

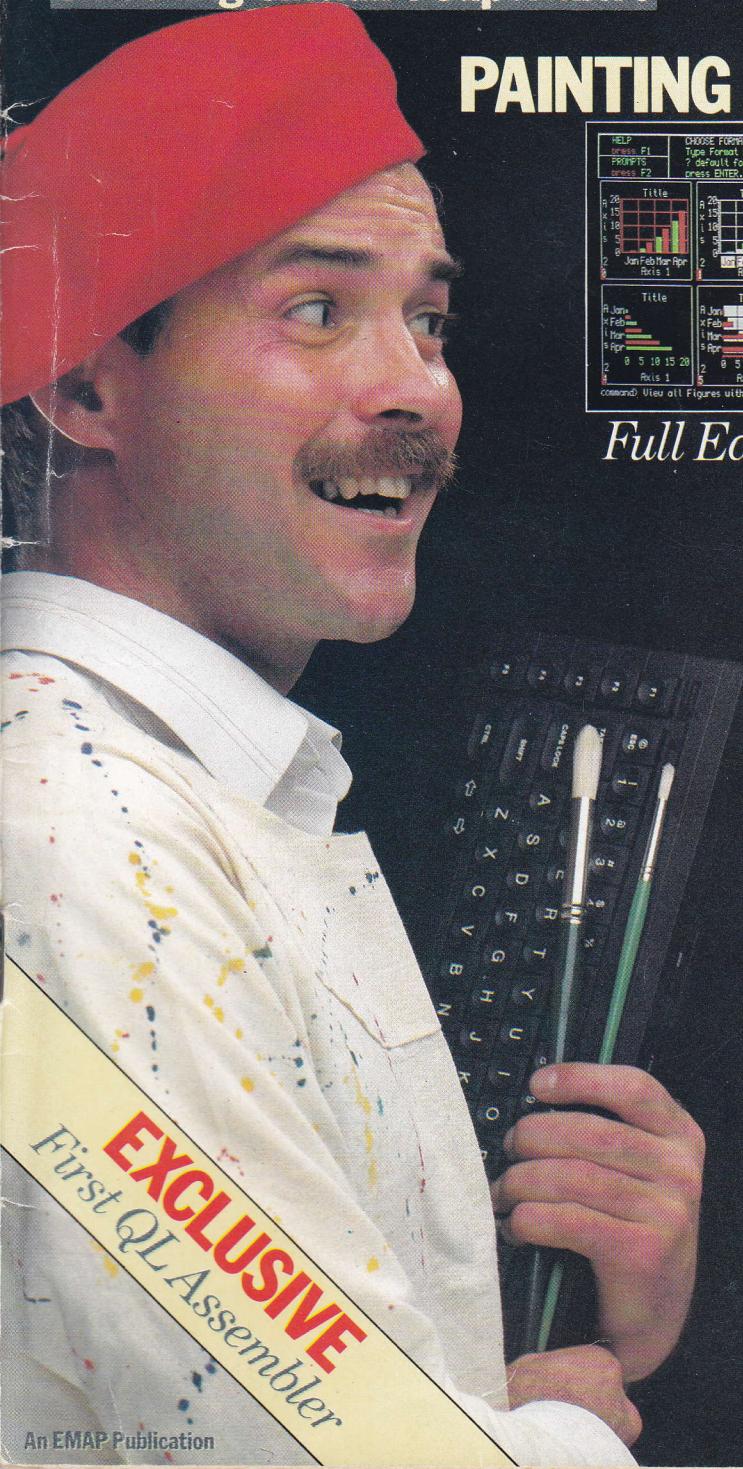
OCTOBER/NOVEMBER

85p

THE ONLY MAGAZINE FOR THE

QL USER

Unlocking Sinclair's Supermicro



MONITOR MONITOR

First 3 Reviewed



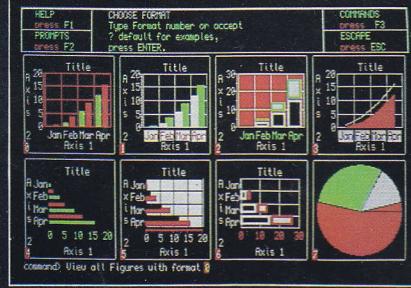
K/OS OR CHAOS

New Operating System

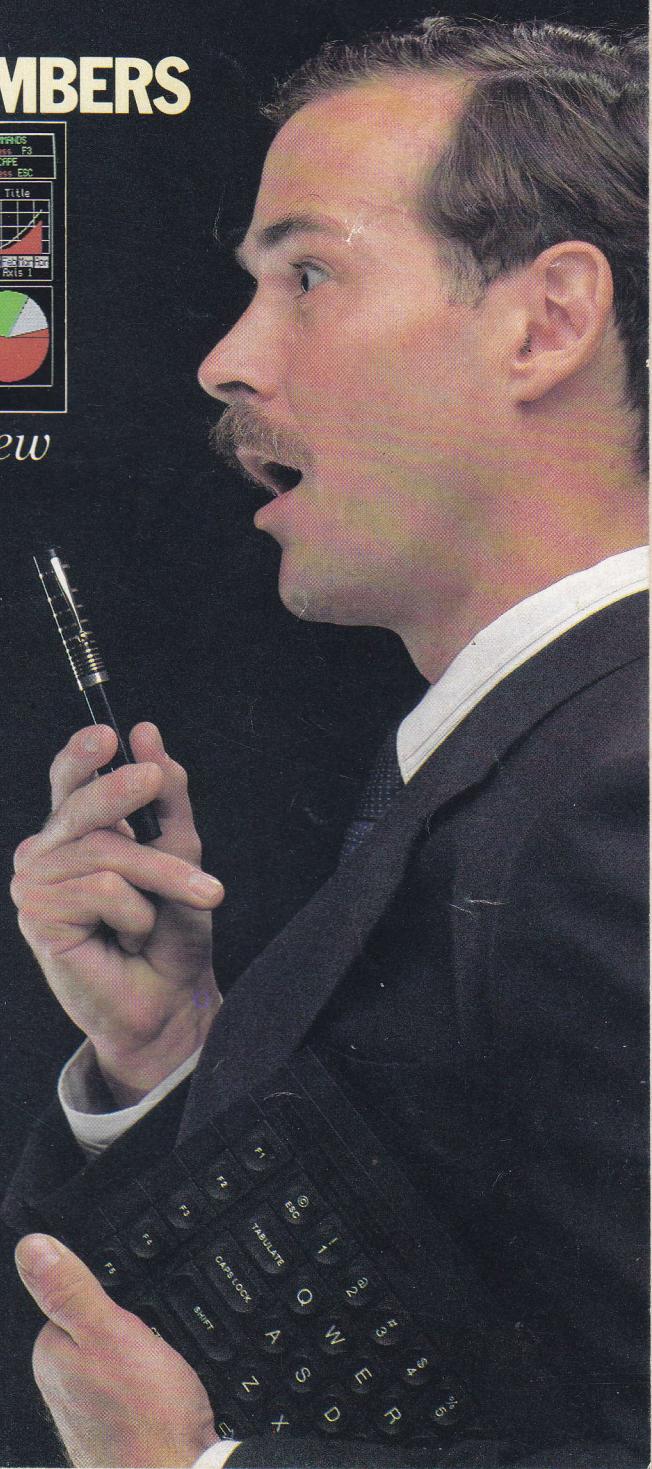
BUGS OUT

A Final QL Update

PAINTING BY NUMBERS



Full Easel Review

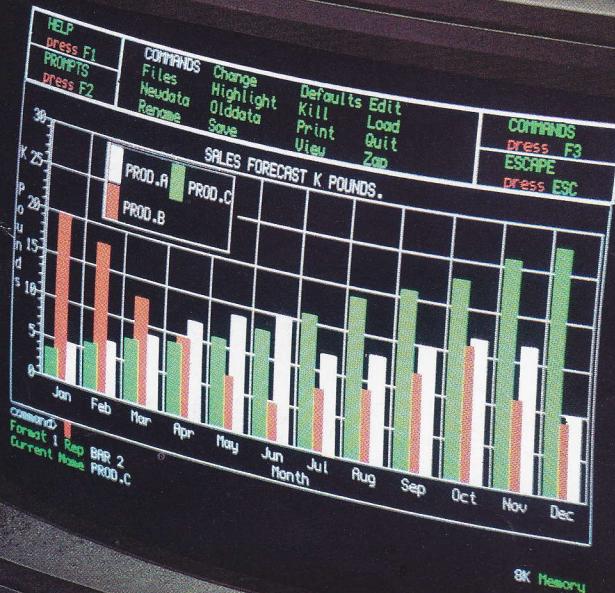


EXCLUSIVE
First QL Assembler

EVEN THE PRICE WILL KEEP YOU IN THE BLACK

This sleek, black Microvitec CUB medium resolution colour monitor has been designed to be totally compatible with the Sinclair Q.L. An ability to display 85 column text is combined with outstanding graphic capabilities – at a price that won't break the bank.

£275
(inc. VAT)



Specification

Model:
CUB 1451/DQ3
14" Q.L. Monitor
RGB TTL input
Tube Resolution (pixels)
653(H) x 585(V)
Dot Pitch 0.43mm
Bandwidth 18 MHz

Also available in Standard Resolution
version for only £225 inc. VAT.



MICROVITEC
CUB
COLOUR DISPLAYS

Microvitec PLC, Futures Way, Bolling Road, Bradford BD4 7TU, West Yorkshire. Tel: (0274) 390011. Telex: 517717



CONTENTS

4 News There's no shortage this month as new products come forward and ex-employees look back.

10 Letters Our readers air their views.

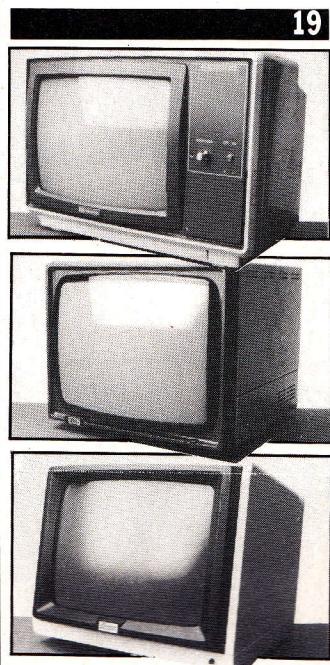
13 Next month

14 Books

Books Matching title to content Nicky Trevitt determines whether quick off the mark means just in for a fast buck.

17 User club

User club Founder member, Leon Heller provides an independent view from an independent club.



Monitor survey With TV's incapable of the resolutions necessary for an 80 column display, Gary Evans looks at the alternatives and surveys four monitors currently available.



Spectrum interface A useful project to link up a Spectrum to act as a printer buffer and driver also sheds light on the problems of networking on the QL.

19



Easel The last of Psion's four packages included with the QL comes under close scrutiny.

32

QL review With the bugs out of QDOS our reviewer stands back and assesses the QL's likely impact on the marketplace.

36

Inside story We take the lid off the QL and examine its innards.

38

68KOS An alternative operating system to QDOS, Adam Denning investigates its potential.

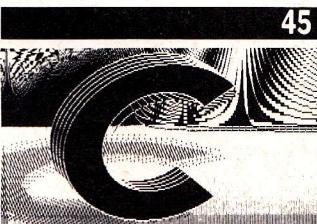
40

Quest Will the QL breathe life on CP/M or is it the other way round? In search of an answer we visit Quest.

43

Assemblers Both GST and Metacomco have released assemblers for the QL, our reviewer weighs one against the other.

45



C series Tentative first steps in C. We look at operators and control structures to construct our first compiled program.

52

SuperBasic course

Continuing his tutorial Adam Denning looks into the intricacies of defining procedures and develops some valuable conversion and backup functions.

54

Machine code The Motorola 68008 with its comprehensive instruction set and general purpose registers whilst extremely powerful is relatively easy to program. Our resident expert demonstrates how.

Competitions

Aug/Sept Issue: The winner of the printer competition is R K Richardson from Hampshire.

This Issue: The prizes will be two monitors for the QL. One donated by Microvitec and the other by Opus Supplies.

Two to one

Confusion is at an end. This magazine is now, categorically, the ONE AND ONLY QL User being published in the UK.

Editor

Paul Coster

Consultant Editor

Peter Rodwell

Art Director

Paul Carpenter

Sub Editor

Caroline Fyshe

Contributors to this issue

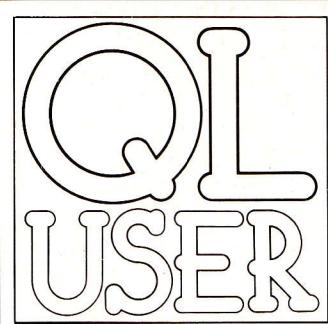
Paolo Baccanello, Adam Denning, Gary Evans, Leon Heller, Mark Jones, Sid Smith, Nicky Trevitt

Advertising

Phil Baker

Publisher

Alfred Rolington



Scriptor Court

155 Farringdon Road

London EC1R 3AD

Telex: 32157 EMAPPB.G

QL User is published and

distributed by EMAP

Business and Computer

Publications Ltd

Typesetting by Crawley

Composition Ltd

Printed by Riverside Press

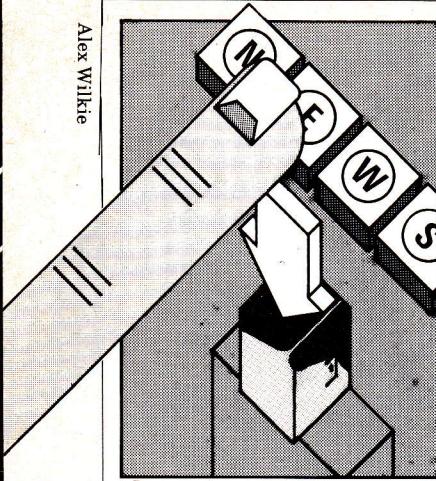
© Copyright QL User 1984

All editorial enquiries to:

01-278 1082

Advertising:

01-833 0531



A resigned attitude...

One of the three people who created the QL resigned in disgust the day it was launched.

The big three were Jan Jones (who wrote the Basic compiler), David Karlin (responsible for the hardware) and Tony Tebby – author of QDOS. It was he who, appalled by Sinclair's promise that the QL would be shipped within six weeks handed in his notice immediately after the machine's January unveiling.

In the event Tony hung on a little longer, eventually fixing a date to leave after Sinclair's management decided to ship those first 89 machines without, he says, telling the software division.

He still takes a lively interest in the QL – so lively that many of his opinions are unprintable! It's worth recording however his belief that the delay was exclusively due to problems with the machinery itself, the apparent difficulties with SuperBasic and QDOS being caused by the firmware writers having to conform with frequent hardware updates. Status conscious QL users, in fact, might care to enliven their one-upmanship over operating systems with a little snobbery regarding the hardware issue of their machines: check the underside of your QL for the two digits following the green D; 07 is good, anything higher than 09 is excellent. Particularly interesting meanwhile are Tony's opinions on the shape the QL's promised new operating system – expected next year – ought to take.

I'm fairly confident that the new version will include improvements to the Basic editor, but the cost in memory of any full screen editor would be disproportionate, especially when you consider what else you could have for the same

NEWS NEWS

A run-down of new products, people and events in the QL market
by Sid Smith of Micronet.

amount of code,' he says.

Preferable alternative goodies, feels Tony, would include direct access file handling in Basic, the ability to read or write data to anywhere in a microdrive file; formatted print using parameters on a print command to describe in detail the format of output; a full set of job control facilities for examining what multi-tasking programs are running, and for killing them, suspending them, changing their priorities, etc; and a spooler, for outputting files to a printer while the user gets on with other tasks.

'All that lot would take about half as much ROM space as the 2K you'd need for a full screen editor,' says Tony, though he's dubious about the likelihood of Sinclair coming to sensible decisions about what to offer in the new ROMs.

'They'll probably be more impressed by marketing appeal,' he remarks, 'than by what the facilities actually do, which is unfortunate because marketing appeal only sells the first few machines, it doesn't sell the next million.'

So what'll happen to people with the old operating system, I asked innocently. Will they get a free upgrade?

'There is no old operating system and there never has been. The operating system which is out now is the same as the operating system which was out in January. Anything else is just a story invented by journalists looking for something to write about.'

Gulp.

'But in any case, QDOS contains linkages to facilitate expansion. All the things I've mentioned can be supplied externally – on microdrive or ROM cartridge, for example – to be booted up on power on, and will look just as if they form part of the machine. They could be built into the new ROM, but they certainly don't need to be.'



All smiles from Psion and Potter.

Slowness in the eye of the user

Quill is slow. I know because Psion Managing Director David Potter told me. He even said why.

But first the good news: In September we'll be releasing a Quill Version 2 which will overcome all the problems of Version 1 thanks to an intensive effort to specially code-down, and using special techniques to compress the software. We've also put special additional features in the screen driver, writing our own outside of QDOS to speed writing to the screen.

Phew! But why are all these hairy techniques necessary?

David explained: 'Quill is actually designed as a very powerful word processor to run very large documents. It was designed for substantial machines. You could criticise it in the sense that you're trying to pack a hell of a lot into a modest machine – the QL.'

'The effects of this are, firstly that the amount of data memory left – because the program is very substantial – is modest.'

'Secondly, we've had to resort to overlaying, which was one of the points made in *QL User*, – when you press for Command the microdrive whirrs to get another overlay in. When the code compression is finished, all the commands will be resident in RAM at all times, and additional memory will be available to the buffer space so that Microdrive 2 will be less used.'

'The third factor is that there is no video chip in the QL, and therefore everything on the screen has to be done by the processor. That is a hardware design feature of the QL which frankly is pretty tricky.'

'And the final factor is the microdrives.'

Now none of these observations about Quill will cause particular surprise to readers of this magazine, who will remember similar remarks in the somewhat critical review in our last issue. What is surprising is that they were made when Mr Potter phoned to complain about said review.

Wafer thin logic?

Don't know if I'm pleased or not. Clive Sinclair has announced that the .5 Megabyte RAM pack, destined for that big square of empty space at the left-hand end of our beloved QLs, will be a wafer scale integrated circuit. And this giant chip, available in late 1985, will be the first product to emerge from Sinclair's very own silicon factory where Sir Clive intends to invest 'tens of millions over several years,' and which he sees as a 'major research effort.'

WSI technology – the notion of producing an integrated circuit which covers the entire several inch diameter of a silicon wafer, not just the small fraction which is broken off to form a conventional chip – has been much discussed recently. Sir Clive's announcement coincides with the news that US hi-tech company Trilogy, after



Adding a RAM pack.

sustaining heavy losses, has been forced to pull out of its own well-publicised efforts in the area.

However, Sir Clive is characteristically confident that the line of research adopted by his company won't turn out to be a similar expensive mistake. 'We have

immense knowledge of integrated circuit design,' he affirms, 'and the technology we'll be using is streets ahead of Trilogy's.'

The technology in question is based on patents lodged some 12 years ago by lone British inventor Ivor Catt. At its heart is a new method of overcoming the central problem confronting any would-be manufacturer of wafer scale circuits – that of component failure.

All current methods of integrated circuit production generate a high proportion of duff chips. Since every silicon wafer contains dozens of chips, and since the duff ones can be broken off and tossed away with no effect on the remainder, manufacturers can afford to see their full reject bins merely as an expensive inconvenience.

Picture the scene though if that errant speck of dust has corrupted not merely one

chip out of dozens, but has guaranteed that the entire wafer is a throw-out...

Trilogy's answer was to build up to four back-ups for every gate on his wafer, a technique which, Clive contemptuously points out, demands around nine times more silicon than would otherwise be necessary.

In contrast to this hardware-orientated approach, the Sinclair/Catt method is to use the intelligence of individual chips on the wafer to build up a chain of working components across its surface.

An off-wafer tester first locates a functioning chip. This component is then instructed to similarly interrogate an adjacent chip. Then the second chip cross-questions a third, and the third gives third-degree to a fourth, and a fifth is forthcoming to the fourth, until full wafer scale integration has been achieved.

'But the real beauty of Catt's approach,' claims Sir Clive, 'is that the wafers can repair themselves once they are installed in the computer – simply by turning the system off and going through the whole linking-up process again. This is fundamental when you consider the problems of repairing an advanced computer system which could have millions of components.'

Well, yes. Though the first, selfish reactions of this particular QL user were, 'Is my .5Mb RAM really dependant on such iffy technology?' and, 'Do I really have to wait till (at least) late '85 for it?'

High street mirage?

By the time you read this the QL will be available in High Street shops. At least, that's the intention though cautions from Sinclair that 'demand will inevitably outstrip supply' might suggest it's a while yet before we'll be in and out of our local retailer.

In fact at the reception held to make this announcement, dealers complained bitterly to Sir Clive that even the long established Spectrum was being delivered at the rate of only one per store per week. (And that of these, a good 20 per cent are faulty.)

However, sitting alongside the QL in Sinclair's dream retailer will be the long-awaited flat TV (complete with 25 per cent price leap to £99), and some rather nice software bundles with the Spectrum and ZX microdrives – all selling madly thanks to a £4m ad. campaign. Remarkably, the odd QL peripheral should be there too. Miracle Systems have already sewn up a deal to get their parallel printer interface stocked in major branches of WH Smith, speaking gleefully of the effect this exposure will have on their many rivals.

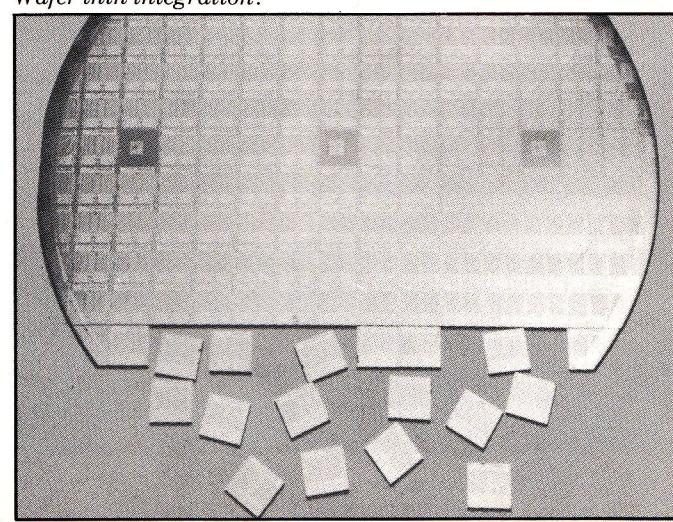
QL User fearlessly

predicts a major shake-up in the QL peripherals business. Four major companies making parallel printer interfaces is just too many. The first symptoms of this industry-wide blood-letting are already apparent, with Miracle whispering in confidence of a 20 per cent price cut to £39.95 for their device, shortly after I hear of a similar axing for the CST Q-Pi – now down to £50 (plus VAT).

Equally significant is the amount of mutual slagging-off to journalists already taking place. Company one tells me that company two uses the QL's serial ports for its device, a method guaranteed to mislay characters on the input side. Company two then says that company one employs the QL's expansion port, thus requiring the Sinclair motherboard before said port is freed for other duties. Then another other company says that another other company doesn't cater for Epson printers, and the other other company tells me in turn that all their rivals have fallen foul of the same bug in the QL's ULA chip.

Me, I say nothing... except, wait for the frank and fearless review of all these devices in forthcoming issues.

Wafer thin integration?



ADD-ONS FOR YOUR QL LOOK NO FURTHER THAN TECHNOMATIC

We've built our reputation on providing a reliable service

* Wide Range

* Competitive Prices

* Deliveries from stock

* 12 months guarantee on all printers

* Accessories and consumables readily available

DOT MATRIX PRINTERS

EPSON

This ever popular range of dot matrix printers set the industry standard and provide reliability and quality 'second to none'. The budget RX80T & RX80FT have 100cps and all standard printing and graphic functions with tractor feed on RX80T and both tractor and friction feed on the RX80FT. The deluxe FX80 has all the above, as well as a 160cps, buffer, programmable characters etc. For wider paper use — upto 15" — the RX100/FX100 are ideal.

RX80T £210(a)

RX80FT £230(a)

FX80 £315(a)

RX100 £345(a)

FX100 £450(a)

EPSON Serial Interface c/w 2K buffer £60(c) (£52.50 if bought with the printer)

Serial Lead QL/EPSON 2m long £12(d)

Spare Ribbon Cartridges: RX/FX 80 £5.00(d)



RX/FX 100 £11.50(d)

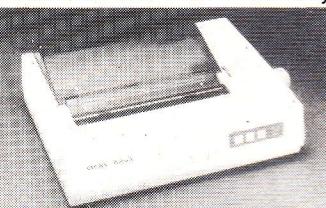
Dust Covers: RX80 £4(d) FX80 £4.50(d)

Paper Roll Holder £17(b)

Tractor Attachment for FX80 £32(b)

KAGA

This new range of printers with EPSON compatible control codes feature NEAR LETTER QUALITY print using a 23 x 18 matrix in addition to the features one would normally expect from good quality dot matrix printers. Text modes include Normal, Italic, Enlarged, condensed Super & Sub script and Proportional spacing. Defined characters can be placed in a rom to give personalised print.



An integral 3K buffer, both friction and adjustable tractor feeds with built in paper roll holder, etc, etc make these superb 'value for money' printers unique.

KP810 (80col) £249(a) KP910 (156 col) £369(a)

Kaga Serial Interface c/w 2K buffer £85(c) (£77.50 if bought with the printer)

Serial Lead QL/Kaga 2m long £12(d)

Spare Ribbon Cartridge £6.75(d)

Enquire about the special font roms.

DAISY WHEEL PRINTERS

JUKI 6100 £349(a)

A reliable printer that provides quality printing and reliability at an economical price. The linear motor used for the wheel carriage ensures little wear and tear. 20cps, 2K buffer (expandable), 10, 12 & 15cpi, proportional spacing, block graphics will meet most requirements of a discerning user.

Serial Interface QL/Juki £65(c) Serial Lead 2m long £12(d)

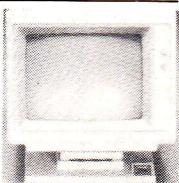
Tractor Feed Attachment £99(a) Single Sheet Feeder £199(a)

Plain Fanfold Paper with extra fine perforation (Clean Edge 2000 9.5" x 11" £13(b); 2000 14.5" x 11" £18.50(b)

SOON TO BE IN STOCK
PARALLEL INTERFACE FOR THE QL £42(b)

MONITORS

The British made MICROVITEC 14" RGB Monitor with high clarity 85 character display offers a consistent and reliable performance. The KAGA and SANYO 12" green monitors offer an ideal solution for word processing applications. The non reflecting screen on the KAGA allows it to be positioned independent of the light source position. All monitors are supplied with a suitable lead at no extra charge.



MICROVITEC 1451DQ3 £239(a)

Slanting Plinth for Microvitec £8.50(b)

KAGA K1201X Green Screen £106(a)

SANYO DM8112CX £99(a): KAGA Amber Screen £115(a)

Swivel Stand with digital clock for KAGA Monochrome £22.50(b)

TECHNOMATIC LTD

MAIL ORDERS TO: 17 Burnley Road, London NW10 4ED
(Tel: 01-208 1177 Telex 922800)

SHOPS AT: NW London: 15 Burnley Road, London NW10 4ED
 • (Dollis Hill 2 mins walk, ample car parking space)
West End: 305 Edgware Road, London W2. Tel: 01-723 0233
 (Near Edgware Road) •

01-208 1177

All prices exclude VAT. Please add 50p carriage unless indicated as follows:

(a) £8: (b) £2.50: (c) £1.50: (d) £1.00.

add 15% VAT to the total order value. For fast delivery telephone your order quoting VISA or Access card or official order number.
(Minimum telephone order £5).

Who's the odd one out?

Sinclair Research has mapped out its technological future to a remarkable degree.

The company has revealed its commitment not only to the 68000 family of chips, but also to the microdrive, the flat screen TV tube and to wafer scale integration.

I'm certainly in no position to argue with all those gentlemen in thick spectacles (many of them at Apple Corp) who assure us that Motorola has produced the finest chips available, but of the other three technologies, all connected with the QL, there's certainly more to say – notably that in each case Sinclair is following a line of development ignored or dismissed by the rest of the industry.

The most vivid example of Sinclair idiosyncrasy is the espousal of Ivor Catt's ideas on wafer scale integration after they'd been tried and rejected over a 12 year period by ICL, STC, Burroughs and Plessey. An official at the government's Alvey Directorate tells me that he will not support the company's application for sponsorship of its WSI project, and mis-directed efforts in WSI recently cost Trilogy a sum estimated at \$200m to \$300m. Could Sinclair Research survive a

similar mistake?

More modest but no less controversial are the company's efforts with the cathode ray tube. Sinclair's flat – or rather sideways – TV was launched a year ago but is only now reaching the shops. In the meantime, one Japanese company has unveiled a full-size 80 column by 25 line liquid crystal display and another has produced a colour liquid crystal pocket TV. In short, the cathode ray tube is under threat of imminent obsolescence.

Considering that the device is scheduled for incorporation in future Sinclair computers (and, until the Dundee production figures proved unacceptably low, was due to form part of the QL), it is no surprise that Sir Clive defended his sideways tube from my charge that its future is now in doubt.

The flat tube uses less power, gives a better display and is cheaper to manufacture than any liquid crystal device. Sinclair Research knows as much about them as anyone; they're simply not the way to go, I hope so.

Now the microdrive. In case you think these devices are merely an intermediate measure until the price of conventional drives falls low

enough, the company revealed plans at the Carlton Towers lunch for microdrives up to one megabyte capacity.

The meeting became at times a eulogy on the benefits of this entire technology. Nigel Searle discoursed on its warm reception amongst software houses (which isn't what they say to me), thanks to the anti-piracy option of embargoed cartridge sales to disreputable companies. David Karlin dilated on the encryption techniques available for software protection, and Clive himself hinted that a second company would soon announce its adoption of the device. (I can reveal that an unfinished twin microdrive device has been spotted at Psion HQ; 'an announcement will be made before the year end,' says the company.)

In fact, of course, the one benefit of microdrives is that they're cheap; they've been the weak point of every QL I've laid hands on, and the machine I'm using to write this will soon go back to Camberley, thanks to a mdv2 which doesn't always recognise a cartridge when it sees one.

Since Sinclair is the one company working to develop this technology (whereas powerful market forces are operating world-wide to improve conventional drives), the microdrive seems likely to remain a low price, low performance alternative to floppy disks, perfect for the home and hacker's machine which the QL has turned out to be, but alienating the users Sinclair most desired – business people and Americans.

The sooner disks are available for the QL the better, and we must hope that future Sinclair machines won't try to push the microdrive into markets which seem unlikely ever to accept it.

Sir Clive has said: 'Sinclair Research won't survive long if we do "me too" products; we have to innovate right across the board.' The real driving-force behind all three of these idiosyncratic technologies however is doubtless his own delight in doing something new. In an industry tolerant only of winners and the people who imitate them, his individuality is terrifying.

Wising up to the business user

Who says the QL isn't a business machine? Not Sagesoft, or Owl Consultants both of whom are launching products for the more formally dressed QL user.

Sagesoft's offering is a microdrive version of the dedicated accounting package they've already configured for 150 CP/M and MS-DOS machines.

Gritty details such as price and availability were not forthcoming, however, because the Sage package will be marketed by Sinclair who aren't ready to say anything. As for a pre-launch peek, 'We're reluctant to show it too early,' said Sagesoft MD David Goldman, 'especially after the unfortunate experiences of Sinclair itself.'

Less shy is Owl Consultants, which runs a QL Training Scheme at the delightful address of Clarkes Spring, Tring.

The company even quote the comments of Owl old-boys about the course – though we've been so corrupted by the marketing hype of other firms that honest assessments like 'really interesting and helpful' and 'I would recommend the course to a friend' sound full of touching innocence.

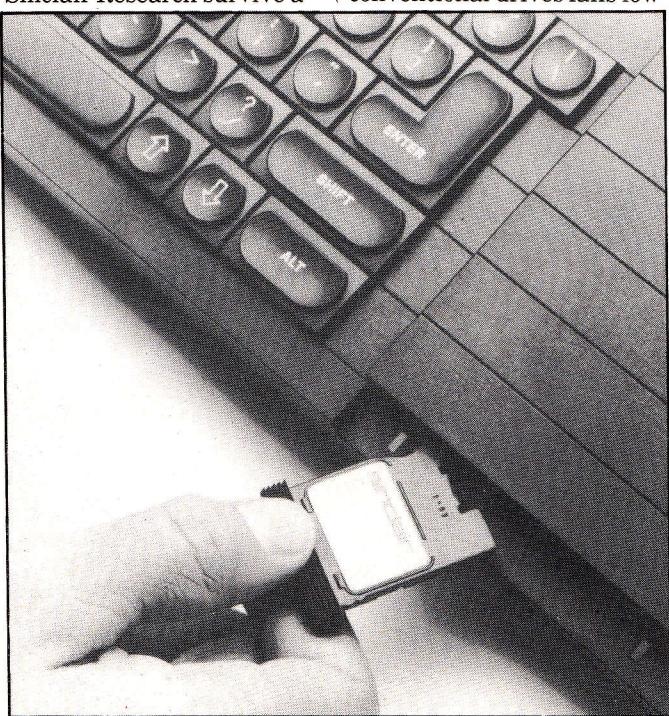
Sagesoft are contactable on (091) 284 7077, with Owl at (044282) 7302.

The silver lining . . .

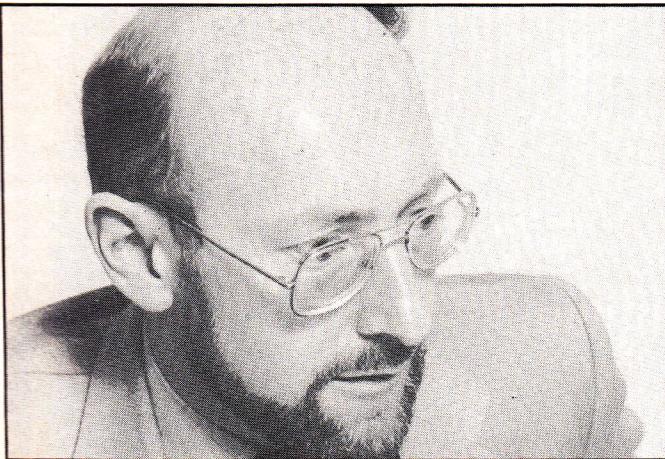
Only 89 QLs were dispatched in that first shipment at the end of April, no more appearing until the half-way decent machines were sent out a month later.

