QLUB

News for the QL user

Keeping up with the QL

NEW peripherals, software and training support for the QL have been launched in the past two months. And many more companies are set to bring out products for the machine before Christmas.

THE peripherals include colour monitors specially designed to handle the QL's 85-character display, printers configured with serial printer inputs for direct connection to the machine, parallel printer interfaces and alternative storage systems. And many of these add-ons are now available in high-street shops, as is the QL itself.

Meanwhile, on the software front, things are moving quickly. Although software companies generally agree that it takes longer to produce a program which properly exploits the QL's unique capabilities than it does to configure existing peripherals to work with new kit, the first end-products of such detailed research work have started to appear. On Page 2, there is news of completed QL software – including Psion's Chess program and GST's Assembly language program.

Sinclair Research itself will not lag behind in the peripheral race. Over the next few months, you'll see the appearance of some add-ons which should greatly enhance the way you use your QL.

Details can't be released just yet – but that would spoil all the fun anyway. We'll let you know more in November.

In addition to all the new add-ons and software for the QL, other forms of support are appearing for the machine. Some businesses have started training courses to teach you how to get the most from your QL's business software, while others have published books on subjects ranging from Games Listings to Interfacing.

You can be sure that increasing numbers of books and training courses for the QL will appear in the coming months. And QLUB News will be here to tell you about them and all the other goodies for your machine.

Discounts begin at home

AS you're probably aware, being a member of QLUB entitles you to this newsletter, software support and lots more. Part of that 'lots more' includes discounts offered periodically by various distributors and manufacturers. So Sinclair Research has decided to start the special offers by offering a 20 per cent reduction on all QL software purchased directly from Sinclair Research.

The savings can be anywhere from a few pence to a few pounds, depending on what you're buying. **On QL Chess and QL Assembler for example, you would save a total of £13.**

Other discounts have already been offered to QLUB members and QLUB News will help you keep track of them through a column at the back of each issue called 'The Special-Offer Office'. This will tell you what the product is, who produces it, how much the discount is and when it will be available.

INSIDE QLUB NEWS

sinclair

A MESSAGE FROM THE EDITOR

IT seems a long time since we last spoke (well, we spoke, you listened) and a lot's happened since then.

The long summer software drought is nearing an end, the strange black boxes sticking out the back of many QLs have disappeared and QLUB News has been fitted with an expansion interface. This column, and all the new features inside QLUB News are part of that refit.

There's much more to this newsletter than its predecessor, but there'll probably be much more to the next than there is to this. After all, we'll have more comments from you, and that can only help improve your publication. And it doesn't stop there.

Being a QLUB member means you're someone special, someone with an appetite for QL news and support, someone who wants to make the most of the QL. You'll not only get the newsletter six times a year, but also special offers on software, support and add-ons, a free update of QL Quill, Abacus, Archive & Easel and the chance to enter QLUB competitions.

Most importantly, however, you get each other. Engineers at Sinclair could cogitate for years and still not come up with the programming ideas and applications you'll think up. QLUB gives you an opportunity to exchange those ideas and combine them with professional advice.

Your ideas help us help you.

When the Spectrum first made its way into people's homes, it was considered a machine to educate people about home computers and perhaps have a bit of fun with. Little did we know that people would soon start designing word-processors, modems, spreadsheets and business graphics systems for this little workhorse. Software houses and the demand people created by buying their products led the machine in all kinds of directions that we couldn't have originally predicted.

With QLUB, you don't have to wait to REact, it's your chance to ACT. Tell us what you want to use your QL for, and we'll do our best to help you do it.

Q LETTERS

QUITE soon after receiving my QL I tried to save a program on a cartridge and it just ran and ran forever. It seemed to me that there were two possible lines of action – BOTH prohibited by the manual.
1) Switch off (and lose the program)
2) Pull the cartridge out (and risk permanent damage to the machine).
Ken Whyld
Caistor, Lincolnshire

The only thing you can do under those circumstances is press the reset button. You'll lose the program in memory, but it's unlikely to do any damage to either your QL or the microdrive cartridge. If the same thing happens while you're trying to access a program or data, again it's unlikely that either the program or data will be corrupted. – Ed.

I am using the Abacus design facility to produce a series of text-based screens, as an alternative to overhead projector slides, in small presentations and lectures.
My needs to improve this are:
1) Colour on Abacus.
2) The facility to delete border and status areas.

3) A single-key screen change to move about different parts of the spreadsheet quickly.
Lt. Cdr. R. Nichol
Plymouth, Devon

You can delete the border and status prompts by pressing F2. Your colour requirements and the need for single-key screen changes suggest you might look to import your data to Easel – see the article later in this issue – "Making Effective Use of EASEL Graphics" – for hints on presenting data. – Ed.

I recently received my QL and am anxious to know when you and other companies will be releasing software for the QL. The product has now been on sale for nine months and yet you have not produced any software other than the four programs supplied with the QL. Why?
M. Kent
Upton Noble, Somerset

Part of the answer to your question is contained on the front page of this issue and on the page opposite. It took some time getting here, but software is now becoming available for the QL and there's lots more in the pipeline. AT least 100 titles are planned in the next 12 months, and those are only the ones we've heard about. – Ed.

This is your page

White space is not a pretty sight, particularly in the eyes of an editor. Only you can prevent this dreaded disease from striking the letters' column of Sinclair QLUB News.

If we say or do something you disagree with, are surprised by or just want to sound off about, put pen to paper (or hands to keyboard) and let us know. Like it says at the top of this column, this is your page and it's up to you to make the most of it.

It isn't the only place in this magazine for your comments – there's the Programmer's Forum for your programming tips and queries, the QL Case Study to air your applications hints and the QL Quirks column to hear QL quips – but the letters page is the first place most people look for letters.

Write to:
The Editor, QLUB News,
Sinclair Research
25 Willis Road
Cambridge CB1 2AQ

HARDWARE UPDATE

THERE are lots of new bits of hardware to hook up to your QL – mainly monitors and printers, but lots more is in the works . . . The first piece of hardware developed by Kaga Electronics Company Ltd. in conjunction with Sinclair Research Ltd. has been launched for the QL. The Sinclair Vision QL Monitor has a 12" non-glare tube, 85 Column text display, and is ideal for high resolution colour graphics. The Sinclair Vision QL Monitor is recommended by Sinclair Research Ltd. for use with the QL. It is manufactured for and sold, guaranteed and serviced by MBS Data Efficiency Ltd.

To place your order, telephone 0442 60155 (MBS Data Efficiency Ltd, Computer Peripherals Division) . . . SCI of Petersfield, Hants is offering a Cosmo-80 serial dot-matrix printer (80 CPS in normal print mode) which promises 'plug-in and go' compatibility with the QL and the ability to print Easel graphics for £219 plus VAT. . . . A world-standard modem that should work with the QL's serial ports is being offered by Miracle Technology of Ipswich. It offers a number of baud rates and the ability to handle both UK (CCITT) and US (BELL) signals. The modem can also take an add-on auto-dial/auto-answer expansion board. It sells for £129.95. . . . Microvitec of Bradford is selling a special colour co-ordinated version of its Cub RGB medium resolution colour monitor for use with the QL. It can handle the QLs 85 column display, offers a resolution of 653 by 585 pixels. Microvitec is selling the jet-black Cub for £275 although it also offers a standard resolution version of the same monitor for £225. . . . JVC's entrant into the monitor sweepstakes goes by the name 'Model 1302-2 High Resolution Monitor' and is available from Opus Supplies of London. It also claims to have cracked the 85-character challenge and offers a resolution of 580 by 470 pixels. The monitor sells for £229 (including VAT, carriage and one year's guarantee) . . . A Centronics printer interface for the QL is being sold by Transform of Kent. Although the QL will work quite happily with serial printers – and most parallel printers can have serial interfaces installed in them – external printer interfaces offer an alternative. The interface sells for £49.95 . . . That's all we've got room for here, but lots more hardware goodies for the QL are either here or on their way – and there will be more details on some of them in the next issue.

