

# ORIC OWNER

INCORPORATING THE TANSOFT GAZETTE

ISSUE 1

JAN/FEB



INSIDE THIS ISSUE:

Forth  
Spreadsheet  
Oric-Trek  
in colour

SPECIAL FEATURES:

We talk to  
Dr Paul Johnson  
How to get the most  
out of your Oric



# ORIC OWNER

Issue 1 Jan/Feb

## Features

Interview:  
Dr Paul Johnson p.10

How to get the most  
out of your Oric 1 p.20

## Regulars

Editor's Comment p.4

Captain Tanex p.23

Construction Corner p.33

Club Page p.35

Oric Quickies p.36

I/O Page p.38

## News

News Brief p.5

Software Scan p.8

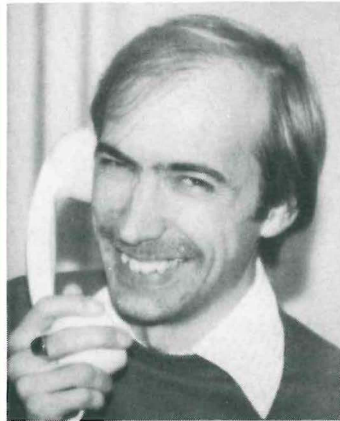
## Software

And Forthly . . . p.14

Oric Trek p.16

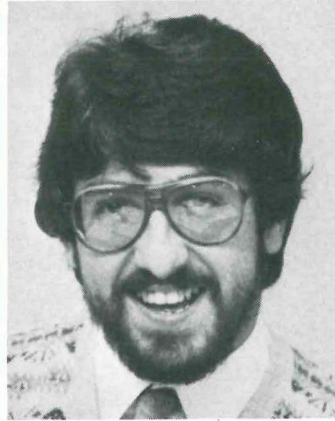
Awari p.24

High Speed sort p.30



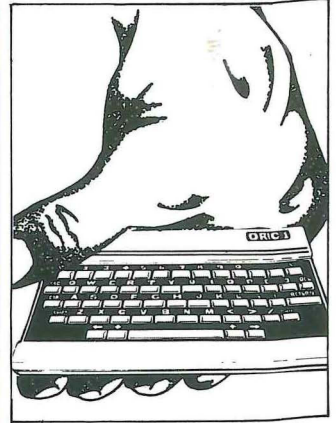
### An interview with Dr Paul Johnson

*Dr Johnson tells us of his plans for the Oric.*



### Editor's Comment

*Editor Paul Kaufman outlines the magazine's aims and philosophy.*



### And Forthly

*Our regular column for all Forth enthusiasts.*



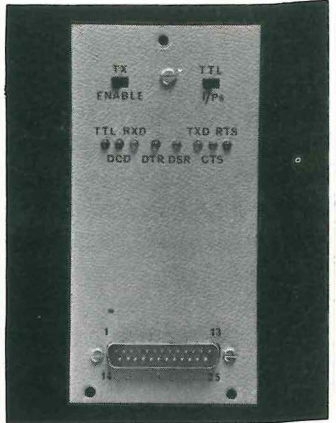
### Oric Trek

*A superb Star-Trek game!*



### Captain Tanex

*Can our intrepid hero uncover the secret of the eye of Tanger?*



### Construction Corner

*Microtan serial input/output option buffer and status indicator.*

## Oric Owner Magazine

Editor: Paul B. Kaufman  
 Graphics and Artwork Manager:  
 Carolyn Groeneveld  
 Technical Consultant: Dr Paul  
 Johnson  
 Printers: Eastern Counties Printers  
 Ltd, Ely, Cambs.  
 Photographic Equipment: Minolta

Oric Owner is published at  
 bi-monthly intervals by Tansoft Ltd,  
 Reg. No. 1632070  
 3, Club Mews, Ely, Cambs.  
 CB7 4NW  
 Advertising Rates are available on  
 request.

# Editor's Comment

## Issue 1 Jan/Feb

Welcome to issue 1 of **Oric Owner**, the magazine for people who have bought or are about to buy an Oric 1 computer.

The aim of this magazine is to keep you informed of the latest developments in hardware and software for the Oric, give you plenty of programs to try, and to open our pages to readers with anything interesting to say about subjects concerning the Oric.

On the Hardware side we intend to give you up-to-the-minute technical information on the latest Oric add-ons as they appear, whether they are produced by Oric Products themselves or from other sources. We will print improvements or modifications to existing hardware to allow you to "tune up" your system to give the best performance and hope to bring you constructional ideas to try if you are electronically inclined.

Software is obviously a very important factor for any new computer and so we intend to print as much software as possible. Apart from giving you programs to type in on your Oric, we will look closely at some of the books which will start appearing soon.



We intend to bring you a number of tutorial articles designed to give you in-depth information on some of the points not covered in the Oric manual, such as advanced graphics or programming the Oric in machine code. Interviews with important Oric personalities will feature from time to time plus there will be plenty of news and gossip to amuse you.

Obviously, as with any new magazine, we have a great need for programs and articles to print. If you have any ideas for articles or have written any interesting program or even discovered a new "undocumented" feature we would be delighted to hear from you. Don't forget our I/O page for your letters and the Club page for user groups.

You may have noticed written on the front cover that this magazine incorporates the Tansoft Gazette. The Tansoft Gazette was the original magazine published by Tangerine for the Microtan System. As there are many similarities between the Oric and the Microtan Systems we will continue to publish articles for the Microtan as long as people send them in.

---

*We are in constant need of good quality articles, programs and hints. If you have any that you think other Oric Owners would be interested in we would be delighted to see them. We do pay for articles printed and are able to help with layout and design of articles where needed. We also look forward to your correspondence for the I/O page.*

---

# News Brief



## Oric Launch Surprise



At the official launch party of the Oric-1 computer on Thursday 27th of January at Oric's Coworth Park Headquarters, Peter Harding, Oric Sales Director, an-

nounced 6 major deals which will bring the Oric into the High Street. Greens will be selling the Oric through their outlets around the country. Chinon will be taking a large chunk of the European export market. Micro-Peripherals who have over 200 distributors will be aiming for the computer enthusiast.

W. H. Smith will be taking Orics from the beginning of April and will be selling them mainly through their larger stores. All of Laskys stores will be selling Orics from the end of February. The Micro C branch of Currys will also be selling Orics from the end of February. These deals total over 200,000 Orics, the majority of which are 48K models.

## New Software House

Firefly Software is a new software house formed specifically to support the Oric. A full range of business and leisure software is under development and a consultancy service is also available.

Full details from Firefly, 01-200 6209.

## New Titles from Tansoft

Tansoft have announced the first titles of their range of Oric Software.

Zodiac is a 36K adventure game that takes place in a strange land populated with weird people and animals. You have to search for the 12 cunningly disguised signs of the zodiac which will lead you to valuable treasures. Zodiac costs £9.99 including V.A.T., postage is 40p.

Oric-Chess plays a powerful game and uses high resolution graphics and features 5 levels of play and allows Castling and En-Passant. Oric-Chess costs £9.99 + 40p postage.

# News Brief



## Intelligent Orics?

Their first business package is Oric-Base, a full database retrieval system, which allows the setting up of complex datafiles, selection and sorting of data and the printing of full reports. Oric-Base costs £15 and includes a full manual, postage is 40p.

Intelligent Software in Somerset have announced 4 new titles for the Oric to be released towards the end of February.

These are an assembler package, a graphics toolkit, a multi-games tape, and a multi-games tape in machine code. Full details from Higher Combe, Combe Florey, Taunton, Somerset.

## Oric Trader

Taipan is a 'Trading Adventure' game based around the China seas in the 19th century. You are involved in buying, selling and transporting goods from one part to another. There are plenty of hazards such as storms, bankrupt customers and the danger of strong-arm tactics from the Emperor. More info from Jaysoft 22 Dane Acres, Bishops Stortford, Herts.

## Full Manual Ready

Oric Products inform us that the full instruction manual has now been completed and is being sent out to all those who only have the preliminary manual and has all those annoying misprints corrected.

---

**News..... News..... News..... News.....**

---

# News Brief

## Salamander Software

Based in Brighton, newly formed Salamander Software are putting together a range of Software for the Oric. Under development are a number of games including Backgammon, Othello, Star-Trek, a maze game and a complex adventure game. They also are intending to release Utility Software in the near future.

More details on:  
(0273) 771942

## Microtan Sale

Negotiations are almost complete for the sale of Tangerine's Microtan range to Microtan Hardware in Dulwich. This will allow Tangerine to concentrate on the Oric computer while allowing the Microtan range to continue to be supported. All requests for technical information, orders, or repairs should be directed to Microtan Hardware on:  
01-693-1137

**ORIC**  
IS INDEED  
INTERNATIONAL  
Trade Distribution  
Benelux

**MICROWORLD**  
**EUROPE**

297 BRUSSELSSE STEENWEG  
B1950 KRAAINEM/BRUSSELS  
PUBLIC ENQUIRIES: (09/00) 02 7675165

## ORIC-SOFTWARE

**16K LUNAR LANDER and ASTEROIDS**  
ONE TAPE/TWO ACTION-PACKED GAMES WITH HIGH RESOLUTION COLOUR AND SOUND

**PLUS** Fully explained program listings - Enjoy the games then learn the tricks of the trade!

£5.00

**16K ASSEMBLER and ASSEMBLER'S**  
**and NOTE-PAD.**

**48K** Auto line-numbering; up to 6 character labels; hex or decimal numbers with simple arithmetic on operands; full syntax checking with error reports; two parse assembler maintains and updates list file; comprehensive documentation AND A4 size pad specially laid-out for clear and rapid assembler programming.

£9.95

(Pads available separately £1.30 inc. p&p)

DURELL SOFTWARE  
HIGHER COMBE · COMBE FLOREY  
TAUNTON · SOM TA4 3JF

News..... News..... News..... News.....

# Software Scan

## Software Conversion

One of the problems facing anyone who buys a computer that is new on the market is getting enough Software to run it. We are fortunate with the Oric 1 that it is based on the 6502. This makes a number of programs already written for the Pet, Apple, Microtan, Vic etc. readily convertible onto the Oric. This is particularly easy if the original computer runs Microsoft Basic. The only problems are likely to be if the program in question uses machine specific features such as Hi-Res graphics commands, monitor sub-routines and peeks and pokes. By careful examination of the program it is often possible to work out what the commands are doing and to convert them to the appropriate Oric commands.

## Acorn and Sinclair?

You will sometimes run into problems when converting programs from machines that don't use Microsoft Basic. Sinclair and Acorn have both written their own versions of Basic which although on the surface appear to be fairly standard you will find subtle differences in the way they work. Differences in the way the computer handles floating point numbers (numbers with a decimal point), how numbers are represented internally in the machine and how numbers or characters are stored in arrays can all cause a fairly simple program to give the most unexpected results.



How the machine handles truth values can cause GOTO's and GOSUB's to go wildly astray if not converted correctly. Oric's Basic uses 0 to represent false and -1 to represent true. This can easily be proved with the following routine:

```
10 A = 3
20 B = 4
30 C = 3
40 PRINT (A = B):REM
DOES A equal B?
50 PRINT (A = C):REM
DOES A equal C?
60 END
```

When executed this program will print 0 at line 40 and -1 at line 50, showing that A does not equal B but A does equal C. However some machines use 0 to represent false and +1 to represent true. Unless a value is negative, Oric Basic won't recognise a true value. So although the program will work correctly if all logic tests come out as False, the Oric won't

be able to recognise them if they are true, this will usually cause the program to go haywire or to go into endless loops when least expected so BEWARE!

Getting back to software for the Oric you will be pleased to know that several software companies are gearing up to start producing software for games and business use. Many of the companies are already well known in the ZX81 and Spectrum fields. Ourselves (Tansoft that is) are completing a number of packages which you should start hearing about very shortly.

Apart from Forth, which some of you will already have, we are looking at some other languages such as PASCAL and LOGO. An extended Basic, which gives the equivalent of B.B.C. Basic is almost ready to go into production - keep in touch for more details. We also

have a number of games under development including Zodiac - a 36K adventure game, a Flight Lander and hopefully a chess game. We also have a Data Management program called, provisionally, DBASE. This enables you to enter data very simply into the machine and to manipulate it with the least amount of fuss. You can sort records, tot figures up, keep running totals and print nicely formatted reports. We have started to use it to keep track of royalty payments to people who have written software for us - so it must be good!

## Coming Soon!

In the next issue I hope to examine some of these packages in more detail and also to bring you news of what other software companies are doing for the Oric.



# Software for the Microtan

### **COMPILED INTEGER BASIC £15**

Faster than Forth, special extensions to improve realtime performance.

### **CRUNCHCALC SPREADSHEET £15**

Fully featured financial modelling program. Needs **TANRAM**.

### **CRUNCH TRANSPORTER £15**

Complete solution to the standard transportation problem. Automatic perturbation and deperturbation.

### **CRUNCHMATH SCIENTIFIC CALCULATOR £10**

Evaluates mathematical expressions. 26 variables store values or complex formulae. Includes standard maths functions e.g. SIN, EXP, FACTORIAL, etc. Extra functions include: combinatorials, permutations, binomial, Poisson, normal and standardised normal distributions.

**TRANSPORTER AND CRUNCHMATH ARE ALSO AVAILABLE FOR THE ORIC**

Send cheques to:

**CRUNCH COMPUTER SYSTEMS LTD,  
76 VICTORIA RD., SWINDON, WILTS**

**The lottery winner is Mr. T. C. Haines of London. A compiler is on its way.**

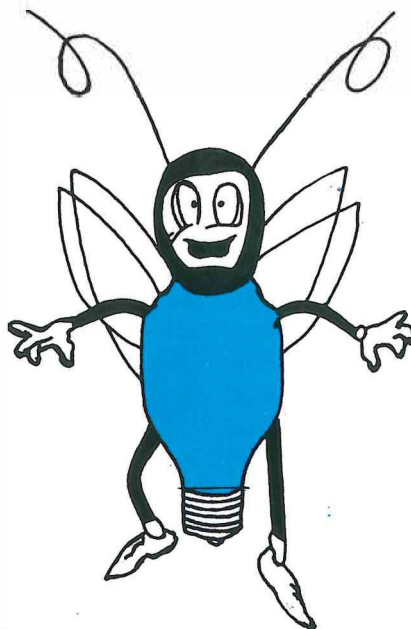
## **FIREFLY SOFTWARE** **Software and Consultancy** **Services**

Programming undertaken for the Oric-1, Microtan and Dragon computers.