So one vanload of half-finished QLs was responsible for the endless series of appalling reviews which appeared in monthly magazines – including this one – throughout the summer . . . (You'll be aware of the production schedules which mean that magazines don't appear until weeks after they've been written).

But look on the bright side: the shipment did enable Clive to say that the machine had only been two months late.



One of the microdrives major benefits is price.



Sir Clive said it

Quote of the month comes from Sir Clive himself. When I mentioned that the QL firmware didn't seem to live up to the promise of the 68008 chip, he actually agreed!

'I think that's true. I think that's because it's been done in some haste; it had to be. I'm sure better operating systems will come out later,' said he.

I didn't want to spoil such a splendidly scandalous admission by asking for an expansion, but he gave it anyway.

'We'll go on improving the firmware indefinitely. The best firmware for the Spectrum is the most recent, and the same will happen to an even greater extent on the QL because there's so

much more learning to be done.'

General surprise was caused amongst QL-watchers when Sinclair produced a successor to the AH operating system only a week or two after Nigel Searle had described it as the one he intended to stick to.

Are owners of outmoded operating systems likely to be offered free upgrades to later versions? 'I don't think so,' said Sir Clive, 'because I don't think there's any need.'

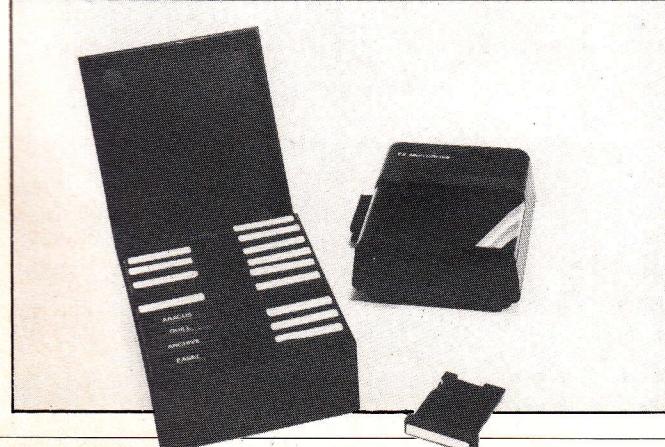
Sinclair Research has since repeated and strengthened that line. Changes to the AH version are officially described as 'insignificant' so you can expect no free upgrades until perhaps, the major re-issue of the operating system in several months time.

Safe and sound

This rather murky photograph depicts the Transform Microdrive Case. For a mere £5.95 the case will warmly embrace 20 of your cartridges, with space in the central compartment for all those sheets of sticky-backed labels.

Ignore the ZX microdrive – the case is 100 per cent QL compatible. Mine's already in use, and counts as one of the most charming freebies I've ever had. I do hope they don't ask for it back.

Orders to Transform at 41 Keats House, Porchester Mead, Beckenham BR3 1TD. (Tel 01-658 6350).



Beeb disappointing

'The BBC have a monopoly position, and they're going to exploit it to their profit.' Thus spake Clive to your reporter when asked about the corporation's decision to continue their exclusive patronage of the Acorn BBC Micro.

The chap seemed genuinely disappointed. The notion that the QL might receive the Beeb's imprimatur was much mentioned by Sinclair Research both at and since the machine's launch. So is he still talking to the

Corporation? 'Oh, the BBC move in mysterious ways. I don't know what they're playing at.'

Is the idea of two-tier sponsorship, part for the Acorn machine and part for the QL, still a possibility? 'They've written back to us and said that they're not going to do a second machine.'

So are negotiations finished? Are you expecting to see them again in, say, the next six months?

'No, no,' said Sir Clive dolefully. I kindly changed the subject.

Blessing from on-high

The 68K/OS replacement operating system has been officially blessed by Motorola, US manufacturer of its host 68000 chip. (See our review in this issue.)

GST of Cambridge, originator of 68K/OS, becomes the first European software house to benefit from Motorola's independent software support scheme, designed to promote software for the 68000 series.

According to Motorola's Tim Coombs, 'UNIX is well

established as the natural multi-user operating system, and we see the need for a software product to fulfil the same requirement in the single-user arena.

'68K/OS can fit this requirement because it's a compact operating system aimed specifically at the M68000, and has been designed from the outset to handle a bit-mapped display – a feature found in more and more systems.'

Welcome to QNET

A database specifically for QL owners has been opened on Prestel. Provisionally named QNET, it has been set up following news that Cumbria-based OEL, commissioned by Sinclair to design and build the QL modem, has already decided to produce many thousands of the devices (see our last issue).

The new database is an

offshoot of Micronet 800, the down-the-phone computer magazine which has been operating on Prestel since early '83. (And for which the present writer has been known to scribble the odd line.)

Along with its other services, QNET is likely to carry telesoftware versions of the listings published in *QL User*.

Sid Smith is news editor of Micronet 800.



Probably the most user friendly model on the market sir.

**Nothing will affect you
(and your school or college)
more than...**



We'll send you
a sample copy of
EDUCATIONAL COMPUTING
for just £1.00 — write to:

Valerie Day
EDUCATIONAL COMPUTING
Durrant House
8 Herbal Hill
London EC1 5EJ

**EDUCATIONAL
COMPUTING**

(Cheques made payable to EDUCATIONAL COMPUTING)

BEST PRICES – BETTER SERVICE
PRINTERLAND
ORDERED TODAY – DELIVERED TOMORROW

FREE WITH ALL PRINTERS SOLD
QL USERS PRINTER GUIDE

Describes how to get the most from your printer/QL. Includes tips like printing 145 columns in ABACUS and 100 different typefaces in QUILL.

DOT-MATRIX

EPSON RX 80	£189.95
EPSON RX 80 F/T	£219.00
EPSON RX100	£329.00
EPSON FX80	£317.00
EPSON FX100	£425.00

DOT-MATRIX PLUS NEAR-LETTER QUALITY

CANON PW1080A	£249.00
---------------	---------

DAISY-WHEEL

JUKI 6100	£325.00
EPSON DX100	£356.00

QL/RS232 PRINTER INTERFACES FROM £27.00

WE ALSO STOCK QL's, MONITORS,
MICRODRIVES, QL BOOKS, QL PROGRAMS
AND QL ACCESSORIES AS AVAILABLE.

SHOWROOM OPEN 9.15 TO 5. SAT 9 TO 12.30

Educational, G'ment plus O'seas Orders Welcome

Please add 15% VAT. Delivery Printers £10.00. Paper £3.00. Other 50p
Printerland, Unit 27, Estate Buildings, Railway St. Huddersfield HD1 1JP

TEL HUDDERSFIELD (0484) 514105

A WHOLE NEW QL WORLD AT YOUR FINGERTIPS

Available Now

QL SUPERBASIC: A PROGRAMMERS GUIDE

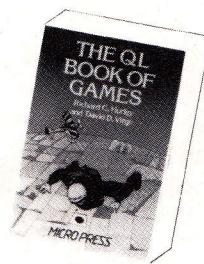
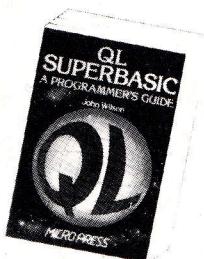
John Wilson — 144 pages £6.95

Freed from the restrictions of most other BASICS, this book guides you into sound programming habits, avoiding the maze of GOSUBs which frustrate many other BASIC programs. The exciting new features in SuperBASICare clearly explained to take you to a high degree of expertise.

THE QL BOOK OF GAMES

Richard G Hurley / David D Virgo — 160 pages £5.95

This book offers you the opportunity to start your own games library with a collection of excellent graphic games. Together the games make full use of the machine's new features and as you key in the programs you will begin to see the way SuperBASIC works. All the programs have been tested thoroughly and will provide hours of fun and entertainment.

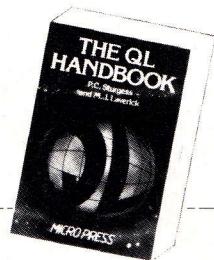


Forthcoming November

THE QL HANDBOOK

P C Sturgess / M J Laverick — approx 140 pages £6.95

This book covers everything the QL owner needs to know from connecting up and switching on to the machine's advanced special features—an essential addition to the QL manual.



Available from Booksellers or direct from the Publishers.

ORDERS TO:

Micro Press, 27 London Road,
Tunbridge Wells, Kent TN1 1BX
Tel: (0892) 39606

Name _____

Address _____

Signature _____ Date _____

Please send me:

QL SUPERBASIC
QL BOOK OF GAMES

£6.95

£5.95

I enclose cheque / postal order for £ _____
(plus £1 for p&p for a single copy and 50p each additional copy).

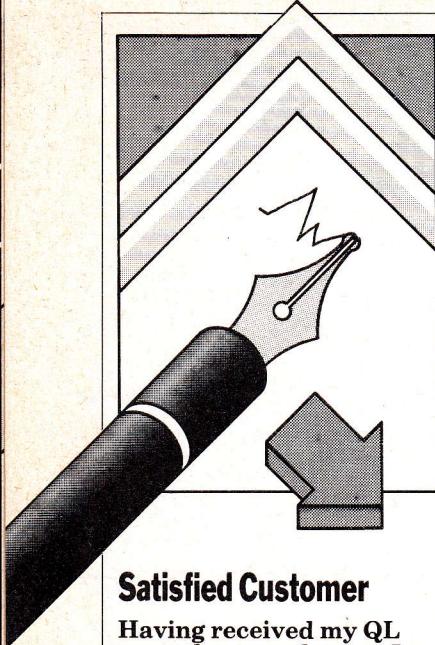
Please debit by Access / Barclaycard / Diners Card:

No: _____

I am interested in THE QL HANDBOOK. Please advise me of publication Please tick.

LETTERS

We welcome letters from our readers but do try to keep them short and to the point – long, boring letters will be cut! Send letters to: The Editor, QL User, Scriptor Court, 155 Farringdon Road, London EC1R 3AD.



Satisfied Customer

Having received my QL some three weeks ago, I must say that I am no less than delighted with the performance. I really don't see how it can all be marketed at the inclusive price of £399 and Uncle Clive still makes a profit!

I use a Shinwa CP80 with a Miracle Systems interface. The results are excellent, better I think than the Epson it is said to be the copy of, and I have to mention how superb the graphs are that I can print from Easel. One thing about the CP80 is that none of its character sets includes both the pounds sign and the hash sign. I have used the following program to simply change the character set, to which ever one I need:

```
10 REM ** English to American character sets **
11 Open #4, SER1
12 DEFine PROCedure AM
13 PRINT #4,CHR$(27); CHR$(82);CHR$(0)
14 END DEFine AM
15 DEFine PROCedure EN
16 PRINT #4,CHR$(27); CHR$(82);CHR$(3)
17 END DEFine EN
```

Then simply typing AM or EN at the start of a program line will change the upper case 3 sign to pounds or hash, e.g.

```
100 AM:print #4,"#" will print a hash sign
110 EN:print #4,"#" will print a pound sign
```

My monitor is the Sanyo med. res. model 3117. This needed no extra conversion work to portray the full 85 characters on screen, and I found the sync. plug alterations very easy. I use the following small

program (time), in 80 column mode, to give me an up-date on how long I have been sitting in front of my QL:

```
1 open #3,con_ :window #3,50,10,420,230
2 paper #3,4 :ink #3,0 :cls #3
3 let aaa$=date$ :print #3,aaa$(13 to 20)
4 window #0,400,53,0,203:cls #0
5 border #0,1,4
```

If the name of the file is mdv1_boot the passage of time will be displayed in the bottom right hand corner. Merge instead of load will include this program at the front of any program you may wish to load, and as the line numbers are so low will not interfere with your programs. Listing with "list 10 to" will not include this program in your own listings. Leaving the tape in the computer all the time does not seem to cause any trouble, and the program will load every time you press the cut out button, without incidentally, zeroing the time elapse display. It is not a good idea to leave the psion programs in your computer all the time, as switching on or off with them in position seems to destroy the overlay file, so that they are of no further use!

*Timothy Fuller
Southampton*

Four in One

Thank you for your acknowledgement of my registration card. I have had my QL for three weeks now (it is the JM version) and being on leave, have had a good opportunity to get to know it. I have mainly used the four Psion packages, and despite the

various bugs listed below, I believe they are good value considering they are included in the price for the QL. I have managed to find a use for all four packages:

a. QUIL. I have written many letters, including this one and use a Silver Reed EX-44 Electronic Typewriter, which although slow, does have its own keyboard, and is quite cheap when one considers its dual role. The program has its bugs: so far I have been trapped in a loop a couple of times and the merge command locks up the computer.

b. ABACUS. I use ABACUS to keep a track of my finances, forecasting peaks and troughs, mainly the latter! I also use it to monitor the performance of Investment Trust Companies on the Stock Market, which has proved very useful, and hopefully could earn me some money. I have been unable however to print the formulae in each cell to my printer as promised by the manual, though the values do print.

c. ARCHIVE. I have written a program to produce manning statistics for my Regiment, an almost daily task in my job. This will I hope, save me hours at home in the evenings.

d. EASEL. By importing the Investment Trust data from ABACUS, I get a good impression of how the various companies are doing. Unfortunately, the program has crashed on me a couple of times.

My other use of the QL so far has been to explore its facilities, and for this, I have written a small program. Not having a monitor, I could not immediately make use of the split window facility to

list my program and see the results. My program achieves this in a simple and readily adaptable way.

Program Listing – Window Generator

```
10 MODE 4 (or 8 if your television won't stand 4!)
20 WINDOW #0,460,42,30,212
30 WINDOW #1,287,210,203,0
40 WINDOW #2,173,210,30,0
50 BORDER #0,1,2: BORDER #1,1,4: BORDER #2,1,4
60 SCALE 100,0,0
70 PAPER #0,0: PAPER #1,0: PAPER #2,0
80 INK #0,4: INK #1,4: INK #2,4
90 DLINE 10 to 90
```

Notes:

1. The program, if saved on mdv1_ as 'boot', will load and run automatically when the F2 button is pressed on start-up.
2. The resulting screen is not affected by NEW or LOAD commands.
3. The values given for the window sizes work on my television but can easily be modified by trial and error for others.

*Capt D N Charles RA
Taunton*

Niggling Omissions

Upon recently receiving my QL I have found numerous faults and ambiguities.

1. Names of procedures that need variables passed to them need to be in upper case.
2. Names of arrays must be in lower case.
3. The microdrives run at different rates. This means that a cartridge formatted on MdV2 will not work on MdV1 and

MM

UNBELIEVABLE SAVINGS

** COMPUTERS **

		EX VAT
APRICOT	F1 - from	£675.00
APRICOT	Point 7 from	£2950.00
APRICOT	PORATBLE from	£1445.00
APRICOT	256K 315Kx2 MONITOR	£1395.00
APRICOT	256K 720Kx2 MONITOR	£1545.00
APRICOT	X1 256K 10MB MONITOR	£2195.00
CIFER	9000 Multi User/21MB	£5095.00
COMMODORE	8250 DISK DRIVE	£785.00
COMMODORE	8296	£695.00
COMMODORE	SX-64 PORTABLE	£675.00
COMMODORE	64	£156.51
COMMODORE	DISK 1541	£165.21
COMMODORE	IBEK PARALLEL INTERFACE	£59.50
COMMODORE	1530 C2N CASSETTE	£32.00
COMPAQ		£1795.00
KAYPRO	II	£45.00
KAYPRO	10MB	£1995.00
OLIVETTI	M20 160KB 2x320KB Drives	£1295.00
OLIVETTI	M24 128KB 2x360KB Drives	£1658.00
OLIVETTI	M24 128KB 10MB Hard Disk	£3315.00
SAGE	II & IV	POA
SANYO	MBC 555 128K 2x160K Drives	£795.00
SIRIUS	256K 10MB	£2850.00
SIRIUS	256K 2.4MB	£2095.00
SIRIUS	128K 1.2MB	£1545.00
SIRIUS	Memory Expansions from	£222.00
SIRIUS	Express Accelerator Boards	POA
PLUS 5	External Hard Disk Drives	POA

** MATRIX PRINTERS **

		EX VAT
ANADEX	DP-6500 500cps	£2019.00
ANADEX	WP-6000	£1808.00
BROTHER	EP44	£199.00
BROTHER	HR5	£129.00
CANON	PW1080A 160cps (NLQ)	£279.00
CANON	PW1156A 160cps (NLQ)	£339.00
EPSON	RX 80T 100cps	£195.00
EPSON	RX 80F/T 100cps	£220.00
EPSON	FX 80 160cps	£324.00
EPSON	FX 100/F/T 160cps	£430.00
EPSON	LQ 1500 200cps (NLQ)	£895.00
HONEYWELL	From	£375.00
MANNESMANN	MT80 80cps	£199.00
MANNESMANN	MT180 160cps (NLQ)	£590.00
NEC	PINWRITER P2(P)	£535.00
NEWBURY	DRE 8850 300lpm	£2095.00
NEWBURY	DRE 8925 240cps	£1385.00
OKI	82A 120cps	£255.00
OKI	84A 200cps	£630.00
OKI	OKI 92P 160cps	£379.00
OKI	OKI 2410P 350cps	£1535.00
SEIKOSHA	GP100A	£165.00
SHINWA	CP80 Model II FT	£175.00
STAR	DELTA 10 160cps	£319.00
STAR	DELTA 15 160cps	£435.00
STAR	GEMINI 10X 120cps	£189.00
STAR	GEMINI 15X 120cps	£285.00
STAR	RADIX 10 200cps (NLQ)	£449.00
STAR	RADIX 15 200cps (NLQ)	£549.00
TEC	1550 120cps	£465.00
TOSHIBA	TH2100H 192cps	£1275.00
TREND	930 200cps NLQ 80cps	£1350.00

** SOFTWARE **

ALL MAJOR SOFTWARE PROGRAMS SUPPLIED AT LOW COST

WORDSTAR	£195.00
OPEN ACCESS	£360.00
LOTUS 123	£295.00
SYMPHONY	£490.00
MULTIMATE	£240.00
D BASE II	£280.00
DMS DELTA	£395.00
FRIDAY	£135.00
FRAMEWORK	£345.00

Not only do we offer top quality products at low prices. We also support and develop Software with the assistance of our long established software dept.

MAYFAIR MICROS

BLENHEIM HOUSE, PODMORE ROAD,
LONDON SW18 1AJ

TEL: 01-870 3255

We accept official orders from UK Government and Educational Establishments. Mail Order and Export Enquiries welcome. Callers by appointment.

** DAISYWHEEL **

PRINTERS

EX VAT

BROTHER	HR1	£445.00
BROTHER	HR15	£329.00
BROTHER	HR15 Keyboard	£135.00
BROTHER	HR15 Sheetfeeder	£185.00
BROTHER	HR15 Tractor Feed	£62.00
BROTHER	HR25	£549.00
DAISYSYSTE	2000 200cps	£240.00
DIABLO	630 API	£1315.00
DIABLO	Sheet Feeder	£490.00
JUKI	6100 18cps	£325.00
NEC	2010 Serial 20cps	£545.00
NEC	2030 Parallel 20cps	£545.00
NEC	3510 Serial 35cps	£1049.00
NEC	3530 Parallel 35cps	£1049.00
NEC	7710 Serial 55cps	£1440.00
NEC	7730 Parallel 55cps	£1440.00
QUUME	11/40 RO	£1185.00
QUUME	11/55 RO	£1370.00
QUUME	9/45 RO	£1550.00
QUUME	9/55 RO	£1900.00
RICOH	RP1300S	£895.00
RICOH	RP1600S	£1190.00
RICOH	RP1600S FLOWRITER 8k	£1249.00
RICOH	RP1600S FLOWRITER 8k	£1249.00
RICOH	IBM PC	£1299.00
RICOH	RP1600S Sheet Feeder	£459.00
RICOH	RP1600S Tractor	£138.00
SMITH	TP1 12cps	£195.00
CORONA	STARWRITER F1040 40cps	£895.00
TEC	STARWRITER F1055 55cps	£1235.00
TEC	Sheetfeeder	£459.00
TEC	Tractor	£138.00

** VDU's & TERMINALS **

CIFER	T4	£760.00
HAZELTINE	ESPRIT Fixed Keyboard	£395.00
QUUME	QVT 103 (VT100 VT131)	£695.00
TELEVIDEO	910	£489.00

** PLOTTERS **

MANNESMANN	PIXY PLOTTER
GOULD	PLOTTER

£495.00
POA

- **Win Doomdark's Revenge**, the epic sequel to The Lords of Midnight, in this month's exclusive Sinclair User competition. Using fragments of manuscript and a map depicting the frozen wastes of Icemark, plot Lord Elt-Mar's journey and you could win the original artwork for the map and a cassette of the adventure. Fifty runner-up prizes also to be won.
- **Sinclair Logo** has finally arrived. Find out what the fuss is all about in our extensive review.
- **Meet the Melbourne mobster** who drew the pix for Mugsy.
- **Looking for the write stuff?** We compare word processing packages for the Spectrum.
- **Learn how to make** and use windows on the QL.
- **Plus all the latest news** and reviews of the software scene and the hardware world, and over 18 pages of programs. All in the October issue of Sinclair User, the biggest and best-selling magazine for owners of the ZX-81, Spectrum and QL machines. Available from your newsagents now.

Sinclair User
INCORPORATING SPECTRUM USER

AN APPOINTMENT NOT TO BE MISSED EVERY MONTH!

1984

Tuesday
260-78 Week 42

16

Sinclair User magazine hits the street on the 16th of every month packed with enough material to keep both you and your Sinclair machine fully occupied until the same time next month.

Sinclair User magazine is devoted entirely to the Sinclair ZX81 and Spectrum. Everything you need to know about the latest peripherals and software releases, as well as letters, book reviews, program listings and competitions. Plus special sections for beginners business and education. We even have a special telephone 'hotline' for those problems that just can't wait.

No wonder **Sinclair User** is the UK's top selling Sinclair magazine.

Available from your newsagents now, only 85p



QL

£39.95 INC

PARALLEL PRINTER INTERFACE

- ★ 12 months guarantee
- ★ Fully self-contained with connectors and 1.5 metre cable
- ★ Plugs into Sinclair QL's RS232C port and
- ★ Drives any CENTRONICS compatible printer, eg Epson, Seikosha, Juki, OKI, NEC, Shinwa, Star, MCP-40, Canon, etc, etc
- ★ Fully QDOS and SuperBasic compatible

To order send name and address with cheque to:

MIRACLE SYSTEMS Ltd

Avondale Workshops

Woodland Way

Kingswood

Bristol BS15 1QL

 Available at larger branches of **WHSMITH**



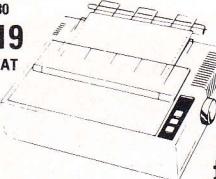
Access orders welcomed

Sinclair and QL are trademarks of Sinclair Research

DOT MATRIX

EPSON
RX80

£219
inc VAT



EPSON
RX80 F/T

£255
inc VAT

inc VAT

Epson RX80

£190.43 + VAT = £219

Epson RX80 F/T

£221.74 + VAT = £255

Epson FX80

£329.57 + VAT = £379

Epson FX100

£511.30 + VAT = £588

KDC FT 5001

£203.48 + VAT = £234

Mannesmann Tally MT80

£195.65 + VAT = £255

Canon PW 1080A

£277.39 + VAT = £319

Serial/Parallel I/F 8K Buffer for

Star Delta 10

£317.39 + VAT = £365

RS232 Interfaces from

£26.09 + VAT = £30

Printer cables from

£10.43 + VAT = £12

DAISYWHEEL

JUKI
6100

£375
inc VAT

inc VAT

JUKI 6100

£326.09 + VAT = £375

Daisystep 2000

£216.52 + VAT = £249

PROCESSORS

BBC Model B + Recorder

£320.87 + VAT = £369*

BBC 'B' with DFS

£399.13 + VAT = £459

Amstrad CPC464

£139.13 + VAT = £160

6V power + UHF

£24.25 + VAT = £28

ISL 8083 (IBM Comp)

£1173.91 + VAT = £1350

DEC 11/23 + 40 MB

£6086.96 + VAT = £7000

DEC 11/73 + 170 MB

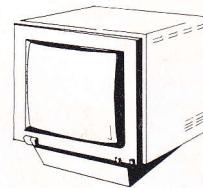
£1043.78 + VAT = £12000

PHONE FOR QUOTATION ON BESPOKE ACCOUNTING

SOFTWARE INVOICING/LEDGERS/PAYROLL

* While stocks last

colour MONITORS



MICROVITEK
1451/QL
Medium Res.
£255
inc VAT

Amstrad CTM640	£146.96 + VAT = £169
Microvitek 1431	£165.57 + VAT = £195
1431 (RGB/PAL/AUDIO)	£195.65 + VAT = £225
Texan Vision EX	£186.96 + VAT = £215
JVC 1302 1 (QL/BBC)	£169.57 + VAT = £195
Microvitek 1451/QL	£221.74 + VAT = £255
Microvitek 1451	£293.13 + VAT = £325
1451 (RGB/PAL/AUDIO)	£295.65 + VAT = £340
Texan Vision II	£234.78 + VAT = £270
JVC 1302-2 (QL/BBC)	£220.00 + VAT = £253
Microvitek 1441	£333.91 + VAT = £399
Texan Vision III	£321.74 + VAT = £370

mono MONITORS

Amstrad GT64	£60.00 + VAT = £69
Sanyo OM2112 (15MHz)	£65.22 + VAT = £75
Sanyo DM8112CX (18MHz)	£86.10 + VAT = £99
ISL 18 (18MHz)	£53.91 + VAT = £62
ISL 20 Swivel 80 col	£63.48 + VAT = £73
Teco with Zoom	£91.30 + VAT = £105

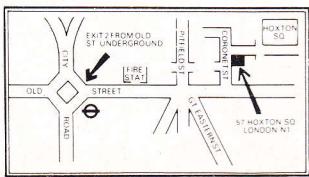
micro FAST

The Experts

57 Hoxton Square London N1

01-729-1778

Prices correct at time of going to press in September



LETTERS

vice versa.

4. Functions may not return strings.

None of these bugs are mentioned in the User Guide, I only found them after much hair pulling over misbehaving programs and cartridges.

These faults may go undiagnosed by beginners and leave them with the feeling that the machine is getting at them.

P R Morris

Spectrum Drives

Sinclair Research have been much criticised in the past for launching products before they are actually available, but I am pleased to say that this trend now seems to have been reversed. I seem to have found a product which is readily available before Sinclair have launched it - extra Microdrives for the QL!

On the right-hand edge of the QL, near the reset button is the slot designated for Microdrives to augment the two built-in to the QL. Page 2 of the Introduction to my QL Users' Guide states "ZX Microdrives are not suitable for use with the QL", but recently I have read a letter in the computing press (sorry, forgotten where), that Spectrum Microdrives could be used as read-only devices with the QL. Filled with trepidation, I was not only able to confirm this, but was able to FORMAT and SAVE files on cartridges in these extra drives. Such files are LOADable in the two original drives, and I have now been using the QL with two extra ZX Microdrives for a week with no apparent snags. I'm glad I made provision for eight microdrives in my QL Utilities package!

To connect them, I unplugged the ribbon cable from the Spectrum's Interface 1, and plugged this into the Microdrive expansion port of the QL. The Microdrives therefore end up with the slots facing away from you. Perhaps this is why Sinclair say

they are not suitable! Is this compatibility exclusive to QLs fitted with 'dongles' (I have QDOS version PM still)? I do hope some of you will try this with later QLs and let us know. Don't sue me if something goes wrong; the User Guide must be right sometimes!

C C Wilton-Davies, MA,
MBCS
Channel Islands

Launch Issue: QUEST Program

I agree with some others in saying that the QL QUEST program as listed in your launch issue would not work satisfactorily as printed. This is due to the odd typing error, for example line 1180 should read '0(5,1)=Z'.

But the main problem is the incorrect use of procedures, for example on line 1310 there is a GOTO 200 statement used to return to the main loop.

Secondly there are extensive numbers of occasions when 'main loop' is called from within lower ranking procedures. This does not RETURN to the main loop but nests another call within the current procedure eg. lines 940, 1000, 1050, 1100, 1150 and 1200 should all be deleted from procedure 'drop'.

Thirdly, jumping around from procedures

with GOTO statements should be avoided even if it is to hide the answer. For example line 2710 GOTO 700. The other major problem is that procedure 'boh' (bucket on head) is not called anywhere.

A totally rehashed version properly structured, with a window displaying score, strength and inventory all the time together with enhancements to fighting is available if you wish.

Aug/Sept Issue: SuperBasic Article

It is interesting that quite a few of the aspects that you chose to comment on as 'worthy' have been dropped from production models.

The ones I have noted so far are:

- 1 SELECT ON does not work with string variables. It would be very powerful if it did.
- 2 I can find no way of setting arrays with array literals as you suggest.
- 3 The TRACE facility has been dropped (except in ARCHIVE) - again very powerful.
- 4 There is a sort of pseudo screen editor, if you use the 'up' or 'down' cursor keys to return a line after editing then the preceding or superceding lines respectively are

automatically brought up for editing. A file can then be panned backwards and forwards a line at a time.

Note for Archive users:

'F5' rediscovers previous instruction. A bonus for panning backwards and forwards through a file rather than typing 'NEXT' and 'BACK' all the time.

D Pewson
Cheshire

Opinion

In nearly every magazine I have read on the computer market, the QL has been set up as a target for criticism. In my humble opinion and as owner of one of these magnificent machines the criticism is unjustified.

There is no other machine in its price range that can be seriously considered as a rival, in comparison overall they are obsolete. Like anyone else I can find plenty of room for improvement in my QL but at what cost?

How any sane critic can start to even contemplate matching a £399 piece of equipment with another at double, treble and in one case seven times the QL's price is ludicrous. Could it be, that this unjustified attack is the result of Sinclair's non co-operation to give the children of the press a taste of the candy first.

The bundled software alone is worth the price paid, and being a novice it was and still is refreshingly easy to use.

The Aug/Sept 1984 issue of your magazine was a joy to read and to be honest I read it cover to cover. That is the first time I have been able to do that with any magazine and must stand for something unique, or is it that the rest are garbage.

P Christie

NEXT MONTH

December/January issue 1st December

In depth interview with
Sir Clive

Can the QL run UNIX?

Printer comparison

Full review of QDOS and
interviews with its creator

Utilities programs

QL the first games

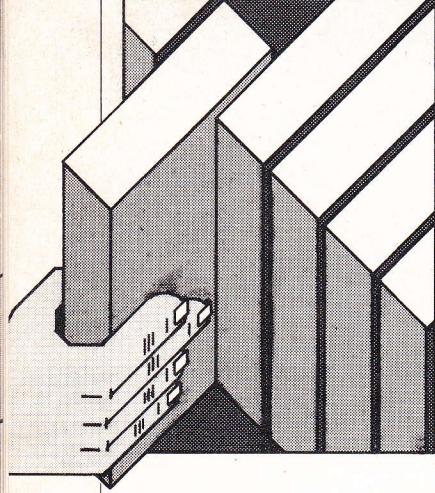
Review of accounts packaging

QL windowing capabilities

Multitasking - potential and
myth

BOOKS

Taking Titles At Their Words



Whatever the doubts about long-term microdrive reliability and non-standard operating systems, it is certainly Sinclair's intention that the QL be viewed as a 'real' computer for professionals with a genuine need for basic computing functions.

'**Desk Top Computing with the Sinclair QL for Professionals and Businesses**' would appear from the title to take Sinclair at his word. The title, however, is largely irrelevant since the book essentially does the job of a number of others including one from Hutchinson called 'Introducing the Sinclair QL', which is reviewed separately here.

To my mind, Barry Miles proves his mettle as someone who genuinely wants to convey useful information rather than someone doing a quick re-write of the manual, meanwhile there is an interesting comment at the end of his preface: '*Many people take the view that a serious business machine should have a CP/M capability and some sort of compatibility with a common discette stand.* In this regard the QL is certainly not a candidate.' Certainly preferable to the 'GOSH! WOW! AMAZING!' that fill the prefaces of books on other and lesser machines.

The bulk of this book is taken up with four long chapters – one for each of the Psion packages. It is not predominantly about the straightforward programming of the machine other than one section that shows just enough about programming and use of microdrives for the user to get the packages up and running. Basically, it's padding.

The first section on Quill is pretty good but ignores this program's biggest problem – the appallingly slow rate of printing to the screen. This obvious de-merit seriously jeopardizes Quill's usefulness.

Having said that, basic Quill ground is covered in the main without merely rehashing the manual. One of the main problems, that of non-Epson compatibility, is mentioned but no solution provided. A short section about getting the various parameters right for a variety of different printers could have been useful.

The next section deals with Abacus in particular and spreadsheets in

Three books reviewed – does the content match up to the title?

general and is good on both. Where this book scores is in that most vital of areas – explaining how to do as little as possible yourself, whilst making the computer do as much as possible. Here Barry points out the use of copy command and special QL functions like 'Net Present Value' (NPV) and 'Internal Rate of Return' (IRR), showing their power and usefulness. The section ends – as do all the Psion package sections – with an admirable weighing up of the advantages and disadvantages together with a list of features that would be useful but are missing. It treats the programs as Sinclair would wish as 'serious business packages'. The book gets 50 per cent of its worth from these sections alone.