SOFTWARE UPDATE

AS you've probably already discovered from reading our front page story, all QLUB members will be offered a 20 per cent discount on software ordered directly through Sinclair. This software includes . . . GST Computer Systems' QL Assembler, an assembler suite comprising both a powerful screen editor for creating and altering program files and a Motorola format-compatible 68000 assembler which converts source files written in M68000 assembly language into machine code or binary files capable of running the QL.
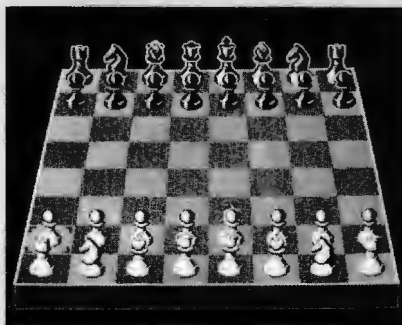
Both the assembler and editor are written in machine code and thus can be multi-tasked in SuperBasic – allowing you to switch between the editor, assembler and SuperBasic instantly. The QL Assembler sells for £39.95 including VAT (£31.95 to QLUB members) and is available immediately . . . QL Chess from Psion is an award-winning chess program featuring a three-dimensional Chess board, real-time clock, easy cursor-driven movement of the pieces and a separate movements screen. The program recently won the 1984 World Microcomputer Chess Championship and could prove very difficult to beat. If you feel you're up to the challenge, the program costs £19.95 including VAT (£14.95 to QLUB members) and is available immediately . . . Phone 0276-685311 to place your order or get further details. While you're deciding which (if not both) of those new software goodies to buy, have a glimpse at the range of upcoming QL software . . . QL Toolkit by Sinclair Research and Tony Tebby – a programmer's toolkit consisting of about 60 programs, the majority of which are linked into SuperBasic from the beginning and then can be used as command extensions or within a program.

Among these SuperBasic enhancements are; printer spooling – to allow you to print a file while running a SuperBasic program, file access – offering full random input/output commands, job control – this allows management of multi-tasking programs including the ability to display, alter priorities and delete multi-tasking jobs from SuperBasic, improved user interface – provides a full-screen editor, defaulted drive names, the amount of free memory displayed and an extra SuperBasic command display, Fitters and pipes – a Unix-like redirectable file input/output system which can be accessed through SuperBasic and a general set of utilities including user-defined graphics, an on-screen real-time clock and format conversions. It will sell for £24.95 (£19.95 to QLUB members) . . . QL Cash Trader from Accounting Software Ltd. – This package from a traditional mini and mainframe software supplier is designed to help in keeping track of transactions with a system that dynamically updates your profit/loss balance and makes the most of the QL's graphics in displaying that balance. Accounting Software promises that you'll only need a minimum of accounting skill to operate this book-keeping system. It will sell for £59.95 (£47.95 to QLUB members) . . . There's lots more software in the works, and much more than there is room for here.

However, this column could not close without mentioning an important new package for software houses; QL Linker . . . QL Linker is a relocating Linker commissioned by Sinclair Research from GST Computer Systems and should be available by January, 1985. It will allow the formation of programs from modules written in compiled languages and M68000 assembler. A standard format for relocatable modules, called Sinclair Relocatable Object File Format (or S-ROFF) has been specified. Compilers and relocatable assemblers available through Sinclair Research will support this format. If you're developing a compiler for the QL independently, however, you might want to consider using the Linker on a licence basis. Details – including the S-ROFF specification – can be obtained by writing to The Editor, QLUB News at the address given in the letters column.

Communications Corner

The Serial Port

THE QL is a very out-going machine and enjoys talking with other computers, modems, printers and storage devices. This particular corner of the QLUB News interfacing section is devoted to using the QL to communicate, particularly with modems and printers through the serial port. In this issue, we look at the device-independent nature of QDOS and how you can use that to your advantage in communications.

The QDOS operating system used by your QL is what's known as 'device independent' in its input and output.

This means that the QL is just as happy sending information to a printer or microdrive as it is sending information to the screen. This is a very powerful facility.

If, for instance, you set up a program to print out a form on your printer, you can 'pre-design' the form on-screen by changing the output device from screen to printer. To accomplish this, you just change the advice you OPEN the channel to and the output will appear in the same manner. (Obviously, the screen window you use must be the same width as the printer.)

A perhaps more useful application is sending 'control codes' to a printer. Control codes determine things like what typeface your printer will select, what the line-spacing will be and what print density is used. On an Epson FX-80, for example, the code for proportional spacing mode is set by sending Escape, "p", "1" in sequence. You COULD do this by typing:

```
100 OPEN #3,ser1
110 PRINT #3, CHR$(27);"p1";
120 CLOSE #3
```

However, if you instead type:

```
COPY con_ TO ser1
```

And then press the ESCAPE key, the p key, and the 1 key, you will have done the same thing. The more codes you have to type, the easier this method is. You can then stop sending codes by using the key BREAK sequence (CTRL and SPACE).

Similarly, another useful trick is used to list a program to a printer, so that you don't need to type:

```
OPEN #3,ser1
LIST #3
CLOSE #3
```

And instead just the command:

```
SAVE ser1
```

SAVE and LIST are almost the same thing, because the QL always stores programs as

SER1		
pin	name	function
1	GND	signal ground
2	TxD	input
3	RxD	output
4	DTR	ready input
5	CTS	ready output
6	-	+12V

TxD Transmit Data
RxD Receive Data .

SER2		
pin	name	function
1	GND	signal ground
2	TxD	output
3	RxD	input
4	DTR	ready output
5	CTS	ready input
6	-	+12V

DTR Data Terminal Ready
CTS Clear To Send

their text representation, so that SAVE and LIST both send ASCII (unformatted ordinary text) to the chosen destination. This accounts for the way that the QL loads Basic programs a block at a time, as it is passing them.

This has the advantage that you can list a Basic program from another machine to the QL's serial port, and LOAD it into the QL. SuperBASIC stores the number of spaces that a line is indented by as a count, so your program is the same length whether you indent it or not.

Finally, if you are communicating with another computer on the serial ports, you may find it useful to use the "z" option, as in "OPEN #3,ser1z". This means that the end of the file being sent or received will finish with the character CONTROL-Z

(ASCII 26). This means that if you are receiving on the QL, you will not have to use BREAK to exit from the file transfer. So, for example, you would type.

```
COPY ser2z TO mdv1_receive_file
```

on a receiver, and

```
COPY mdv1_transmit_file TO ser1z
```

on a transmitting QL. If you are transmitting with another type of computer, then you should persuade it to put CONTROL-Z on the end of the file.

NEXT ISSUE: A terminal program listing to let your QL use a modem to connect to Prestel, Micronet and various computer bulletin boards.

Using Epson FX-80 printers with Easel
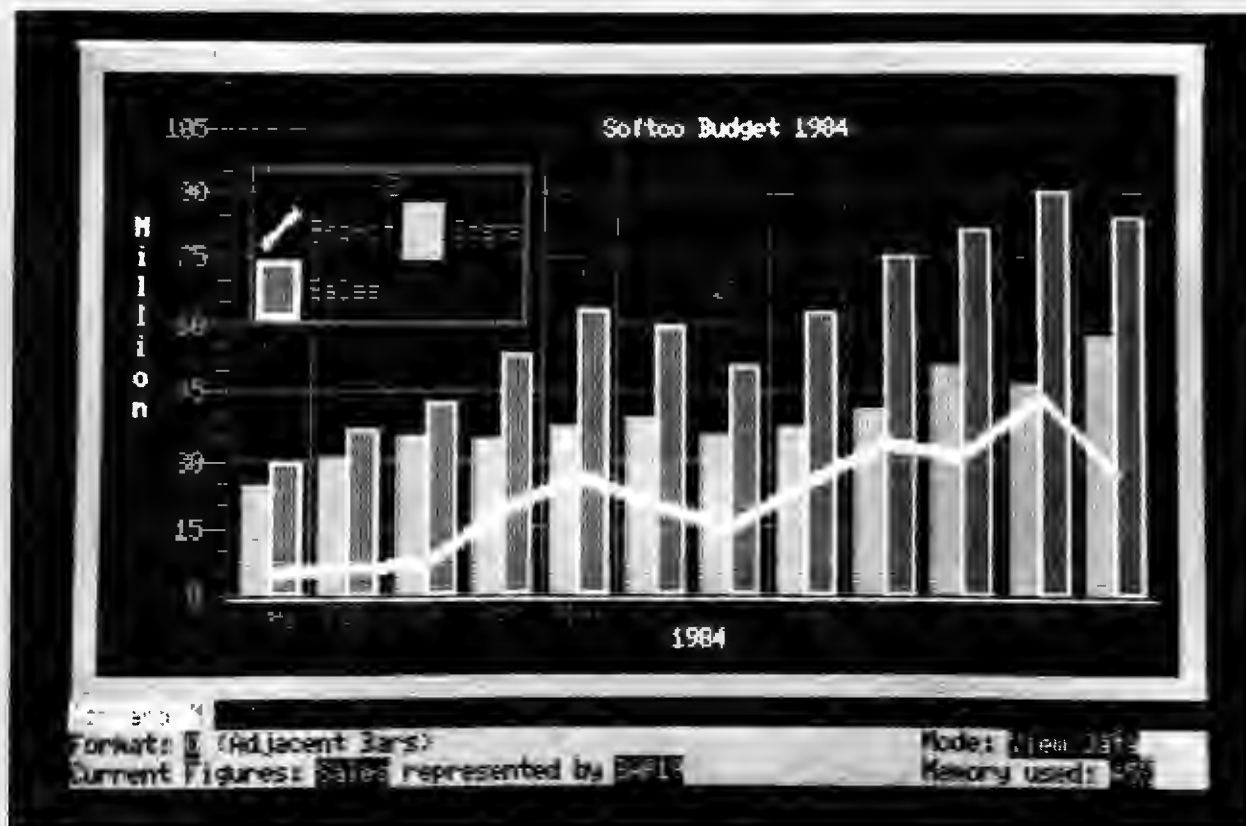
A NUMBER of problems can occur when using Epson FX80 compatible printers with Easel. The first problem is that the printer's default BAUD rate is not set at 9600 BAUD and changing this may involve dismantling the printer to some extent and altering various switch settings inside the printer.