Professional work by experienced analyst/programmer, author of Zodiac/Dbase and contributor for Tansoft Gazette.

Programs listed 1p per line.

Write for details of above to:



Available March: An adventure game for the Oric-1 with proper descriptions, real English commands, for those who are fed up with "GET GOLD" and "I don't understand"!

Coming soon: A series of games, business packages, and utilities for the Oric.

Firefly Software,  
8, Poolsford Road,  
London NW9 6HP.

# An Interview with Dr Paul Johnson

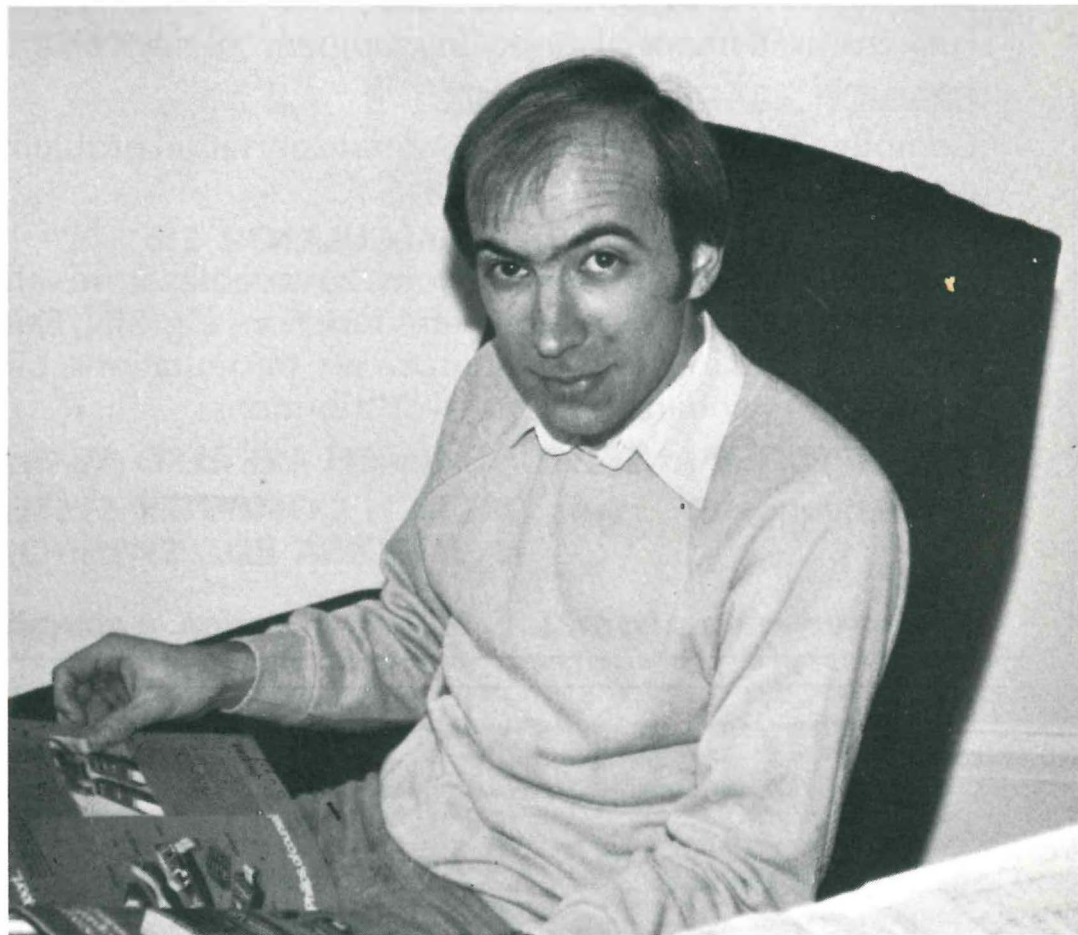
**Q** *Let's start by saying that you are the main figure behind the Oric-1, particularly the ULA. Why do you attach so much importance to the ULA?*

**A** The Gate Array (or ULA) looks after all the video signals. It replaces a large number of discrete and TTL components. Using the Gate Array means that you can save a lot of board space and cut down on component and production costs.

A major advantage of the Gate Array is that you can make the Video far more sophisticated than you could using discrete chips. It allows you to go "over the top" — you can make it do all sorts of things that you could not do any other way. Another advantage is that by using the Gate Array to handle all the memory control signals we can exploit low cost DRAMs (Dynamic RAMs). Without the Gate Array the 16K Oric would cost about £250.

**Q** *In what way do your methods of design differ from those of other Micro manufacturers?*

**A** We took a lot more care over it. We designed the Logic, we breadboarded it, we thoroughly checked it out. We went over to the ULA manufacturers (C.D.I. in California) and sat down with their engineers and checked every stage of development from the logic design, layout, and computer



**“It allows you to go ‘over the top’”**

simulations, even testing the first prototype chips and designing test programs.

Another aspect of the design was the care taken to ensure that the logic circuitry took account of the limitations of the Gate Array. We designed around its limitations to exploit it to the utmost. Whereas other manufacturers couldn't have done so with their Gate Arrays or else they wouldn't have experienced

the problems that we have heard so much of lately. They either used a Gate Array that wasn't up to the job or else it was simply poor logic design.

**Q** *In other machines that use ULAs, due to the complexity of the design, it is often possible to discover new features or ways of using the machine that weren't envisaged in the original design. Have you found any such features with the Oric?*

**A** There are going to be lots of features of the ULA that weren't intentionally designed in. It is almost inevitable that when you design something as complex as a Gate Array someone will find a hidden feature that can be of some use. The same thing has happened with processors, people have found new instructions that aren't documented.

## Features

The danger is that it is never possible to be sure exactly what's going on inside the chip and you could get some strange side effects.

**Q** *Was there any particular reason why you chose the 6502A in preference to, say, the Z80 or 6809?*

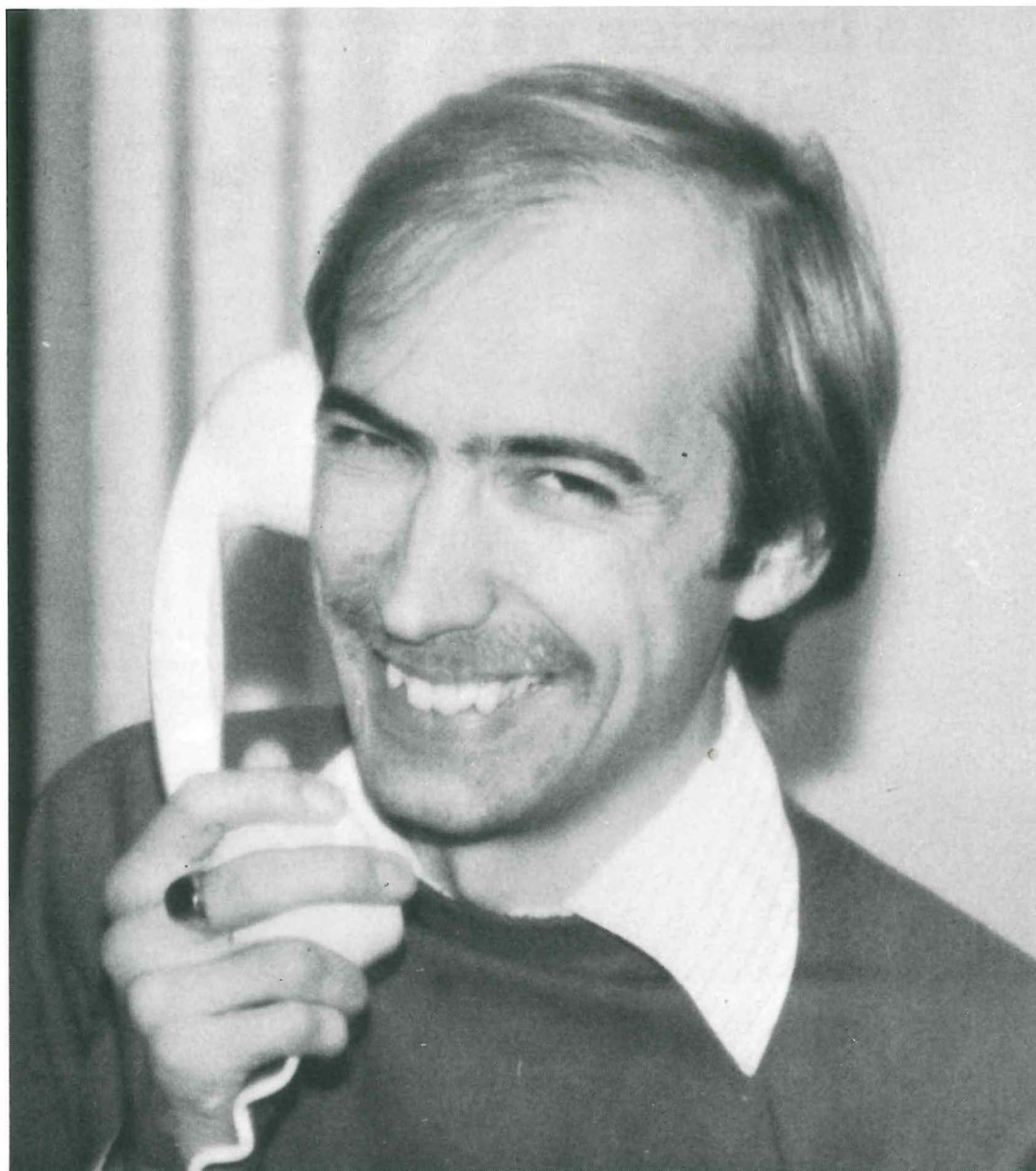
**A** It has always been a better processor to use in a low-cost product. The Dragon uses a 6809 which has a very similar bus structure to the 6502. We could have used a 6809 in the Oric but there's just not enough software around at the moment to make it worthwhile.

**Q** *What is the difference between the 6502 and the 6502A?*

**A** The 6502A runs twice as fast as the 6502 but in fact we're not running it twice as fast. We're using the 6502A because we're doing some funny things with the clock signal and the ordinary 6502 is not guaranteed to work under those conditions. You could knock the speed up a bit but of course you'd start losing the video and the dynamic RAMs. So unless you had the high speed versions of everything there would be no advantage in running the 6502A at full speed.

**Q** *Will any other processors be available in the future?*

**A** We might do a 6809 version of the Oric in small numbers for the real enthusiast. We would then be able to put FLEX on it. Obviously because of the small quantities involved the



### “There's not enough software for 6809”

price would have to be higher.

**Q** *In the Oric brochure a lot of fuss is made about the Tangerine cassette format. Why do you think it is so good?*

**A** It's been proven over about three years with thousands of systems and we have had very little trouble with it. Obvi-

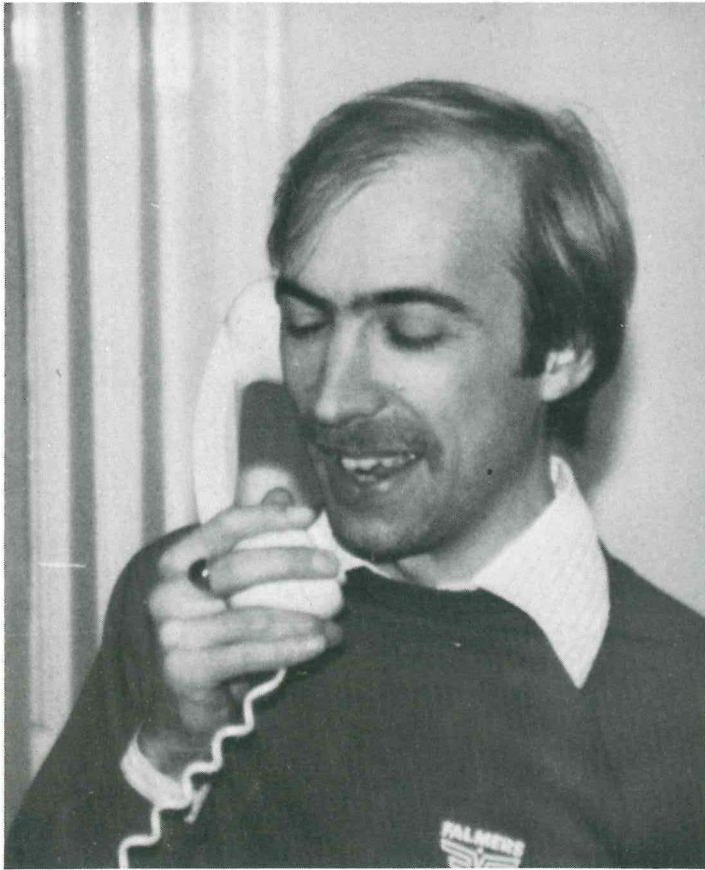
ously at the end of the day the reliability of a cassette system depends on the quality of the Owner's equipment. He may have the best cassette deck in the world but, if the connections aren't right or if the impedances don't match then he's going to start having difficulties. The hardware in the Oric is so simple that there is very little that can go wrong with it. If a chap phones up and says he can't read prog-

rams then 99 times out of 100 the problem is at the cassette recorder end.

**Q** *Which peripherals for the Oric are we likely to see next? How soon can discs be expected?*

**A** The next peripheral you will see for the Oric is the Modem, followed very quickly by 5¼" discs.

### An Interview with Dr Paul Johnson



**Q** *Is there anything you can say about the Modem? I know that one end plugs into the back of the Oric and the other end has a G.P.O. lead.*

**A** It is designed to be a direct-connect Modem to the Post Office line and will use the Viewdata standard so that you can communicate with Micro-Net 800, closed user groups and who knows, Oric may set up their own database in Ascot.

**Q** *Will you be able to put Orics back to back over the phone-line, if the baud rates are programmable?*

**A** YOU may be able to. We've not decided yet. It depends on what chip we finally use. There is a chip that would allow people to have their Orics connected back to back on the phone line. It's a more expensive chip and would probably need its own power pack as the Oric probably wouldn't be able to power it directly.