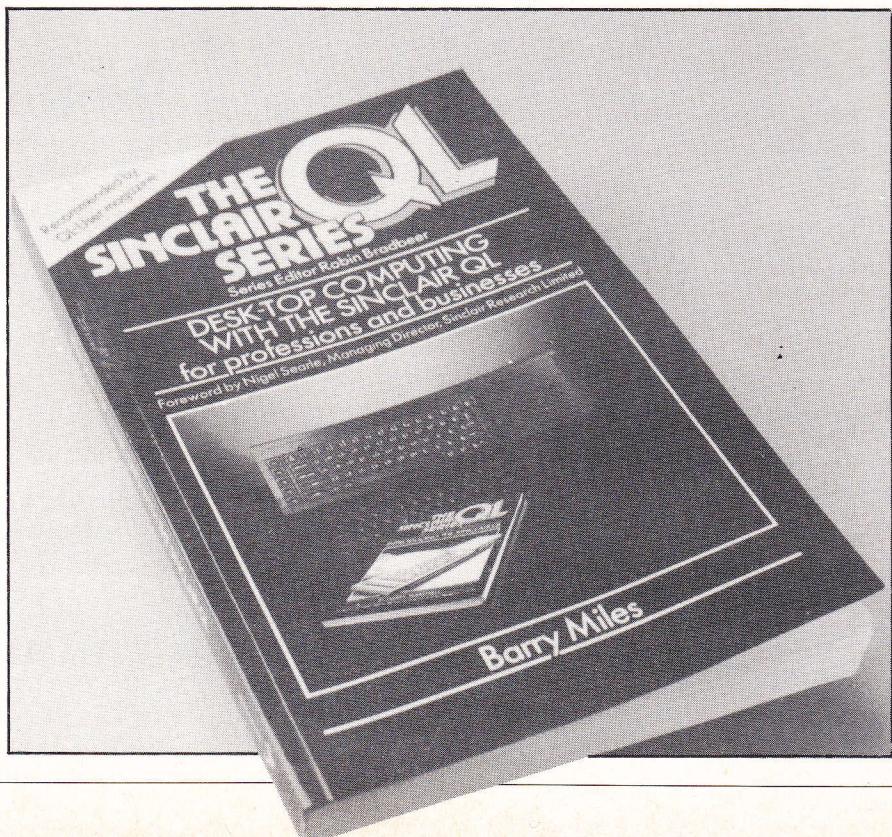
Archive is the most complex of the packages due to its new programming language-like qualities. The subject is

well covered up to a point, but to say as Barry does: '*To get the best out of Archive you will really need to study programming as an intellectual exercise and I suggest you study not only the introduction to Super Basic and Advanced Programming in this series, but also a good text on structured programming*' is not good enough. The book is only really about the four packages so how about a few Archive listings for some of the more common database functions?

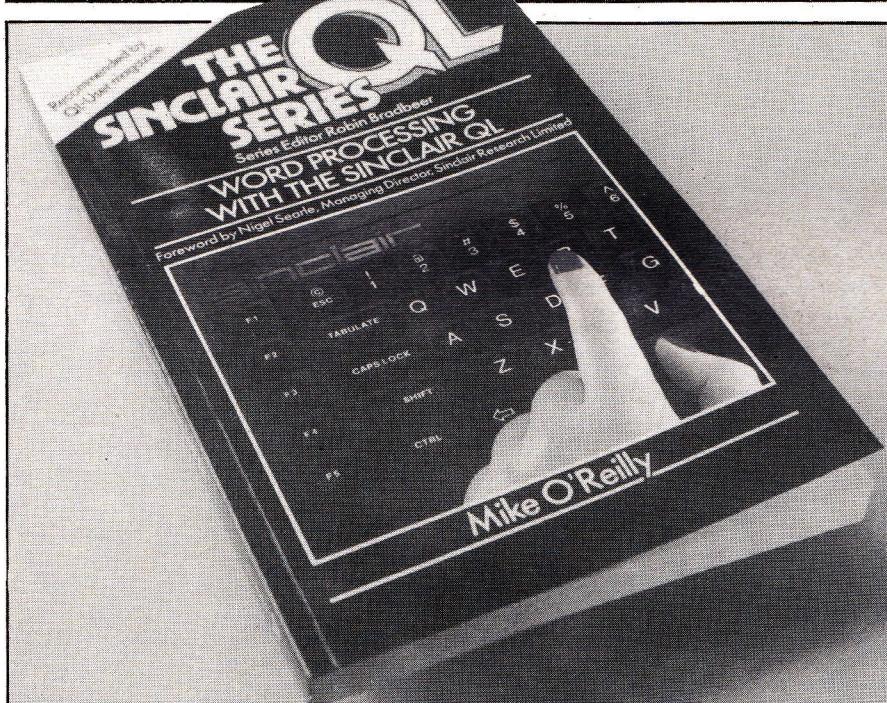
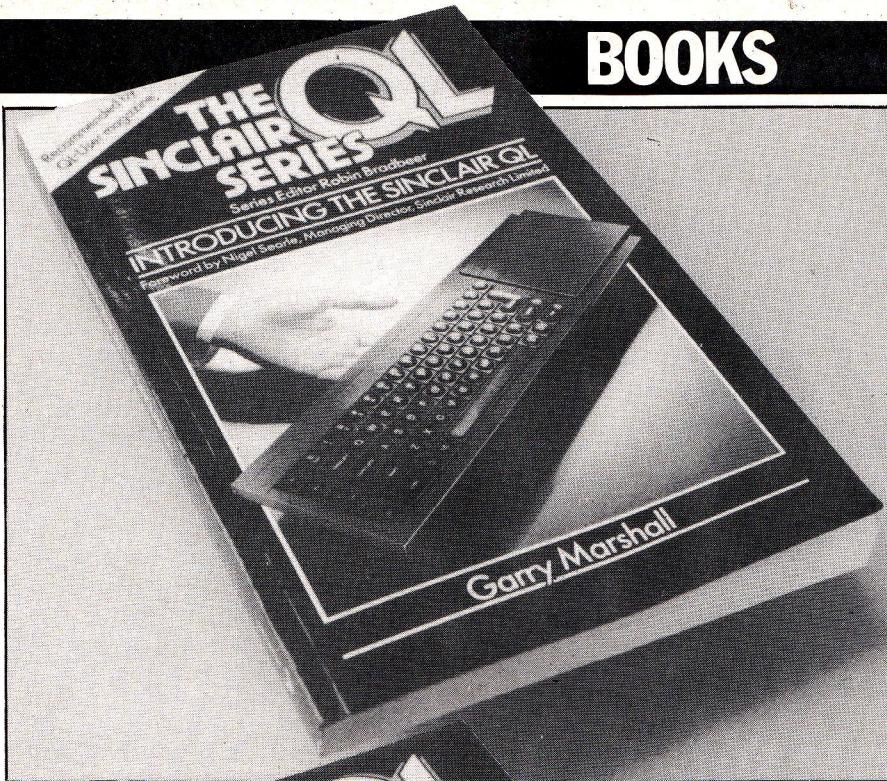
Easel is the most obviously impressive of the Psion packages and is certainly the easiest to use, consequently it receives the least attention but what is said is not just a repetition of the manual.

The book closes with a lengthy section on Basic keywords and a glossary – the former identical to that found in Introducing the QL. Whilst this is better than the keywords section in the manual, it leaves the impression that it's really only there to make the book appear longer.

Overall it's a very useful book that



BOOKS



isn't merely an attempt to make a fast buck. In particular the 'summing up' of the chief qualities of the Psion packages is essential reading for QL buyers trying to confirm in their own minds whether it can really be true that you can get 'all that' for £399.

Doubtful chapters . . .

There are few books on word processing in general so it's a mark of the QL's potency amongst book publishers that there should already be a title devoted specifically to Quill, 'Word Processing with the Sinclair QL' by Mike O'Reilly.

You may well decide that two of the longest chapters are completely irrelevant. VDUs and Health is not specifically about the QL or even word

processors in general while the other doubtful section concerns networking – sending data down the telephone – although this at least mentions the QL once or twice.

The opening section of the book discusses the QL in general and offers some good material concerning TV versus Monitor, the definition of RGB, and so on. The second chapter is all about choosing a printer – this is unusual but actually makes a lot of sense since it is a vital consideration in any word processor choice. It's another good section with information on control codes, bidirectionality, logic seeking, print speeds and the like.

Additional sections provide a genuinely useful guide to using Quill going well beyond the manual in both detail and clarity of presentation. Aside from lengthy sections on obvious things such as Overwrite and Insert there is also excellent material

concerning Search, the way it does not distinguish between upper and lower case, and when and how to use copy.

Other sections of formatting and using Superscript, text painting and the like are clearer than the manual although seemingly not actually containing any new information.

'Beyond Quill' discusses word processing adjuncts such as spelling checkers presumably in the firm belief that they will soon become available, while 'Improving Your Documents' contains some useful remarks on mundane details concerning work presentation. Despite the specialised subject matter and a little padding here and there it is genuinely a very useful book.

Little to add

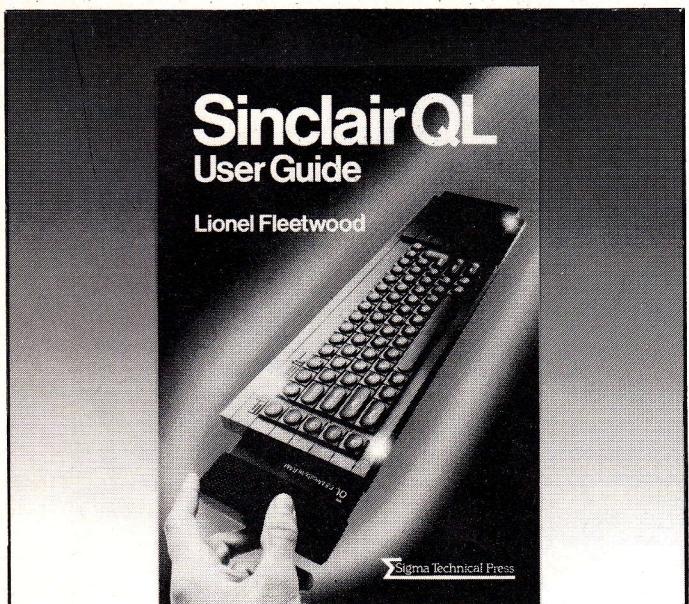
Whatever it may pretend, a more accurate title to 'Introducing the Sinclair QL' should be 'Introducing the four Psion packages that come with the QL', since these form the bulk of the book. This being so, it treads firmly on the feet of 'Desk-Top Computing with the Sinclair QL' published by the same company. It features the same appendix on SuperBasic Keywords and I can't see why anyone would want to buy both.

The opening section, 'Using the QL', consists of absolutely basic material such as plugging in and getting the microdrive working – marginally useful for people who have not been sent a manual by Sinclair. Next comes 'Quill' which is chock full of definitions for 'centring', 'format', 'line spacing' and 'tabulation' – just the kind of thing for a glossary at the back but not the stuff of chapters. Whilst this section explains quite well why a word processor is a wonderful thing, it contributes little more than the manual on Quill.

The section on Archive is better. Again, it discusses generally why databases are useful going on through some different examples from those in the manual, but not delving very deeply. We are told that 'Archive supports a complete programming language for writing procedures and programs' – but nothing more about it. The same is true of the Easel and Abacus sections which provide some different examples from those in the manual. In addition is a nice diagram showing how the TV or monitor screen can be thought of as a 'window' on a much larger matrix of figures cross-linked in Abacus. Easel also benefits from quite a few illustrations showing the various graph designs possible, but generally there is not much you couldn't readily get from the manual.

The section on SuperBasic runs to 10 pages, is bitty and practically useless for the beginner. Three pages are spent on the Let and Print commands and then it's straight into arrays of variables.

The rest of the book is the Sinclair QL SuperBasic Keywords list and all in all, since it so nearly matches Desk-Top Computing, you have a straight choice. For my money (all £6.95 of it) each one is a good buy.



Read before you leap.

Sinclair have taken their leap — now it's your turn. And with the expert guidance of Lionel Fleetwood's *Sinclair QL User Guide* you can take that leap in the right direction.

Step by step it will enable you to exploit the potential of this revolutionary new micro. The self-contained sections make quick overviews possible when you've a particular problem to solve, and real life examples show how to produce letters, keep records, prepare accounts and create useful graphs.

Bits and bytes don't appear until you get to the technical section on Super BASIC, file handling and other advanced QL features.

Even here you'll find the language remains clear and reader-friendly.

Priced £7.50, this guide will prove invaluable reading to both owners and prospective buyers alike.

You can obtain your copy through good bookshops and computer stores or simply by using the coupon below.

Published by Sigma Press (180 pages, ISBN 0905104 927) and marketed by John Wiley & Sons Ltd.

To: Carol Sutton, John Wiley & Sons Ltd, Baffins Lane, Chichester, FREEPOST, Sussex PO19 1YP (No stamp required)

Please send me *Sinclair QL User Guide(s)* at £7.50 each

I enclose PO/cheque for (payable to John Wiley & Sons Ltd.)

OR charge my credit card (Date of expiry)

Card No.

Access/Barclaycard/Visa/American Express/Diners Club (delete)

OR telephone your credit card order — dial 100 (UK only) and ask for FREEFONE 3477

Name/Address

Signature

MARKETED BY JOHN WILEY & SONS LTD (Reg. No. 641132 England) QLU

Sigma Press

EPROM SERVICES

QL, SPECTRUM AND ZX81 HARDWARE

SPECTRUM/ZX81 DATA AQUISITION/ CONTROL CARDS QL ADAPTOR

PRINTER ADAPTORS

RS 232 to centronics adaptor for use with Interface 1 or QL. Complete with cables. Other micros — Please enquire.

£39.95

ANALOG TO DIGITAL CONVERSION

8 channel 8 bit from	£23.95
4 channel 10 bit input	£27.95
8 channel 12 bit input	£57.95
3 channel 8 bit output	£26.95
also latches multiplexers etc.	

ROM BOARD £14.95

Accents 2×276A Eproms or RAM to replace the spectrum ROM. Also contains 4 user definable key switches.

INPUT/OUTPUT CARDS

24 line MK3	£14.95
24 line MK1 + LED's	£16.95
I/O accessories include output latches input multiplexers A/D, D/A and LED indicators	

EPROM PROGRAMMER £54.95

A card for Z×81 or Spectrum WYEL 2716/32/32A/64/128 complete with programming software requires external 27V-40V ie 3×9V batteries or PSU £9.95

COMMUNICATIONS

Single bidirectional RS232	£25.95
Dual bidirectional RS232	£34.95
Dual bidirectional centronics	£26.95
complete with centronics printer driver software	

EPROM CARDS and CARTRIDGES

A range of board cards interfaces and cartridges for ZX81 and Spectrum eg ZX81 card £18.95 also range of software in EPROM

TIME

Counter/timer — 4×8bit channels	£13.95
*Interval timer — 3×16 bit channels	£18.95
*Real time clock	£24.95
*Event timer	£21.95
*Crystal controlled cards	

EXTENSION BOARDS

3 slot mother boards for ZX81	£15.95
Spectrum	£16.95
complete with voltage regulation and power on indicator	

CONNECTORS 23 and 28 way

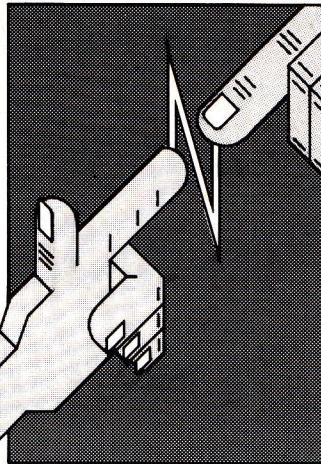
Edge cards	.75p
Edge connectors	£2.50
Gold edge plugs	£2.50
28 way IDC connector	£4.95
Many other assembled cables & parts	

EPROM SERVICES

3 Wedgwood Drive, Leeds LS8 1EF (0532) 667183

Large SAE for details. Export and trade welcome price include UK postage—overseas please add as appropriate

Industrial projects undertaken—please phone for details



Apart from the activities of the Independent QL Users' Group, this new regular feature will also contain information on QL hardware and software, news of competitive products, and items hopefully of general interest.

An Independent view

IQLUG was formed by Brian Pain and myself, Leon Heller, when the QL was first announced back in February. It is non-profit making, with the officers democratically elected by members at an Annual General Meeting. Accounts will be independently audited, and made available to members and the group is affiliated to the Association of Computer Clubs.

Membership is by subscription to Quanta, the group's monthly newsletter, currently containing 28 pages of members' letters, hints and tips, news on the QL scene, program listings and so on. It is edited by yours truly! The group maintains a (mostly) free software library, which currently contains 15 programs. All library software is written by our members, and only non-commercial items will be held there. In addition a free advice service is provided: members can phone in with their problems, and be put in touch with a member who can help them - a register of members with expertise in various areas is kept.

IQLUG is supporting the formation of sub-groups in various parts of the country with members acting as contacts for their area.

A major activity of the group will also be the organisation of workshops, the first of which was in Milton Keynes, and attended by about 50 people who discussed the Psion programs, enjoyed a presentation by Tony Tebby, (responsible for QDOS), and demonstrations by GST of their 68K/OS alternative operating system for the QL. The next workshop will be in Swindon, in November.

The Group has a cordial relationship with Sinclair Research Ltd., and complimentary copies of Quanta are sent to Sir Clive and Nigel Searle, as well as various other individuals at Sinclair.

IQLUG members have been sent a questionnaire to detail features they would most like to see in any future ROM enhancement and Sinclair have expressed an interest in the results.

IQLUG membership details are

available from Brian Pain, 24 Oxford Street, Stony Stratford, Milton Keynes, Bucks. MK11 1JU. Telephone (0908) 564271. That's dealt with the boring stuff, let's get on to something more interesting!

New Software

Metacomco, a Bristol software house recently launched their QL Assembler Development Kit which comprises a full-screen editor, and a micro-assembler all suitably impressive. Metacomco have sent some IQLUG members beta-test copies of this software and we will have a review of it in the October issue of Quanta. It should be available then priced at £59.95. Metacomco also market a Sage-based MC68000/68008 software development system which runs under the TRIPPOS operating system, and is ideal for software houses developing QL software.

Cambridge software house GST have the contract from Sinclair to supply the 'official' assembler for the QL and the cost is likely to be £39.95. Meanwhile GST will shortly be announcing their alternative operating system for the QL, 68K/OS. This will be contained in a couple of EPROMs mounted on a small PCB that plugs into the QL expansion slot, and will allow users to switch between QDOS and 68K/OS. The cost will be £99.95, including 110 pages of documentation. It comes with an editor/assembler package on a microdrive cartridge. A full range of languages will be available to run under 68K/OS and GST are supplying IQLUG with an evaluation copy of the operating system.

Computer One, also in Cambridge, have an interesting range of software for the QL. An assembler (partly written in SuperBasic) comes with a full-screen multi-tasking text editor, as does a Pascal compiler which compiles to a version of p-code, which is then interpreted at run-time, like UCSD Pascal. It had to be done this way since the QL does not have enough memory for a native-code compiler;

one that produces MC68008 machine language. This company also have Forth for the QL and both give the programmer access to most of the SuperBasic features, including full graphics support. The cost for these packages is around £40 each. For two-fingered typists there is a typing tutor program. IQLUG are getting a review copy of the assembler.

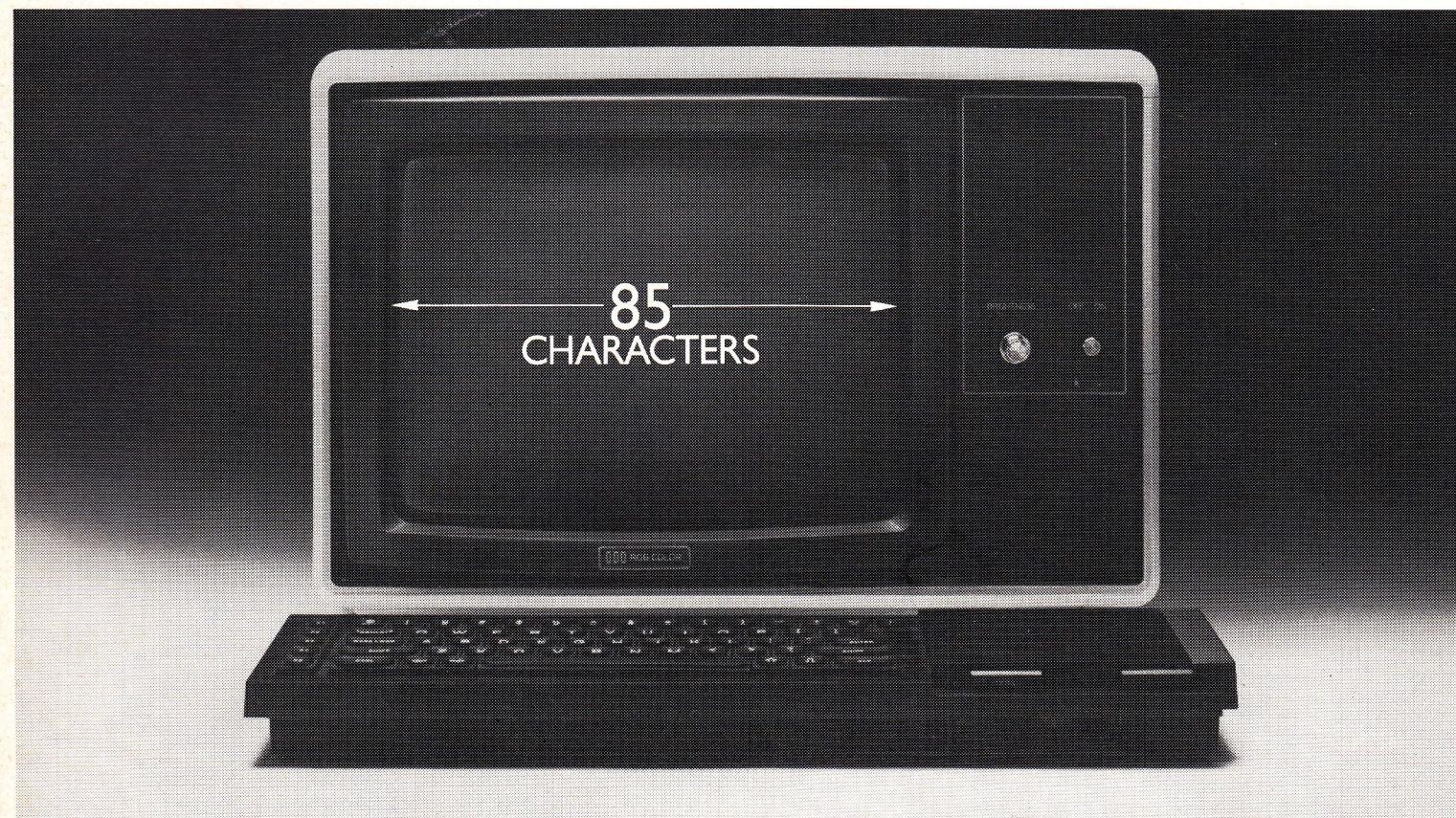
I have a copy of a letter from Alison Maguire, Sinclair's software manager, addressed to a member, in which she states that FORTRAN-77 is going to be made available on the QL. I suppose this makes sense, since FORTRAN is used for most scientific and engineering programming, and there are vast numbers of FORTRAN programs around likely to run on the QL with little or no modification. Fans of COBOL (if there are any) will be disappointed to hear that the most widely used business language will not appear on the QL. Not from Sinclair, anyway.

This high-powered development software is all very well, but most readers will probably be looking for a few games to keep the kids occupied. Unfortunately, very few software houses have owned up to an interest in games for the QL. Kerian (UK) Ltd. of Hessle, North Humberside are working on a golf simulation. They seem to know their stuff and have provided IQLUG with a couple of programs for the newsletter. Also interested in producing other 'intellectual' games, such as Backgammon.

Are you in the QLUB?

No not the pudding variety, but Sinclair's QL Users Bureau. We recently received the first issue of QLUB News which, true to Sinclair form is about six months late. It consists of four pages consisting mostly of blank space and very large headlines, and doesn't say very much. I don't think QLUB members get very much for their £35 subscription ...

A HIGH RESOLUTION MONITOR FOR THE SINCLAIR QL AT A LOWLY £299.95 (INC.VAT).



This is the much acclaimed JVC colour monitor from Opus.

And it gives Sinclair QL owners a lot more than just excellent high resolution.

As you know, the QL gives out information in 85 characters on the screen instead of the standard 80 characters.

And the loss of those five characters means that ordinary monitors simply can't show you the whole picture.

Opus, however, have overcome this problem by adapting the JVC, via a special interface and cable, so that the QL can now work 100% efficiently on it.

All for the lowly price of just £299.95 including VAT, carriage and, of course, a year's full guarantee.

To: Opus Supplies Ltd., 158 Camberwell Road, London SE5 0EE.
Please send me:

_____ High Resolution Colour Monitor(s) at
£299.95 each (inc. VAT).

I enclose a cheque for £_____ Or please debit my credit card account with the amount of £_____. My Access/Barclaycard (please tick) no. is _____

You can order it by post by filling in the coupon below and sending it to: Opus Supplies Ltd., 158 Camberwell Road, London SE5 0EE. (You'll receive it within ten days by courier service.) Or you can telephone 01-701 8668 and pay by credit card.

Alternatively, you can buy at our showroom between 9.00–6.00pm Monday to Friday, or 9.00–1.30pm Saturday.

MODEL REFERENCE	1302-2 High Resolution
RESOLUTION	580 x 470 Pixels
C.R.T.	14"
SUPPLY	220/240v. 50/60Hz.
E.H.T.	Minimum 19.5kv Maximum 22.5kv
VIDEO BAND WIDTH	10MHz.
DISPLAY	85 characters by 25 lines
SLOT PITCH	0.41mm
INPUT: VIDEO	R.G.B. Analogue/TTL Input
SYNC	Separate Sync on R.G.B. Positive or Negative
EXTERNAL CONTROLS	On/off switch and brightness control

Name _____

Address _____

Telephone _____

Opus.
Opus Supplies Ltd.

Monitoring The QL Monitors

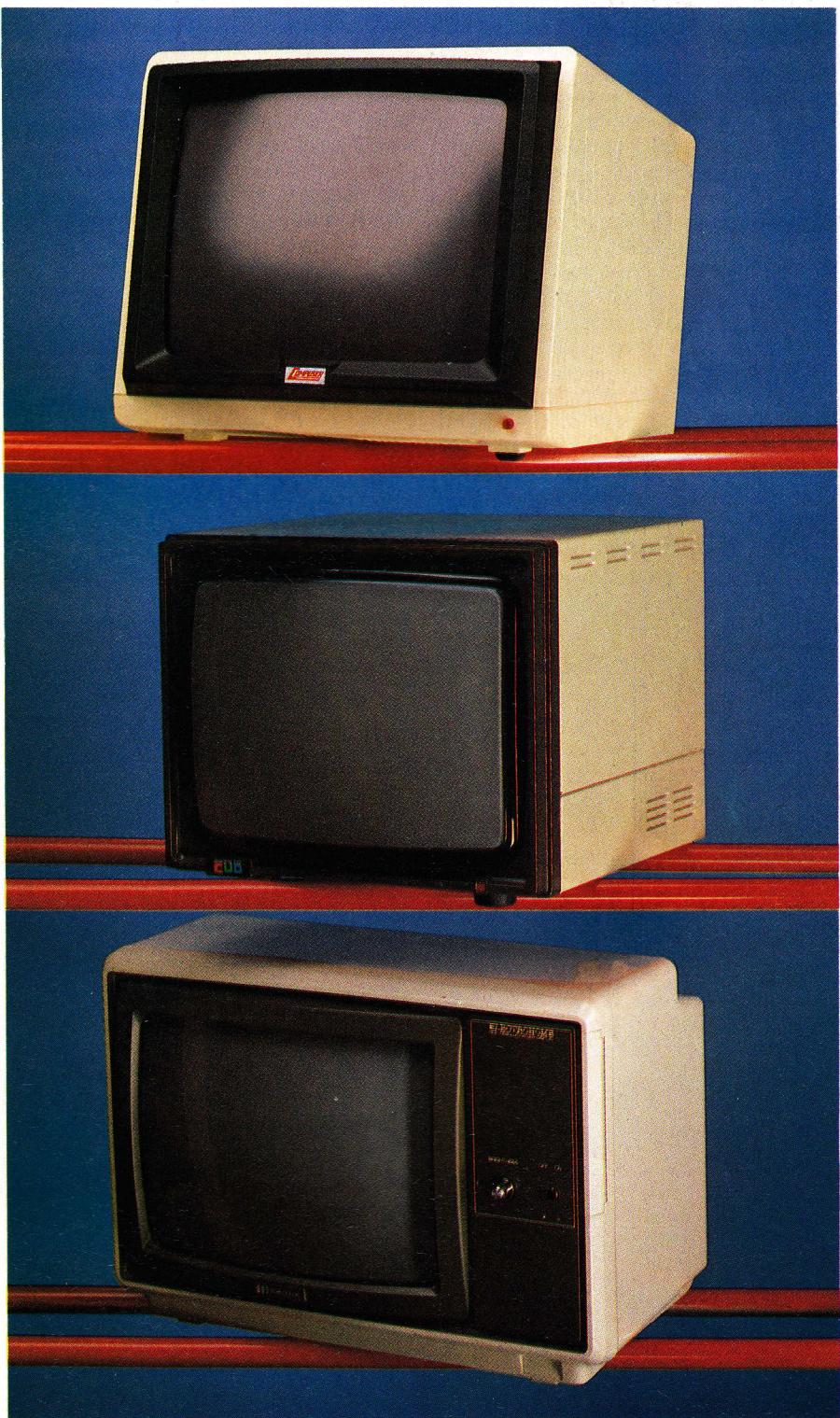
*Highlighting why monitors
are preferable to TV sets and
reviewing those currently
tailored to the QL.*

As part of its design the Sinclair QL computer features a sophisticated video display generator capable of producing high resolution graphics output and a full 80 columns of text. To take full advantage of this impressive capability it is essential that a purpose-built monitor is used to display the output rather than a standard domestic TV which, although capable of producing an adequate output, cannot do justice to the computer's potential.

In this review some of the recently released monitors designed for use with the QL are assessed for performance and value for money. In some cases they have been specifically tailored to the requirements of the QL while in others the QL version of a Q1 monitor is simply an existing model supplied with an appropriate lead. Since the QL's video output does not conform exactly to the standards laid down for video signals, models specially tailored to the QL will have the edge over other 'off-the-shelf' designs, but more of this later.

Before getting down to detail it's as well to explain exactly why it is best to discard the idea of using a domestic TV set in conjunction with the QL although there is no doubt that a TV can quite satisfactorily be used with the QL in some cases. If this were not so Sinclair would hardly have gone to the expense of providing an output suitable for direct TV connection.

Problems arise when the video display section of the computer is used to its full potential, that's when a standard TV set is just not able to keep pace. TV manufacturers produce sets designed for a specific job, namely to receive signals broadcast by the IBA and BBC displaying them in as high a quality as possible. The trouble is that the demands placed upon a set to display 80 column text are far in excess of those required for a standard TV picture. As with all mass produced products, TV sets are built to a price and it would not



make commercial sense to build in performance exceeding immediate needs. A TV set simply will not be capable of displaying a clear image when the QL's video display is being pushed to its limit.

This may indicate that all computers should be used with a monitor and if any computer is to produce its best display, this is so. Many computers though are used in undemanding applications, such as games playing and are only used for relatively short periods of time. Add to this the fact that a monitor is more than the average price paid for a home computer itself and it is easy to see why many users accept the shortcomings of a TV.

Users likely to spend some time at their computer will find a monitor an essential part of their complement of hardware. The problems of eye strain and fatigue associated with extensive time at a computer keyboard will be increased if the screen display is not

A TV set simply will not be capable of displaying a clear image when the QL's video display is being pushed to its limit.

as clear as it might be whilst the displayed image is 'wobbling' or 'jittering' as can happen with TV sets.

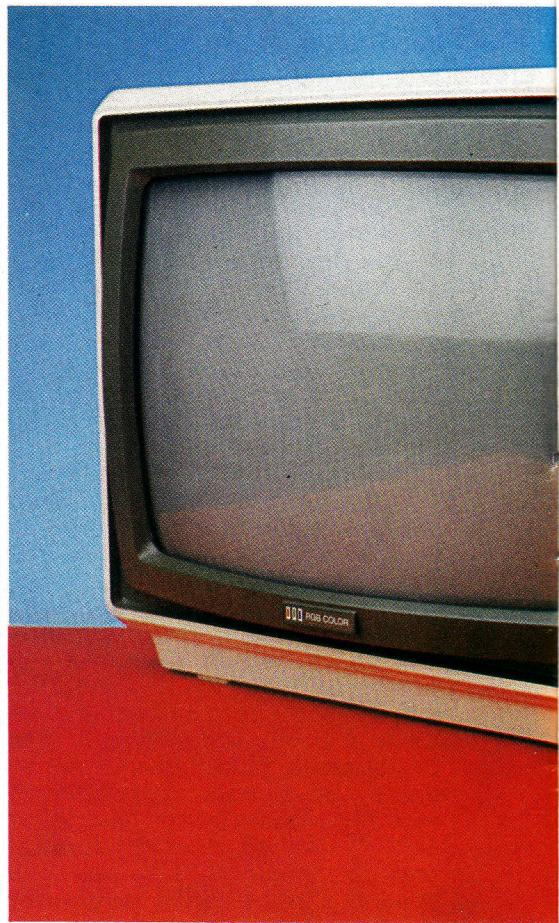
By now the benefits of using a monitor with the QL should be clear, but for full appreciation the operating principles of both TV sets and monitors must be explained. This requires a fairly technical approach which some of you may skip over, but remember, while it is not necessary to know how the engine in your car does its job, some idea of what's going on under the bonnet can be very helpful when getting the most out of the horse-less cart!

Technical background

To understand how a picture is built up on the screen of a TV set or

monitor, an awareness of a number of different principles is called for. The first of these is that any of the colours the human eye is capable of perceiving may, in the case of a video screen, be synthesised by a combination of just three primary colours - red, green and blue. A colour television tube will have three 'guns', each of which can light any point on the screen with a particular colour. The action of the three 'guns' is synchronised in ways which will be described later, but for now accept that the colour of any position on the display screen can be controlled by varying the contribution of red, green and blue.