It is, however, possible to change the BAUD rate at which the EASEL printer driver operates. This is done as follows by using the BAUD command is SuperBasic (SB).

Before loading the EASEL program, go into SB and type–

BAUD number

where number is your printer's required BAUD rate. To set the BAUD rate to 1200 BAUD, simply type–

BAUD 1200.

Now place the EASEL cartridge in drive 1 and run EASEL by typing–

LRUN MDV1_BOOT

Remember that to change the BAUD rate, you must go through the above procedure every time you run EASEL. You may, however, change the SB program–

'BOOT' on a copy of your EASEL cartridge and insert the BAUD command as the first line. Then delete the original BOOT program and save your edited version with the name BOOT.

NOTE: The following problem and solution only apply to EASEL version 1.01.

Running at a slower BAUD rate can introduce a second problem. The EASEL 1.01 graphics printer driver has a built-in 'timeout' which means that if the printer has not finished printing a line within 10 seconds, the printing is aborted.

If the timeout facility wasn't provided and no printer was connected or the printer had a fault (ie out of paper) the program would wait forever. To recover, you would either have to plug in a functioning printer or reset the machine.

On some printers even printing at 9600 BAUD can take longer than 10 seconds making it impossible to print more than 1 or 2 lines.

To change the timeout requires a 'patch' to one of the files on your EASEL 1.01 cartridge this is achieved by running

the following SuperBasic program with a CLONED copy of EASEL 1.01 in drive 1.

```
10 LET tout=3000
20 LET a=respr (3500)
30 LBYTES mdv1_sprint_prt,a
40 POKE_W a+94,tout
50 DELETE mdv1_sprint_prt
60 SBYTES mdv1_sprint_prt,a,2000
```

The variable 'tout' gives timeout in 50ths of a second, hence a value of 3000 gives a timeout of 1 minute. The user may alter this to suit his/her own needs. A timeout of 1 minute should be adequate for most printers.

Running this program creates a new version of the EASEL 1.01 printer driver on your EASEL 1.01 cartidge which can then be used normally.

Note. After running this program it is advisable to reset the machine when the microdrives have stopped going. This is only because the program permanently allocates itself 3500 bytes of memory and the easiest way to reclaim this is to reset the machine.

Quantum Leaps: The QL Case Study

The QL will be used for many different things by many different people. They will either tailor existing software or write their own programs to accomplish specific tasks with the machine. Each of these adaptions is, in its own way, a quantum leap. This column is devoted to highlighting such uses of the QL so that these advances can be shared by all users.

THE first case study has been sent in by H.R.B. Hack and describes how Archive can be adapted for cataloguing scientific references:

"Storing and accessing their references has always been a problem for scientists and research workers in many disciplines. It can give rise to much wastage of time (and emotional stress). Punch hole cards have been widely used, but can be cumbersome and tedious and not exactly user friendly. The use of a mainframe computer is, of course, tempting, but so much of the work can be and is done at home that a home computer which can supply sufficient reliable storage with good accessing facilities can provide the long awaited solution for individual workers outside their place of work.

I enclose an example of an ARCHIVE Record which includes a bibliographical reference to and summary of a published report of an agricultural study of the crop Corn (maize). There are 19 ARCHIVE Fields. The strings naming the Fields are on the left. The variable names are 'A$', 'B$' etc. The insertion of "(CR) (LF)" to produce the new line within the Field is based on a PSION suggestion of using a tilde and the "TRANSLATE" facility of the printing software.

You will notice that, although they include abbreviations, a number of Fields, particularly S1, S2, S3 and S4, have more than 80 characters. Field S1 is most conveniently arranged as four lines. The new line facility is needed where small tables are required.

I do not know how many bytes are required for storage of a Record such as Ref. no. 15 but, speaking from previous experience, similar Records could use 700 bytes. A cartridge would then store some 140 Records which could be in a single file. A greater number of Records would necessitate splitting the file into two or more files, since one file cannot be stored in separate parts on different cartridges. Having at present no extra microdrive, I have made a preliminary trial of subdividing the 19 Fields into four groups, each group comprising the Field of one screen Display and being stored in its own file on MDV2. The groups are linked by the common reference number. There has been little difficulty in devising Procedures which provide ready access, display and printing."

H.R.B. Hack
Shepherds Green
Henley-On-Thames

```
                                                        Ref. no. 15

Senior author: Egharevba, P.N.
Other authors: Horrocks, R.D., Zuber, M.S.

Location:     Columbia, Missouri, U.S.A.
Publication:  Agronomy Journal 68: 40-43
1976

Title:  Dry matter accumulation in maize in response to defoliation

Crop:        Corn
Plant parts: DRYMAT. Leaf. Silk. Tassel. Grain.
Methods:     Field.
Instruments: --
Time:        Growth stages: from 50% silking, at each 10 days for defoliation,
             at each 5 days for DRYMAT measurement.
Space:       40,000 pl/ ha.

Y:  DRYMAT. Yield. Yield components (YDC).
X:  Defoliation, lvs all or above or below ear. Leaf area.
    Cultivars.

S1: Yield loss from defoliat.:
    1. Total, DY% = 101.453 - 2.049*d.
    2. Partial, DY% = 37.927 - 0.745*d.
       (d=10,20...50 days after 50% silk).

S2: Measurable effect on DMA from:
    Total defoliation after 10 d; partial defol. after 20 d.
    Defol. above/below ear has same effect.

S3: No. of kernels / unit area and 1000 kernel wgt affected progressively
    less by defoliation 10,20,30,40 d after 50% silk.

S4: With early defoliation kernels near tip of ear did not develop.
    With total defoliation nearly 50% did not develop fully.
```

If you have any unique application idea for your QL which you'd like to share on these pages, write to:

**Quantum Leaps:
The QL Case Study
The Editor QLUB NEWS
Sinclair Research, 25 Willis Road, Cambridge CB1 2AQ**

Programmers Forum

SuperBASIC is a revolutionary and complex high-level language.

The IF statement is a good example. It exists in different forms and SuperBASIC tells which form is required by the way the initial statement line is ended. If the line terminates without another statement on it, for example:

IF a=b THEN

then SuperBASIC assumes that the long form of the IF statement is required and so expects at some time in the future to see an END IF or an ELSE followed by an END IF.

SuperBASIC is still "a BASIC" and so only considers a single line at a time. It cannot infer whether the programmer requires the long or the short form of an instruction by looking for the relevant END statement.

If the statement ends with another statement, for example:

IF a=b THEN PRINT "a is equal to b"

then SuperBASIC assumes that the short form of the IF statement is used and no terminating ELSE or END IF is expected or desirable.

It should be obvious that the two types of statement cannot be mixed. So the following sequence of lines is illegal but will be processed by SuperBASIC because SuperBASIC is perfectly happy with the IF statement at line 100 although the END IF statement at line 130 will be treated as a null statement.

100 IF a=b THEN PRINT "a is equal to b"
110 LET a=a+1
120 LET b=b+1
130 END IF

This brings us on to an important feature of SuperBASIC. From the following listing we can tell that lines 110 and 120 are really inside a FOR loop simply by looking at the program.

100 FOR count =1 TO 10
110 PRINT count ! count*count
120 END FOR count

SuperBASIC has no idea of the above structure. SuperBASIC sometime in the past has executed a FOR statement and has created control variable, *count*. Sometime in the future, if SuperBASIC finds an END FOR statement or a NEXT statement which refers to *count* then all is well and the processing can continue. Between these two events SuperBASIC (or any other BASIC) has no concept of being inside a FOR loop and is simply processing a sequence of statements a statement at a time.

The important thing is that the structure, as far as SuperBASIC is concerned, is completely defined by the order in which the statements are processed, while for the programmer it is defined by the layout of the statements.

