figures of 16384 bits are not uncommon. Clearly with that number you cannot expect every bit to be transferred using it's own wire - the wires would take up half the table!

Help is at hand - fortunately the very nature of the raster display provides for it. The information about a pixel is only needed when the beam in the screen tube gets to the pixels particular place - and since it 'writes' the pixels in a succession from the upper left to the lower right corner, the data needed can be represented as a series of words, the width of the word in bits being sufficient to describe one pixel. In effect, the pixels are 'shifted' out of the memory serially - much like a superfast serial port. However, in order to keep the flow of pixels going and to keep the display memory free for the CPU or graphics processor to read and write, we need memory which can be accessed by two devices at once.

Seeing is Believing - (cont'd)

In practice this is simulated by a buffer which transfers a whole line of pixels from the memory to itself during the horizontal retrace interval, and then shifts the pixels out as needed, keeping the memory access free for the CPU. For serious applications special VRAM (video - RAM) chips are used which have this buffer on the chip itself. Other designs use a separate buffer and standard DRAM. However, since the VRAM internally has a very wide bus (1024 bits or more - one wire for each bit, but remember the wires are VERY small in a chip) it takes only a single access to fill the whole buffer. Although DRAM also has a wide bus internally, externally it's far smaller, so the same 1024 bits have to be transferred 16, 32 or 64 bits at a time to the external buffer, requiring several, albeit faster accesses.

Unfortunately, the pins which are used for data transfer are the same the CPU would use, and since they are occupied during a transfer, if the CPU wants access at that time, it has to stop and wait until the transfer is complete. In order to match speeds with a VRAM, DRAM would have to have a still wider external bus (to shorten the transfer time). As resolution rises, the increase in the needed external devices for a wider bus, makes the whole design more expensive than an equivalent design using VRAM, although VRAM is more expensive than standard DRAM.

On the other hand, if things are viewed from the standpoint of a certain resolution, the VRAM system is invariably faster than a system using DRAM, again justifying the cost. In some cases, such as the Masterpiece card, the VRAM needed to satisfy the requirement is small enough to fit a single chip, which reduces the number of chips needed for the whole card enough to well justify it's use.

The above reasoning should also explain why it is not possible to use existing video RAM in a setup to get higher resolutions or more colours. For instance, the technique applied inside the original QL display circuitry barely makes the original resolutions work, and at that slowing down CPU access to display memory to half speed. Having it display twice as many pixels would mean that the CPU would not be able to access the memory at all. Making the display memory have a wider bus implies a radical re-design of the whole motherboard and a replacement for the 8301 chip, which would certainly cost far more than the Masterpieces 50 Pound projected price. On the other hand, the memory on the Gold card, or Super Gold card cannot be used because it exactly satisfies the speed requirement of the on-board CPU. Any modification would mean similar slowdown as with the 8301, and a complete re-design of the board, because the existing design does not cater for anything other than the CPU having access to the on-board memory.

Well, I hope this answers any questions as to why the QL's graphics is the way it is and why it could not have been this way or that. I believe the information here should last you for the next year or so (grin). There are a lot of things I have not covered, like displays which do not use cathode ray tubes (like LCD or plasma), and a whole universe of problems to do with altering pixels represented in the display memory in desirable ways (forming characters, lines, shapes, pictures, windows...) have been left aside - for a future article...

PROG Watch !!

Over the past few months, a number of rumours have been floating through the QL community. The first deals with the possible launch of a Pointer Environment compatible DESKTOP PUBLISHING program. The second deals with the launch of a Pointer Environment compatible WORD PROCESSOR. What do the PROGS have to say about all this ?? Absolutely Nothing !! They are keeping information about their future products close to the vest.

The one bit of news to be received from them, is that ALL their software is now available on either Double Density or High Density diskettes. The reasoning behind this move is obvious, with so many QLer's moving up to HD and ED disk drives it will cut down on the disk clutter. Please specify your choice of media when ordering.

QITALY Club - The Italian QL Users' Club

Keeping the QL alive!

If you are an Italian QL User, please consider joining us.

You'll be in contact with almost everything the QL world has to offer, in Italy and abroad. We are not an alternative to Clubs like Quanta or magazines like IQLR and QReview. We strongly recommend to join/subscribe all of them too!

What we offer

QITALY Magazine - electronic magazine
MONDO QL - paper magazine
QITALY BBS - Fidonet BBS
QITALY Program Line - wide range of PD
Hardware repairs
Free support, help and advice

QITALY Magazine comes on 720Kb disk and is PD.

Each issue contains articles, news, programs and screenshots taken from PD and commercial software. Most of QITALY Magazine is viewable onscreen directly from the disk (just BOOT it).

QITALY Magazine is in Italian, with short info in English.

MONDO QL is an A4 paper magazine out at irregular intervals (about 2 to 4 times a year). It is all in Italian and contains more info on particular topics.

QITALY BBS is our own Fidonet node. We carry all the major QL international echomail message areas and a few Italian message areas, all about the QL. Nearly 20-30 QL Users here in Italy regularly connect to QITALY BBS. There are also many PD programs avaliable for free download (about 500 programs as of Dec '94).

QITALY BBS is avaliable on +39 342 590451 24h a day on Saturday and Sunday; 5:00 PM - 7:00 AM GMT, Monday to Friday.

You can connect with any modem up to ZyXEL 19200 bps - soon V.34

Rates

QITALY Club association (1 year): 35.000 ITL *
(You get Mondo QL paper magazine, unrestricted access to the BBS, hardware repairs, etc.)