At any point on the screen the video output of a computer, or video camera for that matter, supply three signals corresponding to the amount of red, green and blue necessary to display the colour required. In the case of an RGB (Red, Green, Blue) monitor, getting



the signal from the video source to the display is straightforward, since the red, green and blue signals of the computer, or camera, are directly connected to the electron guns of the display tube. This is the reason that RGB monitors offer the highest quality display and partly explains why domestic TV sets or monitors described as having a composite video input have difficulty reproducing high definition displays.

The major obstacle preventing domestic TV sets from producing high quality displays for computers is that they are designed to receive transmissions from broadcast stations. The standards laid down for broadcast video signals carry a number of constraints that restrict the ultimate quality of a transmitter's output. A colour signal is broadcast in such a way that it is also possible to receive it with a black and white set. The colour signal takes up no more 'airspace'

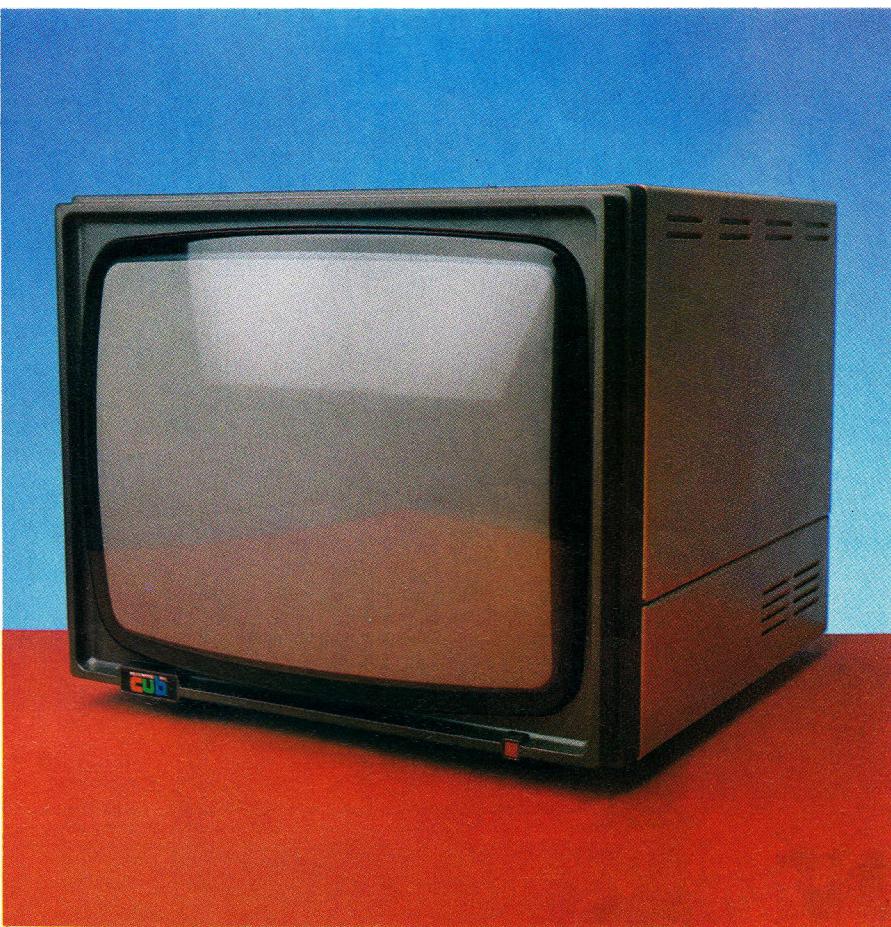
Problems arise when the video display section of the computer is used to its full potential.

than a comparable monochrome transmission. A colour signal is broadcast as a luminance signal (combining red, green and blue information) and as two colour difference signals. The luminance signal will produce a correctly balanced black and white picture achieving compatibility with mono TV sets which ignore colour difference signals. These two signals are cunningly modulated on to a single carrier frequency at 4.43MHz. Any TV set or composite monitor must place a notch filter in the video signal's path to retrieve this information and as this is in the middle of the video bandwidth, it is bound to reduce the quality of the display.

An additional factor that degrades the output of the QL when used with a standard TV set is that the computer must mimic the output of a TV station's transmitter,

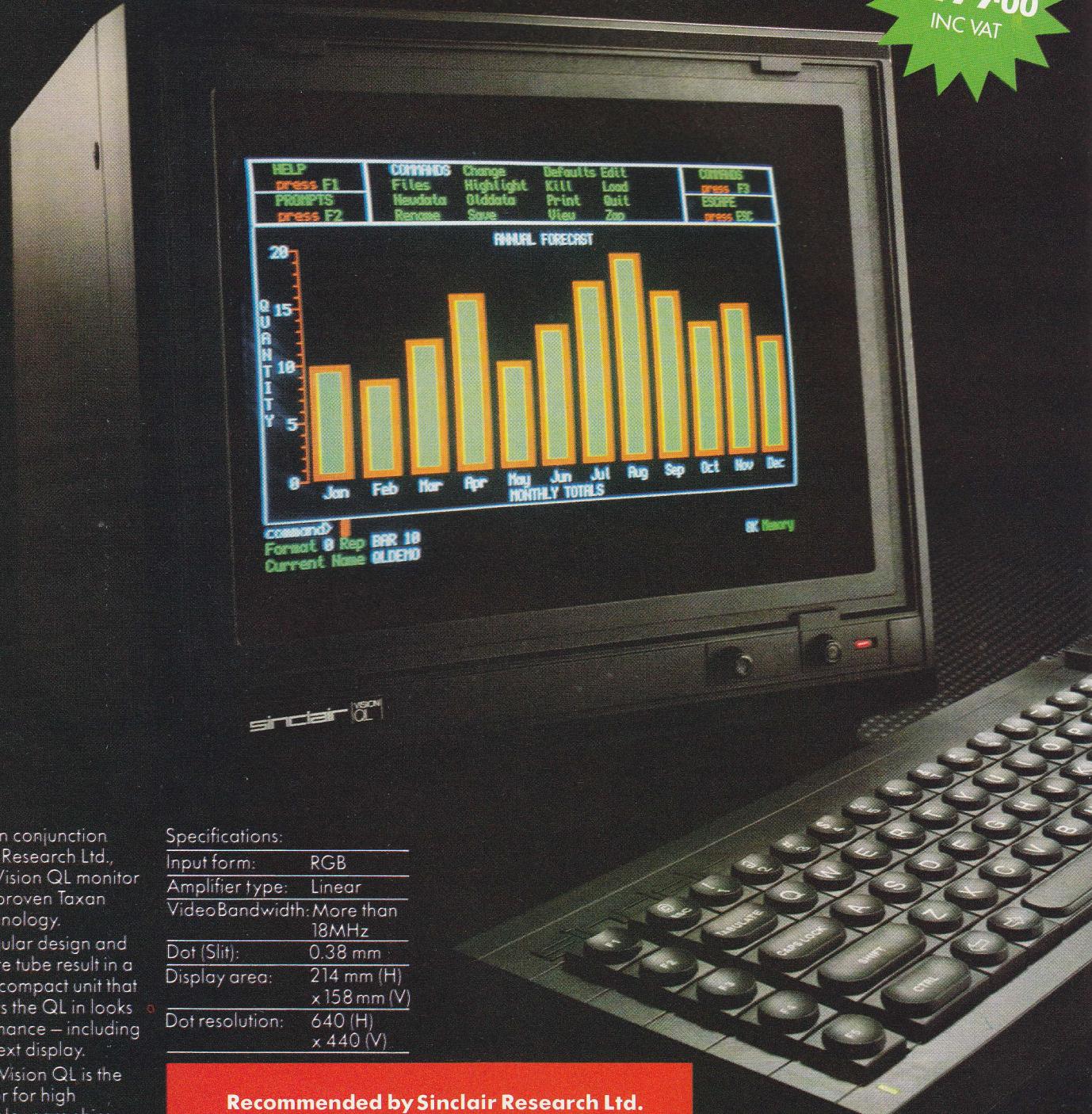
which means the video signal must be modulated on to a radio frequency wave only to be demodulated within the TV set. This process adds to the degradation of the final display.

According to some reports the QL has a potential problem because its video output does not conform to standards for certain aspects of the video signal. Many computers do not meet all of the requirements of a broadcast TV signal and indeed this is not necessary to produce a perfectly acceptable screen result. It has also been remarked upon that the QL does not allow sufficient time for 'line flyback' - jargon for the period during which the scanning beam making up the display is inactive, while returning to the left hand side of the screen to display a new line. This manifests itself when the QL is displaying 80 column text. With some monitors it's possible that some characters at the ex-



THE **sinclair** VISION QL MONITOR

£299.00
INC VAT



Developed in conjunction with Sinclair Research Ltd., the Sinclair Vision QL monitor is based on proven Taxan monitor technology.

Its black angular design and 12" non-glare tube result in a remarkably compact unit that complements the QL in looks and performance – including 85 column text display.

The Sinclair Vision QL is the ideal monitor for high resolution colour graphics and professional business applications.

Comes complete with cable and full 12 months warranty.

Specifications:

Input form:	RGB
Amplifier type:	Linear
VideoBandwidth:	More than 18MHz
Dot (Slit):	0.38 mm
Display area:	214 mm (H) x 158 mm (V)
Dot resolution:	640 (H) x 440 (V)

**Recommended by Sinclair Research Ltd.
for use with the Sinclair QL computer.**

**Manufactured for and sold, guaranteed
and serviced by MBS Data Efficiency Ltd.**

AVAILABLE FROM SELECTED BRANCHES OF MAJOR RETAILERS NATIONWIDE



MBS Data Efficiency Ltd

EXCLUSIVE UK DISTRIBUTOR

Computer Peripherals Division, Maxted Road, Hemel Hempstead, Herts. HP2 7LE Tel: (0442) 60155 (20 lines) Telex: 825554 DATEFF G

MBS DE is a member
of the Micro Business Systems plc
group of companies.

'Sinclair' and 'QL' are registered trademarks of Sinclair Research Ltd.

tremities of the display 'fall off the screen'. In some cases this is correctable by adjusting the screen width control but not always.

Monitors assessed

Of the monitors supplied for review three were RGB types while the fourth was a green screen mono model. We shall first compare the performance of the RGB models before assessing Citadel* Product's green screen unit. There is a considerable difference between the way colour and mono monitors perform. Single colour models offer far higher resolution and if the only use of your QL is to be word processing, for example, then a mono display will probably be adequate. In many applications, however, the extra quality that colour gives the QL's display will be worth any slight sacrifice in definition that use of an RGB monitor entails.

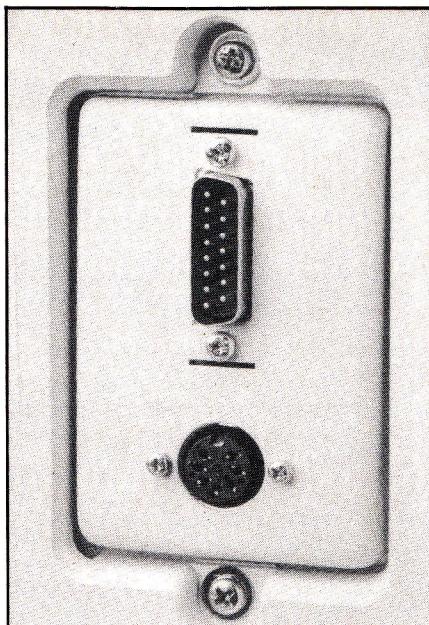
Always bear in mind that monitors are unlike most computer peripherals, for when purchasing the only thing that needs to be considered is the hardware compatibility of the QL and the particular monitor. In most cases this boils down to the provision of the correct connection lead. Unlike so many other QL add-ons there is no question of software compatibility. The QL produces a video signal that it expects a monitor to faithfully reproduce.

The RGB monitors

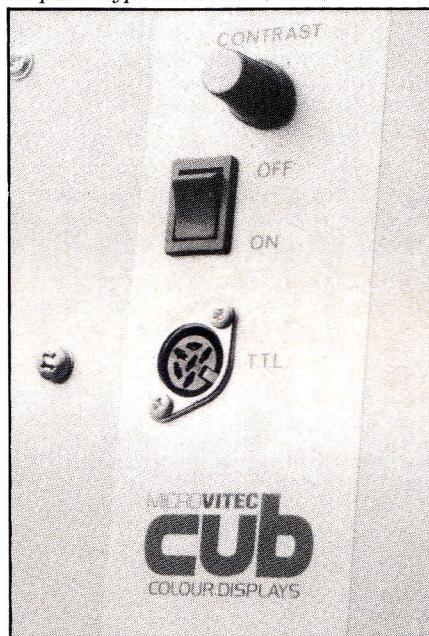
The RGB monitors under review are those from Microvitec, Opus supplies and Compuser. In terms of styling, the Microvitec monitor stands apart as it is finished in matt black to complement the QL. The Opus and Compuser units are finished in a more traditional buff/beige colour which, while acceptable, does not blend in with computer.

In all cases user controls are kept to a minimum, these being the on/off switch and a display brightness adjustment. Access to other controls is presumably possible on all of them but only the Opus instruction manual details the location of additional facilities.

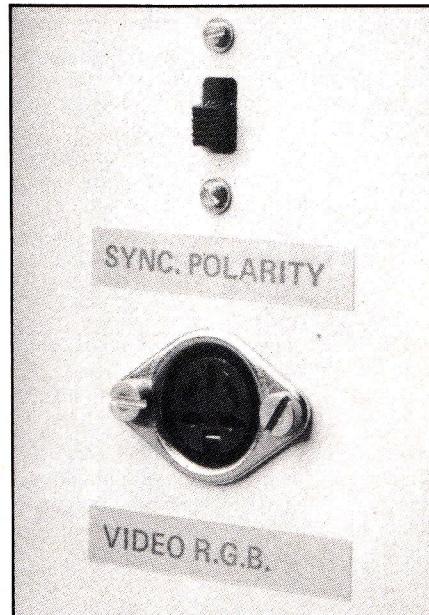
The connection between monitor and QL requires a special lead that is supplied. It is essential to have the correct lead as the plugs on



Opus Supplies: 8 pin DIN (female) and 15 pin D type connector (male).



Microvitec: 6 pin DIN (female) connector.



Compusense: 6 pin DIN (female) connector.

both the QL and the monitors reviewed could well prove difficult to obtain at a local shop, in addition the task of making up the lead is best left to a dealer.

The display produced by all three monitors is, as expected, superior to that of a TV set. Definition is improved and black really is black rather than dull grey that is often the background to a TV set's display. All instruction books are adequate, if brief, and all of the sets perform according to specification.

Green screen monitor

The green screen monitor from Citadel also met the promise of its specification, although there is no doubt a monochrome monitor can mean some programs lose their impact, not so important in a strictly word processing application, as we've said, but if graphics are to be incorporated, colour really is a must. Having said that the Citadel monitor is much cheaper than the RGB designs and offers all the benefits of improved display quality. If printed output from the QL is unlikely to be in colour, such as in many business applications, the green screen display should prove quite acceptable.

Specifications guide

Screen size measured across tube diagonal: 14in is an acceptable size in the majority of applications.

Resolution: a measure of the quality of the monitor in terms of the maximum definition it can provide, usually quoted in pixels, some manufacturers produce monitors of different resolutions, high, medium and standard.

Bandwidth: an alternative measure of the quality of a monitor – the higher the figure the better.

* Since this article was written a question mark hangs over availability of the Citadel model at least until 'supply problems' have been settled.

Microvitec, Futures Way, Bolling Road, Bradford BD4 7TU.

Opus Supplies, 158 Camberwell Road, London SE5 0EE.

Citadel Products, 50 High Street, Edgware, Middlesex HA8 7EP.

The Spectrum as a Pri

Mark Jones investigates the possibilities with varying degrees of success ...



Kevin O'Keeffe

This article and accompanying machine code program is aimed at allowing those QL owners who also have a Spectrum to use that machine as a printer interface and buffer for the QL. The machine-code was written to work with a 'JM' version QL but it should also work with 'AH' and 'FB' versions. To find which version you have, type PRINT VER\$.

When the QL's glossy brochure appeared there were many promises for a machine that was then not even at the production stage. Now that various versions of the QL have appeared those promises can be reviewed.

It appears that QDOS, the multi-tasking system, is not directly accessible from BASIC, which will disappoint many QL owners and may put off some potential purchasers. To be fair access to QDOS was not stated but implied in the

brochure. It is only machine-code programmers who will be able to use QDOS while the BASIC programmer is made frustratingly aware of the power of the multi-tasking system by the microdrives whirring long after his prompt cursor has returned.

Another promise of the brochure was the ability to network both Spectrums and QLs. This I decided to test.

The first problem is getting any sort of communication going between the two computers, because though the QL has exactly the same type of net socket as the Spectrum, its software is quite different. Trying to send a program down to the Spectrum, - using the SAVE neto_ or SBYTES or SEXEC command on the QL LOAD **"n";1 or LOAD**"n";1 CODE or LOAD**"n";1 DATA on the Spectrum - resulted in a

'wrong file type' error on the Spectrum. This is probably not surprising as the machines are vastly different but Uncle Clive might have spared us the tedium of messng about with opening channels and using the INKEY\$ function to move data.

Here is the BASIC program to print out a QL program on the Spectrum screen. A more sophisticated machine-code version of this appears later in the article:

SPECTRUM QL
10 CLOSE #6 SAVE NETO_1
20 OPEN #6,"n";1
30 LET A\$=INKEY\$#6
35 IF A\$=CHR\$(10) THEN LET A\$=CHR\$(13)
40 IF A\$="" THEN GOTO 20
50 PRINT A\$;
60 GOTO 20

Notice that the QL end of line character is 10 whereas the new-line character on the Spectrum is 13. As will be seen from the above listing, the QL listing is saved as a block of ASCII characters. Again the Spectrum differs here in that the machine's listings are made up of character codes, two byte stores for line number and length of line and floating point forms of numbers. As a result listings cannot be directly moved from computer to computer.

Since the QL listings are stored as a string of ASCII characters it should be possible to write QL programs on a word processor and with the right sort of software move directly on to the QL. At this point I had the idea of writing an on-screen editor on my trusty Spectrum and then writing QL programs using this rather than the cumbersome QL line editor. This may appear as rather a step backwards but so far I have not yet mastered writing 68000 machine language without an assembler whereas it would be relatively easy to write such an editor in Z80.

With this idea in mind I tried to send information from my Spectrum to my QL. My dreams of an on-screen editor were shattered: most data sent from the Spectrum arrived at the QL in a corrupted form - on average three bytes out

nter Driver For Your QL

of every five. It seems that the QL listening software is not compatible with the Spectrum output software, perhaps it is the timing that is wrong. Sinclair claimed they did not know of this problem - no doubt they have greater worries on their hands than this minor bug.

As it could be my QL that is faulty the only way to be sure is to try the network on your own machine. Here is a test program which should move 128 bytes from the Spectrum to the QL.

SPECTRUM

```
10 FOR F= 50000 TO 50127
20 POKE F,32+ (F-50000)
30 NEXT F
40 SAVE **"n";1 CODE 50000,128
```

QL

```
10 open #6,neti_1
20 a$=INKEY#(£6)
30 IF a$="" THEN GOTO 20
40 PRINT a$
50 GO TO 20
```

If you have no corruption the QL ASCII character set should print out. Of course it is always possible to send data from the Spectrum to the QL via the RS232 interface but this requires a special lead since Sinclair's QL RS232 has a different connector from the Interface 1 RS232 port - both of which are different from the international standard! Sinclair standards are not even compatible with themselves nowadays!

The machine-code program included does have facilities for sending QL programs and code from the Spectrum to the QL, for those who have the appropriate RS232 lead. Details for this lead are given on page 49 of the Interface 1 manual and should be a relatively easy operation if you do not mind a little bit of soldering.

Other possible solutions to the Spectrum-QL network problem would be to send the same byte of data down the network a number of times and then analyse the data at the QL end to see which is the most commonly received byte.

It seems that Sinclair Research have neglected yet another aspect of the QL system in their hurry to get it launched. The QL is relatively slow over the network compared with the Spectrum, and it lacks the ability to load up with data from the Spectrum. This only goes to further the distinct lack of compatibility between Sinclair products as well as their renown for

Code Creator

```
10 DATA "3e013231773e073236772
a3477222f7721557522b75ccf31ed7b3
d5c3b3b012f75c3b71ec92a3d5c11617
5732372c3f0013a3a5cf0b2821fdcb0
17e2007216175e5c3b712cd0313fd360
0ff2a595ccda711216175e5c3b4122a5
d5c2b7ef2a20d5fd36"
11 DATA 13373
20 DATA "26000cd3a7679fe092826f
e0a284178fe0d2808fd36000bf3a20b
9fd3600ffffdcb017e2042d16175e521b
712e5c3761bc578fe0d2814fd3600ffff
d362600cd7a1cfdfcb007ec1288c18c5c
10e0d18c0c5fd3600ffffd362600cd7a1
cc1fdcb007eca6875c5"
21 DATA 14128
30 DATA "cd7d1cc1fdcb007eca687
5c3a375216175e5793d875f160021197
6195e2356ebe95177c0764377b076b87
63376337616784e78a178fe78f19783a7
879323677c3761b2174760e00060c0cc
5cd5676c1280610f6d535d5cd535d5
c1a7c90605ed5b5d5c"
31 DATA 13106
40 DATA "e51af0e5b38020d620be200
5132310f2afe1f501050009f1c9514c4
95354514c4f4144514c4c53345a58505
2544e4f4e5a5852533233324e4554574
b5153415645515342595451535845435
14845584c51484558503e00323177c37
61b3e01323177c3761b"
41 DATA 11980
50 DATA "2a3477222d773a3677fe0
6cabf7721d65c3601cf2ddde5e122515
cdde5cd61530fbfeff2024dde1060ec
5dd6e122515cdde5cd61530fbddde1c
110eddde5e122515cdde5cd61530fbfd
de12a2d777723222d77dd7e0fa728e3d
d7e13ddbe1420dbcf2e"
51 DATA 14757
60 DATA "2a2d77222d773c3761b000
0007d0100000007d00cd6b0d213177ccb
e3e021807213177cbfe3e03cd01162a3
477c9cd3777cd541f3059cd357928587
e23fe0a20043e0d180cf02030023ec0f
ec138023ec0e5213177fe7e2808cb462
804fe80302b2a7b5ce5"
61 DATA 13202
70 DATA "cb7f281ef5e6705f1600c
b23cb12cb23cb12cb23cb12214c79192
27b5cf1e60fc690d7e1227b5ce118a23
e0df5d73efec0d0116c3761b2c95f32c75
c110c00ed53c35ce5cf1da728fb0effb
9201dcf1dcf1dcf1d57d5cf1dd15fd50
60bc5cf1dc110fad10e"
71 DATA 14665
80 DATA "0013ed53feffe11800e5c
5d5cf1d1c1e1cb792804fe1a280b772
31b7ab320e9cb7920e5222f77c3761b2
10c0022c35c2a3477cd35792800f2a347
72b23e57ecf1ee1cd357920f53e1acf1
ec3761b2a2f77ed5b3477a7ed52ca761
b4d442a3477180acd99"
81 DATA 13862
90 DATA "1ec5cd991ec5e1c13e0c3
2c35cccd9878c57ee5cf1ee123c10b78b
120f3c3761bd5e5c53efffcfleafcf1ea
fcf1ec1c5c578cf1ec179cf1ec1d1c
9e5c542c5afcfc1ec110f9c1e1c9cd707
8160acdb8b78c93e0c32c35cccd991ec5c
d991ec5cd991ec5e1c1"
91 DATA 15101
100 DATA "d1c7d7078e5c5d5afcf1e3
e01cf1e1602cd8b78d1d57acf1ed17bc
f1ec1e11606cd8b78c36078e52132773
600ed6f0f0f0fcdef787ecdef78e1c
```

9c630fe3a38002c607d7c9cd43771803c
d3777cd3579ca761bcd3f793e17d73e0
ad7afdf77ecdda7823cd

101 DATA 14651
110 DATA "3579ca761b3e17d73e12d
7afdf77ecdda7823cd3579ca761b3e0dd
718d2e5a7ed5b2f77ed52e1c9d5e51e2
001f0d8cd2a19c3301a280038043c443
c00384038043c443c00100038043c443
c0028003844444438003840384444443
80000021c2c34245800"
111 DATA 12792
120 DATA "280044444444438000001c2
020201c08187880784444444000000f2c
2fcfs0fe000000fca2bca0fe000810380
43c443c00201038043c443c00d718243
8043c443c002800384478403c00008103
84478403c000280010"
121 DATA 11520
130 DATA "101010000810001010101
00020100010101010001028001010101
000081038444444438002010384444443
800182438444444438000810444444443
800201044444444438001028444444443
800038243824242800000081c28282821
c0004428107c107c00"
131 DATA 10291
140 DATA "00201000000000000014001
82224427e42001c201824427e420010001
824427e420008107e407e407e00283c4
24242423c0078bc424242423c00003c4
62a32225c002842424242423c00003c4
240423c0818005a62524a464200003e5
85cf0909e00007e309c"
141 DATA 11380
150 DATA "90307e000020524c48542
200003c403c42423c00003c427e42423
c000040201018244200004242625e404
0000003e64242424000010385454381
00000100010101000000010101020423
c0000780438403c00103c403c423c023
c0000423c42423c4200"
151 DATA 10888
160 DATA "001428502814000000281
40a1428000000001028100000000000
03e000000000010307e3010000000000
c7e0c0800000010387c1010100000000
0107c3810002a142a142a142a00"
161 DATA 7721
5000 CLEAR 29999: LET c=0: LET f
=50000
5010 FOR h=1 TO 16: READ a\$
5020 FOR s=1 TO LEN a\$ STEP 2
5030 LET a=CODE a\$(s): LET b=CODE
E a\$(s+1)
5040 LET c=c+b+a
5050 IF a>96 THEN LET a=a-96
5060 IF b>96 THEN LET b=b-96
5070 LET a=a-48
5080 LET b=b-48
5090 POKE f,a*16+b
5100 LET f=f+1
5110 NEXT s
5111 PRINT h
5120 READ tot: IF tot<>c THEN P
RINT "Error at line ";h#10: STOP
5125 LET c=0
5130 NEXT h
5135 BEEP .1,.1: CLS
5140 PRINT AT 15,0;"To save code
to microdrive type SAVE *'m';1;
'olist' CODE 30000 ,1700"
5150 PRINT AT 12,0;"To save code
to tape type SAVE 'Qlist'
CODE 30000,1700"

QL network. Leader bytes format.

	SBYTES	SEXEC	SAVE
1	255	255	
2	0	0	data
3	1 - Length of code	-1	starts
4	1 - being moved.	-1	here
5	1 - 3 bytes	-1	
6	0	0	
7	0	1	
8	0	0	
9	0	d - Size of EXEC	
10	0	d - data area.	
11	0	d - 3 bytes.	
12	0	0	
13	0	0	
14	0	0	
15	0	0	
16	... data starts here ..	-	
17	-	-	

lack of compatibility with anyone else's.

Of course it could be argued that compatibility stunts the progress of new computers as, to be so, they need to keep the operating systems of previous generations. Take note makers of MSX machines!

I hope the machine-code program included will make up for some of Uncle Clive's omissions and what is more, ensure that your Spectrum has a reprieve from its dusty fate at the bottom of a drawer.

The above information defines how the first 15 bytes sent out on the network by the QL are formed. When saving a program no leading data is sent out at all, so the first bytes sent out of the one line BASIC program 10 print; - would be ascii code "1", ascii code "0", ascii code "", asc "p", asc "r". etc.

The final program is written in machine code for the Spectrum. It allows the Spectrum to accept data sent from the QL using the SAVE, SBYTES or SEXEC commands. If you are using the network both Spectrum and QL should be assigned as being station 1 i.e. their default values. QL programs once sent down to the Spectrum can be listed to the screen using the command *QLIST or to a printer using *QLLST.

Since data is included for the entire QL character set, listings should appear identical to QL listings on the screen and on the ZX printer. If you are using a printer other than a ZX type *NONZX before you do any printing to allow the printer to interpret the ascii codes in its own way. *ZXPRT allows the program to print out on ZX printer again. Blocks of code and programs from the QL are loaded to address 32000 on the Spectrum. Blocks of code can be dumped to a printer or the screen as QL ascii characters with the normal *QLIST and *QLLST. The current QL file can also be dumped to the screen or printer as hex

bytes using *QHEXL and *QHEXP. The list commands will stop dumping to screen or printer automatically on printing the last byte of the last block of code loaded in. Finally the program allows programs to be sent back up to QL via the RS232.

It could be argued that compatibility stunts the progress of new computers

To input the code, first of all type in Program 1 and then save it. Then run the program and save the resultant code. The BASIC program contains a checksum which should help sort out errors. To use the code, first of all load up the Spectrum with any software your printer might use, and then load the code using LOAD " " CODE. This code is located at 30000 onwards so this will limit the size of the BASIC programs that can be used but it does mean that fairly large QL listings can be stored. To turn the code on, type RAND USR 30000. This will set the code up for a ZX printer, setting the current listing to a null file and doing a CLEAR 29999. To load up with a QL program type *QLOAD on the Spectrum and then type SAVE neto_1 or SAVE SER1z on the QL. To save code from the QL use SBYTES or SEXEC.

SUMMARY OF COMMANDS

*QLOAD To load any QL program or code to address 32000 on the Spectrum. Initially set to network mode. Uses RS232 or network.
 *QLIST To list a previously loaded QL program to the screen or to dump the last block of code to be loaded on the screen as ASCII characters.

*QLLST As above but dumps to printer.

*NONZX Tells program that printer is not ZX.

*ZXPRT Tells program that printer is ZX.

*RS232 Sets routine to RS232 mode. This is set at 9600 baud so you may have to set your QL to this baud rate.

*NETWK Sets routine to network mode.

*QSAVE Saves last program that was loaded back to QL. Type LOAD SER1z on the QL. The 'z' indicates that a control Z is sent to mark the end of file. If you forget this you will have to escape from the communication once the program is loaded. N.B. Control Z is not needed when moving blocks of code. Uses RS232 only.

*QSBYT Takes the form *QSBYT or *QSBYT x,y. With no parameters specified the code block sent is the block of code from 32000 to the end of the current file. Otherwise X indicates the start address of the block of code and Y indicates the length of the code. Type LBYTES SER1, start_address on the QL. Uses RS232 only.

*QSXEC Takes the form *QSXEC start_address, length, size of data area. This command is for sending machine-code programs only. Use RS232 only.

*QHEXL Lists the current QL file to the screen as pairs of hex bytes. I use pairs of bytes as these have more significance in a 16-bit machine.

*QHEXP As above but dumps to printer.

ROUTINE VARIABLES

All of these values can be changed by the appropriate pokes.

ADDRESS

30001 Initial printer mode. ZX=0. NONZX=1. Poke the mode you prefer and resave the code if you wish the program to start in that mode.

30003 Initial Net/RS232 mode. NET=7. RS232=6. Can be changed as above.

30513 Current printer mode.

30518 Current Net/RS232 mode.

30516 Stores address to which all programs and code are stored. Normally 32000.

Psion's Easel: Pie in The Sky

Once you have a word-processor, database and spreadsheet what more could you possibly want? The answer - a dainty little business graphics package to dress up any disastrous results or forbidding forecasts. Psion's Easel is exactly this ...

The old adage that 'one picture is worth a thousand words' takes on a new meaning in business graphics where a single figure can take on a thousand guises. The tabular form is dead, long live the bar chart upright or horizontal, and its cousins the pie chart and line graph.

Signing on

Getting into Easel is no problem at all. Insert its cartridge in the left hand drive and a blank one in the other. Then, if in SuperBasic type "1run mdv1_boot" otherwise make a selection between TV and monitor and the program will automatically load.

Getting out is also simple. Most will exit of their own accord via the 'quit' command on the control ('F3') display while some, who like us received an apparently duff copy of the program, will be shunted out involuntarily and without ceremony whenever any advanced function is attempted! In our case an abortive cloning session (bad mode on GOVR_OVR) and a version number (1.01) well in arrears of the manual example (1.04) should have indicated that all was not well.

The real McCoy

With a working version of Easel up on the screen things are remarkably straightforward. In keeping with the likes of Quill, Archive and Abacus, function keys 1 to 3 are set to perform standard operations, with the two remaining ones taking care of simple screen editing.

For the initiated the functions are:

- F1 Help menu**
- F2 Increase display area by removing prompts**
- F3 Enter a Command**
- F4 Delete a cell**
- F5 Insert a cell**

Standardising key functions (wherever possible) is more common in the business market than on home computers and is highly laudable. It lends coherency to the group of programs as a whole and saves having to learn new techniques for every different application. Full marks to Psion.

Again to their credit is the way in which tedious menu arrangements have been dispensed with. In some cases users are called upon to select from a host of remote and often cryptic numbered options, in this instance with the screen split into three areas for data entry display and prompts respectively, one is never far from one's subject and there's no wandering off on a tangent (fig) or getting bogged down in subsidiary menus.

Small beginnings

To get started on the Easel all that's necessary is to aim a set of cross-hairs at a desired cell or block on the screen and fire off a value and a bar of appropriate size will fill in. Continue this process and scaling along the horizontal and vertical axis takes place automatically to accommodate any new highs or lows.

As many values as you wish can be entered though correspondingly the widths of the bars become

appreciably smaller to the point where they are some two or three dots thick. In addition the greater the number of figures the longer it takes to draw a graph. It's all reminiscent of a self-focusing camera with displays redrawn in the most suitable perspective. The process is visually striking and simple.

Having input a series of values and constructed a simple bar chart a skeleton framework of data and the bare bones of a display are there. There is now the opportunity to extend the former or dress up the latter. New sets of data may be added or an existing set displayed in one of eight different formats. Either way the route is via a series of commands (F3).

A designated command is executed by simply entering the first letter of the prompt that references it. For example, type 's' as in 'save' or 'l' as in 'load'. Where commands require more than one item of information for input, further prompts will be displayed. This single step entry procedure is virtually idiot proof.