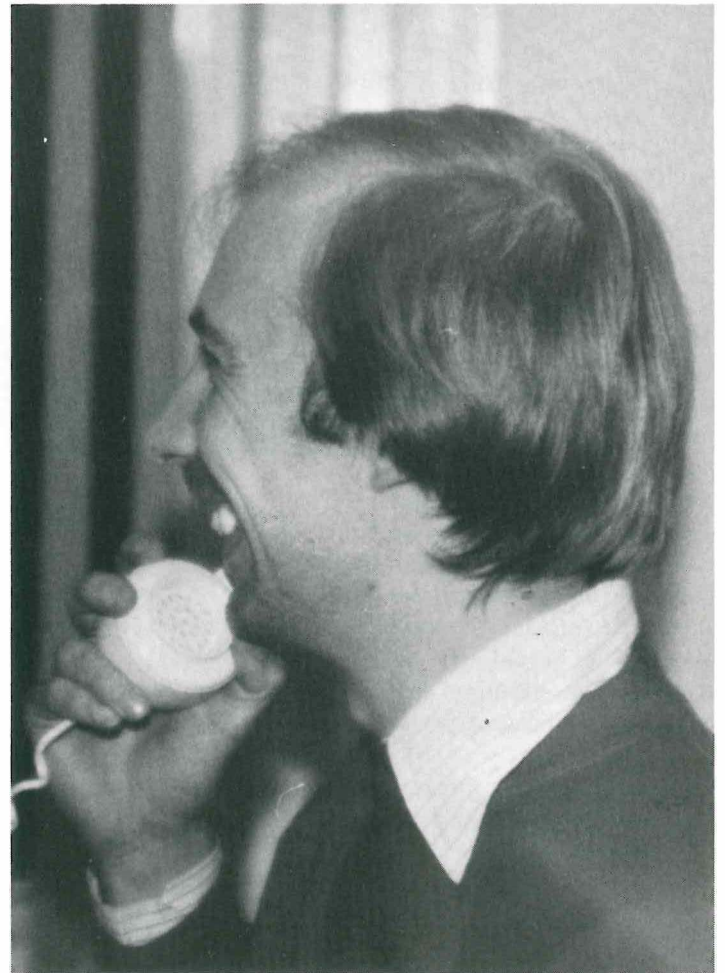
**Q** *What about 3 inch discs?*

**A** We've got to wait until we see who wins. There are three formats at the moment 3", 3½" and

100mm. It would be disastrous to back the loser. What's happening is that 5¼" drives are coming down in price so dramatically, because of the Micro-Floppy, that I think it's fairly safe at the moment to go for the 5¼". It gives people plenty of storage, although at a slightly higher price than Micro-Floppies anyway there's no real problem. We'll go over to Micro-Floppies when and only when, we can see which one is going to be the clear winner.

**Q** *Can you tell us anything about your future plans for the machine?*

**A** Sure as eggs are eggs! We are going to have a product range for the Oric. The marketplace has decided that small computers are now consumer products. There's only one way to sell consumer products and that's through retail outlets and big retail chains. A retailer won't be happy just selling a box with a computer in it. He'll want to sell software, cassette recorders, discs, printers, and so on. So it is essential that we supply him with a full range of accessories. I can't tell you at the moment which add-ons we're working on but you can be assured that they'll make the Oric more powerful than machines costing even treble the price.



*"Thank you very much Dr Johnson, we will wait with much interest to see how your predictions turn out."*

THE FLEXIBLE COMPUTER SYSTEM FOR THE FUTURE

# ORIC 1

CREATED BY

# TANGERINE

How can Tangerine promise you a professional computer for only £129.00? Because, unlike most computer builders, we have designed the U.L.A. ourselves.

This makes the ORIC-1 substantially more reliable and versatile to work with and what's more, at £129.00, you get a professional system well below the price of

all leading manufacturers.

We can give you prompt service, quality, reliability and full technical backup: The ORIC Computer System will guarantee you that and more:



- 8 colour graphic display (8 foreground + 8 background)
- 40 character by 28 line colour text display
- High resolution graphics (240/200 pixels) 240 across screen, 200 down
- 96 User defined graphics symbols
- Microsoft BASIC software
- 6 octaves of music with Hi-Fi output and 4 preset sound effects – Shoot, Explode, Ping, Zap
- Centronics printer interface (compatible with a whole range of standard printers)
- Optional Communications Modem (allowing access to 200,000 pages of Prestel and direct link with other computers)
- Typewriter style keyboard
- Professionally written user manual by well-known computer authors
- Oric Owner Magazine included with each ORIC 1 purchased
- Tan-Forth supplied free with every mail order 48K Model
- Extended Basic (BBC etc) available soon
- Full range of business and leisure software coming shortly.

(Please allow 28 days for delivery) Subject to availabilities.

## Order your ORIC-1 direct from the designers

BY POST: You can pay by cheque, postal order, ACCESS – BARCLAYCARD-VISA

BY PHONE: Just ring our telesales number ELY (0353) 2271/2/3/4

Please delete/complete as applicable.  
 \*I enclose a cheque/p.o. payable to TANGERINE  
 COMPUTER SYSTEMS LTD. For £  
 Please charge my Access, Barclaycard, Visa No.

Name \_\_\_\_\_  
 Address \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

If you require a VAT receipt please tick

Please send me a full colour brochure

TANGERINE COMPUTER SYSTEMS LTD. 3 Club Mews, Ely Cambs CB7 4NW

Item	Qty	Price Inc. VAT	Total
ORIC-1 16K RAM	1	£129.00	1
ORIC-1 48K RAM	1	£169.95	1
ORIC Communications Modem	1	£79.00	1
ORIC Owner Magazine (Bi-monthly)	1	£10.00	1
Postage and packing	1	£5.95	1
		<b>TOTAL £</b>	

# And Forthly...

One of the major advantages of using Forth on a micro is the transportability of programs from one machine to another. Most of the Forth programs described in this column should work quite happily on the Oric, the Microtan, or any other machine that uses FIG-FORTH.

As a simple exercise in writing Forth programs that are actually useful I have attempted to create a spreadsheet program that although fairly basic in concept, allows expansion and improvement in any desired direction. A spreadsheet program simulates on the computer all the rows and columns of a ledger sheet such as used by accountants or bookkeepers. The computer automates the tasks of entering the figures, calculating totals and printing reports.

## The Program

Variables FIRST and LAST are defined for later use as indicators to which columns are to be displayed on the screen. ARRAY is a defining word that is used to create a two dimensional

array of any desired size. This is used in the next line to create a twelve by twelve array named SPREAD. 2FETCH is a word which returns the address of any supplied coordinates. To initialise the array 2CLEAR is used. COLS, SPREAD and DISPLAY are used to format and print the SPREAD array on the screen.

One of the more useful features of this program is the ability to add up rows or columns of figures and store the results at the sides of the array. ROWCALC and COLCALC perform this function on the supplied row or column.

The remaining FORTH words are the higher level commands that are necessary to run the program. ENTER allows data to be stored in the array. INSTRUCTIONS simply prints the various program options and asks for user input. SHOW prompts the user for the start and end columns to be displayed (between 0 and 11). Note that row 11 and column 11 are used to keep totals in and data should not be entered

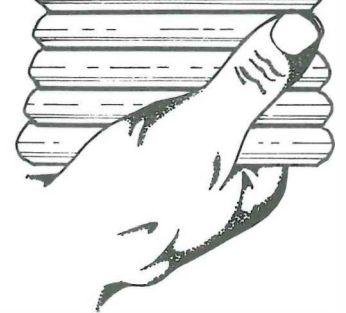
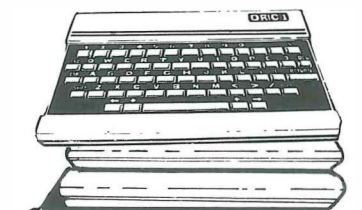
into them directly. Array position 11,11 is used to display the grand totals of rows or columns. SPREADSHEET is the complete program and simply loops around acting on the users input. The IF . . . ENDIF statements could probably be replaced by a CASE statement for greater efficiency.

## Expansion

There are several ways of improving the program which may become obvious after being in use for a while. Apart from simply tidying up the code the array could be made larger to accommodate more data. A facility to allow names of rows or columns could be added, such as months or account dates. Extra calculation facilities could be added such as percentages or multiply and divide and perhaps an option to allow the user to define his own arithmetic treatments.

## Finally

I look forward to seeing any extensions or improvements you may make to the program.



# Software

## SCR # 1

```
0 (FORTH SPREADSHEET 1 OF 3 . Paul Kaufman Dec.82)
1 0 VARIABLE FIRST 0 VARIABLE LAST
2 : ARRAY <BUILDS OVER OVER , * 2 * ALLOT DOES> 4 + SWAP 2 * + ;
3 12 12 ARRAY SPREAD
4 : 2FETCH -2 SPREAD @ * + SPREAD ;
5 : 2CLEAR 12 0 DO I 12 0 DO
6 DUP I 2FETCH 0 SWAP ! LOOP DROP LOOP
7 : COLS 4 SPACES LAST @ 1 + FIRST @ DO I 4 .R LOOP;
8 : .SPREAD (Print Spread array)
9 COLS CR CR 12 0 DO I I 2 .R 2 SPACES
10 LAST @ 1 + FIRST @ DO DUP I 2FETCH @ 4 .R
11 LOOP CR DROP LOOP;
12 : DISPLAY ( First Last)
13 LAST ! FIRST ! .SPREAD;
14 -->
15 (More on next screen)
```



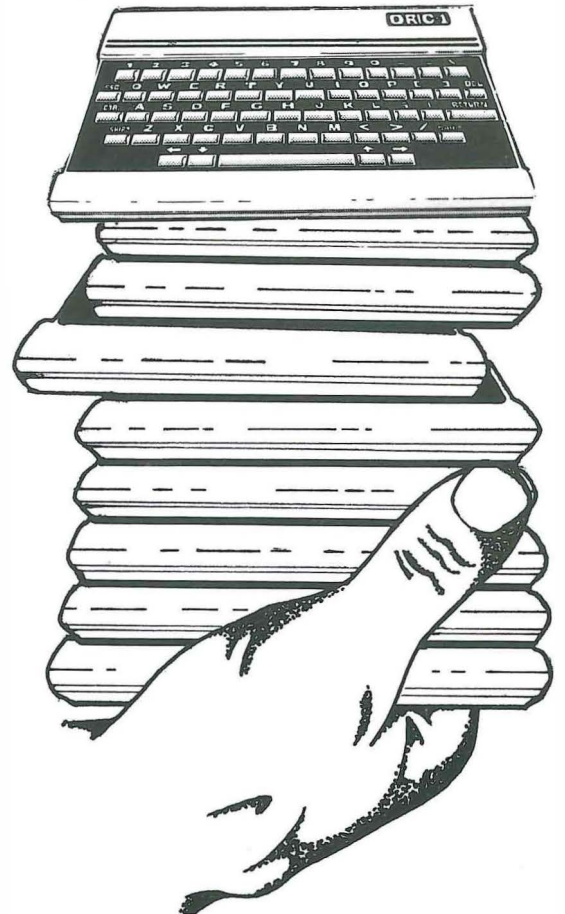
## SCR # 2

```
0 (Forth spreadsheet screen 2 of 3)
1 :ROWCALC ( x ROWCALC)
2 DUP 0 SWAP 11 2FETCH ! 11 0 DO DUP DUP I 2 FETCH @ SWAP 11
3 2FETCH +! LOOP DROP FIRST @ LAST @ CLS DISPLAY ;
4 : COLCALC ( y COLCALC )
5 DUP 0 SWAP 11 SWAP 2FETCH ! 11 0 DO DUP DUP I SWAP 2FETCH @
6 SWAP 11 SWAP 2 FETCH +! LOOP DROP FIRST @ LAST @ CLS DISPLAY;
7 :ENTER ( Val x y ENTER )
8 CR ." Enter Value" IN# CR ." Enter Row "IN# CR
9 ." Enter Column " IN# CR 2 FETCH ! FIRST @ LAST @ DISPLAY ;
10 : INSTRUCTIONS ."1)Clear Array 2)Enter Value"
11 CR ." 3) Display Array 4)Calc Row " CR ." 5)Calc Column"
12 ." 6) End Program " KEY DUP EMIT ;
13
14
15 -->
```

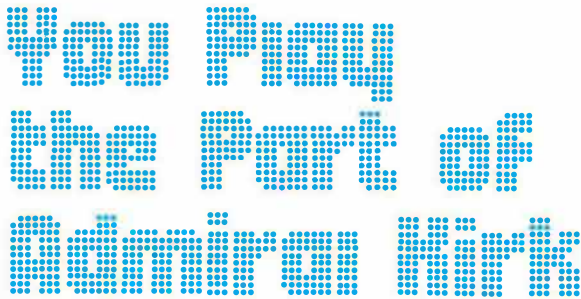
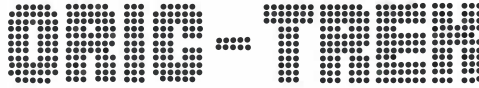


## SCR #.3

```
0 ( Spreadsheet 3 of 3 )
1 :SHOW CR ."Enter Start Col " IN# CR." Enter End Col "
2 IN# CR DISPLAY ;
3 : SPREADSHEET 2CLEAR CLS 0 FIRST ! 4 LAST !
4 BEGIN CR INSTRUCTIONS DUP 49 = IF 2CLEAR CLS ENDIF
5 DUP 50 = IF ENTER ENDIF
6 DUP 51 = IF SHOW ENDIF
7 DUP 52 = IF CR ."Enter Row " IN# ROWCALC ENDIF
8 DUP 53 = IF CR ." Enter Column " IN# COLCALC ENDIF
9 DUP 54 = IF SP! QUIT ENDIF
10 AGAIN ;
11 ;S
12
13
14
15
```



# Software



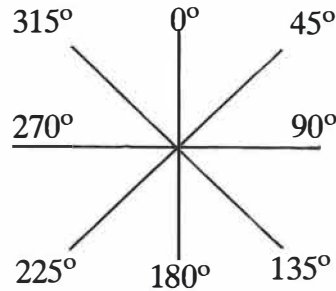
Space. Deep dark Space. All systems functioning, condition green. Aboard the Enterprise all is quiet, the crew looking forward to a few hours relaxation on the starbase just located in an adjacent quadrant. but wait, what's this? A long range scan reveals a Klingon cruiser in the vicinity. Condition yellow, set course and start warp drive on an intercept course. Klingon in phaser range, battle stations, condition red. Phaser targeting enabled. Fire phasers maximum power. Missed. Try a Photon Torpedo, targeting designate 45 degrees. Missed again. Klingon begins to fight back. Engage course computer to give a more accurate firing angle. 42 degrees, fire Torpedo again. A hit! Klingon cruiser destroyed.