This can be demonstrated with the following program:

100 GOTO 120
110 END FOR n
120 FOR n =1 TO 20
130 PRINT n
140 GOTO 110

The GOTOs ensure that the FOR and END FOR statements are processed in the correct order and so as far as SuperBASIC is concerned the structure is correct. Visually the structure is almost undecipherable and should only be used by programmers with a masochistic temperament. And this program will never terminate because whenever the loop finishes, it immediately falls back into itself again.

This feature applies to all the SuperBASIC structure constructs and to the procedure and function constructs.

If a program is stopped while it is executing a procedure or a function SuperBASIC cannot tell that it is inside a procedure or function and so the status of the machine when stopped remains in the "executing procedure" state.

The state would normally return to normal when the END DEF or RETURN statements are executed.

When it is inside a procedure (or function) SuperBASIC is executing SuperBASIC statements with the procedure's formal parameters, parameters replaced by the actual parameters and the actual parameters replaced by the formal parameters (if the actual parameter was an expression then a temporary variable is created to hold the result). This is reasonable enough and is quite useful. However, if the program is not restarted correctly then the variables will always be swopped over and it will be impossible to access the actual parameters.

This is partly responsible for the strange behaviour when a program is edited or changed when stopped inside a procedure. Performing a CLEAR first will ensure that the state of SuperBASIC is reset. SuperBASIC cannot perform this clear itself because it does not know that it has stopped inside a procedure, once again all it knows is that sometime in the past it executed a procedure call and sometime in the future expects to execute an END DEFine or RETurn statement.

Many aspects of SuperBASICs behaviour which at first seem a little strange usually turn out to be because SuperBASIC is trying to make life easier for the programmer and the behaviour easily explained.

> The important thing is that the structure, as far as SuperBASIC is concerned, is completely defined by the order in which the statements are processed, while for the programmer it is defined by the layout of the statements.

QLibrary

YOU will no doubt already have seen a huge array of books appearing about the QL and by now may well have either bought one or considered buying one. To help in this choice, QLibrary will feature a few interesting books in the next couple of issues in order to give you a better idea of what's available.
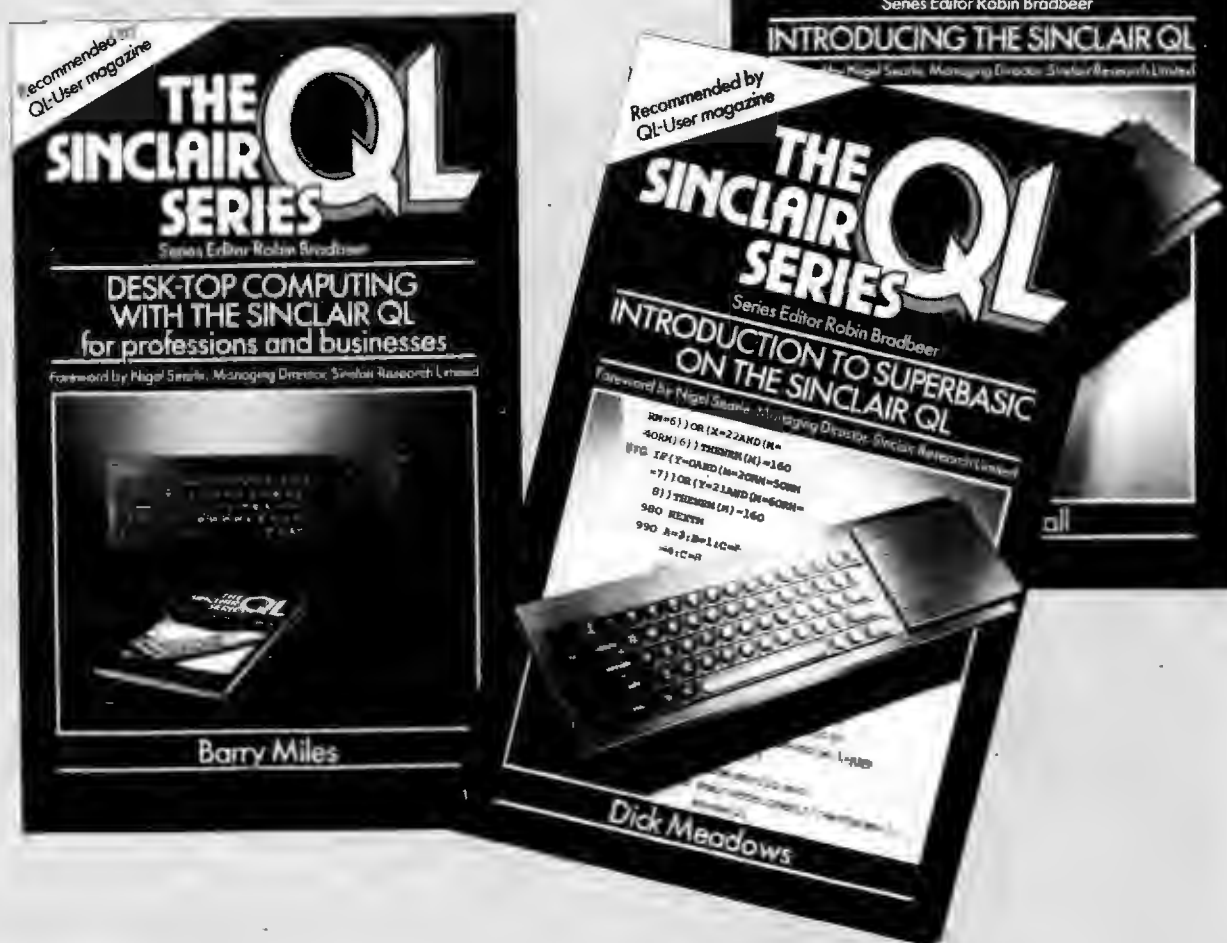
THE information here by no means constitutes book reviews, nor recommendations by Sinclair Research—but rather information that we're passing on. Like a real library, we'll show you a large stock of books and it will be up to you to decide for yourself what you really want to read.

This issue we're featuring a range of books about the QL from TBS. The books have been edited by Robin Bradbeer, longtime computer journalist and consultant. Robin has been around a long time—he wrote the original ZX-81 manual and started up some of the first magazines about Sinclair hardware products.

But getting back to the books, they're in a series simply known as The Sinclair QL Series. You may well have seen the ads for it—they feature the words Quantum Leap in large black letters on a light blue and purple background.

They also feature a quote from Nigel Searle—who said of the series; "I am certain that these books will add enormously to the enjoyment and practical use which QL users will get from their computers." And the series also claims to have been recommended by no less an authority than QL User magazine.

The books are;

Introducing the Sinclair QL—a beginner's guide to what the QL does, how it works, and some idea of how you can make the most of it.

Introduction to SuperBasic on the QL—An attempt to let beginners come to grips with the unique features contained in SuperBasic.

Advanced Programming on the QL—Calls itself 'an essential reference work for users who really want to get to grips with the Sinclair QL'. Among its listed topics are programming techniques, program logic presentation and document design.

Desktop Computing with the Sinclair QL—How to get the most from the four Psion software packages that come with the QL.

Word Processing with the Sinclair QL—This one attempt to explain the whys and wherefores of wordprocessing and how they can best be exploited using Quill.

The books cost £6.95 each and can be ordered from TBS, 38 Hockerill St., Bishop's Stortford, Hertfordshire.

THE TEN COMMANDMENTS OF QUILL

by Martin Stamp

QUILL, the wordprocessor that comes with every QL, is designed to be easy for the first time user.

As far as possible, what you see on the screen is what you'll end up with on your printed copy. Prompts at the top of the screen are there to guide you at every stage, and help messages appropriate to what you are doing are available at any time by pressing F1.

More experienced users will find that QUILL is a sophisticated wordprocessor with such features as 4 different tabs, soft hyphens, word count and dynamic display of both hard and soft page breaks.

Wordprocessing is at one level a very simple concept: you want to put words and numbers onto a page in the same way people have for a hundred years with typewriters. But at another level it is not so easy to grasp and does take some time to learn. For example, the screen can only show you part of the document at one time although, you can move up and down using the arrow keys. QUILL also enables you to modify the text by changing justification, margins and typeface.

The problem for a wordprocessor designer is to make these ideas easy to grasp and simple to use. What follows are some suggestions on how best to use QUILL.

1. First—don't be scared.

Don't be intimidated if you come across a few words you don't really understand; you don't have to understand every QUILL function to produce professional-looking documents.

2. Before using QUILL, make a copy of the software as described in the manual. Never use the original software, keep it as your master copy.

3. If you are new to wordprocessors it's a good idea to practise on a test document to become familiar with QUILL. This way you can experiment without any fear of losing valuable text.