If a default case can be offered the program will supply one. If an entry is invalid, it will inform, whereupon the user can either re-input or 'ESC'. Best of all when you are required to make a choice entry of a '?' will display all the options. Nothing could be simpler, in fact you begin to wonder why a manual is supplied at all!

Datasets

Selecting the command 'newdata' enables further sets of figures to be

entered. These may be named and thereafter manipulated as a whole and this feature makes the Easel a mini spreadsheet in itself for it permits number-crunching as well as charting. Bearing in mind that Easel files can be exported across to both Archive and Abacus, it could well prove to be the most enjoyable as well as illustrative manner in which to enter figures intended for use by all three.

Easel permits six geometric and five arithmetic functions for data manipulation as well as basic arithmetic operations. Functions and operators working upon named datasets can be combined (as well as nested within parenthesis) to make up complex formulae. Also provided for formulae use are two pseudo-variables 'cell' and 'cell-max'. The former is used as a store for a cell's relative position on the screen while the latter's value is the total number of cells displayed.

The inclusion of pseudo-variables combined with a comprehensive range of functions make Easel a particularly sophisticated analytical tool

Tailor-made

When it comes to dressing up whatever figures have been put on the Easel, there is no shortage of disguises. In terms of graphical representations or formats Easel offers three basic types: line graphs, bar and pie charts, with bar charts either horizontal or vertical and where more than one set of figures is being examined bars may be stacked, or placed side by side within a particular cell.

Just as format can be varied so too can display. The background ('graph-paper') can be any one of a variety of colours with eight colours in 40 column mode while in 64 or 80 column mode the choice is red, green, white or black. In addition a grid may be superimposed to improve readability.

Bars can have borders of varying width and contrasting colour and lines may be point plotted or solid, differing in width and colour. Axes, labels and titles may be altered and relocated. And for those not content with the selections offered, in most cases personal display types can be created. Only the three basic formats themselves are invariable.

By far the most powerful display feature is the 'view' command which permits a variety of datasets to be drawn on a single display

providing an ideal means for analysing relationships between them. Choice of a dataset's particular format is of considerable importance for whilst Easel would appear to support any number of bars or lines on a single screen, the constraint of space and number of colours favours a judicious mix of the two. (note: no more than one dataset may be displayed using the pie chart format).

A nice finishing touch to a display meanwhile is the 'highlight' feature which allows emphasis of a particular value by altering the bar structure in the relevant cell, when applied to a pie chart this effect is similar to cutting a slice out of a cake.

Fraught filing

Easel allows for saving and loading of datasets and the process is reasonably painless. The dataset currently being worked on can be saved under any name chosen which in effect means that to save more than one dataset 'olddata' commands must be interspersed between saving or saviour prior to moving on to enter with 'newdata'.

Theoretically any dataset saved can be reloaded back onto Easel at a later stage but the problem on our version (1.01) was that loading a file back onto Easel obliterated every other dataset on it! As only one dataset per file can be saved the most that can ever be reconstituted from a previous session is one set of figures any attempt to load more than one file will result in the Easel storing only the figures contained in the last one loaded. This makes the save and load feature useless for all but the most primitive of operations.

A way round the shortcomings of Easel data files is to save multiple datasets on an export file which when created saves all the figures including any labels that might be attached to cells. If this file is reimported it should be that all datasets will be recovered.

Invisible exports

Exporting data from Easel to Abacus or Archive as well as importing from the other two is by no means simple. The manual devotes some 24 pages to what is easy and simple but when it comes to the least straightforward of operations, little more than a page and a half is set aside.

As it transpires exporting from Easel has the same defects as saving/loading an Easel file in that only a single dataset can be pushed across, at least this was our experi-

ence when exporting more than a single dataset across to Abacus. This elicited an 'error in format' and to Archive, no error but no additional dataset either.

The manual does state that when exporting, certain rules governing file format must be adhered to. These relate to the placement of text within a file, yet with Easel automatically attaching default labels (month names) to every new dataset created, a palatable format for transfer was found to be near impossible. This may have been an idiosyncracy of the particular version reviewed or an operator error - judgement is reserved! It remains true however that exporting from Easel is uncharacteristically complex.

Importing data onto Easel turned out to be considerably easier. Provided any text is reserved exclusively to distinguish one dataset from another, few problems are encountered with either Abacus and Archive though the former provided the most trouble free transfer.

In Abacus, reserving the first column for a dataset name with all succeeding ones containing figures no problems ensued transferring in excess of ten separate datasets in one operation. To illustrate export problems however, the addition of another dataset this time in Easel with the export file overwritten, rendered it useless for a reverse transfer.

In many ways because importing data is considerably more effective than exporting it would seem preferable to use Archive or Abacus as primary sources of information. In a sense this would appear to relegate Easel to a secondary position.

Memory

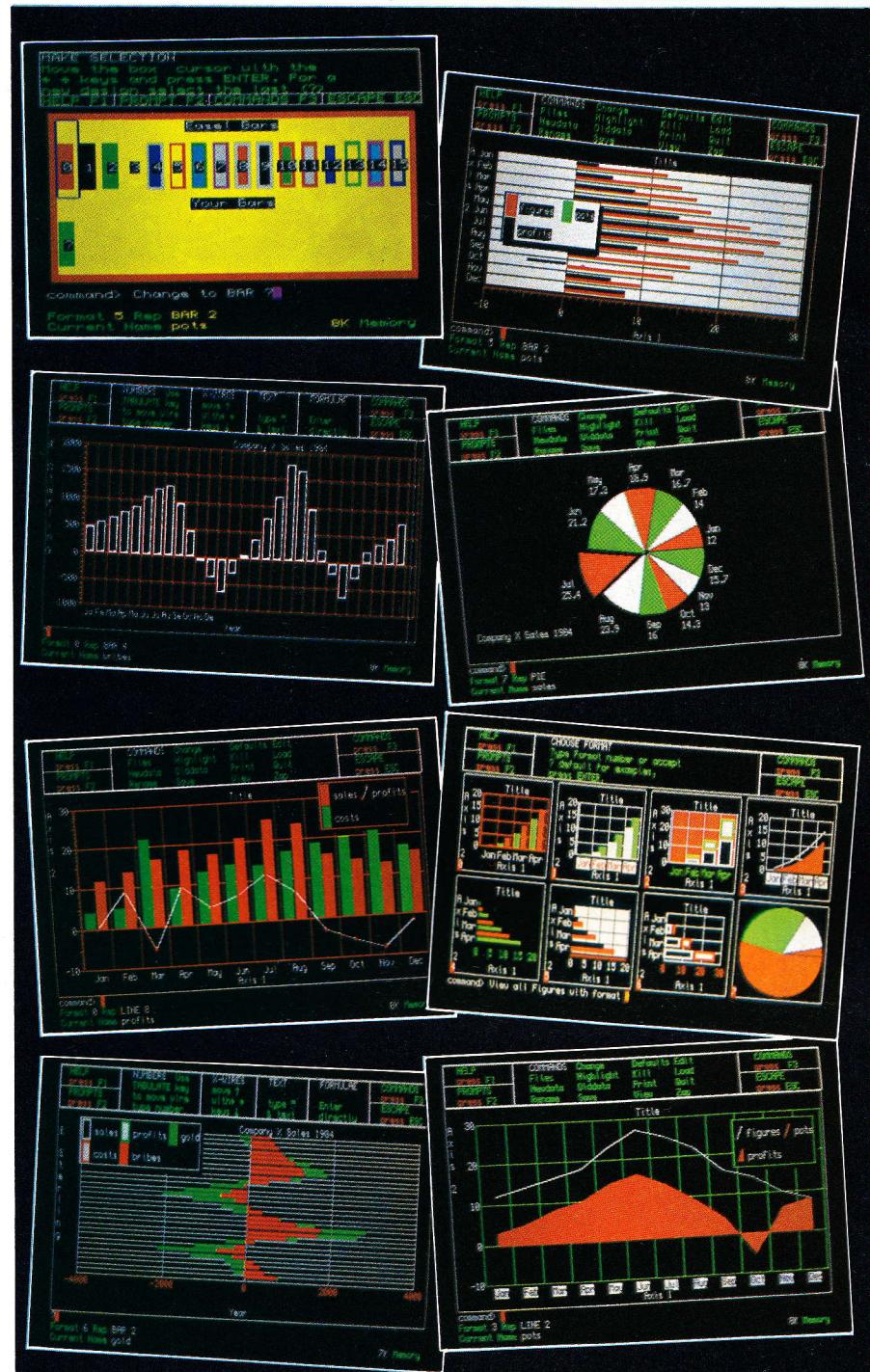
Easel sets aside 8K of memory for storing figures which roughly allows for 40 different datasets containing 12 figures each or some six sets containing 150 figures, which is more than sufficient for even the most exacting sessions.

Free memory is displayed at the bottom right of the input display in 1K units. Exceeding memory capacity is not fatal since the point at which full capacity has been reached is given, leaving users free to make whatever amendments are necessary. Positive and negative figures are displayed to seven decimal places and problems of significant places encountered in SUPERBASIC appear not to apply.

Hard copy

Easel allows for any graph or chart

EASEL



By virtue of a powerful 'view' command Easel permits one to display a number of different sets of figures on a screen in a variety of differing formats. Bars, lines and curves can all be individually varied as can background colour and grid outline. Aside from pie charts any number of combinations are possible.

currently on display to be reproduced on paper using a dot-matrix printer. Using an Epson RX80/FT printer with an RS232 interface some impressive hard copy was obtained.

The manual indicates that printer type selection is an integral part of the print command. This is not the case, but it was found that using the Easel after a session on Quill (where lines had been double spaced) the result was graphs printed in similar fashion. This indicates that printer information is carried over from one program to

another. A blessing and a curse; it does make up for the absence of a resident printer-driver program while proving both time consuming and circuitous in requiring other programs to be run.

In addition to dumping a display to the printer dumping can be to a microdrive file but as the user cannot reload a screen dump via Easel, this is only of any use to those who cannot even indirectly configure to their particular printer. Dumping to tape does mean that you could conceivably construct your own driver in Super-

Basic however.

Both printing and saving to file are lengthy operations as the entire content of screen memory must be transmitted bit by bit to the relevant device. At least a minute per display should be allowed.

Interestingly the manual also contains information as to optimum exposure setting, film speed and aperture for those wishing to photograph displays.

Manual

For the most part the section devoted to Easel in the QL manual is short and sweet – as it should be since most operations are easy. Coverage is more than adequate except for file operations which are poorly documented and in some cases explanations are misleading. For example, the descriptions regarding saving/loading a file give the impression that the feature is more powerful than it is.

Conclusion

Deceptively named, Easel is not the tool of an artist but of a businessman being both cosmetic and functional. By itself it provides a useful analytical tool providing graphic insights into relationships between differing sets of figures. As part of an overall package it is the icing on Psion's cake, by far the easiest to use and some remarkable effects may be achieved with a minimum of effort. Ideally it should be used to add the finishing touches to a report or forecast generated with the help of Quill or Abacus.

If there is any criticism it is that discrepancies in filing operations tend to undermine the concept of an integrated business package. Whilst it is true that Easel meshes with Archive and Abacus, transmission is by no means automatic and the process is uneasy, involving some grinding of gears. Whether later versions (1.02 onwards) will solve these shortcomings remains to be seen but Psion would have done better to have omitted certain unrealised features rather than create false expectations.

All in all Easel is an extraordinarily comprehensive business graphics package the like of which has yet to be seen on any computer remotely near the same price. The fact that it is part of a suite of four included in the cost of a QL cannot but increase its appeal especially when one considers that Easel's MS-DOS version is projected to sell at £175.

by **Recommended**
QL-User magazine

**At Last! THE SERIES THAT MAKES THE
QUANTUM LEAP**

THE SINCLAIR QL SERIES

"I am certain that these books will add enormously to the enjoyment and practical use which QL users will get from their computers."

NIGEL SEARLE

Managing Director, Sinclair Research Limited

5 books to help you get the most from the QL, whether you are still waiting or are lucky enough already to have yours.

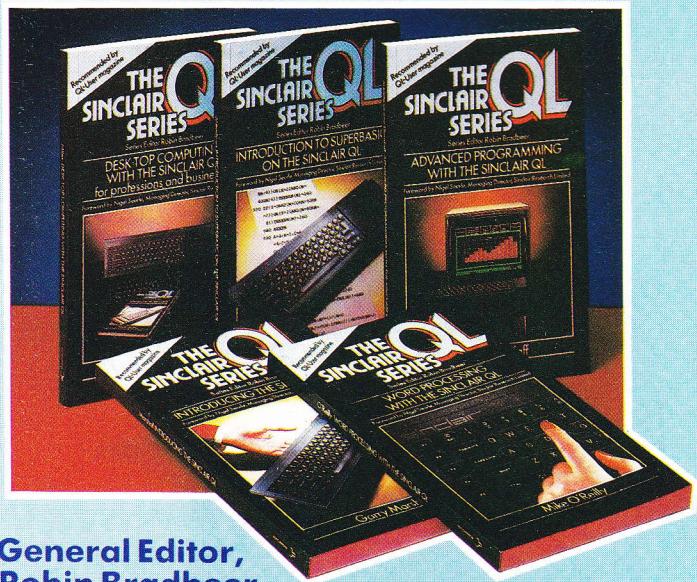
Introducing the Sinclair QL explains how the QL works and what you can do with it.

Introduction to SuperBASIC on the QL

Advanced Programming with the Sinclair QL is an essential reference work for users who really want to get to grips with the Sinclair QL. The book includes such topics as program logic representation, types of commercial program, programming techniques and document design.

Desk-top Computing with the Sinclair QL shows just what can be achieved in business computing using the Sinclair QL and how to get the best out of the four QL software packages: word processing, spreadsheets, database management and business graphics.

Word Processing with the Sinclair QL has been written to explain both the concepts behind the uses of word processing and how the QL word processing package operates and what it can do.



**General Editor,
Robin Bradbeer**
and his team of authors received the
help and co-operation from both Sinclair
and Psion Software – creators of the
QL software – to ensure that these
books really are the ultimate
handbooks for QL users.

Order now – you won't have to wait long for your books! Just fill in the order form and return it to us with your cheque or money order. You can even charge your Access or Barclaycard account. We will send you your books POST FREE within 28 days.

ORDER FORM

To: TBS, 38 Hockerill Street,
Bishop's Stortford, Hertfordshire.

QL

Please send me:

..... copies of **Introducing the Sinclair QL** @ £6.95 each

..... copies of **Introduction to SuperBASIC on the QL** @ £6.95 each

..... copies of **Advanced Programming with the Sinclair QL** @ £6.95 each

..... copies of **Desk-top Computing with the Sinclair QL** (£6.95 ea)

..... copies of **Word Processing with the Sinclair QL** @ £6.95 each

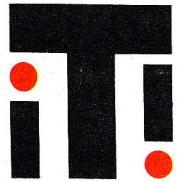
I enclose my cheque/money order made payable to TBS for £ _____

Please charge my Barclaycard/Access account (delete as appropriate) number — 1234 5678 9012 3456

My name and address is

Dealer enquiries to:
Doug Fox, Hutchins

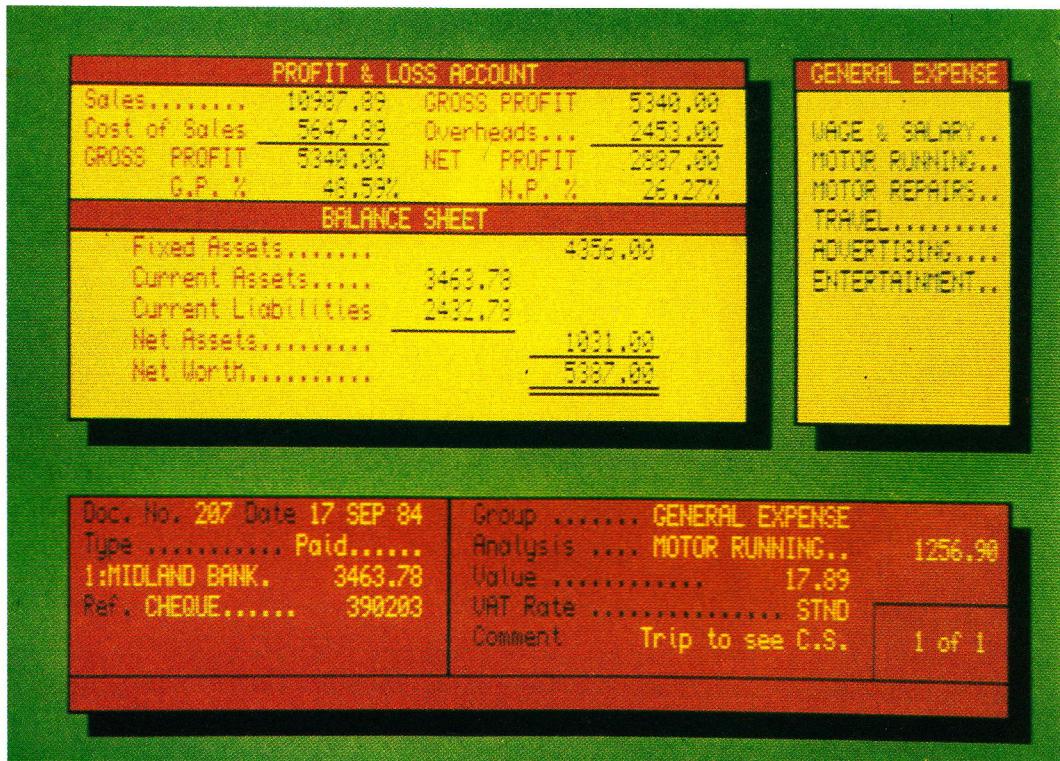




CASH TRADER

**FINANCIAL WINDOWS ON YOUR BUSINESS. AN ACCOUNTING PROGRAM
DESIGNED FOR SMALL BUSINESS USERS OF THE SINCLAIR QL**

- ★ Designed for businessmen, not accountants, typists or programmers
- ★ Entries can be made in any order, the computer does the sorting
- ★ Updated Profit & Loss A/C displayed as each entry is completed
- ★ 18 other Management Reports can be displayed in the Reporting Window these are available even in the middle of making an entry



- ★ Other Management Reports include VAT summary, Bank Account balances, Sales & Overhead analysis, amounts due
- ★ Comprehensive training exercises and 200 page operating manual provide key by key instructions plus tutorial
- ★ Many 'add-on modules' shortly available to expand the system's facilities including Invoicing, Tax etc.
- ★ The Extracted Reports module allows extremely powerful reports to be designed to suit individual user's needs
- ★ Accounting logic and reports approved by Accountants, and the VAT module checked by the Customs & Excise
- ★ User support club provides periodic newsletters with further instructions, program updates, operating help etc.
- ★ 140 user defined Analysis heads supported, including 10 for Bank Accounts and 20 for Credit accounts

ACCOUNTING SOFTWARE LTD

26 Fore Street, St. Marychurch, Torquay, Devon. Tel: 0803 313454

Please supply:			
Qty	Module	Each	Total
	ITI CASH TRADER with VAT.....	£75.00	
	Extracted Reports module	£50.00	
	Support Club.....	£25.00	
Total		£	
Plus VAT @ 15%		£	
Total (we'll pay postage)		£	

I enclose my cheque/PO/Cash for £.....
 Ref EM/09
 Please debit my Access Card No.....

Signature.....

Name.....

Address.....

If you require a VAT invoice, please tick
 Please supply further details only

Please allow up to 28 days for delivery. Sinclair and QL are registered Trade Marks of Sinclair Research Ltd

Bugs Out

*Here is the machine we have
been waiting for – a
completed version of the QL.
When you buy a QL in the
shops what can you expect?*

Some nine months after launch it seems the QL has reached maturity. The dongle has disappeared, the operating system has had most of its bugs removed and the micro-drives no longer seem to have a will all of their own. At long last we can dispense with documenting unfulfilled promises and move on to assess the likely impact of this (hopefully) finished product.

Keyboard & Display

Having removed the QL from its polystyrene, plugged in the external power supply and connected up to a TV or monitor, the first thing you will have to come to grips with is its keyboard.

The 65 key QWERTY layout is a pleasure to behold but much less satisfying to use. Keys are set flat on one plane making the three skimpy plastic feet that tilt the machine at an angle an absolute must for any kind of reasonable access to most of them.

The keyboard mechanism is of the calculator-style membrane variety which should be familiar to Sinclair users. Whilst it is an improvement on previous 'dead-flesh' versions with full travel on the keys, touch typists will find that their skills are lost in a 'quicksand' of missed contacts.

For one or two-finger typists however, the keyboard is quite adequate especially when considering the speed at which Quill operates. Furthermore the plastic membrane underneath serves to prevent dust and ash from getting into the machine's innards which is some consolation.

The QL's video display similar to its keyboard again is quite adequate offering resolutions of 512 × 256 pixels (80 × 25 text) with four colours and 256 × 256 pixels (40 × 24) with eight colours.

The clarity of the TV display seems to vary from make to make, but, none are capable of handling the 80 column display and even the

compromise 64 column display offered by the QL proves to be a strain after a few hours' word processing. Serious users should consider buying a good quality monitor.

Contrary to the accepted industry standard (11.5 microseconds) the QL requires a monitor with a scan flyback of less than 10 microseconds. This means the QL will require a dedicated monitor to avoid overscanning on either edge of the screen. This may come as a bit of a blow to anyone who has already invested in a conventional model for another system.

Data Storage

One of the features of the QL is that it comes with two on-board microdrives. Intended as a halfway house between the hopelessly inefficient and unwieldy cassette-based systems and the more expensive but considerably more efficient disk systems, these microdrives more than fulfil their function.

Anybody who has used cassettes to store data or load programs will realise what an amazing improvement these drives are. Storage and retrieval takes a matter of seconds as opposed to minutes and the media itself is a delight, since five or six cartridges can be carried in a pocket without spoiling the line of a suit or dress!

The cartridges store some 100K and are said to be unreliable, but with some 500K on the review machine we have yet to find a corrupted file. The real problem with the drives is the cost of the cartridges – £4.95 – too expensive for what is supposed to be a cheap alternative to a disk drive. Furthermore, their cost and the fact as yet Sinclair is the sole supplier means that commercial software in microdrive form will probably be quite costly. The fact that the QL does not include a cassette interface aggravates the situation and could lead to a potential software famine.

Bearing in mind the QL's cost,

any comparison to disk is to some extent, academic, but with slave processors looking after the drives, 128K (with a very unlikely further 500K) available and a multi-tasking potential, their operation could conceivably be made virtually transparent. The consequent time saving puts them in contention with some of the cheaper disk systems, in particular those offering only sequential access.

In operation, one niggling problem with the drives remains: if used over an extended period of time say six to eight hours, they tend to seize up. Nothing may be saved or loaded and the machine must be disconnected and given time to cool down. Clearly this is undesirable especially in the middle of a lengthy word processing session where all unsaved text will be irretrievably lost, indicating that being intended for business use the QL is not designed for extended usage.

The nature of the expansion ports at the machine's rear however do little to support the notion that it is aimed primarily at a business audience. These include two serial and two joystick ports which use Telecom-style connectors the likes of which have yet to be seen on any other computer.

Of the two serial sockets both are RS-232-C with one wired as a modem. The problem appears that baud rates cannot be set independently for each channel which causes considerable problems if, for example, the QL is to be used to drive a modem and a printer where data is likely to be transmitted at differing rates.

The two network connectors are in fact thinly disguised serial links. There is little firmware within the machine to effectively control and monitor data transfer but, the possibilities do exist bearing in mind the 68008 capabilities although only for expert assembly programmers.

In addition QL supports a ROM/Cartridge socket at the rear which was initially used to house the



'dongle' containing the earliest operating systems. Also, hidden behind a black plastic shielding, is a 24 pin edge connector likely to act as the machine's major expansion port. Originally intended to house a half-megabyte RAM extension this is more likely to be used as a disk drive connector by enterprising independents.

SuperBasic

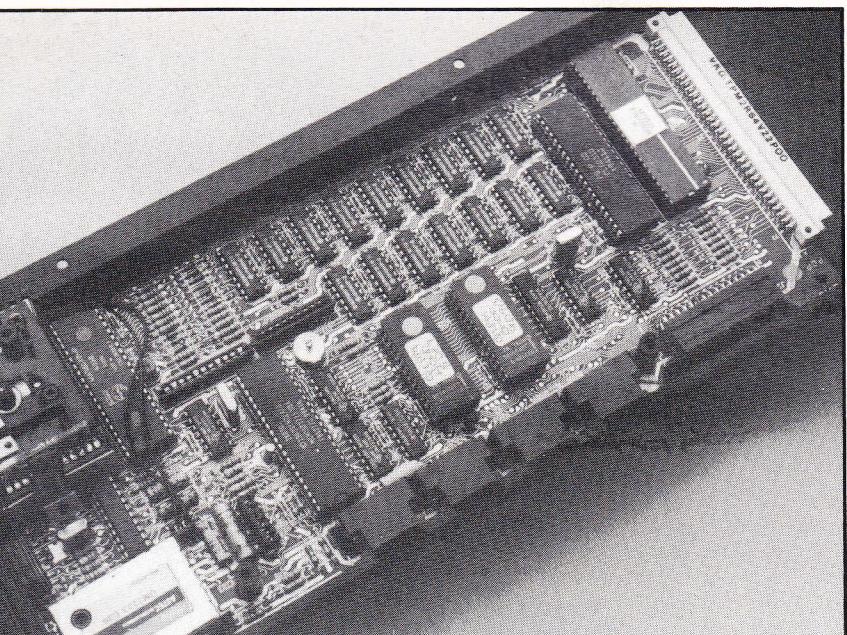
SUPERBASIC, is powerful and extensive offering a wide range of control structures and affording a fair degree of control via QDOS of the machine's various functions. Despite all this, however, the QL's BASIC appears to be a tool without a definite application. For business users it affords only seven digit precision, which means that you cannot display figures in excess of 99,999.99 without rounding.

For the games programmer, it fails to support user-definable characters and sprites. Windowing features are little more than a colourful screen handling gimmick as objects PANed or SCROLLED out cannot be recovered but must be redrawn. Sound features can only be described as experimental.

For the beginner meanwhile, the language is hardly ideal, its syntax is idiosyncratic characterised by long and cumbersome commands and ambiguous error messages. It is also slow!

The Software

Shortcomings in the QL's SUPERBASIC indicate that the machine was never intended to run serious applications in that language. Psion's packages are an ideal example - these could not have been written in BASIC but rather mesh with QDOS on a much lower level and it is here that the QL has enormous potential.



PCB: No empty slots with the promise of better things to come!

The packages have been described as 'the plastic toy in the cornflake packet'. This is hardly the case. Abacus, Easel, Quill and Archive provide the most convincing argument for the QL. A machine capable of running one, to say nothing of four extremely sophisticated packages (word processing, database, spreadsheet and business graphics) at a cost of £399 inclusive is revolutionary. The QL has effectively opened the door for anyone wishing to enter the field of serious computing. The fact that these packages are now being made available on machines costing at least twice the price is a more than adequate testimony that in this area the QL has succeeded.

The Hardware

Much has been said of the Motorola 68008 and supporting slave processor contained within the machine. If the sheer brute processing force

that they contain could be harnessed then the QL would leave its eight-bit competition standing.

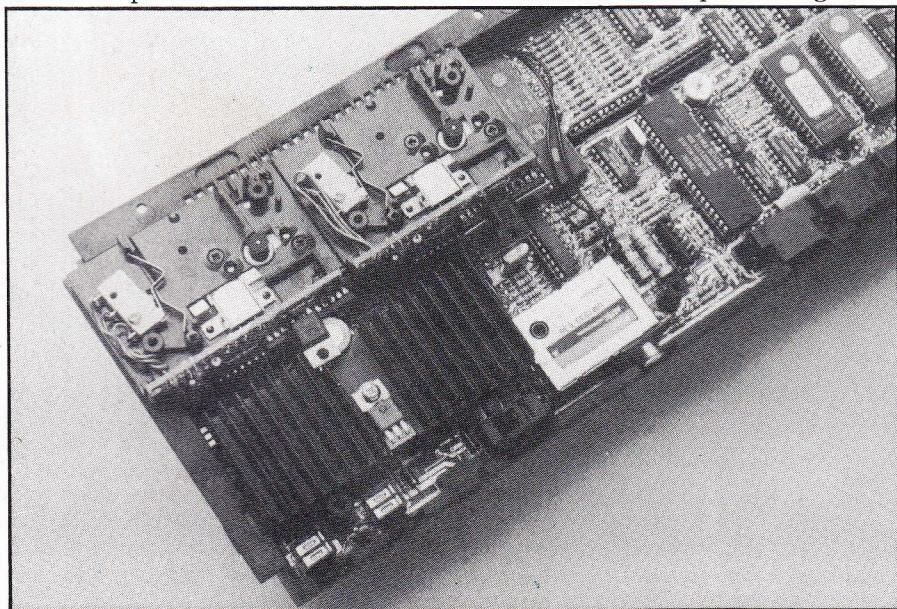
The indications are that software houses are beginning to stir to the QL's potential and already assemblers and alternative operating systems are appearing. More interestingly, these are coming from machines where multi-tasking and true windowing capabilities are a reality. The principal problem encountered is not shortcomings in QDOS but in being able to cram everything into 125K plus of RAM.

In Conclusion . . .

In an attempt to give the punter on the streets the kind of computing facilities more commonly found in the office certain compromises have had to be made namely in its microdrives, SUPERBASIC and keyboard. What is surprising, however, is that in these areas the QL performs more than adequately.

Indeed, any shortcomings can easily be forgiven for the simple fact that the QL offers so much more than any similarly priced machine and has the potential for very much more.

The history of the computer industry has been coloured by constant change. The minicomputers of the past are the micros of today and if any distinction remains between the two it is one of price. The QL is the first to gnaw at the distinction between home computer and office machine. The process is by no means complete, but the battle of the bits is effectively over. Whether the campaign will last five years or two is now in the hands of the software houses and not the manufacturers. The QL won't stabilise the market but it will benefit the consumer.



Despite a giant heat sink the drives tend to overheat after prolonged usage.

SCI UK

7 DAYS
A WEEK

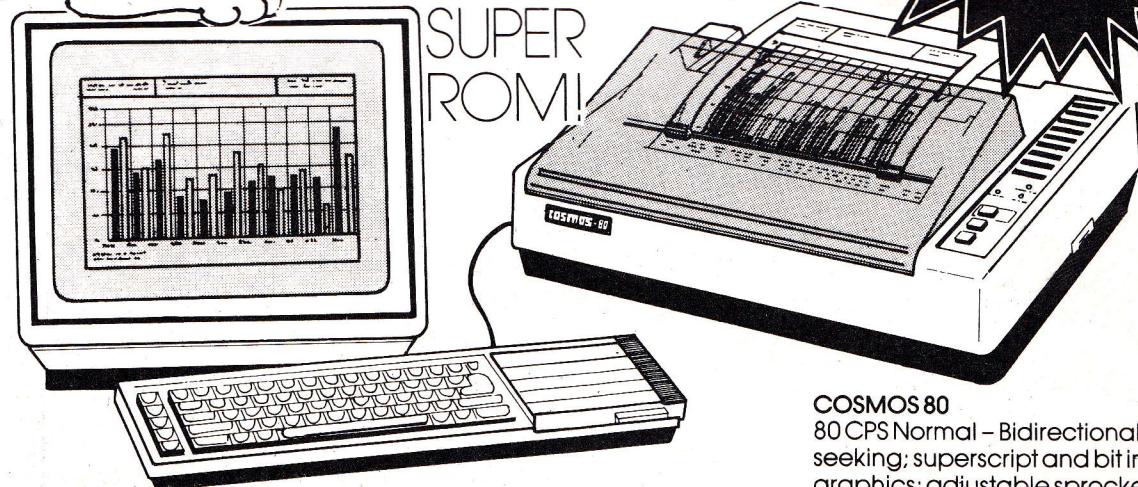
- * SUNDAY OPENING
- * 24 HOUR DELIVERY SERVICE
- * COMPREHENSIVE TECHNICAL SUPPORT



**SCI Cracks the
QL problem!
call 0730 68521**

PLUG IN AND GO! NO INTERFACE PROBLEMS

SERIAL CABLE
SERIAL INTERFACE
COSMOS 80
NOW WITH SCI(UK)
SUPER
ROM!



**£219
+ VAT**

OTHER SUPERB OPTIONS

with EPSON RX80FT

£319 + VAT

with EPSON FX80

£399 + VAT

with CANON PW 1080A

£379 + VAT

All with dipswitches set for you by our experienced staff



24 hour nationwide delivery by Securicor £9.50 + VAT.
Bankers orders; Building Society Cheques; Postal Orders;
same day despatch. All orders covered by the Mail Order Protection Scheme.



MANY MORE PRINTERS
AVAILABLE. 1,000s
OF BARGAINS —
SEND NOW FOR
THE FAMOUS
SCI(UK)
CATALOGUE

FAST Mail Order
SCI(UK) Unit 16, Inmans Lane, Sheet,
Petersfield, Hants GU32 2AN
Freepost Mail Order
SCI(UK) Freepost, Petersfield,
Hants GU32 2BR,
Tel. 0730 68521,
Telex 88626 MYNEWS G

**Demonstrations
every hour!**

**Remember! These prices
are for complete packages
— no more to pay!**

URGENT ORDER
SCI(UK) FREEPST.
PETERSFIELD, HANTS GU32 2BR.
PLEASE RUSH ME
Name _____
Address _____
Credit Card Number _____

Inside Story

Paolo Baccanello pulls the lid off and peers inside

Weighing in at just over 3lb (excluding 2lb power supply), enclosed in an elegant black plastic shell measuring some 18" by 5", the QL hardly comes across as the heavyweight that it is cracked up to be.

Remove the ten screws that hold the casing together and the picture changes. To your right, modulator, microdrives and accompanying heat sink occupy all available space. Centre and left lies a packed PCB with the keyboard mechanism stacked on top.

Motorola 68008

On the far left of the PCB lies the Motorola 68008 Microprocessor. This is the driving force behind the QL. Making a break with the usual choice of '8 bit' CPU's (Z80 and 6502) Sinclair have gone directly to 3rd generation processors leapfrogging the plain 16 bit variety on their way.