The above scenario is simulated by this Basic game. You play the part of Captain Kirk (Sorry, Admiral Kirk!) navigating the Enterprise through the galaxy destroying Klingons whenever you can find them. You will need a 48K Oric to run the program and it should be entered on the Keyboard exactly as the listing below shows. Microtan users can run the program if they remove the colour and sound commands.

On entry to the program you will be asked to enter a number to seed the random number generator. You will then be given your orders by Starfleet. A short range sensor scan will be displayed and a list of possible commands. These function as follows:

## Option 0, Set Course

This allows you to move from quadrant to quadrant or sector to sector. When asked for the course you should enter an angle between 0 and 359 degrees. The directions are as illustrated below:



When asked for warp factor enter a whole number (1-8) to jump quadrants or a fractional number to move sectors within a quadrant (e.g. .2 will move you one sector.)

## Option 1, Short Range Scan

Each dot indicates one sector. E means Enterprise (You!), \* means a star, B, a starbase and K means a Klingon. A display of the current ship status is also given.



## Option 2, Long Range Scan

This displays the quadrants surrounding the quadrant you are currently in. Therefore the number in the middle of the screen is where you are. The numbers are broken down as follows:

- Units : Number of Stars in quadrant
- Tens : Number of Starbases
- Hundreds : Number of Klingons.

If you find a starbase you can dock with it by moving next to it. This will refuel you and recharge your weaponry systems.





ORIC-1



Damn  
Star logic  
space!

### Option 3, Fire Phasers

Will fire on any close object, power diminishes with distance. 100 units are usually enough to damage a Klingon Cruiser.

### Option 4, Fire Torpedoes

A more accurate weapon. Will only work within a quadrant. Requires you to enter the firing angle. This can be calculated accurately by using the course computer.

### Option 5, Damage Report

Indicates state of repair of vital systems. The more negative the number given is, the more severe is the damage.

### Option 6, Course Computer

Calculates distance and angle of object relative to Enterprise. Enter the number of sectors (dots) the object is above or below you for vertical coordinate. A positive number means above you, negative means below. For horizontal coordinate enter number of sector to the left or right of you (+ or -). Distance is printed as warp units.

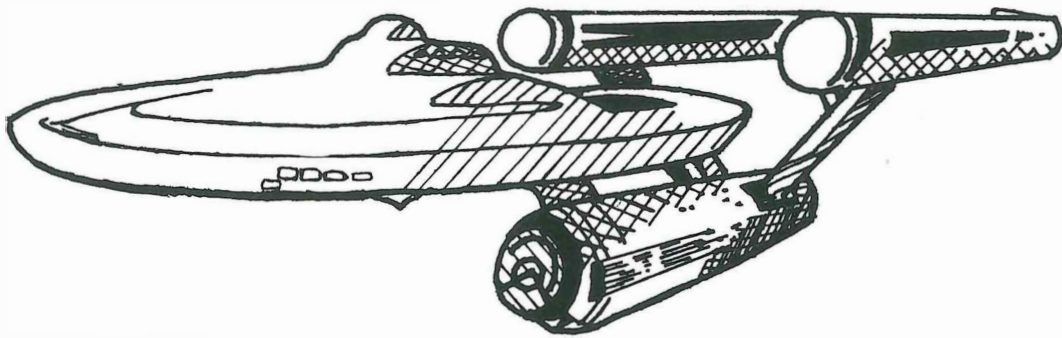
### Option 7, Galaxy Scan

As for Long Range Scan except prints map of entire known galaxy. Should only be used as a last resort.

Thus equipped with the above weaponry and control

systems you are ready to do battle to save the Starfleet Federation. The program leaves some room for improvement such as using better graphics to represent ships or stars.

The Galaxy Scan could be altered so that it only prints quadrants that have been visited. It will take you several hours to learn and master the program so I wish you luck and keep me informed of any new improvements you may make.



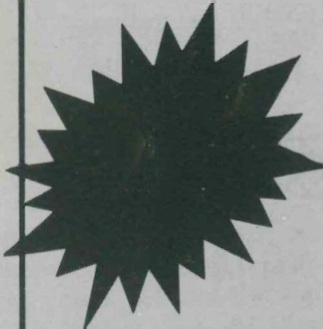
Damn  
Star logic  
space!



# PROGRAM LISTING



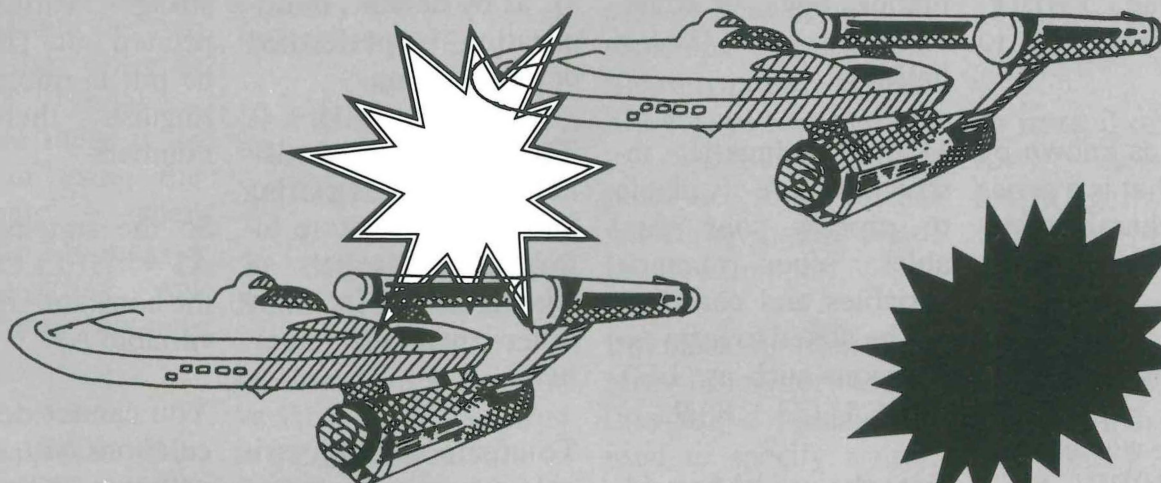
```
0 REM START HERE
50 REM STARTREK
200 REM PAUL B KAUFMAN OCT 1980
250 PAPER7:INK0
400 DIMS$(5),S$(8,8),D$(8,8),R(7),J(10)
500 S$(1)="":S$(2)="*":S$(3)="K":S$(4)="B":S$(5)="E":Q$=""
1100 DEFFNA(X)=INT(RND(1)*X)+1
1200 PRINTCHR$(12):FORI=1TO6:PRINT:NEXT:PRINT"          STARTREK":
1300 WAIT100
1400 PRINT:PRINT:PRINT:PRINT:PRINT: INPUT"Initialise,enter any numb
er":A
1600 FORI=1TOA:B=FNA(I):NEXTI
1900 PRINTCHR$(12):PRINT:PRINT:PRINT:PRINT"Starfleet orders imminen
t...":PRINT
2100 K1=FNA(20)+7:S1=FNA(149)+150:B1=FNA(9)
2400 .FORI=1TO7:R(I)=0:NEXTI
2700 FORI=1TO8:FORJ=1TO8:S%(I,J)=0:NEXTJ:NEXTI
3200 L1=0:L2=0:D1=INT((FNA(1999)+2000)/10)*10:D3=40-FNA(B):D2=D3+D1
3700 FORI=1TOS1
3800 A1=FNA(8):A2=FNA(8):IFS%(A1,A2)>8THEN3800
4100 S%(A1,A2)=S%(A1,A2)+1:NEXTI
4300 FORI=1TOK1:A1=FNA(8):A2=FNA(8):S%(A1,A2)=S%(A1,A2)+100:NEXTI
4800 FORI=1TOB1:A1=FNA(8):A2=FNA(8):S%(A1,A2)=S%(A1,A2)+10:NEXTI
5300 E1=FNA(8):E2=FNA(8):E7=FNA(8):EB=FNA(8):P=3000:C1=0:T1=10
6000 PRINTCHR$(12):PRINT:PRINT:PRINT"  ORDERS: $TARDATE = ";D1:PR
INT
6200 PRINT"Captain Kirk,you must destroy"
6300 PRINT"an invasion force of ";K1;"Klingons"
6400 PRINT"You have ";D3;"Solar days":PRINT"to complete your missio
n."
8100 PRINT:PRINT:PRINT"  PRESS SPACE TO CONTINUE ";GETV9:PRINT
9800 GOSUB19700
9900 GOSUB31300
10000 GOSUB14800
10100 GOTO11300
10300 GOSUB31300
10400 PRINT"Command:";GETC2$:C2=VAL(C2$):PRINT
10600 REM
10800 IFC2>7THEN10400
11200 ONC2+160TO26600,14600,23200,43200,47600,13300,54500,52100
11300 Y=1:GOSUB12900
11600 GOSUB12900
11700 PRINT"0=Set Course  1=S/R Scan"
11800 GOSUB12900:GOSUB12900:GOSUB12900
12100 PRINT"2=L/R Scan    3=Fire Phasers"
12200 GOSUB12900:GOSUB12900
12400 PRINT"4=Fire Torps  5=Damage Report"
12500 GOSUB12900
12600 PRINT"6=Course Comp. 7=Galaxy Scan":PRINT
12800 GOTO10300
12900 Y=Y+1:RETURN
13300 REM  DAMAGE CONTROL
13400 IFR(6)<0THEN14400
13500 PRINTCHR$(12):PRINT:PRINT"  Device  State of Repair"
13600 PRINT"  ===== "
13700 FORC2=1TO7:GOSUB39300:PRINT"  ";TAB(16)R(C2)
13800 NEXTC2:PRINT:PRINT:GOTO10300
14400 PRINT"Damage Control not available"
14500 GOTO10300
14600 GOSUB14800
14700 GOTO10300
14800 REMSHORT RANGE SCAN
14900 IFR(2)>0THEN15200
```



```

14700 GOTO10300
14800 REMSHORT RANGE SCAN
14900 IFR(2)>=0THEN15200
15000 PRINT"*S/R SENSORS ARE OUT**"
15100 RETURN
15200 GOSUB19700
15300 PRINTCHR$(12)" SHORT RANGE SENSOR SCAN":PRINT
15500 PRINT" 12345678"
15600 FORI=1TO8:PRINTSPC(3);:FORJ=1TO8
15800 Z9=DZ(I,J)+1:PRINTS$(Z9);:NEXTJ
16100 ONIGOTO16200,16400,17600,17800,18000,18200,18400,18600
16200 PRINT"1 Stardate ";D1:GOTO18500
16400 PRINT"2 Condition ";:IFC1OR0THEN16800
16600 PRINTCHR$(27)"RGreen":GOTO18500
16800 IFC1>1THEN17100
16900 PRINTCHR$(27)"S Yellow":GOTO18500
17100 IFC1>2THEN17400
17200 PRINTCHR$(27)"O Red":GOTO18500
17400 PRINT"Docked":GOTO18500
17600 PRINT"3 Quadrant ";E1;"- ";E2:GOTO18500
17800 PRINT"4 Sector ";E7;"- ";E8:GOTO18500
18000 PRINT"5 Energy ";(INT(F*10)/10):GOTO18500
18200 PRINT"6 Torpedoes ";T1:GOTO18500
18400 PRINT"7 Klingons ";K1
18500 GOTO18800
18600 PRINT"8 Days left ";D2-D1:PRINT
18800 NEXTI:RETURN
19000 REM STRING OF DASHES:PRINT
19200 FORI1=1TOC2:PRINT"-";:NEXTI1:PRINT:RETURN
19700 REM COMPUTE QUADRANT LAYOUT
19800 IFL1<>E1THEN20100
19900 IFL2<>E2THEN20100
20000 RETURN
20100 L1=E1:L2=E2
20300 FORI=1TO8:FORJ=1TO8:DZ(I,J)=0:NEXTJ:NEXTI
20800 DZ((INT(E7+.5),INT(E8+.5)))=4
20900 IFS$(E1,E2)-INT(S$(E1,E2)/10)*10=0THEN21600
21000 FORI=1TOSZ(E1,E2)-INT(S$(E1,E2)/10)*10
21100 E3=FNA(8):E4=FNA(8)
21300 IFDZ(E3,E4)<>0THEN21100
21400 DZ(E3,E4)=1:NEXTI
21600 IFINT(S$(E1,E2)/10)-INT(S$(E1,E2)/100)*10=0THEN22300
21700 FORI=1TOSZ(E1,E2)/10-INT(S$(E1,E2)/100)*10
21800 E3=FNA(7.9):E4=FNA(7.9)
22000 IFDZ(E3,E4)<>0THEN21800
22100 DZ(E3,E4)=3:NEXTI
22300 IFINT(S$(E1,E2)/100)=0THEN23100
22400 FORI=1TOSZ(E1,E2)/100
22500 J(I)=300
22600 E3=FNA(7.9):E4=FNA(7.9):IFDZ(E3,E4)<>0THEN22600
22900 DZ(E3,E4)=2:NEXTI:RETURN
23100 RETURN
23200 REM LONG RANGE SCAN
23300 IFR(3)=0THEN23600
23400 PRINT"*L/R Sensors are out**":GOTO10300
23600 Y=1:GOSUB12900
23800 PRINTCHR$(12):PRINT:PRINT:PRINT"Long Range Sensor Scan"
23900 PRINT" Quadrant";:PRINT" ";E1;"-";E2
24100 C2=24:GOSUB19000
24300 I=E1-1

```



# How to get the most

## How do I store my data?

This first article will explain the different types of data that can be stored within the Oric 1 computer.

Although primarily a beginner's guide, the information may help those familiar with the workings of the Basic.

## What is a Variable?

A variable is so called because the value it represents is not fixed. The opposite of a variable is naturally a constant, such as 1, 2 and 3.

Obviously the number 1 can never be equal to 2, or the value "FRED" change suddenly to "JIM".

A variable is known by its name, that is a series of unique characters defined by the programmer.

For example, `LET NUMBER = 6` will set the variable which is called 'NUMBER' to a value of 6.

At this stage, I should explain that only the first two characters of a name are important, but `NUMBER` is more meaningful than `NU`.

## What sorts of Variables are there?

There are two main classes of variables, numbers and character strings.

A number, such as 453 is put in any variable name with a statement such as `20 LET B = 453`.