A beginner might start by typing a couple of lines of text. Observe how the text is formatted on the screen with a straight right margin using 'word wrap'

(word wrap simply means you only have whole words on a line). Remember QUILL automatically starts new lines, there is no need to press [ENTER]. Move round the text with the four arrow keys, the cursor shows you where you are. Try deleting some text using CTRL & ‹—. Notice how the text reformats immediately. Press [ENTER] to start a new paragraph and notice how it automatically starts with an indent (that is to say the first line of each paragraph is set in from the rest of it). Press CTRL & ‹— immediately after [ENTER] to see how to delete unwanted paragraphs.

Continue using these functions, inserting and deleting text till you feel confident. They try out a couple of commands—MARGINS and JUSTIFICATION are easy to pick up. By moving the cursor to the top of the document and pressing the sequence

[F3] M I ‹— ‹— ‹— ‹— ‹— [ENTER]

you can set the text so that the indent margin is the same as the left margin.

The sequence

[F3] J L]ENTER]

enables you to give your text a straight left margin and ragged right margin (called left justification).

They try out some of the other commands such as SAVE, LOAD and PRINT which are necessary for serious use.

4. Before starting to work with QUILL in earnest, try out all the features you will use—if you are going to have columns of figures use TABS; if you want bold text, use TYPEFACE.

5. Make a copy (or in computereze, a backup) of all important documents and files. Use the sequence

[F3] O F B

to set up the BACKUP command—then type the first file name followed by [ENTER] followed by the new name and [ENTER].

6. Once saved, each document has a name; if you use the same name twice you will overwrite the first one and its contents will be lost for ever. QUILL warns you that you've chosen the same name by asking if you really want to overwrite

(if you don't press ESC).

7. If there is a particular format (that is to say margin setting, justification, tab settings) that you use repeatedly set up the format on a blank document and save it: then whenever you want that format all you have to do is load that document.

8. QUILL runs faster if the Microdrive cartridge in MDV2 (the one on the right) is not too full of data, at least 50 free sectors is recommended. More RAM memory (when available) also speeds up operation as will the new software release which gets rid of overlays.

9. For the more advanced user.

When you load a document into QUILL, it makes a second copy (called def-doc) so that the original remains untouched. It is safest if these two versions are on different cartridges. To achieve this:

Start with a formatted cartridge in MDV2. Go into the LOAD command ([F3] L) and when both drive lights have gone out replace QUILL with the cartridge containing the document. Type MDV1-NAME (replacing NAME with the name of your document)—QUILL then copies the document onto MDV2. When the command is finished replace QUILL in the left hand drive.

To save the document to the original cartridge go into the SAVE command and again wait for the drives to stop and replace QUILL. Press [ENTER] and Y to overwrite your original version.

10. There is an old computer saying which may be worth remembering: "If all else fails, read the manual."

Note: [F3] means press Function key 3, [ENTER] means press the enter key, and ‹— is the left arrow key.

The spaces in the examples are for clarity, don't type them in!

---

1. **Don't be scared.**
2. **Make a copy of the software first.**
3. **Practise on a test document.**
4. **Try out all the features first.**
5. **Back up all important documents.**
6. **Never use the same document name twice.**
7. **Save customised formats in a separate file.**
8. **Keep a half empty cartridge in Microdrive two.**
9. **Use the def-doc procedure.**
10. **If all else fails, read the manual.**

Hints on using Abacus

by Colly Myers

MEMORY SIZE

THE basic unit of storage in ABACUS is a single cell. Regardless of whether a cell contains numeric information or text information each cell will require 14 bytes (characters) of memory storage. Thus in Version 1.02 of ABACUS the total capacity of the spreadsheet is approximately

$$16k/14 = 16*1024/14 = 1170 \text{ cells.}$$

Being a very memory efficient program, ABACUS does not store any information for cells which are empty. Thus if there is a value in cell A1 and a value in cell BL1, ABACUS only uses 28 bytes of storage whereas many other spreadsheets would store all the missing values giving a storage requirement of 896 bytes. In this way ABACUS allows you to use large areas of the grid without prematurely running out of memory.

Indeed surveys of other spreadsheets show that they require between 30 and 50 bytes of storage for every cell in the spreadsheet. Consequently ABACUS allows you to have at least twice as many cells for each 1k of memory than any other spreadsheet. Thus although 16k may not seem to be a very large amount of space it is in fact quite large enough for most requirements.

Finally you may have noticed that after doing a ZAP command you are not left with as much memory as when the program was first loaded. Do not worry as this is quite natural, and occurs because of some initializations, which never get reset by the ZAP command. You always have the same true amount of memory as when the program first started. The amount of this information varies from 1k to about 2k depending on what's happened prior to the zap command.

UNITS AND JUSTIFICATION

If you set the units on a cell to say MONETARY, before a value has been given to the cell, you may have been surprised when you entered the value to find that it came out as if nothing had been changed.

The reason for this, as was mentioned above, is that cells with no values in them are not actually stored in memory. As the format for the cell, is stored with the cell, then obviously a cell which does not exist cannot have a format.

What actually happens is that the program holds what is known as the default for such things as the UNITS of the cell and the JUSTIFICATION of the cell. When an entry is added to the grid which did not exist before then the format of the cell is made up from the current default's. This is why the commands UNITS and JUSTIFY ask if you want to change the default or cells.

If you change the default then all new cells created will be given the formats you set up. If however you specify cells then all the cells which already exist in the range you give, will be changed to the new format.

It is probably best to leave the default alone while you set up your spreadsheet and then just before printing, go through the spreadsheet and change all formats to suit your requirements.

FORMULAE AND NAMES

One of the special facilities that ABACUS provides to make using the spreadsheet faster and easier is the ability to use whole rows and columns in a formula. This is a lot more efficient than other spreadsheets, where you have to enter a formula in one cell and then echo it to all the cells which you want to contain the formula. ABACUS can be instructed to do this in one go by specifying ROW=formula or COL= formula.

As well, any cells which contain text may be used as labels for columns and rows of cells. Thus a formulae may be entered as

PROFIT = SALES — COSTS

One final advantage that ABACUS can offer as a result of using these whole row and column operations is that ABACUS can remember that all the cells which were created using the same formula must be kept together.

Thus you can amend any cell which uses the formula and all the cells using the formula will change to the new formula. All other spreadsheets insist that you must echo the formula over the cells that you wish to use the formula every time you change it.

In order to take maximum advantage of automatic naming in ABACUS it is as well to remember that the use of names is a "TOOL" and like any tool will work as well as you choose to use it. E.G. Trying to knock a screw into the wall with a chisel would convince you that both the screw and chisel were not particularly effective tools.

Firstly, what constitutes a valid name? The rule is simply:

A name must start with a letter and can be followed by any number of letters or digits. Thus the following are some good names

PROFITS
SALES83

And some invalid names

83SALES (starts with a digit)
PROFITS/FIRST (/ is not a letter or digit)

This probably appears very restrictive, but look at the example of PROFITS/SALES and you can probably see why there is this limitation. Of course ABACUS cannot tell if you meant a name PROFITS/FIRST or PROFITS divided by SALES.

Finally there are some special names which although they obey the rule for names may never be valid names. These are the actual cell names A1, A2, BL3 etc. Thus even if you put TEXT of A1 into cell B2 and then do a goto A1 ABACUS will go to cell A1 and not cell B2.

PRINTING

In many cases the spreadsheet you have is a lot wider than the size of the paper you have. Don't worry, just print the whole size of the grid and ABACUS will print the grid in two or more parts as necessary.

However it is quite likely that your printer can probably print more than 80 characters across if it is a dot matrix printer. You might be wondering how to make use of this facility.

All that is required is that you run the PRINTER INSTALL basic program and change the PRE-AMBLE string to include the necessary codes to put your printer into condensed print. On the EPSON FX80 range of printers this is CTRL O or 15 decimal or 0f hex. Install the modified printer. This will have created a file on the microdrive called PRINTER-DAT. Copy this file onto all the cartridges that contain ABACUS.

From now on whenever you print the printer will automatically go into condensed mode when printing. Don't forget to change the setting for the number of characters across the page in ABACUS by using the DESIGN command or else ABACUS will still think that is the default of 80.
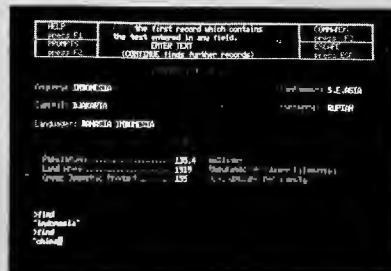
Archive Files

By Charles Davies

ONE of the most common questions asked about Archive on the QL concerns the maximum number of records an Archive file (an _dbf file) may contain.