The 68008 is one of a family of sophisticated '16/32 bit' processors. Introduced in 1979, these were designed with the programmer in mind. The series offers 16 32-bit general purpose registers, a 32-bit program counter, and a 16-bit status register. Five basic data types are supported, these are bits, BCD Digits (4-bits), bytes (8-bits), words (16-bits) and long words (32-bits). In addition there are 14 flexible addressing modes which include the capability to postincrement, predecrement, offset and index. Finally, the processor supports some 56 different instruction types of which several themselves include variations.

The CPU's highly regular structure makes assembly language programming both easier and more reliable. Operations on registers and memory are independent of the data. Separate special instructions that operate on byte, word and long word are not necessary. Where previously tens of operations were required, here a single operation will, in most cases, suffice. As a corollary, compilers on the 68000 are efficient and indeed, operating systems such as QDOS and Unix are for the most, written on such.

The 68000 also includes numerous features to aid in error correction. For example there are built-in hardware traps to detect certain common error conditions. There is even a TRACE facility that enables one to step through a program instruction-by-instruction.

Where the 68008 differs from its brothers the 68000 and 68010 is

that its external databus is only 8 bits wide instead of 16. This slows the operation of the CPU because loading and storing of registers is done by halves. However, this has meant that the cost of memory chips has been kept down and even at reduced speed the 68008 is considerably faster than its competitors.

2nd Microprocessor

Because the 68008 works on interrupts making timing near impossible using the CPU's internal clock a further microprocessor has been incorporated. An Intel 8049, this is located on the far right on the PCB and controls keyboard, sound and RS232 receiver and leaves the main CPU free to run user programs.

The effect of the 2nd processor is quite noticeable if, for example, you type ahead while using the Quill. The slave processor continuously scans keyboard entry putting characters into a queue. Consequently nothing typed-in is ever lost.

QDOS

The QL's resident operating system began life on a collection of three 16K EPROMS (27128's). Two were on the PCB and a third plugged into the machine's ROM socket. Now, whittled down from 48K to 32K, final versions of QDOS (JM) occupy two masked ROM's to the rear of the PCB. Owners of earlier bug ridden systems (ie. versions FB, PM and KB) may send in their QL's to get an upgrade installed. To find out which version you have type PRINT VER\$.

128K RAM and expansion ports

The QL comes with 128K of RAM which is contained within 16 identical chips occupying two rows at the front of the PCB. Of the 128K, 32K is devoted to the screen bit map.

As the 68008 is capable of addressing a maximum of 1 megabyte of memory, RAM extensions to make up this total should eventually be available. These along with such peripherals as disk drives will make use of the 64-way

expansion port on the far left of the machine.

At the back of the QL a variety of ports have been incorporated. Moving left to right the first of these is a ROM cartridge socket providing up to 32K extension of ROM.

Next come two joystick ports. These use two BT 600 sockets which will probably be used with either games or to control the cursor (ie. with some sort of mouse device).

After the joystick ports come two RS232-C ports. One is wired for use with a modem (DCE) and the other, to drive a printer (DTE). Both are capable of full duplex transmit/receive at seven rates up to 9600 baud.

Unfortunately, as there is no provision to set differing baud rates for either port it is unlikely that users will be able to run both devices simultaneously.

Next comes a standard UHF socket and beyond that there is an 8 pin DIN type RGB socket. The latter supports displays of up to 85 columns and makes full use of the QL's 512 by 256 pixels resolution.

Finally, on the far right hand side are two network sockets. These are supposed to allow up to 64 QL's or Spectrums to be linked up. However, no network driver is as yet incorporated in QDOS so that these ports are little more than poorly implemented RS232's operating at 100K baud.

Microdrives

Without a cassette interface the QL is wholly dependent upon microdrives. Two of these are built into the QL. A further six may be connected in tandem via a concealed port on the far right.

For their storage medium the drives use a compact cartridge measuring some 450×350×5mm containing 200ft of tape in a continuous loop. Theoretically, this permits a storage capacity of some 255 sectors containing 512 bytes per sector. In practice, storage capacity varies from a minimum of 80K to a maximum of 100K.

The drives load at a rate of 15K bytes per second and take on average 3.5 seconds to locate data.

Sound

A Piezo - Electric Speaker is concealed under the microdrives. Sound quality, in many respects is similar to that found on the Spectrum and falls well short of that common to most home computers.

COLOUR MONITORS

MICROVITEC

1431 — 14" STANDARD RES. (BBC etc.)

1451 — 14" MEDIUM RES. (BBC etc.)

1431MZ — 14" STANDARD RES (SPECTRUM/BBC)

1431MQ/DQ — 14" MEDIUM RES (Q.L.)

PRICE INCL. VAT.

£215

£330

£255

£275

ADD £10 FOR DELIVERY



SUPERB MONITORS FOR Q.L. AND SPECTRUM

(1451MQ/DQ)

(1431 MZ)

MONITOR/PRINTER PACKAGES FOR SINCLAIR Q.L.

Monitor	with Printer	Package Price INCL. VAT
MICROVITEC	— BROTHER HRS	£425
1451MQ/DQ	+ MAINS ADAPTOR	£585
	— EPSON RX80F/T	£715
	— BROTHER HR15	£699
	— HONEYWELL S11 CQ	£725
	— EPSON FX80	£899
	— EPSON FX100	
	★ Epson printers supplied with RS232C interface fitted.	
	★ RS232CI/F board £33.45	
	★ Convert your Epson parallel printer to RS232C.	
NOVEX 12/1800		
MONOCHROME	— Brother HR15	£549

Prices include Monitor and Printer Leads
Add £12 for carriage.

ZEAL MARKETING LIMITED

VANGUARD TRADING ESTATE STORFORTH LANE CHESTERFIELD S40 2TZ
TELEPHONE 0246 208555 TELEX 547697

68K/OS BRINGS POWER TO YOUR QL

The revolutionary 68K/OS operating system for advanced home users, software developers and OEMs is available now as a plug-in card for your Sinclair QL.

Features

- ★ Plug-in card with switch to select 68K/OS or QDOS
- ★ Spare EPROM slots for applications software
- ★ Real-time multitasked software with multiple screen windows
- ★ Much faster microdrive access times
- ★ Utility software on microdrive including an editor and file maintenance programs
- ★ User manual and applications programmer's manual

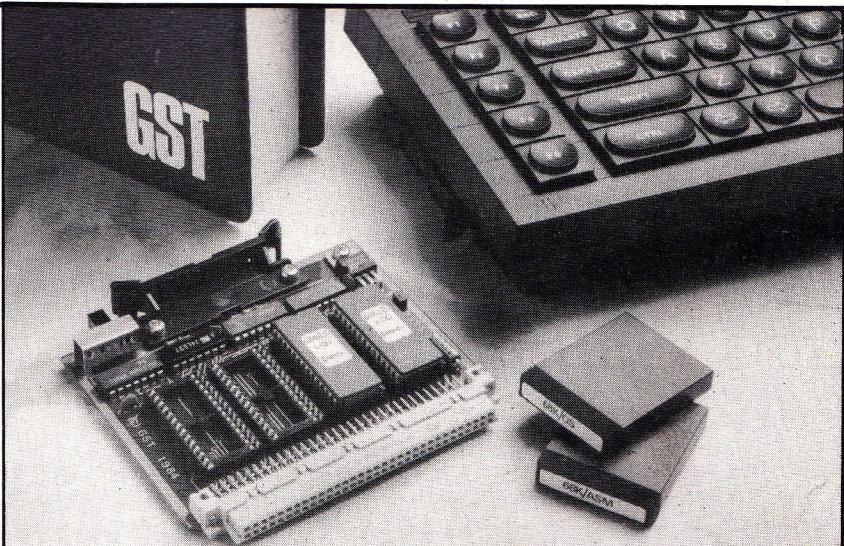
Also available

- ★ 68K/ASM assembler
- ★ Systems programmer's manual

Coming soon

- ★ Full WYSIWYG Word Processor
- ★ Pascal, Fortran IV and C
- ★ Disk Support
- ★ and lots more ...

TM68K/OS is a trade mark of GST Computer Systems Limited.
TM Sinclair, QL and Microdrive are trade marks of Sinclair Research Limited.



To: GST Computer Systems Limited
91 High Street, Longstanton, Cambridge CB4 5BS

Quantity	Item	Item price	Total
.....	68K Operating System	£99.95
.....	68K/ASM Assembler	£39.95
.....	Systems Programmer's Manual	£4.95

I enclose a cheque payable to GST Computer Systems Limited for:

£..... Signed

Name

Address

Telephone

Expected delivery within 28 days

GST

An Expensive Alternative?

The alternative to the QL's built-in operating system QDOS, called 68K/OS is now available as a £99.95 package which turns out to be a premium price for a product whose viability must be under question.

Why pay £100 on top of the QL price for an operating system that turns out to be merely an alternative as opposed to a 'go-for-it' replacement?

The suppliers, GST of Cambridge had originally planned that K/OS would become a de-facto standard operating system for Motorola 68K systems, and Motorola's support goes some way to ensuring this, but only for the OEM market.

There can be only one reason for a QL owner (version AH and above!) purchasing K/OS – to write applications and possibly systems programs for the potentially large K/OS market. To buy because it seems preferable to QDOS turns out to be a false investment.

The system supplied by GST to end users will plug into the QL's expansion port with a switch to select between QDOS and K/OS. The version tested consisted of two 16K EPROMs that had to be swapped for the QDOS EPROMs already in the machine, a trivial process on a 'dongled' QL as later versions require three PCB alterations.

A number of utility programs are supplied on a microdrive cartridge and the package reviewed also included the K/OS version of GST's QL assembler – the one Sinclair is adopting as the official QDOS assembler. Unfortunately!

When a system containing K/OS is booted up a red screen is drawn with five screen layout options. Pressing the requisite function key sets the layout, which is where the first problem arose. All five of the screen modes got the windowing

... That seems to be the verdict on 68K/OS – the QL's 'other' operating system.

single user multitasking environment, just like QDOS, so more than one program can be present (and possibly running) at one time. Whenever a program is loaded it must be derived from something other than the initial ROM setting as this only contains the ADAM program. Instigating another device is known as **mounting** and must be done each time a new microdrive cartridge is put in. This process sets the default pathname to the device specified, and if this supports a directory structure (such as floppy discs and microdrives) then this directory is defaulted too. Once the pathname has been established (on either the program, data section or both) files and the like can be loaded and run simply by specifying their file name.

If a file is to be loaded from another device then as much of the full **device:directory/filename.extension** path as is required by that device must be used. This is standard and very similar to Unix™.

68K/OS supports all the normal QL devices but tends to use different names: SCR_ is SCREEN_; CON_ is a mixture of SCREEN_ and KEY_; microdrives are MD_ and the serial ports are TXn_ and RXn_ depending whether they are being used for input or output. There is also the PIPE_ device, which exists within QDOS but is not advertised by Sinclair.

Documentation consists of two A4 folders, one describing K/OS to the novice and one describing the system and all its calls to the programmer. The former manual also contains a description of the assembler. These manuals are written in an easy style with the odd bit of flippancy – mainly knocks at QL hardware! The intro-

slightly wrong losing the extreme right hand edge of the display, on both an RGB monitor and a domestic television. No doubt this could be cured with Microvitec's special QL monitor.

After selecting the screen mode the system's **data** and **program** devices are set to **ROM:** and the command program is initiated. This is called **ADAM** and provides a basic interface between the user and the machine. The command program operates in two distinct modes, system and user. In the system mode programs can be suspended, killed, released and alternated between, and OS functions such as window size can be altered.

The user mode is that in which individual programs are run. The system is naturally a time sliced,

ductory manual is a little thin on facts and tends to dedicate more time to K/OS utilities than the operating system.

Being eager little beavers this manual was followed to the letter, which resulted in getting nowhere fast – an attempt to run any program re-booted the machine. The fault was eventually discovered in the date and time program, DATE.PROG, which runs concurrently with anything else the machine is doing. This caused any other program to crash, so the date program was ignored and progress was a little faster.

One gets the feeling that GST is proud of getting the operating system to fit into 32K, but it must be remembered that a substantial number of essential utilities are held on the supplementary microdrive cartridge.

The manual offers a relaxed tutorial which includes the backing up of the system cartridge using the COPY.PROG program. This uses standard wildcard syntax and can have its parameters passed to it on the command line, such as

COPY.PROG *.* BACKUP/

or each parameter can be entered in turn in response to the prompts. Unfortunately errors or misunderstood commands at this stage result simply in a further 'Copy from?' prompt. No indication of the actual error is given and the unfriendly syntax can confuse the novice, but it is at least less verbose than QDOS.

The use of COPY.PROG is invariably preceded by the use of FORMAT.PROG, which formats the microdrive cartridges prior to use. This program is far more impressive as it does not allow the formatting of previously-used cartridges without authorisation, and it reports the state of affairs if it thinks the cartridge contains data saved under ZX Spectrum format. It is unfortunate that it regarded QDOS format as Spectrum format too!

The utilities provided on the microdrive cartridge are:

1) **COPY.PROG:** which copies files from one device to another.

2) **DATE.PROG:** which (when it works) displays the time in the extreme bottom right hand corner of the screen.

3) **DRAW.PROG:** a utility graphics design program designed to show off the capabilities of the QL's operating system. It is rather tedious to use through having to constantly switch between two menus when drawing a figure at a specified position, but it is something that doesn't come with the QL. The QL on the other hand does come with four rather more useful pieces of software.



GST's K/OS in full flow – the startup screen, the starting position just after mounting a drive and one of the utilities in action.

user to select a pre-defined font (of which a few are provided) on a selected channel. Could be interesting.

9) **IOSSMENUPROG:** probably the most interesting program for hackers. It makes all the K/OS calls for input and output directly accessible but screen layout is unfriendly and error messages unhelpful.

10) **MXFILTER.PROG:** converts Motorola S-records to K/OS procedures and programs. Useful if you're serious about program development and use a system that creates Motorola S-records to assemble your source, otherwise a triviality.

11) **PRINT.PROG:** essentially a printer driver for the screen editor, again permanently configured for an Epson FX-80.

12) **RENAME.PROG:** an obvious function.

13) **SLIDES.PROG:** another piece of trivia that displays screens full of information as a sort of slide show. It may be useful for the odd shop-assistant programmer. Pictures displayed are created with DRAW.

14) **T.PROG:** written in Pascal (all the others are in 68000 code) and serves absolutely no useful purpose at all, as GST admit, although considering it to be of great amusement value themselves.

Another microdrive cartridge which costs a further £39.95 contains the assembler and two 'include' files holding K/OS definitions and keys. See the review elsewhere in this issue for views on that.

Briefly the main objection is that K/OS is an alternative to, not a replacement for QDOS. Now prove us wrong!

For the technically minded, QDOS uses a system of 68000 TRAPs to perform most of its functions, register D0 being used to hold the function key and return the result error code. This ensures that certain operations are not interrupted by other programs requiring the same facilities, so that time sensitive programs can be simply implemented.

68K/OS on the other hand follows the time honoured system of vectored addressing to each routine. Each method has its advantages and shortcomings, but TRAPs seem so much easier to write around. Being purely an operating system with no interface to the BASIC language, K/OS also lacks a set of floating point routines, or at least any present do not appear to be available to the user. With the Sinclair system of 6 byte numbers with offset exponents and un-normalised mantissas, this may be an advantage!

CP/M-Something Old Quest-Something New

Poked history

CP/M was conceived in a garden shed in the early seventies. It began as a simple routine to control an intel 8080 linked to an early floppy drive. Within ten years it had become a world wide success, establishing its creator Gary Kildall and the company he founded, Digital Research as frontrunners in the microcomputer industry.

The reason for CP/M's success was that it was cheap and more importantly portable. It provided a means to transport application programs from one machine to another, providing the machines used 8080 compatible chips. As most microcomputers were 8 bit Z80 based (an enhanced version of the 8080) CP/M went from strength to strength.

The early eighties and the introduction of 16 bit microcomputers saw a turn in CP/M's fortunes. Concentrating on developing a multi-user and multi-tasking version of CP/M (Concurrent CP/M), their 16 bit single user version CP/M-86 arrived late on the scene. In the meantime Microsoft's MS-DOS had stolen a march. Adopted by IBM for their PC, Microsoft's system with its superior editing, batch processing and error handling facilities has come to dominate this particular market.

As a result of their experience within the 16 bit market Digital Research have not been slow off the mark to release CP/M-68K. C based it is a complete rewrite of their system for the Motorola 68000. To date, however, it seems to have fared little better than its predecessor CP/M-86 (or CP/M-80 as it is now known). The reason for this is that in the main micros incorporating series 68000 chips have dedicated operating systems ie. QDOS, Apple DOS 3.3) or Unix, a powerful multi-user, multi-tasking system that has dropped down from the minicomputer field.

Here the story ends but only until November this year when Quest International Computer Technology Ltd launch CP/M-68K for the QL. Will this breathe life into Digital Research's somewhat ailing product? We went down to Camberley to find out.

CP/M has been around for a while the QL has not. Are they well matched? Paolo Baccanello investigates.

Question time

With the product still under wraps we spoke with Quest director Ralph Warren. He informed us that the system would retail at £49.50, be sold on microdrive cartridge and occupy no more than 30K of storage. Included in the price would be Digital Research's standard C compiler and a 68000 assembler.

The first question that sprang to mind was why the QL? Bearing in mind CP/M's dominance in the 8 bit field why had Quest opted for the 68000 based machine? Why not an Amstrad or MSX machine? Z80 based surely these would present a safer bet?

The answer was that Quest believed Sinclair was a household name the world-over and that as a consequence its market potential was enormous. Furthermore it was felt that Quest would benefit from the considerable publicity that Sinclair machines seemed to generate of their own accord.

Finally Warren intimated that 8 bits were a 'dead-end' and afforded no scope to move with the times. The possibility of implementing Concurrent CP/M on the QL at a later date as well as the opportunity to move quickly across to the more powerful members of the 68000 family had weighed heavily in deciding their choice.

Why CP/M? . . . Quest informed us that they had considered Unix as the only commercially available alternative but that in this situation 'it was like using a sledge-hammer to crack a nut'. This would seem a fair comment, bearing in mind that Unix requiring considerable disk space is best suited to systems running Winchesters as well as multiple terminals.

Moving on to their implementation of CP/M, Quest were unable to know exactly how fast the package would run, as trade-offs for speed against buffer space had not yet been finalised. However, 'as any product runs only as fast as its

lowest common denominator', for users relying on microdrives, the drives, not the operating system would be the principal determinants of speed.

An encore

As well as CP/M, Quest will be launching 5.5" floppy and hard disk systems running off the QL. These will use the 64-way expansion port currently earmarked by Sinclair for his proposed .5Mbyte RAM expansion. As a bonus, Quest indicated that disks will be formatted so that data files could be carried across to an IBM PC.

So that the addition of disk storage will not preclude RAM expansion, Quest plan a 1 to 4 adaptor for this port. (Incidentally, it would appear that Quest are considering developing their own ULA based RAM upgrades).

As regards the extras that come with CP/M-68K these turned out to be a mixed blessing. In effect both add a new language to the QL's repertoire but neither is as comprehensive as it might be. The assembler, though relocatable and including a linker, is not Macro based. The C compiler is simply adequate lacking some of the more advanced features found in compilers running under Unix (ie. no 'struct'). Both are dependent on CP/M's notoriously inefficient editor.

Soft spot

Moving onto Software support we were informed that Quest were currently involved in discussions with some leading CP/M applications software producers. However, no names were mentioned. It would appear that there seems to be some psychological resistance on the part of the better known houses against being associated with Sinclair products matched only by an appreciation of the potential markets which such products might enjoy.

Certainly, Quest have done everything to draw software houses over to the QL. Their implementation of CP/M is 'bog standard' without any idiosyncrasies or embellishments. In addition, the

RELEASE THE TRUE POWER OF THE QL!

Metacomco's QL Assembler Development Kit provides a high specification assembler and full screen editor for the Sinclair QL.

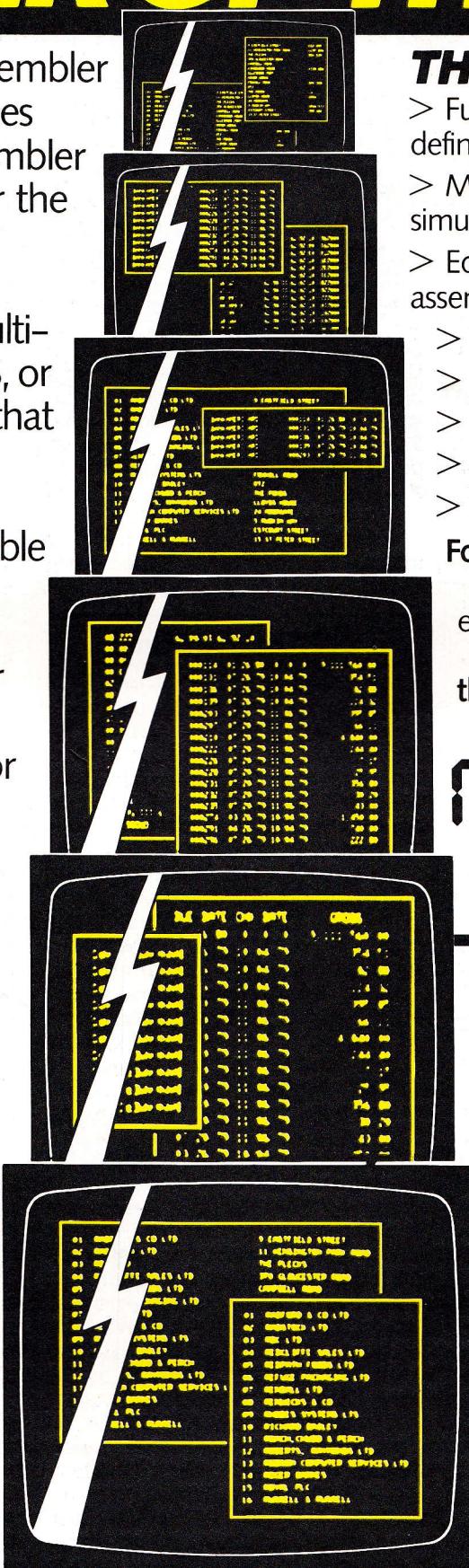
Now you can write programs to use the multi-tasking power of QDOS, or to gain the extra speed that assembler provides.

This sophisticated development kit is available exclusively from Metacomco, a leading producer of software for microcomputers.

Release the power for
only £59.95!

THE ASSEMBLER

- > Standard Motorola 68000 mnemonics.
 - > English error messages.
 - > Produces code which can be EXECed, and run as a concurrent job.
 - > External references allow linkage to high level languages and other assembler modules.
 - > Macro expansions.
 - > Position independent, absolute or relocatable code can be produced.
 - > Conditional assembly.
 - > Large range of directives.
 - > Fully formatted listings.
 - > 30 character variable names and 32-bit values.



THE EDITOR

- > Full screen editor within a user defined window.
 - > Multiple versions can be run simultaneously.
 - > Edits any ASCII file including assembler and SuperBasic source.
 - > Horizontal and vertical scrolling.
 - > Block copy and delete.
 - > Insert file – merges files.
 - > Find and Replace.
 - > Automatic word wrap.

For only £59.95 (incl VAT + P & P).

The kit includes the assembler, editor, and full documentation.

**Can you afford not to release
the true potential of your QL?**

Just clip the coupon or phone.

METACOMCO

26 Portland Square, Bristol BS2 8RZ
Tel: Bristol (0272) 428781
QL, QDOS, and SuperBasic are trade marks of Sinclair Research Ltd.

POST TODAY TO:
Metacomco, 26 Portland Square,
Bristol BS2 8RZ

Please send me ____ copies of the
QL Assembler Development Kit
at £59.95 each. I enclose a cheque
for £ _____ or please debit my
ACCESS/VISA Account No.

Card expiry date _____

SIGNATURE

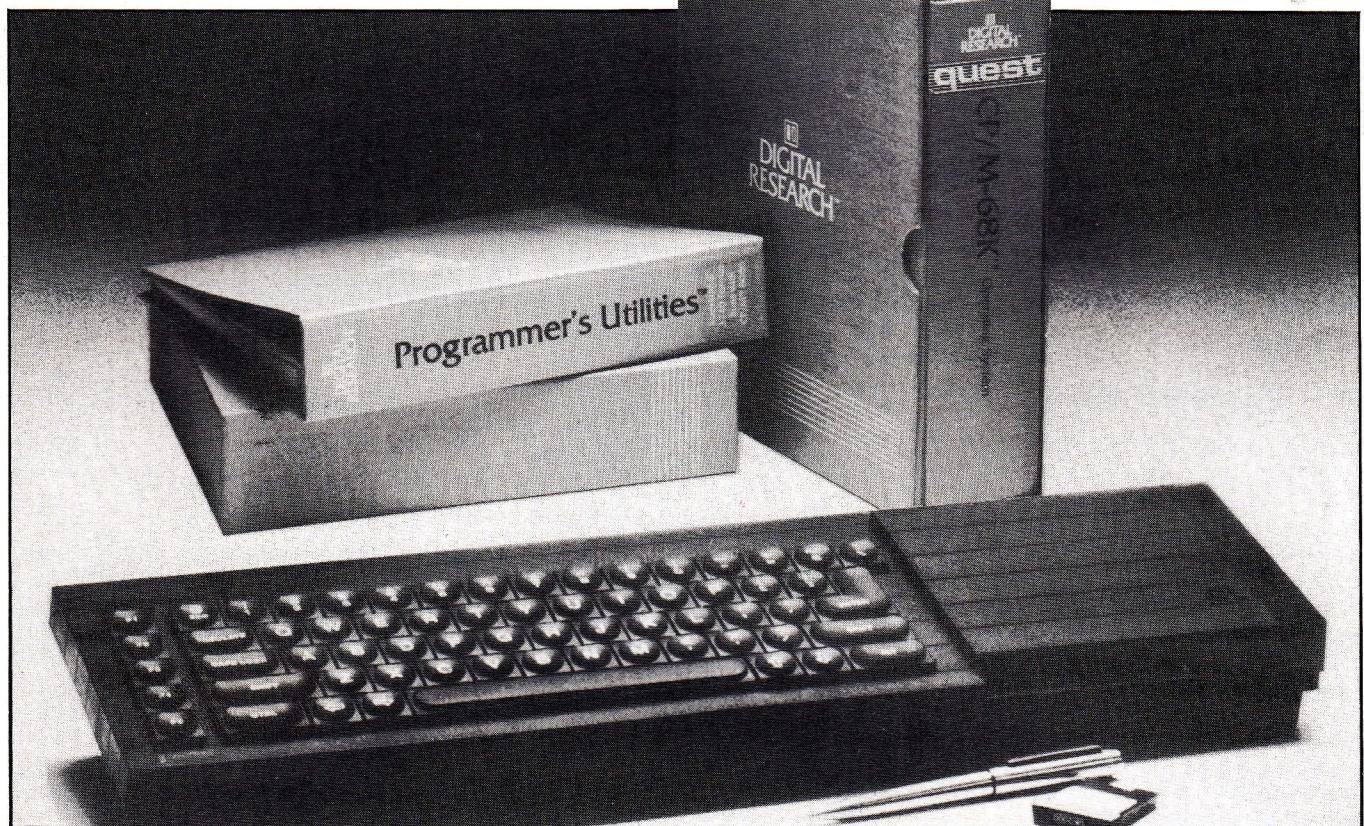
NAME

ADDRESS

POSTCODE

TEL NO

PRICE INCLUDES VAT, POSTAGE AND PACKING U.K.
MAINLAND ONLY. DELIVERY ALLOW 28 DAYS.



parallel development of disc drives and RAM extension for the QL should prove of enormous benefit as many CP/M applications could not possibly work within the 128K RAM and 200K microdrive environment.

A major attraction, however, must be the incorporation of a dongle device with the package - a small PCB to plug into the QL ROM socket. This was initially designed to protect the operating system itself. CP/M will not run without it present. Doubtless, software producers will be permitted to take advantage of this device.

However, even if Quest does manage to entice software houses under the banner of CP/M this will not mean that a huge amount of software will be instantly available. Packages written under CPM-68K are nothing like as plentiful as those available on micros with fewer bits and CP/M portability is wholly chip dependent. Consequently, those whose packages run on other processors will have to reassemble their programs from scratch, a task that will require both time and money.

To ensure that a basic minimum of software is available, Quest are taking over an existing software house (Padmede) and will themselves be marketing an accounting package running under CP/M. It comprises of five standard modules and is projected to sell for a total of £500. To make the package more attractive and taking into account the fact that CP/M does not afford access to the QL graphics, all modules will have facilities to generate data

files which may then be transferred across the Psion's Easel and Abacus.

Thorny issue

When asked about whether CP/M would supplant QDOS Quest maintained that CP/M was simply an alternative not a substitute. The two were intended to co-exist and together afford the user a far greater choice. CP/M would provide sorely missed disk handling capabilities, additional languages and an established software base. QDOS on the other hand could offer graphics and SuperBasic. Quest added that with CP/M resident on microdrive, users would find upgrading simply a matter of exchanging drives. With QDOS, for which Sinclair has extended no explicit upgrade facilities, the process would involve exchanging ROM's.

This view, however, would seem somewhat naive. First, it is rumoured that in the long term Sinclair will be looking to enhance QDOS. If at present they are non-committal this must be to protect sales of existing operating systems. Second, it is hard to believe that the success of one of the systems would not stunt the growth of the other. Say, for example, Sinclair was to enhance his system to include true multi-user and multitasking techniques, then it is unlikely that Quest would develop concurrent CP/M on the QL. Survival of both systems would seem to depend upon each one undertaking a separate course.

Conclusion

In the past CP/M owed much of its success not to technical excellence but to market forces. Today, circumstances have changed, CP/M is only one of a group of competing operating systems. Of these, it is possibly the least user-friendly and has the smallest repertoire of built-in functions. Whilst there are upgraded versions of CP/M that remedy its shortcomings, CP/M-68K is not one. In addition the 68K version does not enjoy the large software base that many believe synonymous with its name. Its selection, therefore, as one of the first alternative operating systems on the QL, would seem unfortunate from the point of view of a user.

For business software suppliers the situation is different. Many grew up with CP/M and are used to it. Few if any will be acquainted with QDOS, even fewer with 64K-OS the only other single user single task system in the market.

It is to this group that Quest has addressed itself. The consideration here is not how CP/M works but whether the QL will support their software and whether it will sell in sufficient numbers to justify the cost of transferring across to it. In providing both hard and floppy disks for the QL Quest have effectively stifled any opposition based on the QL's dependence on microdrives. In making a firm commitment to market CP/M on the QL worldwide, Quest have indicated that they believe the QL to be a winner in the business market.

In Search of The Perfect Assembler

*And finding there's only one
for the QL to date that comes
close to meeting all the
criteria.*

Anyone intending to delve into the intricacies and delights of QDOS and 68000 machine code is going to find it far simpler with an assembler at hand. This month has seen the launch of three radically different products.

Computer One's assembler, which we intended to review, has still not been delivered so must rate as an 'also ran'. Since rumour has it this is written in SuperBasic its only advantage over the other two is likely to be its price, £29.95.

The others are from Bristol based Metacomco and Cambridge old hands GST and cost £59.95 and £39.95 respectively. Before discuss-

sing the individual merits or otherwise it's useful to discuss just what an assembler is and what facilities can be expected.

What it must do . . .

An assembler has one simple task to do: it must convert source code containing mnemonics that (some!) humans can understand into binary numbers that microprocessors can understand. For instance, it might be written:

MOVE.W CA.GTLIN,A2
JSR (A2)
BNE.S EXIT_PROC

but an attempt to run this as machine code is extremely unlikely to work since not many computers can make head nor tail of it. This is the source code which an assembler turns into a sequence of numbers, usually printed out in hex (base 16) alongside the source code like this:

3478 0118 MOVE.W CA.GTLIN,A2
4E92 JSR (A2)
6616 BNE.S EXIT_PROC

where the odd-looking numbers are known as the object code and is a readable-ish representation of what the computer can actually understand.

An assembler therefore simply

converts from one to another, but there are varying degrees of ease with which it can do this from the human point of view. For a start, an assembler could be expected to accept and understand source code using industry defined mnemonics so that it's not necessary to learn a new set for every assembler. But it goes rather further than this. It would be very convenient if certain values and addresses related to names that we choose could be defined so that code is more readable and makes a little more sense. Such names are known as labels and any assembler that cannot accept them is not worth having, especially on a 68000 system. There has to be a defined and standard way of using labels, ensuring that they only consist of certain allowed characters such as letters, numbers and underscores and that a certain number of characters in the label are defined as significant.

Next the ideal assembler must be a nice easy layout, so that an imposed file structure and special symbols don't have to be used to define the end of each field. In addition if comments could be inserted enabling the user to read code a couple of months later and remember what was being done and how it would be nice.

An assembler might also be expected to provide a number of options to specify the format of any listing produced and to specify data areas within the code, etc. Furthermore it would be very nice to be able to specify where the assembler must look for its source file, where it must put its object code and where it must direct its listing. And on a multitasking system such as the QL it would be useful if the assembler were written in such a way that other programs could be run at the same time.

Other good features might be the ability to specify the area of screen that it takes, and the ability to insert other source files in specified places within the current source file. Clear and easy to understand error messages enabling correction with the minimum of fuss would also be good to see, and finally, it would be truly wonderful if an editor supplied with the assembler could produce source files, and the

versatility of this editor was such that it could be used for other tasks such as document creation and source code creation for other languages besides assembly.

The ideal package?

The only package that fulfils all these requirements is naturally also the most expensive - the Metacomco. It is very much the best package and is worth every penny. The package is supplied on microdrive cartridge with a user guide for both the editor and the assembler, both having a common front end so that window definition and general set up is similar on both.

The package was written in BCPL under the acclaimed Tripos operating system, and all Metacomco had to do was alter the input and output procedures and 'port' the program across to the QL. The fact that they're written in BCPL does have the disadvantage that a BCPL run time system must also be ported across, but this has long term advantages in that other BCPL applications programs can be moved across with the same ease. This list is therefore soon to include a BCPL compiler and a LISP system of mammoth proportions, at present awaiting the 128K additional RAM.