In fact, B can be set to any value positive, negative, fraction or integer.

For those non-mathematicians, an integer is any normal number such as 882 or -272, whereas a fractional number could be something like 0.5 (i.e. a half).

Many arithmetic instructions are available to process your variables. Your numeric variables and constants can be mixed to form expressions such as: `LET C = 6.4 * (23 + K) / 9`.

Note the use of \* to signify multiplication and /

for division. When brackets are used in this way the calculation in the innermost brackets is done first. This is important because  $3 * 5 + 8$  is different from  $3 * (5 + 8)$ , as by default, multiplication is performed before addition.

The other type of variable is known as a string and is used to store information, or sets of characters (possibly other than just numbers).

To create a string variable, you specify a name as before, and add \$ to

the right hand side. For example, `FRED$`, `NAME$` or `A$`.

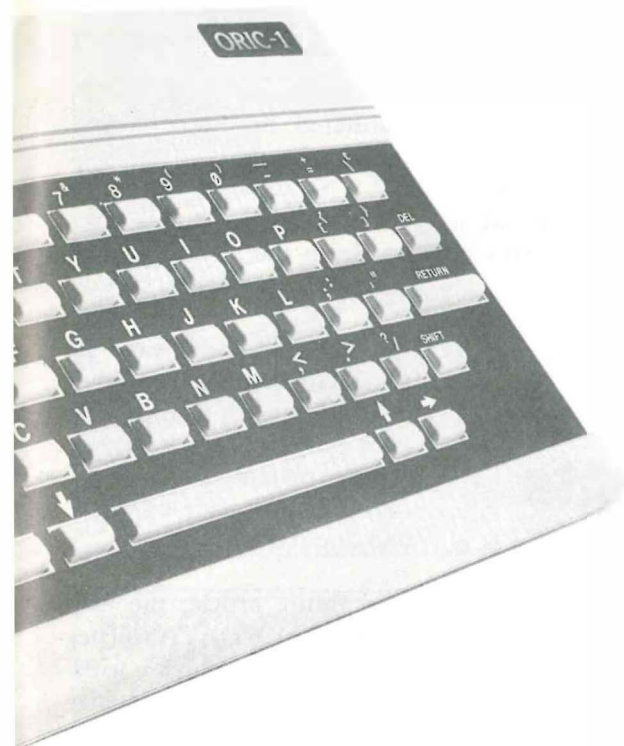
A string constant is used where fixed information is to be moved into a string variable or printed out. These must be put in quotes to distinguish them from numbers.

So the statement `LET A$ = "HELLO"` assigns the constant HELLO to variable A\$.

You cannot do any calculations with strings as you can with numeric variables, but there are



# out of your ORIC - 1



ways of manipulating your strings, such as chopping them up and joining them together again.

Here is a short program to demonstrate the two uses of the differing types of variables. Notice that there is no problem in using the same 'Name' - when you add the \$ to make a variable a string, you create a completely unique entity.

```
10 REMARKS DEMONSTRATE USE OF VARIABLES
```

```
20 PRINT "PLEASE ENTER YOUR NAME"
30 INPUT NAME$
40 PRINT "AND HOW OLD ARE YOU";
50 INPUT AGE
60 PRINT 1983-AGE; "WAS A GOOD YEAR TO BE BORN"
70 PRINT "GOOD-BYE "; NAME$
```

In line 60 the numeric variable AGE was used in a simple calculation to find the user's date of birth, whilst NAME\$ is

only used in the PRINT at line 70.

In line 70 you can see a demonstration of a constant being used in conjunction with a string variable. The program always says 'Goodbye' followed by the name previously entered.

There are other types of variable yet to be discussed, one of which is the integer.

Remember that a variable such as AGE can contain any number including decimal values such as .5. The integer, which is signified by a % sign after the name of the variable, can only contain whole numbers between -32768 and +32767.

Integer variables like K% are less efficient from a processing point of view and occupy the same space as a normal 'floating point' variable. They would be useful if you wanted to force a number to remain whole, saving the use of the INT function.

But most importantly, in terms of memory usage, they are a boon when used to specify arrays, which we come to next, where they use only

2/5ths of the space that a floating point array would require.

## Using Arrays

Supposing we wanted to record a list of ages against names by using INPUT.

Just by using string and numeric variables our program would look like:

```
10 INPUT "What is your name and age"; N1$, A1
20 INPUT "What is your name and age"; N2$, A2 etc.
```

Thankfully, there are such things as arrays, whereby lists of data (either numeric, string or integer) can be stored very easily.

So instead of specifying A4% meaning the fourth age, we can just state AG%(4).

Unless you intend to get away with less than ten ELEMENTS of AG%, then you must first (and only once) dimension the array using the DIM statement.

# How to get the most out of your ORIC-1

Here is our revised program to ask for 20 names and ages and store the information away.

```
10 DIM NA$(20), AG%(20)
```

```
20 FOR I = 1 TO 20
```

```
30 INPUT "Please enter a name and age"; NA$(I), AG%(I)
```

```
40 NEXT I
```

See how another variable I, is used to SUBSCRIPT or identify specifically each element of the NAME array NA\$ and the AGE array AG%.

Each element of a numeric array uses up five bytes of memory when floating point is used. For something like storing ages, we only need the capability for small integers, and the Integer Array fits the bill nicely.

Only two bytes are used for each Age stored, and we can cope with anyone who is younger than Methuselah!

Our string array will occupy at least three bytes per name, plus the actual value of the name. The three byte overhead is always incurred on

any string and covers the actual address and length.

Later on I will demonstrate the savings that you can get (In terms of memory, not speed) by using strings.

## Fast Processing Hints

The fastest processing is always done with normal floating-point variables. Integers

and constants have to be translated by Basic before they can be used.

Normal arrays are quite fast, but keep the subscript as a variable for the fastest speed.

The best response time is obtained for variables who are referenced earliest in your program, and a saving (both in memory and in program speed) is made by using one character data names.

In a future article, the use (and abuse) of DATA will be discussed, along with methods of memory saving and efficient techniques for fast programming.



# CAPTAIN TANEX

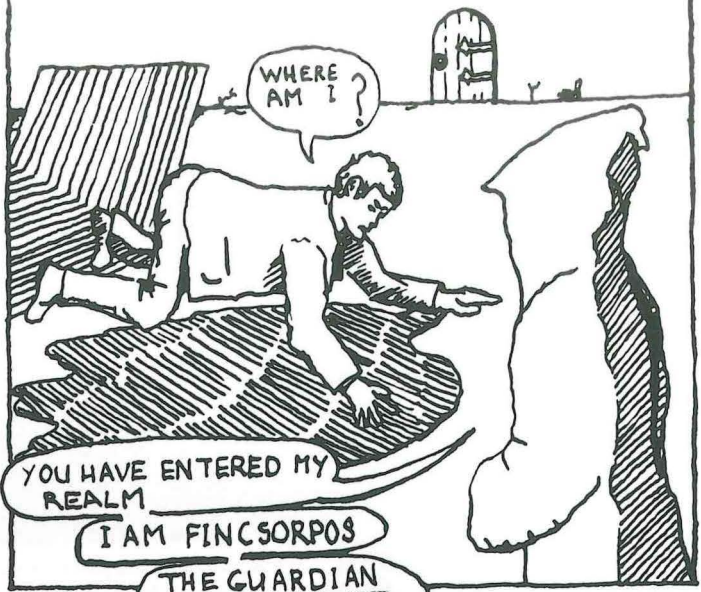


ARGH!

I'M FALLING INTO THE MACHINE

OUR INTREPID HERO HAS UNSUSPECTINGLY SET IN MOTION A COURSE OF EVENTS WHICH HAVE TAKEN HIM INTO ANOTHER REALITY. EVERY ATOM OF HIS BODY AND SOUL HAS BEEN INTEGRATED WITH THE RUNNING ADVENTURE PROGRAM CALLED

## THE EYE OF TANGER



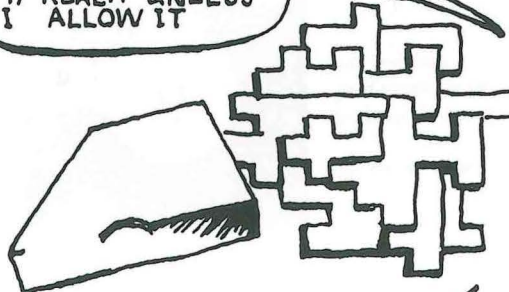
WHERE AM I?

YOU HAVE ENTERED MY REALM

I AM FINCSORPOS

THE GUARDIAN OF THE GATE

YOU CANNOT LEAVE MY REALM UNLESS I ALLOW IT



AND BEFORE I ALLOW IT YOU MUST PERFORM A FAVOUR FOR ME

YOU MUST UNCOVER THE SECRET OF THE EYE OF TANGER

IN ORDER TO ATTAIN THIS GOAL YOU WILL NEED CERTAIN TALENTS

BUT YOU MUST WIN THOSE TALENTS

AND THE FIRST TALENT CAN BE WON IN THE CATACOMBS



I DON'T UNDERSTAND

YOUR TIME IS LIMITED AND YOU MUST NOW...

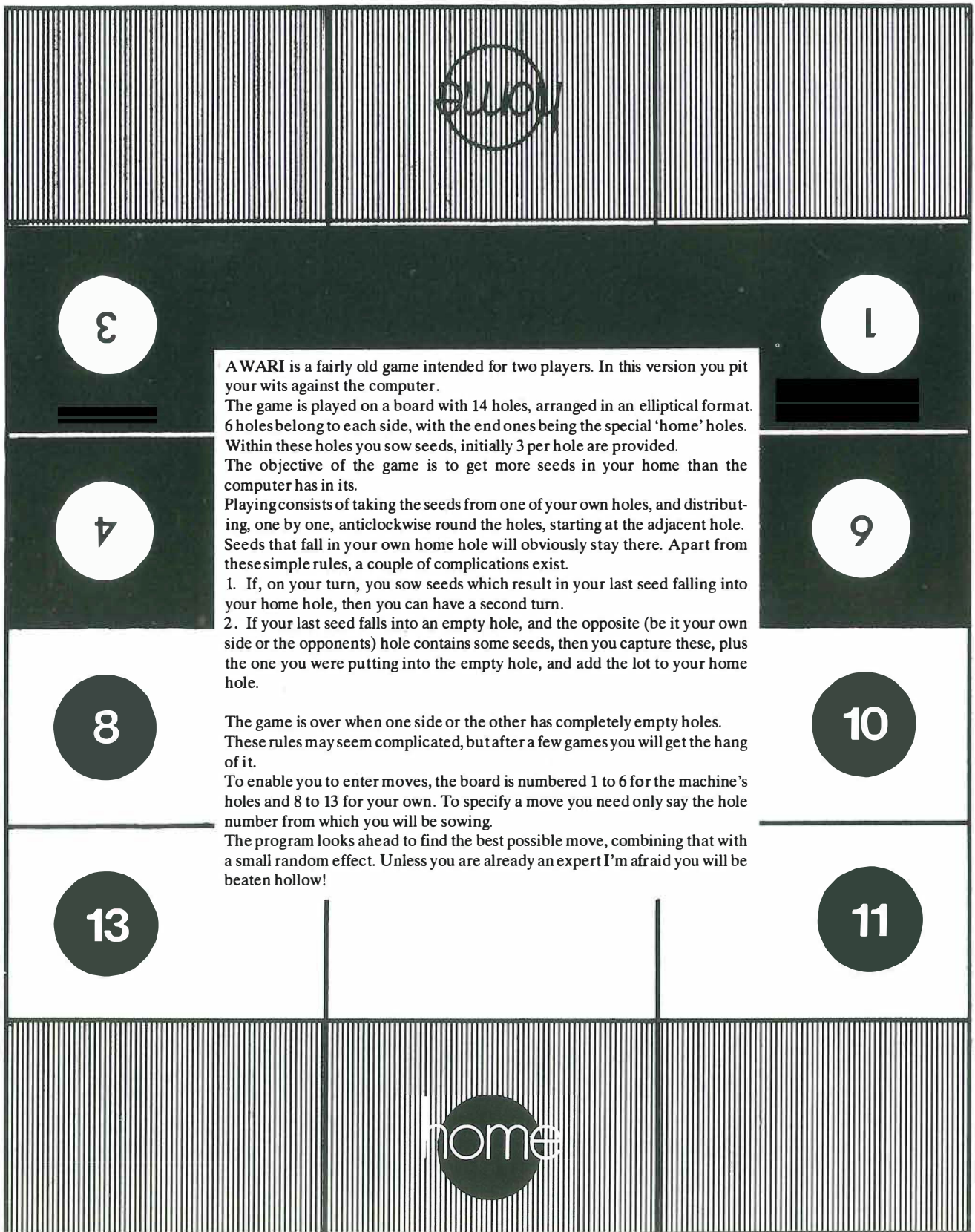
...GO!

BUT BEWARE OF THE ISSD POL !!!



# AWARI

by G. Phillips



AWARI is a fairly old game intended for two players. In this version you pit your wits against the computer.

The game is played on a board with 14 holes, arranged in an elliptical format. 6 holes belong to each side, with the end ones being the special 'home' holes. Within these holes you sow seeds, initially 3 per hole are provided.

The objective of the game is to get more seeds in your home than the computer has in its.

Playing consists of taking the seeds from one of your own holes, and distributing, one by one, anticlockwise round the holes, starting at the adjacent hole. Seeds that fall in your own home hole will obviously stay there. Apart from these simple rules, a couple of complications exist.

1. If, on your turn, you sow seeds which result in your last seed falling into your home hole, then you can have a second turn.
2. If your last seed falls into an empty hole, and the opposite (be it your own side or the opponents) hole contains some seeds, then you capture these, plus the one you were putting into the empty hole, and add the lot to your home hole.

The game is over when one side or the other has completely empty holes. These rules may seem complicated, but after a few games you will get the hang of it.

To enable you to enter moves, the board is numbered 1 to 6 for the machine's holes and 8 to 13 for your own. To specify a move you need only say the hole number from which you will be sowing.

The program looks ahead to find the best possible move, combining that with a small random effect. Unless you are already an expert I'm afraid you will be beaten hollow!