Archive uses a sophisticated method of record management which has many advantages. A consequence of this sophistication is that it is quite complicated to estimate the maximum number of records an Archive file may hold. To do so requires some understanding of the structure of Archive files. In this article, I will explain some of the details of the structure of Archive files on the QL and how to estimate the maximum number of records.

Files, Records and Fields

As with nearly all database programs, the information which makes up the database is structured into a number of files where each file is split into records and where each record is split into fields. What interests us here are the records.

Variable Length Records

The files in SuperBASIC and QDOS are handled as simple sequence of bytes with no other structure imposed on the files. A sequence of bytes is too loose for Archive which needs to split the files up into RECORDS where each record contains many bytes. Most large operating systems (on minicomputers and main-frames) do this splitting for you using an operating system program called the RECORD MANAGER. Not unreasonably, QDOS does not have a record manager so Archive has to do its own record management.

Most other database programs use FIXED LENGTH RECORDS. This is much easier to program than VARIABLE LENGTH RECORDS but is not so easy on the user. With fixed length records you make the record length that of the longest record you are ever going to need and waste the space you don't use when your records are not all of the same length. In addition, when a file is created, the fixed length record manager needs to be told how long the record is going to be before any of the data can be entered. This is specified by the user by specifying the maximum length of each field. Often the user does not know what the maximum length is

(e.g. what is the longest surname?) and tends to overestimate the maximum length especially if he or she does not appreciate the implications on the efficiency of the stored data.

Archive uses variable length records where the records are packed together. This has the principal advantage that you, the user, are freed from having to specify field lengths and secondly that the data is stored more efficiently. Archive has to work harder because it is no longer trivial to find where each record starts and it does this by keeping an index of the record positions and record lengths in memory. This memory resident index is also optionally used to keep a key.

Indexes and Keys

A key is defined in Archive by using the order command. For example, using gazet_dbf, the command:

order country$;a

makes the key the field country$ with an alphabetical order going from a to z (i.e. ascending). Apart from putting the records into an order, the key provides a means for faster record access via the locate command. For example (having previously ordered as above),

locate "TURKEY"

which uses the index is much faster than

search country$="TURKEY"

Try it! The locate method finds the exact file position by doing a binary search on the index (which lives in memory) and going straight for the required record. The search method slavishly scans each record stopping when it has found one which satisfies the search condition. The larger the file, the greater the difference between the speed of the two methods.

How Many Records?

We are now in a position to answer the original question of how many records an Archive file can have. There are 3 limiting effects:

(1) **Internal structure of the Archive program**

This limits the number of records to 64K. However this is a theoretical limit only which is superseded by the following two effects.

(2) **Microdrive capacity**

The approximate length of an Archive file is equal to the sum of the records lengths. The record length may be calculated from the following:

Each numeric field costs 8 bytes

Each string field costs one more byte than its length.

Since the length of each string field may vary, the average expected length must be used to estimate the file length. The file will actually be longer than this because the index and various other information must also be stored. Microdrive capacity is approximately 100K bytes (1K=1024) but the I would not recommend using files larger than about 80K because write operations are likely to slow down if the nearly all the microdrive blocks are used.

(3) **Memory capacity**

As described above, Archive uses a memory resident index to keep track of records and to keep keys. You do not have to have a key in which case each record requires 6 bytes in memory. If a key is built (using the order command), it may be built from 1 to 4 fields adding 8 bytes to the index for each field. To summarise, each record requires the following amount of memory when the file is opened:

Key	Requirement (per record)
No key	6 bytes
1 field	14 bytes
2 fields	22 bytes
3 fields	30 bytes
4 fields	38 bytes

To find out how much memory you have, enter:

print memory()

In the current version of Archive, you have about 10000 bytes of free memory. This translates to a capacity of 1666 records with no key and 714 records with a key consisting of one field.

It is not as simple as this because there are other parts of Archive which compete for memory but the only other one of significance are programs which have been loaded into memory. To find out how much memory a program uses up, load the program before doing a print memory().

Whether you are limited by memory capacity or of microdrive capacity depends on your average record length— the breakeven point is a record length of about 50 bytes based on a maximum file length of 80K bytes with no key.

If you are limited by file length, all you can do is wait for hard disks to become available for the QL. If memory space is a limitation, then you can either wait until extra RAM boards are available and buy one or it may be sufficient to wait for the next release of Archive. This will be 100% hand machine coded and will increase the available memory by between 50% and 100%.

Make Effective Use of EASEL Graphics

by Martin G. Brown

Computer graphics are a powerful way of conveying ideas, and people are more able to retain graphically communicated ideas. But like any other medium graphics will only be really effective if the ideas are succinctly expressed.

If you want to put over a point with a chart, make sure it is the most appropriate one for the task, and don't clutter it with confusing detail. Below are some tips on how best to graphically present different ideas using EASEL.
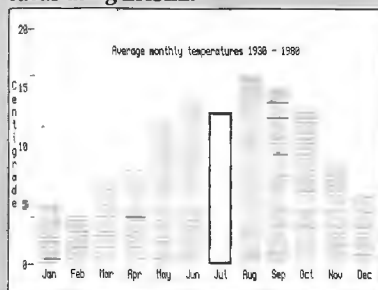

Figure 1

CHOOSE THE RIGHT FORMAT FOR THE JOB

Use the format which best helps convey the point you are trying to make with your graph. First, consider what category your data fits into. Usually it will be of the times-series or position-series type, or else of the comparative type, where order is not important and you are representing different things at the same time.

An example of time-series data would be average monthly temperature figures, such as in figure 1. Here the order of the data is important, we can look for trends and oddities in the figures, such a set of figures could also be represented by a line graph.

Data of the comparative type is shown in figure 2. There is no significance in the order of the bars here, and it would not make sense to show this metal production data as a line graph. A bar chart was used in this example because it clearly shows the actual production figures, which can be read of the axis, and it gives a powerful impression of the relative magnitudes of the production of the different metals.
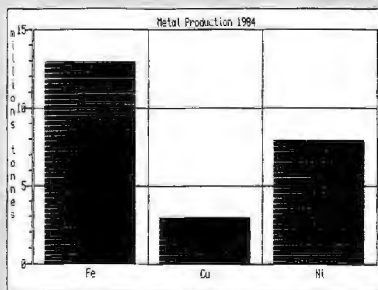

Figure 2

The eye is also very good at measuring angles, pie charts take advantage of this. Pies are also used for representing comparative rather than series data, but it is a special type of comparative data where the idea of parts of a whole is being expressed. In figure 3 the actual values of the sales figures for the different products are not important. The idea that is being put across is rather that of how many products there are and what fraction of the market each product possesses.
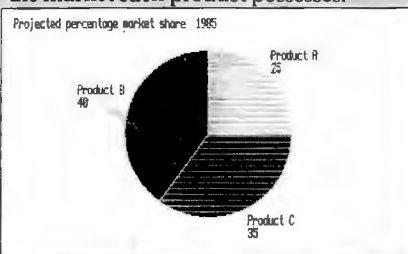

Figure 3

Stacked bar charts are a more sophisticated way of showing parts of a whole. The example shown in figure 4 has quite a lot of information in it, and using stacked bar charts with too many bars or too many values in each stack can be very confusing. From the total height of each stack of bars for 1983 and 1984 we can see the total of the sales figures in all three countries for each year.
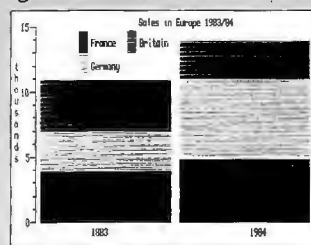

Figure 4

MAKE YOUR POINT CLEARLY

The whole point of using a chart rather than a table of numbers is to convey some aspect of the data to the viewer efficiently. If you clutter charts with too much data or too much text you risk losing the attention of your audience and not conveying your ideas to them. It is best to avoid jargon and technical terms on your charts, these can be better explained in the text or verbally if you are giving a talk.

Be most careful to avoid giving undue emphasis to one set of figures at the expense of another, unless this is what you are trying to achieve, you may end up giving entirely the wrong impression. For example, if you used very bold bars for your competitors sales figures and weak ones for your own you will leave a lasting impression of the size of the competitors sales in the mind of your audience.

ADVANTAGES OF USING EASEL GRAPHICS

Graphics can bring out relationships in data and also reveal hidden or unexpected relationships. The interactive nature of EASEL and the use of the EASEL evaluator for calculating new sets of figures from old ones are particularly important features here. It is most impressive, at a small meeting for example, to be able to calculate and graph new aspects of the data very quickly. If you had planned production figures and the actual ones you could easily graph the difference between the two and made speculations on the results, for example.