QITALY Magazine subscription (4 issues): 25.000 ITL *
(You will get just the disk magazine)

* or rough equivalent in your local currency. Countries outside EEC please add 5.000 ITL for each of the above. You can pay by Cheque, Eurocheque, International Postal Order or Cash. Please send Cash by recorded delivery only.

How to contact us

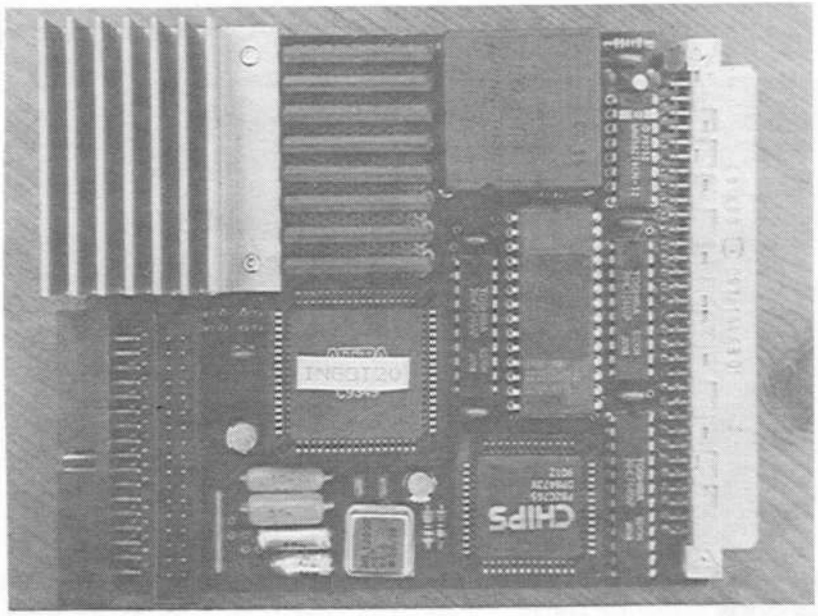
Foreign contact section:

EROS FORENZI - Via Valeriana,44
23010 Berbenno (SO) ITALY

phone: Eros Forenzi +39 342 590450 (5:00 PM - 10:00 PM, GMT)
FAX: Eros Forenzi +39 342 590451 (5:00 PM - 7:00 AM, GMT)
e-mail/Internet Eros.Forenzi@f123.n331.z2.fidonet.org

Do not hesitate to call us for more info!

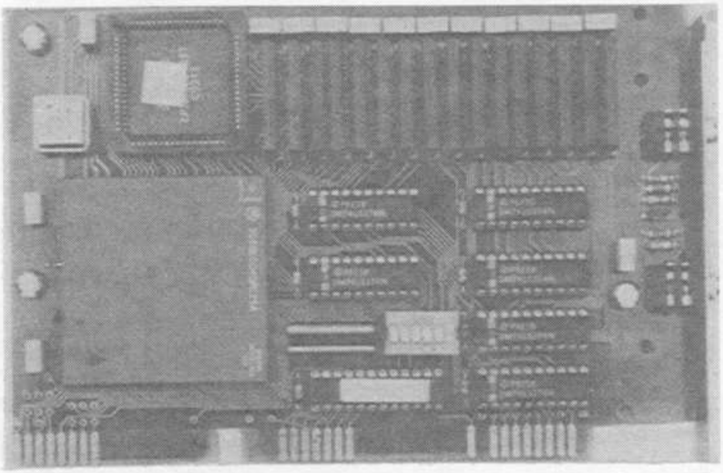
QL Forever!


ALTERA
CHIPS

MIRACLE SYSTEMS

QXL
Now With
SBASIC

SuperBasic Compatible Interpreter



- *68EC040*
- *4M or 8M of RAM*
- *Multitasking SBASIC*
- *QL Network ports*
- *Toolkit II*
- *QDOS or MSDOS floppies*
- *Uses PC's keyboard, floppy & hard disks, parallel/serial ports and mouse.*

This is the card that plugs into a standard 8 or 16 bit ISA slot on a PC and allows the PC to run QL programs - FAST. A new QDOS compatible operating system from Tony Tebby called SMSQ, which is supplied on a disk, includes Toolkit II and gives you the familiar QL environment. SMSQ includes SBASIC a multitasking SuperBasic compatible interpreter.

Installation is simple; plug the QXL into a spare slot and copy 2 files from the supplied disk onto the hard drive and you're ready to go. From the DOS prompt type QXL and the PC will transform itself into a QL before your very eyes. If at any stage you wish to return to DOS just press CTRL-ScrollLock. You can later resume the QL session by typing QXL/ which takes you back to where you left off. For POINTER ENVIRONMENT programs SMSQ can be configured to handle 3 screen resolutions in addition to the standard 512x256 QL screen. Your PC must have EGA or VGA graphics. EGA allows 640x350 whereas VGA also allows 640x480. Most SVGA cards will allow SMSQ to use 800x600 as well.

PRICING:

QXL (4M)	£280 including VAT - (£245 outside EU)
QXL (8M)	£395 including VAT - (£345 outside EU)

or you can send us your GOLD CARD and

£180 including VAT - (£155 outside EU) for a 4M - QXL

£295 including VAT - (£255 outside EU) for a 8M - QXL

You can also deduct a further £15 for a returned QL CENTRONICS and/or £10 for a DISK ADAPTER.

We are happy to accept payment by sterling cheque made payable to "MIRACLE SYSTEMS", or by quoting your MASTERCARD/VISA/SWITCH credit card number and expiry date (SWITCH card holders please also quote issue number).

TELEPHONE/FAX: (01454) 883 602

MIRACLE SYSTEMS LTD - 20 Mow Barton, Yate, Bristol, BS17 5NF,