Those familiar with the BCPL system released by Acornsoft will be delighted to know that the editor supplied by Metacomco is a derivation of the one on that system, and was written as before by Dr Tim King, a one time associate of Dr Martin Richards in Cambridge and co-author of a very popular book on the 68000.

The editor and assembler are invoked using EXEC or EXEC_W and can run concurrently with other programs. The sheer size of the assembler (60K) means that only very short programs will run with it, but the facility is there! Both give an option to alter the window size that the package is going to use and this is done with the cursor keys and various combinations of ALT and the cursor keys, very simple, very straightforward and very pretty.

Every other part of the Metacom-

co assembler follows the definition given already, with the exception that periods (.) are not allowed in label names as they can be mistaken for instruction length specifiers in Motorola standard syntax. At £59.95 the package may be 'professionally priced', but the quality and the level of support that Metacomco says it will provide make it by far the best value for money. Metacomco is at 26 Portland Square, Bristol, England.

The GST assembler meanwhile is the one that has been officially adopted by Sinclair Research as its brand named product, but it does not meet the high standards set by Metacomco. One feels that perhaps Sinclair is 'making it up' to GST for rejecting K/OS.

Deficiencies

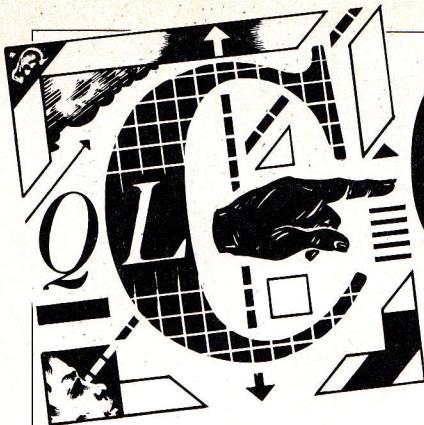
The main deficiencies are a lack of prompts for source file, object file and listing file - they are all included on the command line, which is unfriendly, and the window size is altered by specifying a CON_ window definition rather than the visual approach adopted by Metacomco.

Of more major interest though is the error message system. The assembler appears to have 51 errors reported by number only, and the number is in hex! The meaning of the error has to be looked up in the user guide. The main reason for this is that GST's assembler is only 17K long, which seems to be false economy on a 128K machine and comes across as unnecessary byte pinching.

At the time of writing there were no firm details on the editor that may be supplied with the package, but rumour has it that Metacomco has been asked to supply it. A turn up for the books, and sloppy commissioning by Sinclair Research.

This review implies there is really no choice in the assembler market at present, and the fact that QL User opts for the Metacomco assembler we hope is an indication of putting our mouths where other people's money is!

● Examples of source and object code produced on Metacomco's assembler can be seen elsewhere in this issue.



QL C SERIES

*Taking you deeper into the art
of C programming*

Having examined C generally we now move to specifics. C is unique amongst programming languages as it is self-documenting and structured yet at the same time affords a degree of control akin to that offered by assembler and it appeals to both high and low level programmers.

Programming in C is a pleasure as the language is designed to encourage the most succinct and concise form of expression. C is the art of the understatement.

The best way to learn a language is to start writing programs so we begin by examining the logistics of getting a C program up and running, moving on to construct a simple program and finally, moving on to C proper examining its operators and control structures.

Basic differences

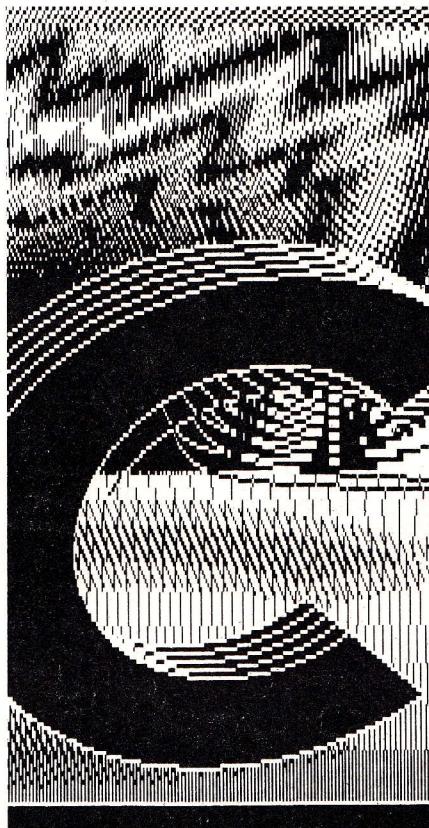
Unlike Basic which is an interpreted language, C is compiled, so the actual procedure for running programs differs. With SuperBasic already resident on the QL the steps for creating and executing a Basic program are for the most part transparent, being carried out automatically by QDOS. Steps for Basic are:

1. EDIT

(a) Enter Basic editor to create a Basic program. Here the user will enter numbered lines of Basic and the editor will automatically check each one for syntax errors.
(b) Save Basic program. This step is optional. The QL will automatically execute any program in memory, but if a program is to be executed by name then it must be saved.

2. (L)RUN

(Load and) Execute a Basic program. Here a named program is loaded into the Basic memory and



each line is then individually translated/executed using a Basic interpreter.

With C, a different course is followed:

1. CREATE AND EDIT SOURCE FILE

(a) First, we use a text editor, a program that creates and modifies files of text to produce a source file. Such a file could rightly be described as the C program proper in so far as all modifications and amendments are made here.
(b) Having finished editing, a file will be saved under a particular name. Convention has it that source file names are followed by a '.c' distinguishing them from others i.e. 'intro.c'. Most text editors will automatically save a file at the end of a session, overwriting any similarly named file.

2. COMPILE SOURCE INTO OBJECT

Next we compile source code to create the executable object code. The compiler is usually invoked with the command 'cc' followed by

the source file name. The compiler will make a number of passes over the program. Ultimately if your program is syntactically correct a file containing machine executable code will be created. On some compilers, the file can be named in the 'cc' command. Otherwise it is usually given the default name 'a.out'. It is at this stage that any errors in syntax will be detected, compilation will be aborted and error messages or 'diagnostics' will be generated. As well as some indication of the nature of the error, the line at which it was detected will be displayed. Usually commonplace errors such as the omission of a ';' will occur in the line immediately preceding or succeeding the one reported.

3. EXECUTE OBJECT FILE

After successful compilation, the object file may then be 'run'. On machines running C-based operating systems this is achieved by simply typing 'a.out' or whatever the object file is called. It is unlikely C compilers running under QDOS will follow the same method.

The first step

We can now move onto writing a simple program. Enter the text editor and create a source file called 'intro.c' containing the following:

```
/* Program to print text to screen */
main()
{
    printf("This is C\n");
}
```

Now compile the source file and run the object file. "This is C" should be displayed on screen with the cursor positioned on the line below.

PRINTER BARGAINS FOR THE QL

Prices include Interface/Cable, VAT and Carriage

DOT MATRIX

Mannesmann Tally MT80 80cps.....	£249
Canon PW1080A 160cps—draft 27cps—NLQ.....	£339
Shinwa CP80 80cps 13x9—dot matrix.....	£225
Seikosha GP100AS 50cps.....	£169
Seikosha GP250X 50cps graphics	£215
Seikosha GP700A Colour.....	£365

DAISY WHEEL

Daisy step 2000 18cps underline bold shadow etc.....	£275
Juki 6100 20cps underline bold shadow 2K buffer.....	£409
Smith Corona TPI 12cps	£209

MONITORS

Philips V70001 12in 18MHz high res. green phosphor.....	£92
---	-----

INTERFACE/CABLES

Miracle System QL Parallel printer	£39-
Miracle System QL Serial printer	£12
Fidelity QL-CM14 & Philips V7001-QL	£12

STRONG COMPUTER SYSTEMS

Bryn Cottage, Peniel, Carmarthen, Dyfed SA32 7DJ

Tel: 0267 231246 for further information

QLPRINT

Your QL documents/listings printed £2 per file any length
inc. p&p. Send your tape and cheque or PO to

**QLPRINT, Chubbs Cottage, Buddle Lane, Hatherleigh,
Devon EX20 3HX or phone Hatherleigh 535**

To
reserve
your
classified
space
please
phone
01-833 0531

For the Sinclair QL:

- CP/M 68K
- Hard and floppy disk drives
- Extra memory
- Applications software
- Other hardware products

**Available soon
from Quest**

For details send name
and address to:

Grant McEwan,
Quest International Computers,
School Lane, Chandler's Ford, Hants.
SO5 3YY

QL UTILITIES

FOUR Programs on Microdrive for Sinclair QL to prevent Directory overflowing the screen, provide single-key LOADING or DELETION of files, repeat FORMATING of cartridges and back-up COPYING of whole or part of cartridge.

£10 from **WDSOFTWARE**, Hilltop, St Mary, Jersey, C.I.
Tel: (0534) 81392

QL - EARLY LEARNING

Shapes, colours, numbers and simple arithmetic for ages three to eight.

Difficulty options and merry tunes!
£14.95 (inc p&p) on Microdrive

**Key Software, Newlands, Durley Brook Road
Southampton SO3 2AR Telephone 695182**

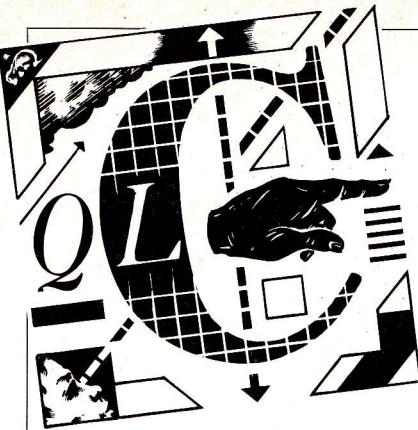
**THIS
MONTH
IN**

**Sinclair
Programs**

● COMPETITION

● **Sinclair Research** are providing a great first prize for our easy-to-enter competition. Will it be a Microdrive, an Interface Two or a QL computer? Read **Sinclair Programs** to find out. Second prize will be a light pen and new keyboard, third prize will be a joystick and Spectrum interface. A hundred runners-up will win software from any one of ten top companies.

- **PLUS** Thirty new program listings for the Spectrum and ZX-81, including a full-scale professional program for the ZX-81.
- **PLUS** Hard-hitting reviews of all the latest programs.
- **PLUS** A fresh look at some of the adventure games on the market: a section designed especially for beginners: and the next part of our programming course.



Now running through the source code step by step:

`/*....*/`

Enclosing characters within `/* ... */` achieves a result similar to REM in Basic. All text enclosed in this way is ignored by the compiler. This device is used to annotate a given C program.

'main'

All C programs irrespective of length consist of a series of functions or procedures performing various operations. 'main' is one such function but unlike all others its purpose is preset and its name can never be varied. 'main' represents the point at which any C program begins executing and need never be explicitly invoked by name. Execution begins with the first statement (or function) following the opening brace '{' of 'main' and ends with the closing '}'. A ';' marks the end of each particular step in execution.

`'(...)'`

The () following 'main' are normally used to contain arguments that will be passed to the particular functions. In the case of 'main' there are none, but in the case of the library function 'printf' the characters contained within the quotes constitute the first argument passed and as such will be displayed on the screen.

'Printf'

'printf' is a library function which will format and print output to your screen (unless some other destination is specified.) Unlike PL/1 or Fortran, functions need not be called but are simply invoked by name.

The '\n' contained within 'printf's' argument is a special character used for formatting. Other special characters are:

- '\n' newline
- '\t' tab
- '\b' backspace
- '\r' carriage return
- '\f' form feed
- '\' backslash
- '"' single quote

So far we have used 'printf' simply to print a string, a more significant use is to print a variable or numeric expression using an output format. To illustrate this insert:

```
printf("What is the value of %d  
in octal?",10);  
printf("%s! %2d decimal is  
0%03o octal\n","Right",10,10);
```

this will print "What is the value of 10 in octal? Right! 10 decimal is 0012 in octal".

This example demonstrates a variety of aspects of formatting using type and precision specifiers. 'Printf' allows a wide range of these, such as:

ASCII character	%c
ASCII string	%s
Decimal integer	%d
Unsigned integer	%u
Octal Integer	%o
Hexadecimal Integer	%x
Floating Point	%f
Exponential Notation	%e

In addition to basic types on numerical displays we can also specify the width and precision. For example, we can display numbers up to 999,999.99 with the format '%10.2f'. One place is set aside for a possible leading minus sign though commas after thousands cannot be specified. Leading zeros can be obtained simply by placing a zero before the output field width.

Variable declarations and types

C has four fundamental types of variable. They are:

int	integer
char	one byte character
float	single precision floating point
double	double-precision floating point

In addition there are also arrays and structures of these basic types, pointers to them and functions that return them all. All of which we will encounter in due course.

All variables in a C program must be declared before use. This is usually done at the start of a function before any executable statements, though declaration can sometimes be done implicitly by context. In the program that follows

```
'int a, b, c, sum'  
declares a,b,c,sum to be integers.  
/* simple program adds two  
integers - print sum */  
main()  
{  
    int a,b,sum;  
    a = 10; b = 40;  
    sum = a + b;  
    product = a * b;  
    printf("sum is %d\n",sum);  
    printf("product is %d",a * b);  
}
```

Variable names may be of any length and must be composed of

letters, numbers or '_' and begin with a non-digit. The compiler distinguishes between upper and lower case. Stylistic conventions reserve upper case names for parameters and names beginning with '_' to system variables.

Examples of legal variable names would be:

i, l, x1, total, acc_no.

Illegal names would be:

_increase, 3feet, auto.

'auto' is illegal as it is a C keyword and as such reserved. The complete list of such keywords, all of which we will encounter, is: auto, break, case, char, continue, default, do, double, else, entry, extern, float, for, goto, if, int, long, register, return, short, sizeof, static, struct, switch, typedef, union, unsigned, while.

Arithmetic and assignment operators

C includes the following operators:

()	Parenthesis
-	unary minus (i.e. -99)
* / %	multiply, divide, modulo
+ -	plus, minus
=	equals

These are listed in the order in which they would be executed i.e. in order of precedence.

Assignment and increment/decrement operators

C provides a number of extensions to the simple assignment operator '='. It allows it to be combined with either an arithmetic or bitwise operator. (For the moment we will ignore bitwise operations). The results of these extensions are:

a += x	Add x to a
a -= x	Subtract x from a
a *= x	multiply a by x
a /= x	Divide a by x
a %= x	get remainder of a divided by x

In all cases the result is stored in a. Thus

a += x

provides a faster way of performing

a = a + x

This is but the first of many effective short cuts that C permits, enabling code to be compact and concise as well as efficient.

Another short cut is provided by the increment operators '++' and '--'. These add one to the value of a variable. When this operation occurs depends upon which side of the variable the operator is placed.



For example if initially $b=9$ then:

a=++b will produce a equal to 9 and b equal to 10

if on the other hand we write instead:

a=++b then both a and b will equal 10.

A good rule is that if $++$ comes before the variable it's done before the assignment, if it comes after then it's done after.

Simple input/output

In addition to 'printf', two further simple I/O functions are in the basic library. These are:

a) `getchar()` which fetches the next input character each time it's called, returning the character as the value of the function and

b) `putchar(c)` which prints the contents of variable C to standard output (usually the terminal) each time it's called. Using these functions we can begin to write some useful code.

Assuming that standard input is from the keyboard and that standard output is to the screen, the following program will echo characters to the screen and count the digits, upper and lower case letters in a line of input.

```
#include <stdio.h>
main()
{
    int c, up_c, low_c, digit_c,
        count;
    up_c=low_c=digit_c=0;
    count=0;
    while((c=getchar()) != '\n') {
        putchar(c);
        if ('0' <= c && c <= '9') digit_c++;
        if ('A' <= c && c <= 'Z') up_c++;
        if ('a' <= c && c <= 'z') low_c++;
        count++;
    }
    printf("\n numerals %d",digit_c);
    printf("\n upper case letters %d",up_c);
}
/* end */
#include <stdio.h>
```

This is a compiler control in-

struction ('macro') that indicates that you wish to tag onto the start of your program a series of preconstructed functions from the standard C library file 'stdio.h'. These will be compiled along with your code. Functions 'getchar' and 'putchar' are sometimes contained within 'stdio.h'.

Logical and relational operators

C has a full set of relational operators. These are:

==	equal to
!=	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

It should be noted that C unlike BASIC distinguishes between assignment '=' and equivalence '=='. Furthermore, great care should be exercised with the placement of a space about the assignment operator. $x = -10$ will set x to 10, $x = -10$ will subtract 10 from x, $x = -10$ will also decrease x by 10. It should also be noted that assignment operators come after all others in order of evaluation, hence the need to put brackets about $c = \text{getchar}()$.

The results of relational operators can be combined using the logical operators:

&&	and
!!	or
!	not

for example

'0' $\leq c \&\& c \leq 9$ ' which would be true if c lay outside the range 0-9, otherwise it would be false. The precedence of logical operators is below that of relational operators doing away with any need for parenthesis. In fact the table of precedence for C operators from highest to lowest looks like this:

Arithmetic
Relational
Bitwise
Logical

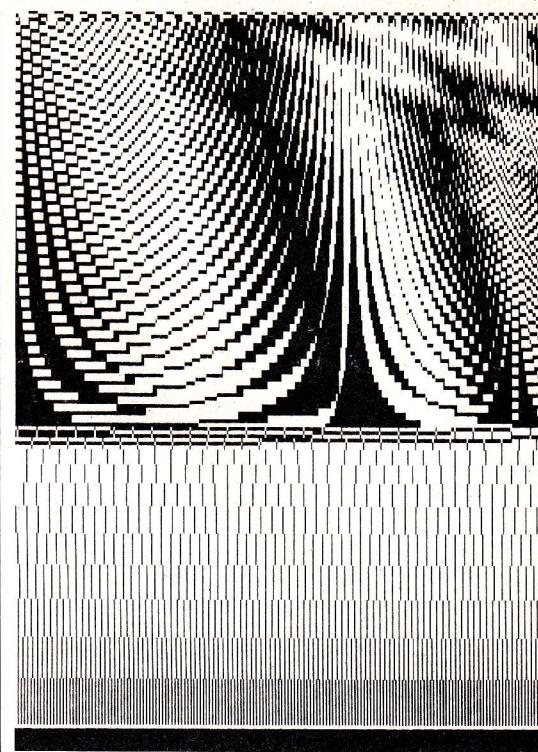
Control structures

WHILE STATEMENT

The While statement is a loop in the form

```
while (expression){
    1st statement;
    2nd statement;
    .....
    nth statement;
}
```

where Step 1) The expression is evaluated 2) if its value is true then the statements enclosed will be performed and then control will



pass back to step 1. Because evaluation takes place first the statement part need never be executed if a false result is obtained in the first pass.

The expression part of the 'while loop' in the program demonstrates a particularly powerful feature of C, which is that assignments can be nested within expressions. Like any expression an assignment carries a value, being that in the right hand side. i.e. c has the value returned by `getchar()`. By implication this means that C will also permit multiple assignment i.e. `up_c = low_c = digit_c = 0`. This is a remarkable notational shortcut that means in effect we get two operations for the price of one. The reason being that an assignment has a value just as any other expression does. This device produces extraordinarily concise and clear code. Indeed, much of the satisfaction of coding in C stems from the fact that so much can be achieved from so little. A line in C save nine in BASIC.

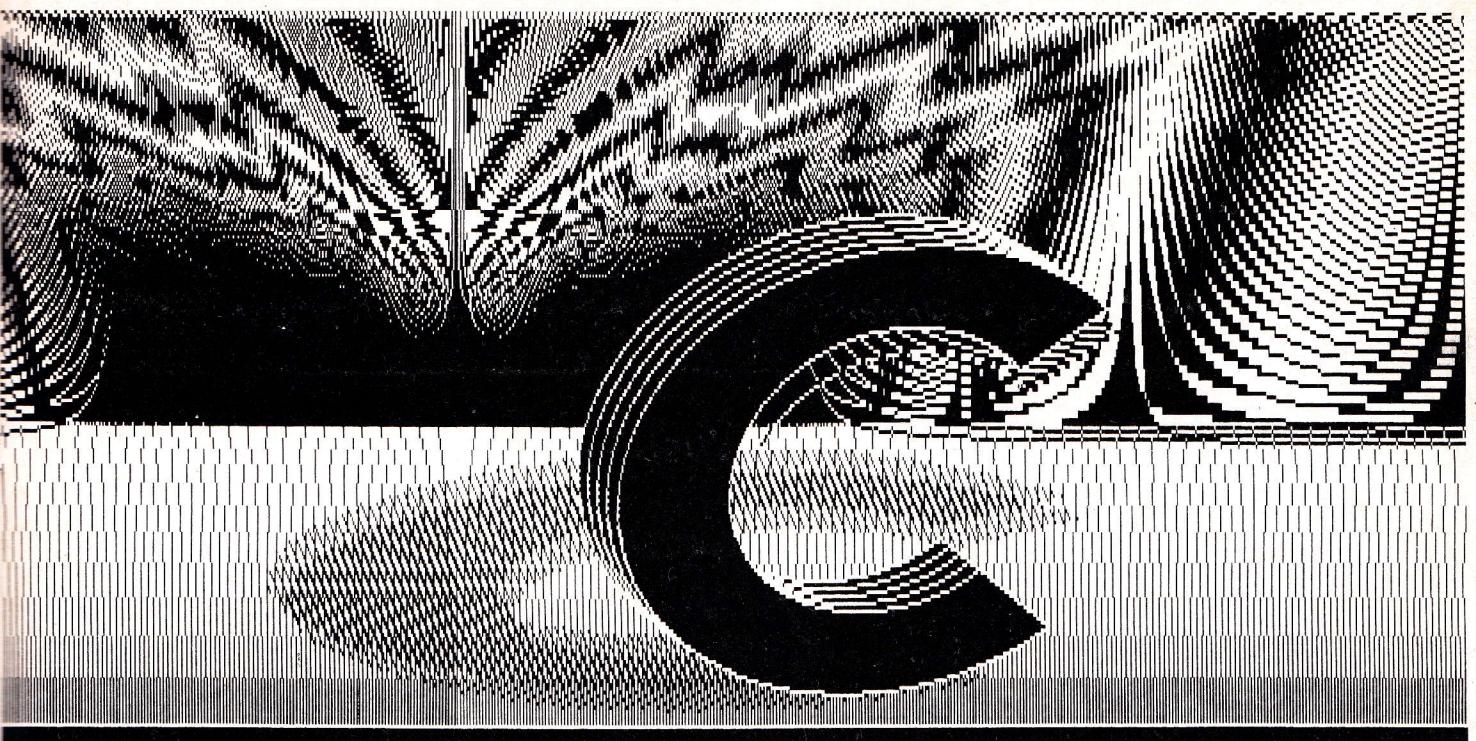
FOR LOOP

General form:

```
for (initialisation; expression;
    increment){  
    1st statement;  
    .....  
    nth statement;  
}
```

As is now characteristic of C, the 'for loop' is another example of succinctness. It enables us to initialise, increment and test all in a single statement. If we return to our example we can replace lines:

```
count = 0;
while((c = getchar()) != '\n') {  
    with the line:  
    for((count=0;(c=getchar()) != '\n')
```



```
n'; count++) {
```

Choosing between for and while is a matter of style, but generally where there is an initialisation one should opt for a 'for loop'. In the example this saves allocating a separate statement to zeroise count.

Any or all 3 parts of the for statement may be left blank. For example amending:

```
for (i = 0; i < 100; i++)
printf("%d", i);
```

to one of the following will result in infinite loops:

- a)

```
for (i = 0; ; i++)
printf("%d", i);
```

 prints ever increasing value of i
- b)

```
for (i = 0; i < 100; ;)
printf("%d", i);
```

 prints 0 all the time
- c)

```
for (i = 0; ;)
printf("%d", i);
```

 same as example (b)
- d)

```
for (;;)
printf("%d", i);
```

 print whatever value i has been set to previously

C provides a means for jumping out of for, switch, while and do...while to the next statement. This is achieved by the instruction 'break'. Thus if we alter a) to:

```
for (i = 0; i++)
printf("%d", i);
if (i == 100)
break;
};
```

the loop performs as originally intended.

THE IF. . . . ELSE STATEMENT

Taking the general form:

```
if(expression){
1st statement;
.....}
```

```
nth statement;
}
```

```
else{
1st statement;
.....;
nth statement;
}
```

this is the fundamental condition testing statement in C. Similar to the while expression, if true, a non-zero value is returned and the succeeding statement contained within the first brace are executed.

The else part of the test is optional, the statement part executing only when the test is false. In the program, for example, no use was made of else.

C also allows for multiple choice else if's so that program flow can be directed along several separated logical courses. For example:

```
If(...){
...
else if(...){
...
else if(...){
...
else
{
...
};
```

Here each condition is tested in order of appearance. If any condition is met the set of statements immediately following are executed and the test is over with program control passing to the statement immediately beyond the ';'. The final else acts as a 'catch-all' case, executing if no condition has been satisfied.

Where a single statement is to be executed braces may be omitted.

The if statement is not restricted

to single test. Relational operators may be strung together to achieve a variety of different tests i.e.

```
if (low_c > up_c && low_c > digit_c);
```

printf("There are more lower case letters than either upper case letters or digits");

THE CONDITIONAL EXPRESSION

C provides an alternate to the basic If... else ... construction which is both more concise and easier to read. Taking the form:

```
expression_1 ? expression_2 : expression_3
```

it means evaluate expression_1 if true then return the value of expression_2 else return that of expression_3. Note that the values returned are not logical but actual values.

To illustrate the use compare the two expressions both of which achieve the same result:

```
if(a < b)
x = a;
else
x = b;
.....
x = a < b ? a : b
```

Conclusion

In the course of this article we have sought to provide users with the basic tools for writing programs, which are tentative first steps into a completely new field for most QL users. As a consequence we have picked contractions that find an easy parallel in Basic. Next issue we delve deeper into the intricacies of C to look at arrays, pointers and structures. As we do so users will be able to more fully appreciate how subtle a distillation of low level control and high level structure C is.

Microdrive Microdrive Microdrive Microdrive Microdrive

NEW

ZX Spectrum and QL Microdrive Storage Box £5.95



- * HOLDS 20 CARTRIDGES
- * FULLY INTERLOCKING
- * CENTRE SECTION FOR INDEX CARDS

- * DESIGNED TO MATCH SPECTRUM AND QL

Transform Ltd has now produced a smart new Storage Box for the ZX Spectrum and QL

The Transform Microdrive **Storage Box** is attractively designed to match both the Spectrum and QL

It will hold 20 Microdrive cartridges (enough to store 1.6 megabytes of data!)

The Microdrive Storage Box is fully interlocking and will allow you to expand your system with all the new programmes on microdrive as well as enhance the 'smart' efficient look of your office/study.

Simply write or telephone:

TRANSFORM LTD

DEPT X, 'SWATLANDS', LUCKS LANE
PADDOCK WOOD, KENT TN12 6QL 089 283 4783

For fast delivery quote your Credit Card/Access
Barclaycard number

Let your Sinclair QL teach you to type!

Type faster! -- Prepare documents quickly -- enter programs effortlessly!

In just a few days, Computer One's new QL-TYPING TUTOR will enable you to learn true touch-typing on the Sinclair QL.

The computer ONE QL-Typing Tutor is a quick and easy way of mastering the QL keyboard.

In a series of simple and enjoyable lessons the QL-TYPING TUTOR provides a personal and friendly way of learning to type in the comfort of your own home; at your own pace.

The QL-Typing Tutor is a valuable educational tool for users of all age groups.

QL-Typing Tutor

- Self-paced tutorials
- Interactive colour graphics
- Speed checking
- Error diagnosis
- Comprehensive manual

PRICE £19.95
incl. VAT & p. & p.

QL-Typing Tutor guides the user through a series of easy learning steps using the traditional and well-proven 'horizontal method' as used by Pitmans.

QL-TYPING TUTOR comes with a comprehensive manual outlining the recommended learning sequence, and providing a variety of information on how to improve your typing technique.

QL-TYPING TUTOR has been developed and tested extensively in schools and colleges.

I would like to order:

QL-TYPING TUTOR £19.95

Please send
further
details

METHOD OF PAYMENT

By cheque or PO (no cash) made payable to COMPUTER ONE



By Access
 Barclaycard

Enter card no.

Sign below

Credit card sales UK buyers only
Please allow 28 days for delivery

Sinclair QL are Registered Trade Marks of
Sinclair Research Limited

TOTAL TO SEND £ _____

computer ONE

QL10

SCIENCE PARK, MILTON ROAD, CAMBRIDGE

NAME _____

ADDRESS _____

AT LAST! the MISSING link from ARCHIVE to QUILL!

QUILMERG

QUILMERG will take a document prepared by QUILL as a skeleton, and automatically inserts fields SELECTed from an ARCHIVE database, printing a complete document for each ARCHIVE record. Ideal for mail shots, newsletters, reminder letters, standard contracts etc.

Price £12.50 plus 25p p&p

UTILITY PACK

PITCH ASSOCIATES UTILITY pack for the SINCLAIR QL, comprises of 12 functions, procedures and programs. Including a print formatter similar to conventional BASICs PRINT USING, data INPUT & VALIDATION, RAM & ROM HEX & ASCII dumps and microdrive FILE editor, HEX-DECIMAL conversion and HEX maths.

Price £10.00 plus 25p p&p

SPECIAL OFFER – QUILMERG and UTILITY
for only £17.50 (plus 25p p&p) when ordered together.

ALL PITCH ASSOCIATES software comes supplied with an UPGRADE voucher. When new releases are announced, send us a microdrive cartridge and the voucher for the latest version.

PITCH ASSOCIATES

39 Rockleigh Avenue, Leigh-on-Sea, Essex SS9 1LA
Telephone (0702) 73002



**CENTRONICS
PRINTER
INTERFACE
£39.00 INC.**

Simply plugs into the 'ser1' port of the QL and any CENTRONICS compatible printer (e.g. Epson, Kaga, OKI, Seikosha, Juki, Canon, Mainsmann etc.). Requires no special software.

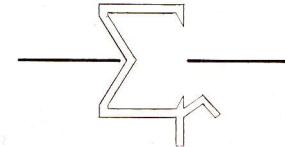
JOYSTICK ADAPTOR £8.95 INC.

Enables any Atari compatible joystick (e.g. Kempston, Quickshot 1/2 etc.) to be used with the QL joystick ports.

Joystick adaptors £6.95 each when ordered with the printer interface.

All products come complete with instructions and a 12 month guarantee.

Make cheques/P.O.'s payable to Sigma Research. All prices include V.A.T. and P & P within U.K.



Sigma Research,
95 Vinery Road,
Cambridge,
CB1 3DW.

QL is a registered trademark of Sinclair Research Ltd.

Languages for the Sinclair QL

Take advantage of the full power of the Sinclair QL with Computer One's new range of languages.

QL-FORTH and QL-Assembler are the essential languages for writing fast and compact programs. QL-Pascal brings you the versatility of a structured programming language.

QL-Assembler

QL-Assembler is a 68008 assembler for the Sinclair QL. It assembles Motorola format source files, producing executable code and listing files.

It may be used to generate machine-code programs, or to extend QL SuperBASIC with machine-code procedures and functions, a facility of great use to anyone writing applications programs in SuperBASIC.

- 68008 Assembler
- Full screen editor
- Relocatable output
- Full syntax checking
- Comprehensive manual

PRICE £29.95
incl VAT & p & p

QL-Forth

- FORTH-83 system
- Full screen Editor
- QL-graphics & sound extensions
- Example games & utilities
- 48-page manual

PRICE £29.95
incl VAT & p & p

QL-Forth combines the simplicity of BASIC with the speed of machine code. This compiled language runs very fast (typically ten times faster than BASIC) and is the ideal language for exploiting the full power of the QL. QL FORTH is an essential language for learners and enthusiasts alike.

QL FORTH is a complete implementation of the FORTH language to the latest 1983 FIG standard.

QL-Pascal

- Program Development Environment
- Full screen Editor
- Comprehensive 48-page manual

PRICE £39.95
incl VAT & p & p

QL-Pascal at last provides the Sinclair QL with a truly structured and powerful language. PASCAL is a well established educational language in schools and colleges, and is widely used by professional programmers. This package provides the essential tools for writing serious software on the QL.

QL Pascal includes all the standard Pascal features plus various QL specific extensions. The package also includes an advanced screen editor integrated with a complete menu-driven program development environment.

I would like to order:

QL ASSEMBLER	£29.95	<input type="checkbox"/> Please send
QL FORTH	£29.95	<input type="checkbox"/> further
QL PASCAL	£39.95	<input type="checkbox"/> details

{

METHOD OF PAYMENT

By cheque or PO (no cash) made payable to COMPUTER ONE



By Access
Barclaycard

Enter card no.

Sign below

computer ONE

QL10

SCIENCE PARK, MILTON ROAD, CAMBRIDGE

NAME _____

ADDRESS _____

TOTAL TO SEND £ _____

Credit card sales UK buyers only

Please allow 28 days for delivery

Siemens QL are Registered Trade Marks of

Siemens Research Limited

After last month's introduction to procedures and functions it's time now to take a look at SuperBasic and the features it needs. After deciding on these added extras, it will be possible to see which ones can be usefully written as procedures or functions in this language. It is perfectly possible to add procedures and functions written in 68008 machine code to SuperBasic but this is rather beyond the scope of this series. Some examples of this more advanced technique can be seen elsewhere in this issue and in the November issue of our sister magazine *Electronics & Computing Monthly*.

From the point of view of the BASIC language SuperBasic does not leave much to add, but from the point of view of the QL, it does, apparently because it was originally intended not to supply a BASIC with the QL - it was added as a marketing afterthought.

As a result SuperBasic does not take full advantage of the micro-drives or its printer port (RS232) facilities. Wouldn't it be nice to be able to backup microdrive cartridges in one go rename a microdrive file, or set up a printer connected to a serial port so that any style the printer supports can be printed? All this (and more!) can be done. It's also useful to be able to use numbers of all different bases on the machine. The QL is limited to decimal whilst programmers often need to be able to specify things in binary and hexadecimal.