In figure 1 the Highlight option of EASEL has been used to emphasise the average July temperature. This is because although the chart shows the expected seasonal variation of temperature July at this location is unaccountably cooler than June. This kind of feature would have been impossible to spot from the raw temperature data.

MAKING SLIDES WITH EASEL

If you have a good quality TV set or better still a colour monitor (these are coming down a lot in price at the moment as their use is more widespread) you can make very high quality slides from EASEL with tremendous impact. It is very easy to do if you follow these guidelines; all the advertising shots for PSION software were in fact done in-house using these techniques.

(1) For best results use a lens with fairly long focal length, around 100mm. This will eliminate curvature effects caused by the curved front of the tube which become more pronounced the nearer the camera has to be to the TV. A 50mm lens can still give satisfactory results if no other is available.

(2) Use film of 64 ASA or more, if you use very slow film you will get odd colour effects from having to use longer exposures.

(3) Mount the camera on a tripod so that it is at the same height as the centre of the screen. Move the camera right up close to the screen to make this adjustment. This is to avoid distortions caused by looking at the screen from above or below.

(4) A good way to get the exposure perfect is to use the graph paper design screen of EASEL to produce a screen that is approximately 50% black and 50% full white by typing 'C' for change and 'G' for graph paper and ENTER. Now press ENTER and the example graph paper will turn white. Adjust your cameras exposure for this light level making sure that you use an exposure time of at least 1/4 second. If you try and use a shorter time you will have problems as you may only capture part of the TV picture, which is refreshed every 1/25 second. As an example, using 64 ASA film with a fairly bright monitor an exposure of 1/4 second at f5.6 might be expected.

(5) Use a cable release or the auto timer to avoid blurring the picture by handshake.

In forthcoming articles I will suggest further ways of making effective use of EASEL's graphics and delve deeper into its powerful features.

QLose–up

David Karlin:

DESIGNING A DREAM

ABOUT two years ago, David Karlin was given a rare opportunity.

Shortly after joining the Sinclair team, the young hardware designer was asked to think about the minimum requirements for his 'dream computer' and then go about designing that computer. David decided that he wanted a fast, but well-supported processor (the Motorola 68008), a professional keyboard, high-resolution graphics and an 80-column display, an inexpensive, but quick, storage medium and finally the ability to use windows and multi-task.

One thing he didn't originally envisage for the QL was a colour display, as his own uses for a computer didn't dictate the need for one. "Luckily, a few other people persuaded me otherwise," he says now. David points out modestly that development of the QL was very much a collaborative process, although the basic specifications of the machine came from him.

Although some aspects of the machine's design could be controlled, others—such as the size and shape—were more or less predetermined. David says that although keeping the size as small as possible was a concern, you cannot go smaller than a certain size in order to fit a workable keyboard and two microdrives into a box.

After some research to reduce them, the size of the microdrives was fixed and the keyboard had to fit certain ergonomic standards, requiring a minimum spacing of keys and dictating a minimum size.

Some reviewers have criticised the keyboard, but David defends it by pointing out that although it may not initially feel like a traditional professional keyboard, it is designed to be adjustable and its characteristics changeable within software. He adds that keyboards are very much a matter of personal taste and that one person's dream keyboard may be another's clunker. He is amused by the way some people use the keyboard. Because it is not 'stepped,' those little plastic feet are provided for use with the QL—but David says some people just don't bother with them. He adds that he had considered using IBM PC-style flip-down legs with the keyboard, but decided against it because he couldn't find any ·sturdy enough. He also defends the microdrive technology, which has been generally misrepresented as either a slow disk drive or a fast cassette system. David points out that it is, in fact, a new storage technology in its own right. So what do you do for an encore with a first machine like the QL? David says he's just keeping his head down and working on QLs for North America and getting UK add-ons ready.

Your chance to win a Sinclair Vision QL Colour Monitor

NOT only do you get discounts and information with your QLUB membership, but now you also get a chance to enter competitions. This issue we're offering you a chance to win one of three Sinclair Vision QL high-quality colour monitors – each worth £299. All you have to do is fill out the questionnaire below (don't be afraid to be honest, it won't harm your chances) and then answer the two skill testing questions in the entry-form at the bottom of the page. We'll put the forms of all the entrants who have answered all the questions in the questionnaire and have correctly answered the two 'skill' questions into a hat and then draw out three winners.

The winners will be announced in the next newsletter and a presentation ceremony will happen a short time after that.

Here are the questions:

1. How do you rate the general quality of the Psion software bundled with the QL (Quill, Abacus, Archive and Easel)?

a) Excellent   b) Good   c) Fair   d) Poor

2. Which program do you use most often (if more than one, circle the appropriate names indicating with a 1, 2, 3 etc. which you use the most)?

a) Abacus   b) Quill   c) Archive   d) Easel

3. Which, if any, do you not use?

_____

_____

4. How often do you use the data transfer facility of the software (import, export, etc.)?

a) Often   b) Once in a while   c) Seldom
d) Never

5. Do you use a printer with any software? If so, indicate which package(s).

_____

_____

6. Do you use a TV or a monitor for display?

a) TV   b) Monitor

7. What additional business software – if any – do you plan to buy when it becomes available?

_____

_____

8. How many blank cartridges have you bought?

_____

9. Which of the following peripherals are you most likely to need in the near future?

a) Modem   b) Memory   c) hard disc interface   d) Printer   e) Parallel printer interface   f) IEEE 48 8 interface

That was the easy bit. Now comes the tough part – answer the two following questions correctly, then fill in the other details and you'll be in with a chance to win one of the monitors.

1) Name the two processors in the QL.

_____

_____

2) Where is the QL manufactured?

_____

_____

Send this completed form with your membership no. to:

Membership no. [_____]

QLUB Questionnaire
The Editor
QLUB News
Sinclair Research Ltd.
25 Willis Road
Cambridge CB1 2AQ

Machine Code and More

In every pursuit there are people who seek out information at a level not generally pursued by others–specialists who work hard to be as enthusiastic and well-informed as possible. This page, Machine-code and more, is for just those people. While other sections of QLUB News will deal with SuperBasic programming and questions of more general interest, this page will be reserved for those with a keen eye for–and an interest in– machine code– programming. The QL will indeed be a challenge for you, but by sharing information we can help you meet that challenge.

THE gauntlet is thrown down in this issue with a digital clock/calendar program that uses the multi-tasking facilities of the QL. The clock has its own window, which appears at the top half of the default command window (#0) and its position can be altered by a change to the program.

The program consists entirely of calls to QDOS (of which more next issue), and is well-documented in this listing and the description below;

Lines 6 to 13 set up meaningful names for the various TRAP keys and utility sectors rather than using 'magic numbers'. Lines 16 to 20 form the standard job header, which can be read by various job utilities to show the name of the job. The initial BRAnch instruction could jump to a fixed address, allowing several jobs with their own data areas to execute the same code.

Lines 23 to 26 set the priority of the clock to a very low value, so it doesn't use a lot of CPU time when SuperBasic is running. The next lines, 27 to 32, set up a screen window for use by the clock: parameters for this are in lines 62 to 69 and can be altered if you don't like the clock where it is.

The main loop of the clock is in lines 34 to 53. The first two lines read the system clock into D1: this is then printed to the clock's window (lines 40 to 45) and the cursor reset to the top lift of the window by lines 46 to 52.

Lines 55 to 60 should never be executed–they are an error routine to printer an error message and then kill the job off.