# Program Listing

```

10 DEFFNA(X)=X+14*(X>14)
15 DEFFNE(X)=(X<8)OR(X>13)
17 DEFFNX(X)=A(1)+A(2)+A(3)+A(4)+A(5)+A(6)
18 DEFFNY(X)=A(8)+A(9)+A(10)+A(11)+A(12)+A(13)
19 DEFFNR(X)=RND(9)*5-2+X
20 DIMA(14),B(4,14)
30 FORI=1TO13:A(I)=3:NEXT:A(7)=0
40 PRINTCHR$(12):PRINT:PRINTTAB(10);"A W A R I
50 PRINT:PRINT"      BY G. PHILLIPS 1983
60 FORI=1TO2000:NEXT
65 GOSUB4000
70 PRINT"Would you like to start?"
80 GETZ$:IFZ$<>"Y"THEN$60
300 FORU=1TO2
305 GOSUB4000
305 PRINT"Please enter your move":INPUTA$:V=VAL(A$):IFV=0THEN$300
310 IF FNE(V)THENPRINT"You can only move from 8 to 13":GOTO300
320 IFA(V)=0THENPRINT"There are no seeds there!":GOTO300
330 X=V:GOSUB2000
335 IFFNX(0)=0ORFNY(0)=0THEN$5000
340 IFY=14THENIFU=1THENPRINT"Second move allowed..":NEXT
345 GOSUB4000
360 GOSUB500
370 IFFNX(0)=0ORFNY(0)=0THEN$5000
380 GOTO300
500 REM COMPUTERS MOVE
510 B1=0:B2=0:B3=0:B4=0:S1=-91:S2=-91:S3=-91:S4=-91
520 FORA1=1TO6:IFA(A1)=0THEN900
525 PRINT".":
530 GOSUB1000:X=A1:GOSUB2000:F=0:IFY<>7THENF=1:GOTO600
540 FORA2=1TO6:IFA(A2)=0THEN800
550 GOSUB1000:X=A2:GOSUB2000
600 FORA3=8TO13:IFA(A3)=0THEN700
610 GOSUB1000:X=A3:GOSUB2000:IFY<>14THEN680
620 FORA4=8TO13:IFA(A4)=0THEN660
630 GOSUB1000:X=A4:GOSUB2000:IFA(14)-A(7)>FNR(S4)THENS4=A(14)-A(7):B4=A4
640 GOSUB1010
660 NEXTA4
670 X=B4:GOSUB2000
680 IFA(14)-A(7)>FNR(S3)THENS3=A(14)-A(7):B3=A3
685 GOSUB1010
700 NEXTA3:X=B3:GOSUB2000
705 IFF=1THEN810
720 IFA(7)-A(14)>FNR(S2)THENS2=A(7)-A(14):B2=A2
730 GOSUB1010
800 NEXTA2
805 X=B2:GOSUB2000
810 IFA(7)-A(14)>FNR(S1)THENS1=A(7)-A(14):B1=A1
820 GOSUB1010

```

```

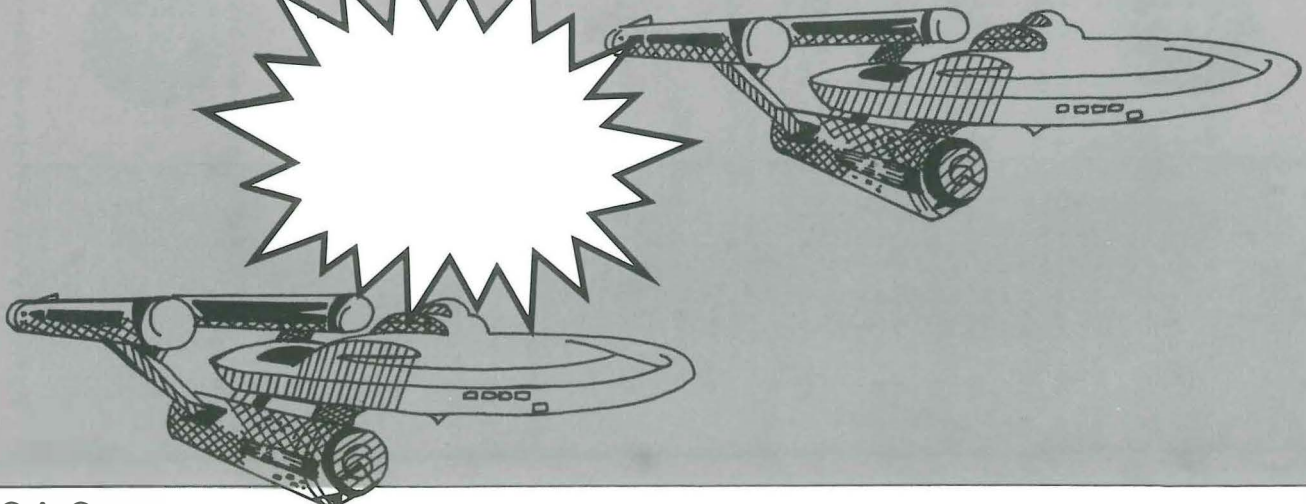
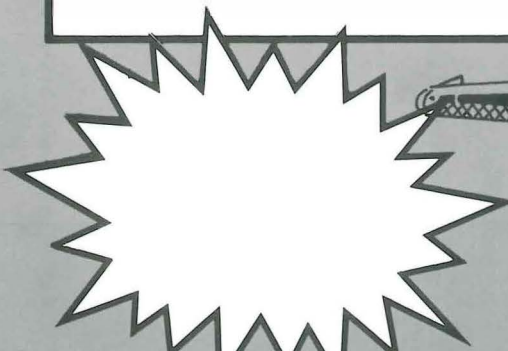
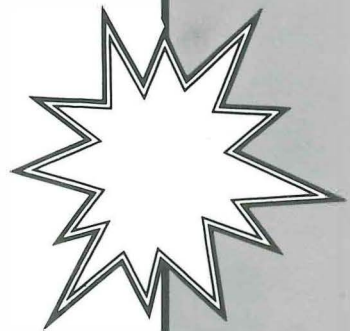
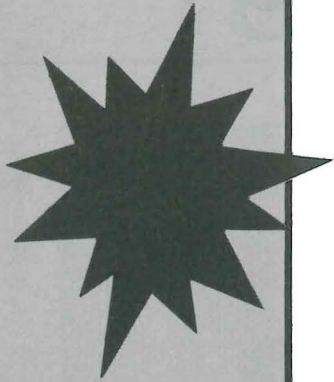
900 NEXTA1
910 X=B1:GOSUB2000:PRINT"My move":B1;
911 IFFNX(0)=0ORFNY(0)=0THEN$5000
912 IFY=7THENX=B2:GOSUB2000:PRINT"And "B2
915 FORI=1TO3000:NEXT
920 PRINT:RETURN
1000 T=T+1:FORZ=1TO14:B(T,Z)=A(Z):NEXT:RETURN
1010 FORZ=1TO14:A(Z)=B(T,Z):NEXT:T=T-1:RETURN
2000 REM MOVE POSITION X
2010 FORZ=X+1TOX+A(X):Y=FNA(Z):A(Y)=A(Y)+1:NEXT
2025 A(X)=0:Q=-1
2026 IFY=7ORY=14THEN2060
2030 IFA(Y)=1ANDA(14-Y)>0THENQ=A(14-Y):A(14-Y)=0:A(Y)=0
2040 IFX<7THENA(7)=A(7)+Q+1:GOTO2060
2050 A(14)=A(14)+Q+1
2060 RETURN
4000 REM PRINTOUT
4002 FORK=1TO1000:NEXT
4005 PRINTCHR$(12)
4010 PRINT"      13 12 11 10  9  8
4020 PRINT"Your -----My
4030 PRINT"Home!":FORI=13TO8STEP-1:A$=STR$(A(I))
4040 IFLEN(A$)>2THENA$=RIGHT$(A$,2)
4050 PRINTA$":":NEXT:PRINT"Home
4060 PRINT" -----
4070 PRINT" ----
4080 PRINT"!":A$=STR$(A(14)):IFLEN(A$)>2THENA$=RIGHT$(A$,2)
4090 PRINTA$":":A$=STR$(A(7))
4100 IFLEN(A$)>2THENA$=RIGHT$(A$,2)
4105 PRINTA$":
4110 PRINT" ----
4120 PRINT" -----
4130 PRINT"!":FORI=1TO6:A$=STR$(A(I)):IFLEN(A$)>2THENA$=RIGHT$(A$,2)
4140 PRINTA$":":NEXT
4145 PRINT
4150 PRINT" -----
4160 PRINT"      1  2  3  4  5  6
4200 RETURN
5000 PRINTCHR$(12)
5005 IFA(14)=A(7)THEN$100
5010 IFA(14)>A(7)THEN$060
5020 PRINT"I won by "A(7)-A(14)
5030 GOTO$200
5060 PRINT"You beat me! Though only by"A(14)-A(7)
5080 GOTO$200
5100 PRINT"We have equalised at"A(7)
5200 PRINT"Would you like to try again?"
5210 GETZ$:IFZ$="Y"THENRUN
5220 END

```

```

24400 IFI>E1+1THEN26000
24500 J=E2-1
24600 IFJ>E2+1THEN25600
24700 IFI<10RI>8THEN26300
24900 IFJ<10RJ>8THEN26100
25100 IFS%(I,J)<=9THENPRINT"  ";S%(I,J);
25200 IFS%(I,J)<=99ANDS%(I,J)>=10THENPRINT"  ";S%(I,J);
25300 IFS%(I,J)>=100THENPRINT"  ";S%(I,J);
25400 J=J+1:GOTO24600
25600 PRINT:GOSUB19000:I=I+1
25900 GOTO24400
26000 PRINT:PRINT:GOTO10300
26100 PRINT"  ";0;:GOTO25400
26300 PRINT"  ";0;"  ";0;"  ";0;:GOTO25600
26500 REM SET COURSE
26600 PRINTCHR$(12):PRINT:PRINT:INPUT"Course      ";C2
26800 INPUT"Warp Factor ";C3
27000 IFC3<=0THEN10600
27100 IFC3>8THEN26800
27200 IFR(1)=0THEN27600
27300 IFC3<=.2THEN27600
27400 PRINT"Warp drive damaged":PRINT"Max warp=.2":GOTO26600
27600 P=P-16*C3:N1=INT(8*C3+.5):N2=-COS(C2*3.14159/180)
27900 IFABS(N2)>.01THEN28100
28000 N2=0
28100 N3=SIN(C2*3.14159/180):IFABS(N3)>.01THEN28400
28300 N3=0
28400 A1=1
28500 IFA1>N1THEN30300
28600 E3=E7:E4=E8:P1=E3+N2:P2=E4+N3
29000 IFP1<.5ORP1>.5THEN36200
29100 IFP1>.5THEN36200
29200 IFP2<.5ORP2>.5THEN37300
29400 IFD%(INT(P1+.5),INT(P2+.5))=0ORD%(INT(P1+.5),INT(P2+.5))=4THE
N29700
29600 GOTO33200
29700 D%(INT(E3+.5),INT(E4+.5))=0:D%(INT(P1+.5),INT(P2+.5))=4
29900 E7=P1:E8=P2
30100 A1=A1+1
30200 GOTO28500
30300 D1=D1+1:E7=INT(E7+.5):E8=INT(E8+.5)
30600 FORI=1TO7:IFR(I)=0THEN31100
30800 R(I)=R(I)+1:IFR(I)<0THEN31100
31000 R(I)=0
31100 NEXTI:GOTO38500
31300 REM
31400 FORI=E7-1TOE7+1:IFI<10RI>8THEN32200
31700 FORJ=E8-1TOE8+1:IFJ<10RJ>8THEN32100
32000 IFD%(I,J)=3THEN33000
32100 NEXTJ
32200 NEXTI
32300 C1=0:IFS%(E1,E2)<100THEN32700
32500 C1=2:GOTO33600
32700 IFP>500THEN33600
32800 C1=1:GOTO33600
33000 C1=3:P=3000:T1=10
33300 FORI=1TO7:R(I)=0:NEXTI
33600 RETURN
33700 REM
33800 C2=RND(1):IFC2<.25THEN34900
34000 IFC2<.9THEN36000

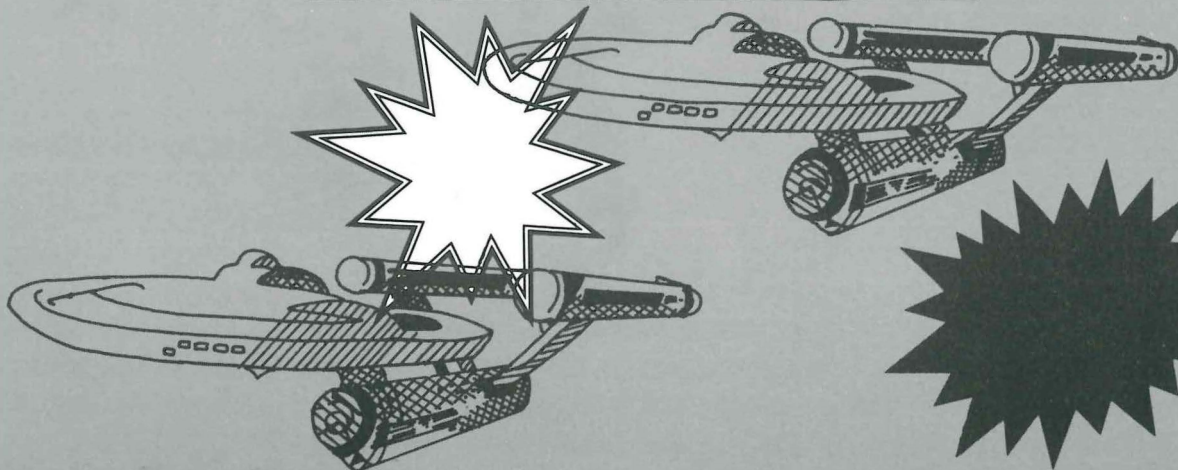
```



```

34100 PRINT" **SPACE STORM**":IFC1<3THEN34500
34300 PRINT" Starbase shields Enterprise":GOTO36000
34500 GOSUB39100:PRINT" Damaged **":R(C2)=R(C2)-5*RND(1):GOTO36000
34900 FORI=1TO7:IFR(I)<>0THEN35300
35100 NEXTI:GOTO36000
35300 PRINT" **TRUCE**":C2=I:GOSUB39300
35600 PRINT" State of repair improved":R(I)=R(I)+2*RND(1)
35800 IFR(I)<0THENRETURN
35900 R(I)=0
36000 RETURN
36100 REM
36200 S2=S6N(P1-1):S3=E1+S2:E7=INT(P1+.5)-8*S2:L1=0
36600 IFS3<1ORS3>8THEN36900
36800 E1=S3
36900 IFF2<=.5THEN37100
37000 GOTO37300
37100 IFF2<=.5THEN38000
37300 S2=S6N(P2-1):S3=E2+S2:E8=INT(P2+.5)-8*S2:L2=0
37700 IFS3<1ORS3>8THEN38000
37900 E2=S3
38000 GOSUB19700:GOTO30100
38200 PRINT" Enterprise blocked at";INT(P1);"-";INT(P2)
38400 GOTO30300
38500 GOSUB31300:GOSUB33700:GOSUB40800
38800 IFF<0THEN57000
38900 IFD1>02THEN57200
39000 GOTO10000
39100 REM
39200 C2=FNA(6.9)
39300 UNDC26GOTO39400,39600,39800,40000,40200,40400,40600
39400 PRINT"Warp drive "":RETURN
39600 PRINT"S/R Sensors "":RETURN
39800 PRINT"L/R Sensors "":RETURN
40000 PRINT"Phaser Ctrl 1":RETURN
40200 PRINT"Photon tubes":RETURN
40400 PRINT"Damage Ctrl 1":RETURN
40600 PRINT"Course capt":RETURN
40800 IFS%(E1,E2)<100THENRETURN
40900 IFC1<>3THEN41100
41000 GOSUB34300
41100 G=1:H=0
41300 FORJ=1TO5%(E1,E2)/100
41400 H=H+1:IFH<=8THEN41800
41600 H=1:G=G+1
41800 IFD%(G,H)<>2THEN41400
41900 QJ=G-E7:Q2=H-E8
42100 D4=SDR(Q1*Q1+Q2*Q2):P5=FNA(J(I)-1)
42300 J(I)=J(I)-P5:IFC1=3THEN42900
42500 P6=P5/D4:P=P-P6
42700 PRINT"Enterprise hit:";INT(P5*10)/10:"units"
42800 PRINT"Klingon at sector";G;"-";H
42850 FORQ=1TO500:NEXT
42900 NEXTI
43000 IFF<=0THEN57400
43100 RETURN
43200 REM PHASER CONTROL
43300 IFR(4)=0THEN43600
43400 PRINT"*** PHASER MALFUNCTION ***":GOTO10300
43600 PRINTCHR#(12):PRINT"Phaser targeting enabled"
43700 PRINT"Power status=";INT(P*10)/10:"units"

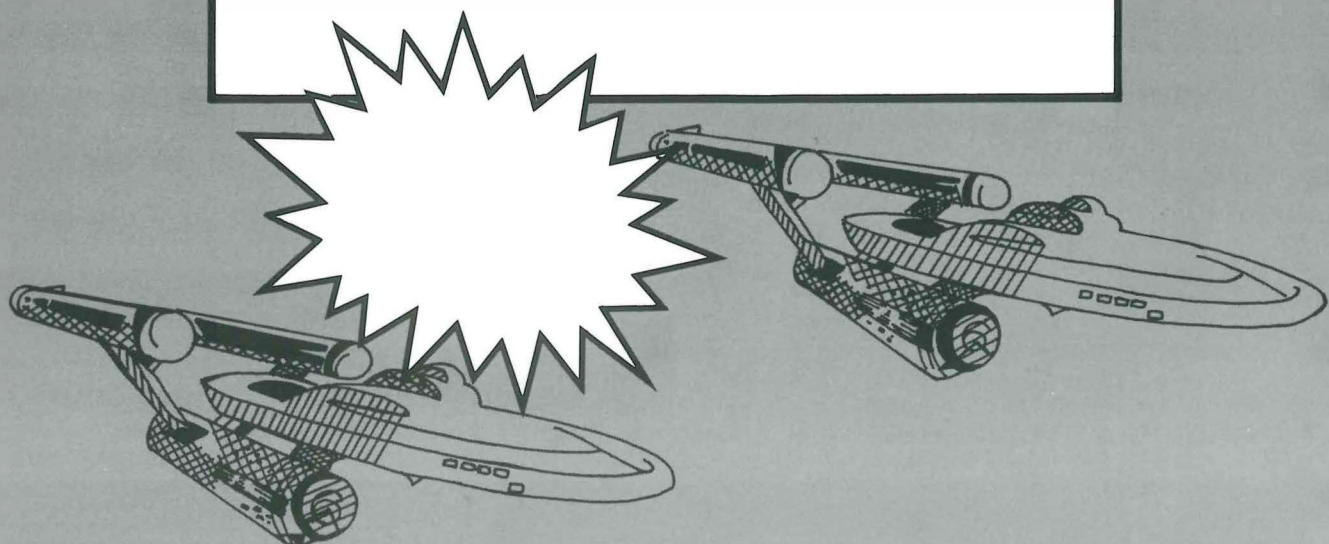
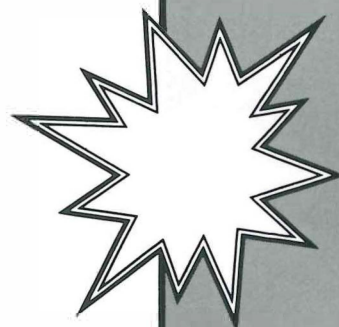
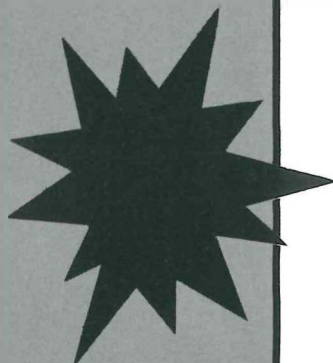
```