Base conversions

A simple hex to decimal function was described last month. This will be extended this month to allow it to convert any base from two to 36 to decimal. It's important to note here that there is a defined convention for writing down bases above decimal. Any base above 10 will need to have some way of representing numbers greater than nine using only one character. This sounds difficult but all that is required is to consider how hexadecimal numbers are represented. It has already been seen that base uses the capital letters from A to F to represent the numbers from 10 to 15. It follows then that an arbitrary base **n** (in which **n** is greater than 10) would use the letters from A to CODE(CHR\$(A+n-1)). In other words base 19 would use the letters from A to I to represent the numbers from 10 to 18.

This is a convention that actually makes base conversion a simple process. All that is needed is a function to convert a string representing a number in base **n** to a decimal number and a function to convert a decimal number to a

Learning Super Basic

*The teach-yourself course:
this month delving deeper
into the language features.*

string representing that number in base **n**. These can be combined so that a number in base **n** can be converted to a number in base **m**.

The function HEX which as we said last month is ripe for extension - only a couple of changes have to be made and it can convert any base to decimal. This is how it's done:

```
1000 DEFine FuNction DEC(num$,base)
1010 LOCAl a,b,dec
1020 dec=0
1030 FOR a=1 to LEN(num$)
1040 b=(num$(a) INSTR
'0123456789ABCDEFHIJKLMNOPQRSTUVWXYZ'(1 TO
base))-1
1050 IF b<>-1
1060 dec = dec * base + b
1070 ELSE
1080 dec=0:EXIT a
1090 END IF
1100 END FOR a
1110 RETurn dec
1120 END DEFine
```

This new version of the function works in exactly the same way as the simpler HEX function last month, but it selects which series of characters to work on by splitting the string of **01234567890 ABCDEFGHIJKLMNOPQRSTUVWXYZ** into a smaller string consisting of just the characters in the specified base. String slicing like this is very important, and Sinclair's way of doing it is elegant and straightforward while other machines use awkward functions like LEFT\$, RIGHT\$ and MID\$ to specify which parts of a string are wanted. With SuperBasic (and ZX81 and Spectrum BASICS) you merely have to specify the range over which the string is to be sliced.

Using the function is simplicity itself. Whenever it is necessary to convert a number into decimal, type:

x = DEC(numstring,base)

where numstring is either a 'quoted string constant' (which is a string constant itself - so is 'ABCD23') or a string variable such as x\$, and base is a decimal number which tells the function which base the number is in, so

that it can convert it into decimal. With a string like '238DFK' representing a number in base 24, to get its decimal equivalent, type the following:

x = DEC('238DFK',24)

which makes x hold the decimal value of that number. Alternatively say

PRINT DEC('238DFK',24)

or even

a\$='238DFK'

x = DEC(a\$,24)

And the other way

Now how about converting the other way, with a decimal number to turn into a string representing that number in a different base. As the result is always going to be a string, it is necessary to write a function that returns a string result. This is done by adding a \$ sign to the end of the function name. So **OFDEC**, for example, would become **OFDEC\$**. The easiest way to write the function is to build up the result string character by character, starting with a null (empty) result string. Say the result string is called **res\$**, this can be done by typing **res\$=""** somewhere early on in the function.

To convert the number it is useful to first think how decimal numbers are written down. The number 2345 is written down as 2345. But this means the number is $2 * 1000 + 3 * 100 + 4 * 10 + 5 * 1$. To put it another way, 2345 is $2 * 10^3 + 3 * 10^2 + 4 * 10^1 + 5 * 10^0$

This is a general rule, so if the 2345 was actually a number in base 8, it would be equal to

2 * 8^3 + 3 * 8^2 + 4 * 8^1 + 5 * 8^0

(which equals 1253_{10} as it happens). It follows that in order to convert the number to base **n** it is necessary to calculate how many times each power of **n** occurs in the number. The function thus becomes:

```
2000 DEFine FuNction OFDEC$(number,base)
2010 LOCAl num,t1,t2,res$build
2020 num=number:res$=""
2030 REPeat buildstring
2040 IF num<base:EXIT build-
string
2050 t1=INT(num/base):t2=t1
2060 t1=num-t1*base:num=t2
2070 res*=chr$(t1+48+7*(t1
>9))&res$
2080 END REPeat buildstring
2090 res$=chr$(num+48+7*
(num>9))&res$
2100 RETurn res$
2110 END DEFine
```

Here the number is repeatedly divided by the base to get each occurrence of each power of the base as a digit in t1. This digit in t1 is then converted into the QL charac-

ter that represents it. This is easy. Adding 48 to a pure number converts it into the code of the character representing that number (if it's between zero and nine) as the code for zero is 48 and each number is continuous after that. Numbers greater than nine however run into problems. Ten plus 48 gives the code for a colon, which is obviously wrong. A quick look at the character set in the concepts section of the user guide shows that capital letters start seven numbers after this, so, if the number is greater than nine, seven as well as 48 has to be added to get the right code. The logical operators of the QL's SuperBasic are used to doing this. The construct `t1>9` has a value. If `t1` is not greater than nine it will have a value of zero, while if `t1` is greater than nine it will have a value of 1. So multiply the value of this expression by seven (which will give us either zero or seven and add that plus 48 to the number).

There are other ways of coding this function, but this method is as good as any. So these are the functions which will allow conversion of numbers in decimal to numbers in any base from two to 36 (after all, although the coding looks as though it will support it, base one only has a valid digit of zero, so it's a fairly pointless base to use!), and function which does the reverse. Now all that's needed is to write a function that converts a number from one base, to any other – within the range of bases from two to 36, of course. This is obvious – just combine the two functions, like so:

```
3000 DEFine FuNction BASE$  
    (num$,base1,base2)  
3010 RETurn OFDEC$(DEC  
    (num$,base1),base2)  
3020 END DEFine
```

What could be simpler?

Importance of 'locals'

Notice how in both the main functions the `LOCal` statement was used as the first line. Why? If all the variables and identifiers used in each procedure and function were not declared as local, then the use of each one would result in some nasty side effects. If there was a variable or identifier in the program that had the same name as one in the procedures or functions being used, the act of calling the function or procedure would change the value of the variable in the main program, which is generally not the aim. By declaring each one as local to the procedure or function in which it is used, any variables or identifiers with the same name in the main program will not be affected.

The need for identifiers to be local is often overlooked. Every time a `FOR...END FOR` loop, a `REPeat...END REpeat` or a `SElect ON` structure is used, an identifier must be used too. Using one with the same name elsewhere in the program would result in its corruption too – so make them local!

At the same time there are instances where the aim is to use the procedure or function to alter the values of variables elsewhere in the program. When this occurs the variables (and / or identifiers) are known as global variables.

Support functions

Now to consider the microdrive support functions, and write two procedures – `BACKUP` and `RENAME` – and one function `CHKFILE` to add to the set of procedures SuperBasic already provides. `BACKUP` will do just that – it will backup the contents of a specified drive onto a cartridge in another specified drive. `RENAME` will rename a file on a specified drive with a new specified name, but the rename will only work on the drive containing the original file. In short, both the source and destination for the file to be renamed must be the same. This is standard and is really meant to protect files. The `CHKFILE` function will return a value of zero or one (FALSE or TRUE), depending on whether or not the specified file is on the specified drive. FALSE means it isn't and TRUE means it is. We'll write this first:

Taking advantage of the fact that the `DIR` command can be used to send the directory of a microdrive to any open channel, if a microdrive file is opened the directory can be sent there. Each entry can then be read back and verified against the filename that's being checked. The same technique is used in the `BACKUP` procedure too. So:

```
4000 DEFine FuNction CHKFILE(fname$,drive)  
4010 LOCAl chk$,check,found  
4020 OPEN_NEW#3,'mdv'&  
    drive'__dir_tmp'  
4030 DIR#3,'mdv'&drive'__':  
    CLOSE#3  
4040 OPEN_IN#3,'mdv'&drive  
    &'__dir_tmp'  
4050 INPUT#3;chk$;chk$  
4060 found=0  
4070 REPeat check  
4080 IF EOF(#3): EXIT check  
4090 INPUT#3;chk$  
4100 IF chk$==fname$:found=  
    1:EXIT check  
4110 END REPeat check  
4120 CLOSE#3  
4130 DELETE 'mdv'&drive'__  
    dir_tmp'  
4140 RETurn found  
4150 END DEFine
```

Once the directory has been copied onto a file called `dir_tmp`, we then read in two entries the cartridge name (the name assigned to it when it was formatted) and the number of free/good sectors. We don't want either of these. Then a loop is entered in which we first check if the end of the directory file has been reached. If so then the loop is exited, otherwise a file name is read in and compared against the file name assigned to the function of the parameter. Notice that two adjacent equals signs ('=='') are used here – this causes the QL to ignore the case of each name, so that if a filename of 'QLUser' was sent to the function and there was a file called 'qluser' on the drive, it would return the correct value of 'yes, this file is on the drive'. If the file being looked for is found the variable 'found' is set to one (TRUE). When the loop has finished the `dir_tmp` file is used and deleted and the value of 'found' returned to the program.

`BACKUP` is much the same:

```
5000 DEFine PROCedure BACKUP(drive1,drive2)  
5010 LOCAl a$,b$,d$,f$,movefiles  
5020 d$='mdv'&drive2&'__  
    dir_tmp'  
5030 a$='mdv'&drive1'__'  
5040 b$='mdv'&drive2'__'  
5050 OPEN_NEW#3,d$:DIR  
    #3,a$:CLOSE#3  
5060 OPEN_IN#3,d$  
5070 INPUT#3;f$f$  
5080 REPeat movefiles  
5090 IF EOF(#3):EXIT movefiles  
5100 INPUT#3:+$  
5110 PRINT#0;'Copying'!f$  
5120 COPY a$&f$ TO b$&f$  
5130 END REpeat movefiles  
5140 CLOSE#3:DELETE d$  
5150 END DEFine
```

Here each file name on the source drive is copied to the destination drive with the same file name until the end of the directory is reached. Notice that the directory file is opened on the destination drive so that there's no copying of that across as part of the backup process.

`RENAME` is rather simpler:

```
6000 DEFine PROCedure RENAME(old$,new$,drive)  
6010 LOCAl a$,b$  
6020 a$='mdv'&drive'__'&  
    old$  
6030 b$='mdv'&drive'__'&  
    new$  
6040 COPY a$ TO b$  
6050 DELETE a$  
6060 END DEFine
```

Here the old file is just copied to the new file then the old file is deleted.

That's it for this month, but those who are interested might like to know that a lot more can be done by delving into machine code – and it's great fun!

Adding to Superbasic

Loading is into the **resident procedure** area of the machine, using the RESPR function. This means that once loaded you can type NEW as often as you like but they'll never go away! The only way to remove them is by resetting the machine. In fact SuperBasic treats these machine code routines as if they were built in, so there's no need to merge in a file full of procedures written in Basic for every program.

The routines are primarily meant for people who want to write

Although the QDOS documentation can be difficult to follow at times it shows how simple it is to add machine code procedures and functions to SuperBasic, as demonstrated here by adding two procedures and one function to the machine.

their own games and all they do is look after a high score table which is in the same code space as the procedures, so there's no need to create one - it's all done for you. The high score table consists of 200 bytes which can hold ten scores. There is also room for ten names of up to 16 characters each and the scores can be anything from 0 to just over 4,000,000,000, which should be enough for most players!

These are the procedures:

SET_TAB this is called with one parameter. All it does is set the

Procedures and Functions for Games

Extensions to SuperBasic

By Adam Denning Copyright © 1984 Adam Denning

Operating system vectors

UT_CON	EQU	\$C6	Opens a pre-defined console device
UT_SCR	EQU	\$C8	Opens a pre-defined screen device
UT_MTEXT	EQU	\$D0	Prints a string to a channel
BP_INIT	EQU	\$110	Routine to link procedures
CA_GTLIN	EQU	\$118	Gets arguments as long integers

Operating system offsets and equates

BV_RIP	EQU	\$58	Offset from A6 of RI SP
ERR_BP	EQU	-15	'Bad Parameter' error
RET_INT	EQU	3	Function return type integer

Trap keys

IO_OPEN	EQU	1	Open a channel
IO_FBYTE	EQU	1	Fetch a byte from a channel
IO_CLOSE	EQU	2	Close a channel
IO_FLINE	EQU	2	Fetch a line
IO_FSTRG	EQU	3	Fetch a string of bytes
IO_SBYTE	EQU	5	Send a byte to a channel
IO_SSTRG	EQU	7	Send a string to channel
SD_TAB	EQU	\$11	Tab cursor in channel

SET_UP	LEA.L	PROC_DEF,A1	Link in procedures and functions
	MOVE.W	BP_INIT,A2	
	JSR	(A2)	
	MOVEQ	#0,D0	
	RTS		

PROC_DEF	DC.W	2	Number of procedures
	DC.W	SHOWTAB-*	
	DC.B	5	
	DC.B	'HIGHS'	
	DC.W	SETABLE-*	
	DC.B	7	
	DC.B	'SET_TAB'	

DC.W	0	
DC.W	1	Number of functions
DC.W	FN_SCORE-*	
DC.B	5	
DC.B	'SCORE'	
DC.W	0	

scores in the table to the parameter and all the names to blanks. It is called from Basic like this:

SET_TAB initial_score

initial_score is any number that you choose from 0 to 4,000,000,000 and to get into the table a player has to score more than the initial value.

The next step, **HIGHS** is called with no parameters and simply opens up its own window on the screen printing out all ten scores followed by all ten names, then closing the window. I haven't bothered inserting any checks for which screen mode the machine is in, but this procedure only really works well in Mode 8.

The function in this package is called **SCORE**, and it is called from Basic like this:

position = SCORE(new_score)

where **new_score** is the latest score being put in the table. The result returned as **position** is the position of this new score in the table and if it's zero it hasn't been high enough to register. Other values from one to ten give the actual position – ten makes it the highest score.

This is not intended as a 68000 machine code tutorial, but each section of the code and every call to

QDOS will be explained. I have not produced the hex of the assembled code as it's a very tedious job to type it all in, and on a machine as advanced as the QL one really ought to have an assembler! Personally I recommend the Metacomco product, the first professional one available, and by far the best so far. All the source follows the format required by that assembler.

utility vector **BP_INIT**, which needs A1 pointing to a table in the following format:

number of procedures (word)
offset from here of procedure start (word)
length of procedure name (byte)
characters of procedure name (bytes)

This is repeated for each procedure being linked and then:

a word of zero

number of functions (word)
offset from here of function start (word)
length of function name (byte)
characters of function name (bytes)

This too is repeated for each function, and the entire table is ended with a word of zero. As the 68000 family much prefers talking to even addresses, names which might result in an odd address (a name whose length is even) should be followed by a further byte (of arbitrary content) to re-align them.

First procedure

QDOS has two ways of being called: one is by a limited number of utility vectors, as in **SET_UP**, the

A procedure to initialise the high score table to ten equal scores of the passed parameter, with no names shown (a line feed is inserted).

DIGITS	LEA.L	BUFFER,A1	Point to buffer for string
	LEA.L	TENTAB,A2	And table of powers of 10
	MOVEQ	#0,D0	Digit counter
	MOVEQ	#0,D5	Leading zero counter
	MOVE.L	(A0),D2	Get score in D2
ONEDIG	MOVEQ	#0,D1	Iteration counter
DIGONE	SUB.L	(A2),D2	Subtract power of 10
	BCS.S	GOTDIG	Gone negative? Branch if so
	ADDQ.B	#1,D1	Else increment counter
	BRA.S	DIGONE	And repeat
GOTDIG	ADD.L	(A2)+,D2	Add power of 10 back in ONCE!
	ADDQ.B	#1,D0	Increment digit counter
	TST.B	D1	Is digit zero?
	BNE.S	NOTLDZ	No
	TST.B	D5	Have there been any valid digits?
	BNE.S	NOTLDZ	No
	CMPI.B	#10,D0	Is this the last digit?
	BEQ.S	NOTLDZ	Yes
	MOVEQ	#SPACE,D1	ASCII space in D1
	BRA.S	LDZYES	Jump to store it in buffer
NOTLDZ	ADDI.B	#‘0’,D1	Convert D1 number to ASCII digit
	MOVEQ	#1,D5	No more leading zeros
LDZYES	MOVE.B	D1,(A1)+	Store digit / space
	CMPI.B	#10,D0	Finished yet?
	BNE.S	ONEDIG	No; so repeat
	RTS		

Table of all the powers of 10 from 10^{-9} to 10^0

TENTAB	DC.L	1000000000
	DC.L	100000000
	DC.L	10000000
	DC.L	1000000
	DC.L	100000
	DC.L	10000
	DC.L	1000
	DC.L	100
	DC.L	10
	DC.L	1

more common way being through one of the 68000's **TRAP** instructions with certain values in pre-defined registers. Each call to QDOS can result in an error, and this is reflected in D0 on return. If D0 is not zero it indicates an error. These errors are all shown in the user guide, and have numbers from -1 (\$FFFFFFF) to -21 (\$FFFFFEF). To return to SuperBasic without error we must ensure that D0 holds zero.

The first procedure is **SET_TAB**. This starts at the **SETABLE** label. The first thing to do is collect the parameters passed to the procedure using the **CA.GTLIN** utility vector which gets all the parameters passed as long integers (32 bits) and puts them on the RI stack. The RI stack is the main interface with SuperBasic, and it uses A1 as its stack pointer. All arithmetic operations are carried out here, and all return parameters are put on this stack before returning. The RI stack pointer is **always** used relative to A6 – as is everything referenced by Basic.

CA.GTLIN returns with the number of arguments collected in D3, and each argument on the RI stack, first argument first. The important thing to remember when writing any sort of machine code is to make its position independent if at all possible – that's why all Basic things are relative to register A6 and all code is program counter relative. It may occasionally be slower – but it's always safer.

The first thing **SET_TAB** does is collect the arguments and check that there is only one. If something

is wrong then it leaves straight away with a bad parameter error while if everything is OK it collects the argument in D1 and uses a DBRA loop controlled by D2 to load this value into every score location. At the same time it puts a line feed into the first byte of every name in the table, so that the **PTNAME** part of the **HIGHS** procedure will show a null name. It then zeros D0 and returns to Basic. This procedure is very simple, but **HIGHS** is not. This starts at the **SHOWTAB** label.

The first thing this does is take advantage of the **UT.SCR** utility vector to open a window (using the SCR device) using the parameters supplied at **PBLOCK**. This routine returns with the channel ID in A0. Incidentally the channel ID has nothing to do with the #n in Basic – it's a long word that holds a cyclic tag and another number. This ID is needed in all communications with that channel. Window size is being traded on here, as it's been deliberately selected so that (in Mode 8!) every score entry starts on a new line.

The next thing to do is set D4 to zero and save the channel ID on the stack – probably bad planning as with a bit of alteration to the code there would be no reason to save it! Next set A0 (see what I mean!) to hold the address of the start of the first (highest) score entry. D4, which is incremented by 20 each time round the loop, is then subtracted from it, to point to the entry that's being dealt with at the moment. Next jump to a subroutine called **DIGITS** which con-

verts the long word score pointed to by A0 into an ASCII string so that it can be printed, using another utility vector – **UT.MTEXT**. First retrieve the channel ID *without* resetting the stack, so that it can be reached each time round the loop. Next use the **SD.TAB** trap to tab the cursor in the window to the 15th column (notice that D3 is set to -1 which implies infinite timeout, the routine will not return until it has succeeded. This comes in handy when using the multi-tasking capabilities of the QL). A routine called **PTNAME** comes next to print the player's name and finally add 20 to D4, check that the end of the table hasn't been reached yet, and loop back to continue printing entries.

If it's finished close the screen channel and reset the stack. There is no need to reset D0 as the **IO CLOSE** trap will do it.

The **DIGITS** subroutine is a standard decimal to ASCII routine using progressive subtraction to reach the result. Leading zeros are converted into spaces and each character is stored in **BUFFER**. The word before this is initialised to ten by the assembler, as we're always going to print ten characters.

The **PTNAME** routine uses the QDOS **IO.SBYTE** trap to print each character in turn on the screen, and it also uses register D2 as a DBRA counter to print up to 16 digits. If it finds a line feed before it gets to the end it prints this, finishing prematurely so that a line feed anywhere in the name field marks the end of the name.

Routine to print a string of up to 16 characters unless a LF is met

LINE_FEED	EQU	10	Ascii line feed character
PTNAME	LEA.L	TABLE+184,A2	Point to highest name
	SUBA.L	D4,A2	Subtract D4 counter
	MOVEQ	#15,D2	Up to 16 characters
MRNAME	MOVEQ	#IO_SBYTE,D0	Send character to channel
	MOVEQ	#-1,D3	
	MOVE.B	(A2)+,D1	Get character
	CMPI.B	#LINE_FEED,D1	If LF then print it and end
	BNE.S	NAMEPT	
	MOVEQ	#0,D2	Clear counter
NAMEPT	TRAP	#3	Do the SBYTE
	DBRA	D2,MRNAME	Repeat
	RTS		

Buffer for long integer to ASCII conversion.

SETABLE	MOVE.W	CA_GTLIN,A2	Get argument as a long integer
	JSR	(A2)	Exit if error
	BNE.S	EXIT_TAB	
	MOVEQ	#ERR_BP,D0	Just 1 parameter?
	CMPI.W	#1,D3	No – so leave with error BP
	BNE.S	EXIT_TAB	Get parameter in D1
	MOVE.L	0(A6,A1.L),D1	Put address of TABLE in A1
	LEA	TABLE,A1	Use D2 as counter
	MOVEQ	#9,D2	Store parameter in table
CLRTABLE	MOVE.L	D1,(A1)+	Put line feed in name field
	MOVE.B	#LINE_FEED,(A1)+	Make A1 point to next score
	ADDAL	#15,A1	Repeat 10 times
	DBRA	D2,CLRTABLE	No errors
	MOVEQ	#0,D0	Leave
EXIT_TAB	RTS		

Score function

All of this brings us to the SCORE function, the most complicated piece of code here. This first gets and checks its argument and then saves the RI stack pointer on the real stack. It then calls CHKSCORE (to be described next) and retrieves the RI SP and increments it by two to leave enough space for an integer result to be returned. CHKSCORE will have returned the position of the score in the table in register D1, which goes on the RI stack, saving the RI SP in BV.RIP (relative to A6!!), set the return type to integer (type 3) and return to Basic.

CHKSCORE has three main components: one to check where (if at all) the score should go in the table, one to move the other scores, putting the new one in its place, and one to get the player's name from the keyboard. These start at labels CHKSCORE, REPLACE and GETSCORE respectively.

D0 holds the score and D1 holds its potential position in the table. First check it against the lowest score – if it isn't greater then it isn't in the table, rendering an

immediate return. Otherwise use D2 to act as an offset into the table and check the score against each of the others in turn. Finally finding one which is greater, or reaching the end of the table, go to the replacement section. At this point D1 holds the correct position for the score and D2 contains 20 more than the offset required.

If decrementing D2 by 20 results in it being zero then the score must be the lowest in the table, and there's no need to move anything, otherwise move all the scores below down one entry (20 bytes, which happens to be five long words – take note!), thus losing the bottom one which is no longer valid.

More exciting

Once the table has been shifted put the score (still in D0) into its correct place in the table and call the GETSCORE routine for the player's name which is then inserted.

GETSCORE is rather more exciting. It saves D1 and A0 on the stack and uses the UT.CON vector to open the console device de-

scribed in DEFBLOCK. Again the channel ID is returned in A0 and this is used to print a prompt saying 'Enter your name:'. Next we use the QDOS fetch a line trap (IO.FLINE) to fetch a line from our console device. As it is a console device IO.FLINE allows normal cursor editing facilities when entering the name, but as a buffer length of 16 (in D2) has been specified, when the 16th character is entered that's it – no more editing. In fact the trap returns with a 'Buffer full' error, but we ignore this and finish the subroutine. This finishing involves closing the console device and retrieving A0 and D1 from the stack.

That's how simple it is to extend SuperBasic – the friendliness of QDOS and the beauty of the 68000 instruction set makes such code writing pleasant and enjoyable.

● The November issue of *Electronics & Computing* features more procedures and functions: allowing random access filing on micro-drives; better cataloguing facilities; cursor control and user defined graphics – which all QL reviews say the machine doesn't possess ... Oh yes it does – with a vengeance!

A procedure to show the current top ten high scores on the screen works rather better in Mode 8

LEN_ONE	EQU	20	Length of 1 high score entry
LEN_TAB	EQU	200	Length of high score table
SHOWTAB	LEA.L	PBLOCK,A1	Open screen device
	MOVE.W	UT_SCR,A3	
	JSR	(A3)	
	MOVEQ	#0,D4	Use D4 as offset pointer/counter
	MOVE.L	A0,-(A7)	Save screen channel ID
	LEA.L	TABLE+180,A0	Point to highest score
	SUBA.L	D4,A0	Subtract counter
	BSR.S	DIGITS	Convert this score to ASCII
	LEA.L	STRING,A1	Print this string of digits
	MOVE.L	(A7),A0	
	MOVE.W	UT_MTEXT,A3	
	JSR	(A3)	
	MOVEQ	#-1,D3	Tab cursor to position 15
	MOVEQ	#15,D1	
	MOVEQ	#SD_TAB,D0	
	TRAP	#3	
	BSR	PTNAME	Print the player's name
	ADDI.W	#LEN_ONE,D4	Increment counter
	CMPI.W	#LEN_TAB,D4	All scores done?
	BNE.S	TABSHOW	No – continue
	MOVEQ	#IO_CLOSE,D0	Close screen channel
	TRAP	#2	
	MOVE.L	(A7)+,D1	Reset stack
	RTS		Leave
Window definition block			
PBLOCK	DC.W	\$102	2 pixel wide blue border
	DC.W	\$602	yellow paper, red ink
	DC.W	\$182	window width
	DC.W	\$6E	window height
	DC.W	\$38	window X origin
	DC.W	\$4C	window Y origin

Routine to convert an unsigned long integer to ASCII

Long integer pointed to by A0. Leading zeros converted to spaces.

SPACE	EQU	,	
STRING	DC.W	10	Always 10 characters
String space			
BUFFER	DS.B	10	

A function to determine the position of a long integer score passed as a parameter within a table of 10 scores, and to return this as its result. The score will be correctly positioned in the table and the player's name will be collected. Other scores/names will be moved as appropriate.

FN_SCORE	MOVE.W	CA_GTLIN,A2	Get argument as long integer
	JSR	(A2)	
	BNE.S	EXIT_SCO	
	MOVEQ	#ERR_BP,D0	
	CMPIW	#1,D3	Just 1 parameter?
	BNE.S	EXIT_SCO	No – leave with error BP
	MOVE.L	0(A6,A1.L),D0	Get parameter in D0
	MOVE.L	A1,-(A7)	Save RI SP
	BSR.S	CHKSCORE	Check D0 score
	MOVE.L	(A7)+,A1	Retrieve RI SP
	ADDQ.L	#2,A1	Only returning 16 bits
	MOVE.W	D1,0(A6,A1.L)	Put D1 on RI stack
	MOVE.L	A1,BV_RIP(A6)	And save the pointer for Basic
	MOVEQ	#RET_INT,D4	Returning an integer
	MOVEQ	#0,D0	With no errors
EXIT_SCO	RTS		
CHKSCORE	MOVEQ	#0,D1	Counter
	CMP.L	TABLE,D0	Is D0 > lowest score?
	BHI.S	CHECKTAB	Yes
	RTS		No
CHECKTAB	MOVEQ	#LEN_ONE,D2	Table entry length
	ADDQ.W	#1,D1	Increment counter
	MULU	D1,D2	Real offset into table
	CMPI.B	#10,D1	End of table?
	BEQ.S	REPLACE	Yes
	CMP.L	TABLE(D2.L),D0	Check D0<=score
	BHI.S	CHECKTAB	No
REPLACE	LEA.L	TABLE,A0	
	SUBI.W	#LEN_ONE,D2	
	BEQ.S	SCOREGET	Find offset from start of table
SHIFTAB	MOVE.L	LEN_ONE(A0),(A0)+	Move all other entries down 20
	SUBQ.W		
	BNE.S	SHIFTAB	
SCOREGET	MOVE.L	D0,(A0)+	Save new score in table
	BSR.S	GETSCORE	Get player's name
	MOVEQ	#3,D2	
	LEA.L	INPBUF,A1	
TABSHIFT	MOVE.L	(A1)+,(A0)+	Move name into table
	DBRA	D2,TABSHIFT	Repeat
	RTS		
BUF_LEN	EQU	16	
GETSCORE	MOVEM.L	D1/A0,-(A7)	Save D1 and A0
	LEA.L	DEFBLOCK,A1	
	MOVE.W	UT_CON,A3	Open console device
	JSR	(A3)	
	LEA.L	MESSAGE,A1	Print prompt
	MOVEQ	#IO_SSTRG,D0	
	MOVEQ	#-1,D3	
	MOVEQ	#MESEND_MESSAGE,D2	
	TRAP	#3	
	MOVEQ	#IO_FLINE,D0	Fetch an editable line
	MOVEQ	#BUF_LEN,D2	
	LEA.L	INPBUF,A1	
	TRAP	#3	
	MOVEQ	#IO_CLOSE,D0	Close console device
	TRAP	#2	
	MOVEM.L	(A7)+,D1/A0	Restore D1 and A0
	RTS		And leave
DEFBLOCK	DC.W	\$102	2 pixel wide blue border
	DC.W	\$602	Yellow paper, red ink
	DC.W	\$18A	Width
	DC.W	\$E	Height
	DC.W	\$3C	X pos
	DC.W	\$F0	Y pos

The high score table

TABLE	DS.B	LEN_TAB	
INPBUF	DS.B	BUR_LEN	
MESSAGE	DC.B	'Enter your name:'	Prompt
MESEND	EQU	*	
	END		



HAVE YOU REGISTERED YET?

If you are a QL owner you can register your machine with us and we will send you details as we receive them, of up to the minute information concerning your machine.

Also we will offer you a substantially reduced subscription offer on *QL User* magazine. Just tear the card, fill in the details and drop it in the mail. This is a free service to all QL owners.

Remember if you have already registered there is no need to send this card.

Name _____

Address _____

Occupation _____

My QL serial number is _____
(on the base of the machine)

Other computers I own, please specify:

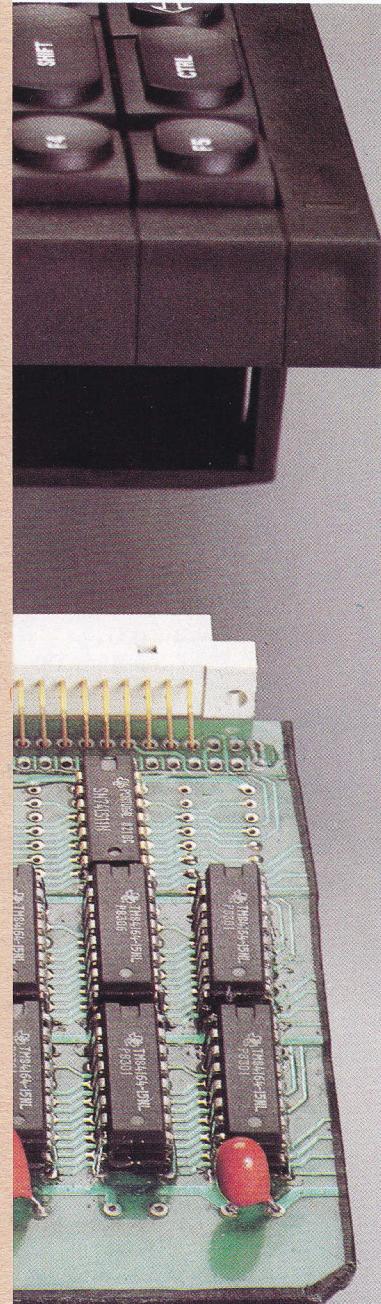
- | | |
|-----------|--------------------------|
| Spectrum | <input type="checkbox"/> |
| IBM PC | <input type="checkbox"/> |
| Commodore | <input type="checkbox"/> |
| ZX81 | <input type="checkbox"/> |
| Dragon 32 | <input type="checkbox"/> |
| Apple II | <input type="checkbox"/> |
| Oric | <input type="checkbox"/> |

Please name others _____

computing power to
your existing system.

The majority of home computers have an in-built potential of using more memory than they are actually fitted with. The new memory expansion board from Simplex Data Ltd, takes full advantage of that wasted potential enabling you to expand your

Memory More



er system — creatively! Two quarter megabyte memory expansion boards can be slotted together to expand your system to a further 512k bytes, giving you half a megabyte more. Costing only £198.00 (inc. VAT, postage and packing), this unit is simple to use, just remove the port cover located at the end of the Sinclair QL Keyboard and slot the unit in, replacing the port cover after insertion.

Exclusively available from Simplex Data Ltd.

TO: SIMPLEX DATA LTD, FREE POST, 432 GREENFORD RD, GREENFORD, MIDDLESEX, UB6 8SG

Please send me: _____ (qty) QL Memory Expansion Unit(s) at £198.00 (inc. VAT, Post and Packing) I enclose Cheque/Postal Order for £ _____

Name: _____

Address: _____ Tel: _____

1 year warranty included. Please allow 28 days for delivery

Payment by ACCESS or BARCLAYCARD phone 01-875 7531

QUEST ANNOUNCE 6 WAYS OF GETTING THE MOST FROM YOUR SINCLAIR QL

-
- CP/M Operating System
 - Business Software
 - Disk Storage
 - Mass Storage
 - Expansion Unit
 - Memory Add-ons
-

**Already Available - all essentials for
your QL range... ■ Sinclair QL's
■ Printers ■ Monitors ■ Cables, paper etc**

For priority information pack write, telephone
or see us at COMPEC stand 938

Quest International Computers Ltd.
Quest House, School Lane, Chandlers Ford, Hants SO5 3YY
Tel: Chandlers Ford (04215) 66321 Telex: 47326

The Computer Professionals
QUEST