The program and the SuperBasic routine to call it are presented:

```
100 a=RESPR(256)
110 RESTORE
120 FOR byte=0 TO 111 STEP 8
130   cs=0
140   FOR o=0 TO 7
150     READ x:cs=cs+x
160     POKE a+byte+o,x
170   END FOR o
180   READ x:IF x<>cs THEN PRINT "Error in
       line"!byte/8*10+1000:STOP
190 END FOR byte
200 SEXEC mdv1_clock,a,128,256
1000 DATA 96,0,0,14,74,251,74,251,760
1010 DATA 0,5,67,108,111,99,107,200,697
1020 DATA 112,11,114,255,116,1,78,65,752
1030 DATA 67,250,0,74,54,120,0,200,765
1040 DATA 78,147,74,128,102,48,47,8,632
1050 DATA 112,19,78,65,67,250,0,96,687
1060 DATA 147,206,54,120,0,236,78,147,988
1070 DATA 32,87,211,206,54,120,0,208,918
1080 DATA 78,147,74,128,102,16,66,65,676
1090 DATA 66,66,118,255,112,16,78,67,778
1100 DATA 74,128,102,2,96,210,54,120,786
1110 DATA 0,202,78,147,38,0,112,5,582
1120 DATA 114,255,78,65,0,0,0,7,519
1130 DATA 0,240,0,10,1,16,0,200,467
```

Basic Listing

```
 1                    ;    continuous date/time display
 2                    ;
 3          3F000 R         ORG     3F000H
 4                          GEN.L
 5                    ;
 6              B     MT.PRIOR EQU   OBH          ; key to set priority
 7              13    MT.RCLCK EQU   13H          ; key to read clock
 8              5     MT.FRJOB EQU   05H          ; key to force remove Job
 9              10    SD.POS  EQU    10H          ; key to set absolute cursor position
10              D0    UT.MTEXT EQU   ODOH         ; vector to print a message
11              C8    UT.SCR  EQU    OC8H         ; vector to open a screen window
12              CA    UT.ERRO EQU    OCAH         ; vector to print error message to #0
13              EC    CN.DATE EQU    OECH         ; vector to convert long integer to date
14                    ;
15                    HEADER
16 0003F000 6000000E        BRA     START
17 0003F004 4AFB            WORD    4AFBH        ; pad out extra space
18 0003F006 4AFB            WORD    4AFBH        ; flag to show Job is in standard format
19 0003F008 0005            WORD    5            ; length of Job name
20 0003F00A 436C6F63        ASCII   'Clock'      ; name of Job
         6B
21          3F010 R         ORG     /2
22                    START
23 0003F010 700B            MOVEQ   #MT.PRIOR,D0 ; set priority...
24 0003F012 72FF            MOVEQ   #-1,D1       ; ...of this Job...
25 0003F014 7401            MOVEQ   #1,D2        ; ...to 1 (very low !)
26 0003F016 4E41            TRAP    #1           ; set it
27 0003F018 43FA004A        LEA     CHAN(PC),A1  ; point to screen parameters
28 0003F01C 367800C8        MOVE.W  UT.SCR,A3    ; get the address of the routine
29 0003F020 4E93            JSR     (A3)         ; set up screen channel
30 0003F022 4A80            TST.L   D0           ; was there an error ?
31 0003F024 6630            BNE.S   ERR_KJ       ; yes, give up
32 0003F026 2F08            MOVE.L  A0,-(A7)     ; save channel id
33                    LOOP
34 0003F028 7013            MOVEQ   #MT.RCLCK,D0 ; read system clock
35 0003F02A 4E41            TRAP    #1           ; time to D1
36 0003F02C 43FA0060        LEA     BUFFER(PC),A1 ; point to space where string can go
37 0003F030 93CE            SUB.L   A6,A1        ; make it relative to A6
38 0003F032 367800EC        MOVE.W  CN.DATE,A3   ; date conversion routine
39 0003F036 4E93            JSR     (A3)         ; convert to date
40 0003F038 2057            MOVE.L  (A7),A0      ; get channel id back
41 0003F03A D3CE            ADD.L   A6,A1        ; point A1 to absolute address of string
42 0003F03C 367800D0        MOVE.W  UT.MTEXT,A3  ; message print
43 0003F040 4E93            JSR     (A3)         ; print date
44 0003F042 4A80            TST.L   D0           ; was there an error ?
45 0003F044 6610            BNE.S   ERR_KJ       ; yes, give up
46 0003F046 4241            CLR.W   D1           ; column 0
47 0003F048 4242            CLR.W   D2           ; row 0
48 0003F04A 76FF            MOVEQ   #-1,D3       ; infinite timeout
49 0003F04C 7010            MOVEQ   #SD.POS,D0   ; set cursor to absolute position
50 0003F04E 4E43            TRAP    #3           ; do it
51 0003F050 4A80            TST.L   D0           ; was there an error ?
52 0003F052 6602            BNE.S   ERR_KJ       ; yes, give up
53 0003F054 60D2            BRA.S   LOOP         ; again
54                    ERR_KJ
55 0003F056 367800CA        MOVE.W  UT.ERRO,A3   ; print error message to channel 0
56 0003F05A 4E93            JSR     (A3)         ; do it
57 0003F05C 2600            MOVE.L  D0,D3        ; return error code, and...
58 0003F05E 7005            MOVEQ   #MT.FRJOB,D0 ; ...force remove...
59 0003F060 72FF            MOVEQ   #-1,D1       ; ...this Job
60 0003F062 4E41            TRAP    #1
61                    CHAN
62 0003F064 00              BYTE    0            ; border colour
63 0003F065 00              BYTE    0            ; border width
64 0003F066 00              BYTE    0            ; paper
65 0003F067 07              BYTE    7            ; ink
66 0003F068 00F0            WORD    40*6         ; width
67 0003F06A 000A            WORD    10           ; height
68 0003F06C 0110            WORD    512-40*6     ; x-origin
69 0003F06E 00C8            WORD    200          ; y-origin
70                    BUFFST
71 0003F070     1E     BLOCK   30
72                    BUFFER
73 0003F08E     2      BLOCK   2
74                          END
```

Machine Code Listing

QL Quarks

A sense of humour is important. Even managing directors of large computer companies have to have them.

Witness Nigel Searle, Managing Director of Sinclair Research, displaying his shining wit at the recent Personal Computer World Show at Olympia 2 in London. The occasion was a panel discussion for Radio 1's Chip Shop program, including Roger Foster – ACT's head man, Hermann Hauser – part of the Acorn managing duo and our good Mr Searle.

THE discussion went on uneventfully for some time discussing the highbrow issues involved in placing computers in society, until the presenter invited questions from the floor. One floored individual – namely Sid Smith from Micronet – decided to throw the cat among the pigeons by asking Roger Foster and Nigel Searle what they thought of Mr Hauser's new range of ABC business machines.

Messrs Searle and Foster replied politely with the intrepid head of Sinclair Research even wishing luck to his rival at Acorn. (Don't worry folks, the punch line is coming soon.) When the man on the floor (the aforementioned Sid) suggested that

Mr Searle's comment amounted to false amiability and that he should say what he really thought our Nigel said (yes, here comes the punch line);

"I said I wished him luck, but I did't say what kind of luck!"

We can all speculate...

A Hertfordshire company has started offering training courses for the QL (see story below) and it seems the GPO must have had a premonition this was going to happen. The last two letters in the company's postal code are QL!

QL Quirk challenger of the month: Come up with the funniest alternative answer to what QL stands for and we'll give you two free Microdrive cartridges. Address it to The Editor, QL Quirks at the address given in the letters column.

SPECIAL-OFFER OFFICE

Welcome to the Special-offer office. In each issue of QLUB News this will be THE place to look for news about cut-price software, hardware and support for your QL.

As you'll no doubt be aware from the front-page story about it, Sinclair Research is from now on offering a 20 per cent discount to QLUB members on ALL software sold through QLUB. And since there's lots of software in the works, this could turn out to be a substantial saving.

But we aren't the only company offering discounts to QLUB members. Just by joining QLUB, you've already become popular with EMAP Business and Computer

If you want to subscribe, send your cheque or money order for £4.20 to:

PRQL Subscription Department
QL User
Priory Court
30/32 Farringdon Lane
London EC1

THE enterprising people at EMAP are also starting their own electronic magazine all about the QL, called (not surprisingly Q-NET). The word is that they will be offering a discount on subscriptions to this system for QLUB members – but you'll have to wait until the next issue to find out

Owl Computer Training of Tring are launching a regular series of QL training courses for people in business. The courses consists of an introduction to the machine, its operating system, (QDOS) and the four application packages – Quill, Abacus, Archive and Easel.

The first courses began in early August and run about every five weeks. Owl Computer Training is an accredited company under the Manpower Services Commission's Small Business Scheme for Microcomputer Training and grants for QL training are available for certain course combinations.

Elayne Coates
Principal
Owl Computer Training
15 Clarkes Spring
Tring
Herts HP23 5QL

Next Issue
– A first look at the QNET electronic information service for the QL.
– Details on new software for Christmas.
– The QL Christmas wish list – a look at the Xmas presents for your QL.
– A terminal listing to allow the QL to use modems and access bulletin boards.
– And your letters, comments, ideas and much more...

Publications Ltd, with which we have negotiated a discount of 15p per issue off the normal 85p cover price of QL User magazine. You would normally have to pay that 85p for the magazine alone, never mind the postage involved in a subscription – but if you subscribe through QLUB the magazine will be delivered for an all-in (postage-included) price of 70p a copy or £4.20 for the next six issues.

just how much that discount is.

The final item on this increasingly-cluttered discount desk is really more of a timesaver than a money-saver. If you're using the QL for business, but can't spend the time to teach yourself how to use the machine and all its attendant software on your own, you might want to consider a series of courses being offered for the machine.