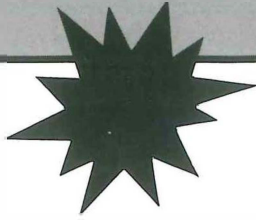


```

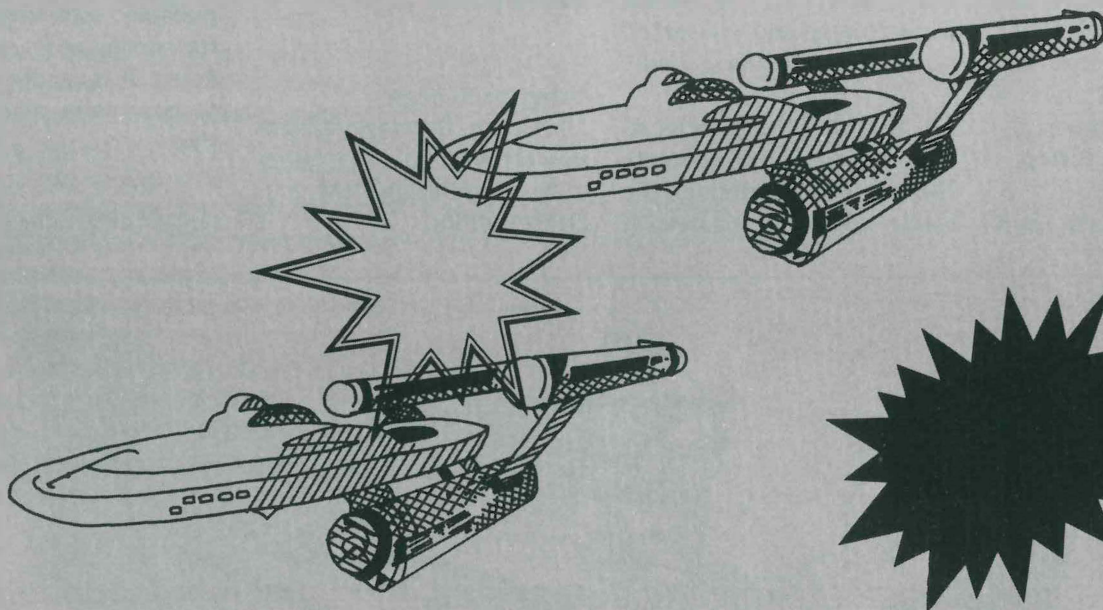
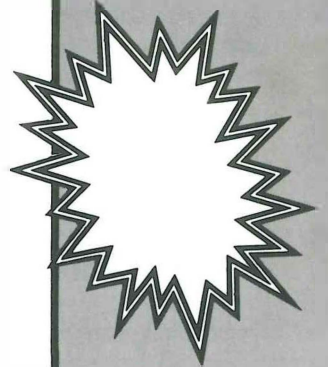
43800 INPUT"Enter units to fire ";C2
43805 ZAP:ZAP:ZAP:ZAP
44000 IFC2>PTHEN43700
44100 IFC2<=0THEN10300
44200 P=P-C2
44300 IFS%(E1,E2)<100THEN47200
44400 P5=C2/INT(S%(E1,E2)/100):G=1:H=0
44700 FORI=1TOS%(E1,E2)/100
44800 H=H+1:IFH<9THEN45300
45000 H=1:G=G+1
45200 IFG>8THEN47200
45300 IFD%(G,H)>2THEN44800
45400 Q1=G-E7:Q2=H-E8
45600 IFD%(G,H)<2THEN44800
45700 D4=SQR(Q1*Q1+Q2*Q2)
45800 P6=P5/D4:J(I)=J(I)-P6
46000 PRINT"Klingon at";G;"-";H;"hit"
46005 SHOOT
46100 PRINT(" ; INT(P6*10)/10;"units) Left=";INT(J(I)*10)/10
46200 IFJ(I)>0THEN47100
46300 EXPLODE:PRINT"*** DESTROYED ***"
46400 FORI2=I+1TOS%(E1,E2)/100:J(I2-1)=J(I2):NEXTI2
46700 S%(E1,E2)=S%(E1,E2)-100
46800 K1=K1-1
46900 D%(G,H)=0
47000 I=I-1
47100 NEXTI
47200 IFK1=0THEN56400
47300 GOSUB31300
47400 GOSUB40800
47500 GOTD10300
47600 REM PHOTON TORPS
47700 IFR(5)=0THEN48000
47800 PRINT"*** PHOTON TUBE MALFUNCTION ***":GOTD10300
48000 IFT1>0THEN48300
48100 PRINT"Torpedoe supply exhausted":GOTD10300
48300 INPUT"Enter targeting designate";C2
48500 T1=T1-1
48600 N2=-COS(C2*3.14159/180)
48700 IFABS(N2)>.01THEN48900
48800 N2=0
48900 N3=SIN(C2*3.14159/180)
49000 IFABS(N3)>.01THEN49200
49100 N3=0
49200 P1=E7:P2=EB
49400 P1=P1+N2
49500 P2=P2+N3
49550 SOUND1,P1*50,15
49600 PRINTINT(P1*10)/10;"-";INT(P2*10)/10
49700 IFF1<.5ORP1>8.5THEN51900
49900 IFF2<.5ORP2>8.5THEN51900
50100 IFD%(INT(P1+.4),INT(P2+.4))=0THEN49400
50200 OND%(INT(P1+.4),INT(P2+.4))GOTO50400,50700,51200
50300 GOTD49400
50400 EXPLODE:PRINT" Star Destroyed":S%(E1,E2)=S%(E1,E2)-1:GOTO5140
0
50700 PRINT" KLINGON DESTROYED":S%(E1,E2)=S%(E1,E2)-100:K1=K1-1:GOS
UB31300

```

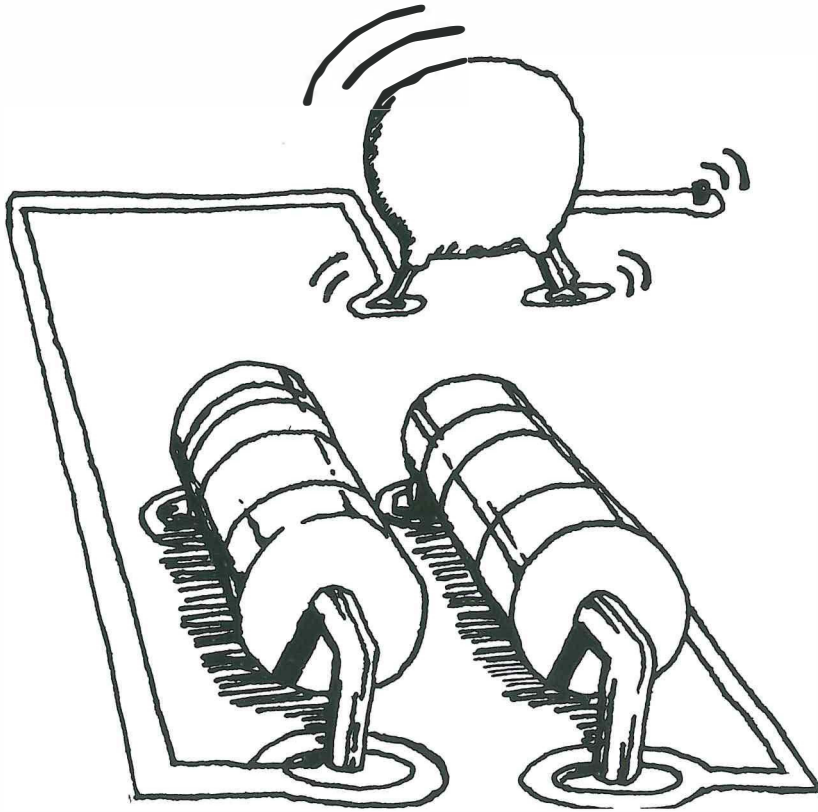




```
50705 EXPLODE
51100 GOT051400
51200 PRINT "Starbase Destroyed":% (E1,E2)=% (E1,E2)-10
51400 DX=(INT(P1+.4),INT(P2+.4))=0
51500 IFK1=0THEN54400
51600 GOSUBS1300:GOSUB40800:GOTO10300
51900 PING:PRINT"Targeting error-Torp missed":GOT051500
52100 REM GALACTIC MAP
52170 PRINT"SWITCH PRINTER ON"
52200 LPRINTTAB(22)"Galactic Scan":LPRINTTAB(22)-----
52500 I9=1
52600 IFI9.8THEN53700
52700 REM
52900 FORJ9=1TO8
53000 IFS%(I9,J9)<=9THENLPRINT":":S%(I9,J9):
53100 IFS%(I9,J9)>=10ANDS%(I9,J9)=99THENLPRINT":":S%(I9,J9):
53200 IFS%(I9,J9)>=100THENLPRINT":":S%(I9,J9):
53300 NEXTJ9:LPRINT
53400 I9=I9+1:GOTO52600
53700 GOSUB12000:GOTO10400
54500 REM COURSE COMPUTER
54600 JFK(7)=0THEN54900
54700 PRINT"Course computer malfunction":PRINT"manual control requ
red":GOTO10300
54900 INPUT"Enter vertical coordinate":C2:
55100 INPUT"and horizontal coordinate":C3:
55300 C4=C4+(C3/C2)*57.2958
55400 IFC4=0THEN56000
55500 IFC4<0THEN56200
55600 PRINT"Course designate=":INT(C4*10)/10:"degrees"
55700 C5=SQR(P2*2+D3*2)*.125
55800 PRINT"sl warp":INT(C5*10)/10
55900 GOT010300
56000 C4=C4+18
56100 GOT05600
56200 C4=C4+36
56300 GOT05600
56400 REM END GAME
56500 PRINT:PRINT:PRINT"Hi is stardate":D1
56600 PRINT"The Klingons have been destroyed":PRINT"The Federation
is safe"
56800 PRINT"Congratulations!":GOT057800
57000 PRINT"Power malfunction":GOT057500
57200 REM DESTROYED ENTERPRISE
57300 PRINT:PRINT:PRINT"Hi is stardate":D1
57400 PRINT"The Enterprise has been wrecked"
57500 PRINT"The Federation will be conquered"
57600 PRINT"Klingons Left = ":K1:PRINT"You are dead!!!"
57900 REM
57810 PRINT"Another Game,Captain Kirk ":GETZZ$:PRINT
57820 IFZZ$="Y"THENCLEAR:RUN
57830 PRINT:PRINT:PRINT:END
```



# High speed sort



One of the most time consuming tasks for a computer to perform is that of sorting data into order. When the program is written in BASIC, not a particularly fast language at the best of times, it can take several minutes for just fifty items to be sorted.

This machine code sort

routine is called by Basic to sort one dimensional character arrays. It is significantly faster than the equivalent Basic routine and will sort 50 items in less than ten seconds.

The machine code is relocatable (there is no absolute JMPs or JSRs) although the listing shows it located at

\$1F00 the Basic program puts the routine at \$400 which is the start of the Oric's spare memory area.

#### Using the Routine

1) Create the single dimension array to hold the values to be sorted using DIM, e.g. DIM A\$(100).

2) Fill the array with the words or strings you want to sort.

3) Assuming the routine has already been entered into memory set up the USR vector to the start address using DEFUSR (POKE34,0: POKE35,31 on the Microtan).

4) Jump to the routine using: P = USR(A) where A is the name of the array to be sorted with the Dollar sign (\$) removed.

No action will be taken by the routine if:

- a) The array cannot be found.
- b) It has more than one dimension.
- c) There are less than two items in the array.

The Basic Program below sets up the machine code into memory and demonstrates the sorting of an array of ten items. It is readily adaptable to cope with arrays of any size.

Note: In the listing £ = #



# Program Listing

```
10 REM ARRAY SORTER
20 REM SORTS SINGLE DIMENSION CHARACTER ARRAY
30 REM MIKE BLAYNEY 1983
40 REM
50 REM SET UP MACHINE CODE ROUTINE...
60 FORI=£400 TO £4CB
70 READ D:POKEI,D:NEXT
80 DATA £A5,£9E,£85,£40,£A5,£9F,£85,£41,£A5,£B4,£85,£42,£A5,£B5,£9,
£80
90 DATA £85,£43,£A0,£0,£B1,£40,£C8,£C5,£42,£D0,£6,£B1,£40,£C5,£43,£
FO,£1C,£C8
100 DATA£18,£B1,£40,£65,£40,£48,£C8,£B1,£40,£65,£41,£85,£41,£68,£85
,£40,£C5
110 DATA£A0,£D0,£DC,£A5,£41,£C5,£A1,£D0,£D6,£60,£C8,£C8,£C8,£B1,£40
,£C9,£1
120 DATA£D0,£F6,£C8,£B1,£40,£85,£45,£C8,£B1,£40,£85,£44,£18,£A5,£40
,£69,£7
130 DATA£85,£4E,£A5,£41,£69,£0,£85,£4F,£38,£A5,£44,£E9,£1,£85,£46,£
A5,£45
140 DATA£E9,£0,£30,£D2,£85,£47,£D0,£6,£A5,£46,£C9,£2,£90,£C8,£A9,£0
,£85,£50
150 DATA£A0,£5,£B1,£4E,£AA,£96,£48,£88
160 DATA£10,£F8,£C8,£C4,£48,£F0,£27,£C4,£4B,£F0,£8,£B1,£49,£D1,£4C,
£90,£1D
170 DATA£F0,£EF,£A9,£1,£85,£50,£A0,£2,£B1,£4E,£48,£C8,£C8,£C8,£B1,£
4E,£AA,£68
180 DATA£91,£4E,£88,£88,£88,£8A,£91,£4E,£88,£10,£EB,£18,£A5,£4E,£69
,£3,£85,£4E
190 DATA£90,£2,£E6,£4F,£A5,£46,£D0,£2,£C6,£47,£C6,£46,£D0,£B5,£A5,£
47
200 DATA£D0,£B1,£A5,£50,£D0,£85,£60
204 REM
205 REM TEST PROGRAM
206 REM
210 DIM A$(9)
220 PRINTCHR$(12)"Enter 10 random words"
230 FORI=0TO9:INPUTA$(I):NEXT
240 PRINTCHR$(12)
250 FORI=0TO9:PRINTA$(I),:NEXT
255 PRINT
260 PRINT"NOW SORTING..."
270 DEFUSR=£400
280 P=USR(A)
290 FORI=0TO9:PRINTA$(I),:NEXT
300 END
```

```

0001 SA EPZ $9E 1F00
0002 EA EPZ $A0 1F00
0003 VN EPZ $B4 1F00
0004 BV EPZ $40 1F00
0005 AN EPZ $42 1F00
0006 ND EPZ $44 1F00
0007 CNT EPZ $46 1F00
0008 L1 EPZ $48 1F00
0009 A1 EPZ $49 1F00
0010 L2 EPZ $4B 1F00
0011 A2 EPZ $4C 1F00
0012 WV EPZ $4E 1F00
0013 SWAP EPZ $50 1F00
0014 LDA SA 1F00 A5 9E
0015 STA BV 1F02 85 40
0016 LDA SA+1 1F04 A5 9F
0017 STA BV+1 1F06 85 41
0018 LDA VN 1F08 A5 B4
0019 STA AN 1F0A 85 42
0020 LDA VN+1 1F0C A5 B5
0021 ORA ##80 1F0E 09 80
0022 STA AN+1 1F10 85 43
0023 S1 LDY ##0 1F12 A0 00
0024 LDA (BV),Y 1F14 B1 40
0025 INY 1F16 C8
0026 CMP AN 1F17 C5 42
0027 BNE S2 1F19 00 06
0028 LDA (BV),Y 1F1B B1 40
0029 CMP AN+1 1F1D C5 43
0030 BEQ S3 1F1F F0 1C
0031 S2 INY 1F21 C8
0032 CLC 1F22 18
0033 LDA (BV),Y 1F23 B1 40
0034 ADC BV 1F25 65 40
0035 PHR 1F27 48
0036 INY 1F28 C8
0037 LDA (BV),Y 1F29 B1 40
0038 ADC BV+1 1F2B 65 41
0039 STA BV+1 1F2D 85 41
0040 PLA 1F2F 68
0041 STA BV 1F30 85 40
0042 CMP EA 1F32 C5 A0
0043 BNE S1 1F34 00 DC
0044 LDA BV+1 1F36 A5 41
0045 CMP EA+1 1F38 C5 A1
0046 BNE S1 1F3A 00 D6
0047 SERR RTS 1F3C 60
0048 S3 INY 1F3D C8
0049 INY 1F3E C8
0050 INY 1F3F C8
0051 LDA (BV),Y 1F40 B1 40
0052 CMP ##1 1F42 C9 01
0053 BNE SERR 1F44 00 F6
0054 INY 1F46 C8
0055 LDA (BV),Y 1F47 B1 40
0056 STA ND+1 1F49 85 45
0057 INY 1F4B C8
0058 LDA (BV),Y 1F4C B1 40
0059 STA ND 1F4E 85 44
0060 S4 CLC 1F50 18
0061 LDA BV 1F51 A5 40
0062 ADC ##7 1F53 69 07
0063 STA WV 1F55 85 4E
0064 LDA BV+1 1F57 A5 41

```

```

0065 NDC ##0 1F59 69 00
0066 STA WV+1 1F5B 85 4F
0067 SEC 1F5D 38
0068 LDA ND 1F5E A5 44
0069 SEC ##1 1F60 E9 01
0070 STA CNT 1F62 85 46
0071 LDA ND+1 1F64 A5 45
0072 SEC ##0 1F66 E9 00
0073 BMI SERR 1F68 30 D2
0074 STA CNT+1 1F6A 85 47
0075 BNE S4A 1F6C 00 06
0076 LDA CNT 1F6E A5 46
0077 CMP ##2 1F70 C9 02
0078 BCC SERR 1F72 90 C8
0079 S4A LDA ##0 1F74 A9 00
0080 STA SWAP 1F76 85 50
0081 S5 LDY ##5 1F78 A0 05
0082 S6 LDA (WV),Y 1F7A B1 4E
0083 TAX 1F7C AA
0084 STA L1,Y 1F7D 96 48
0085 DEY 1F7F 88
0086 BPL S6 1F80 10 F8
0087 S7 INY 1F82 C8
0088 CPY L1 1F83 C4 48
0089 BEQ S9 1F85 F0 27
0090 CPY L2 1F87 C4 48
0091 BEQ S8 1F89 F0 08
0092 LDA (A1),Y 1F8B B1 49
0093 CMP (A2),Y 1F8D D1 4C
0094 BCC S9 1F8F 90 10
0095 BEQ S7 1F91 F0 EF
0096 S8 LDA ##1 1F93 A9 01
0097 STA SWAP 1F95 85 50
0098 LDY ##2 1F97 A0 02
0099 S8A LDA (WV),Y 1F99 B1 4E
0100 PHR 1F9B 48
0101 INY 1F9C C8
0102 INY 1F9D C8
0103 INY 1F9E C8
0104 LDA (WV),Y 1F9F B1 4E
0105 TAX 1FA1 AA
0106 PLA 1FA2 68
0107 STA (WV),Y 1FA3 91 4E
0108 DEY 1FA5 88
0109 DEY 1FA6 88
0110 DEY 1FA7 88
0111 TXA 1FA8 8A
0112 STA (WV),Y 1FA9 91 4E
0113 DEY 1FAB 88
0114 BPL S8A 1FAC 10 E8
0115 S9 CLC 1FAE 18
0116 LDA WV 1FAF A5 4E
0117 ADC ##3 1FB1 69 03
0118 STA WV 1FB3 85 4E
0119 BCC S9A 1FB5 90 02
0120 INC WV+1 1FB7 E6 4F
0121 S9A LDA CNT 1FB9 A5 46
0122 BNE S9B 1FBB 00 02
0123 DEC CNT+1 1FBD C6 47
0124 S9B DEC CNT 1FBF C6 46
0125 BNE S3 1FC1 00 05
0126 LDA CNT+1 1FC3 95 47
0127 BNE S5 1FC5 00 B1
0128 LDA SWAP 1FC7 A5 50
0129 BNE S4 1FC9 00 85
0130 RTS 1FCB 60

```

# Program Listing

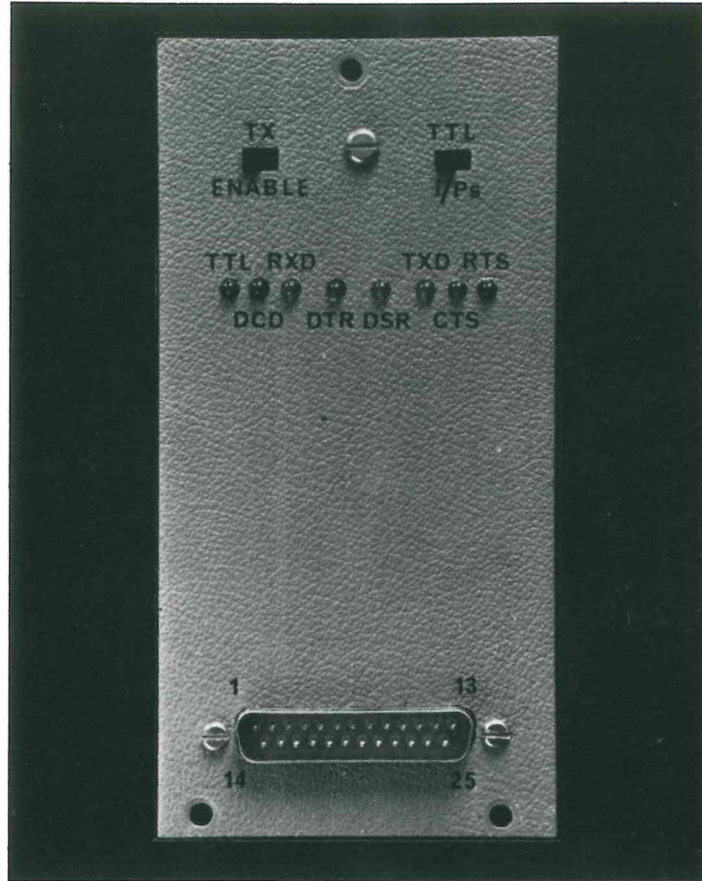


## Construction Corner

### Microtan Serial Input / Output Option Buffer and Status Indicator

This unit was designed and built to:—

- 1) Enable indication of the status of all RS232 modem control and data lines.
- 2) Allow normal RS232 levels for DCD, DTR, DSR, CTS, and RTS.
- 3) Allow front panel access to all the i/p / o/p option lines from Tanex, i.e. both RS232 and TTL.
- 4) Terminate the option with the standard 25W D-Type connector.



The unit was constructed on vero-board mounted on the front-panel of my system rack and linked to Tanex via a short length of 14W ribbon cable, terminated at both ends with a header plug. Power is linked in directly from the mother-board by a 4W ribbon cable.

The circuit allows selection from front-panel switches on 1) Transmitter enable, if a modem is not used and 2) TTL levels to be used on all input lines instead of RS232. Use is made of the 25W D-Type plug pins normally designated as "secondary channel", Ext Tx Clock, Data Rate Select and Busy Lines, to accommodate the TTL level inputs/outputs and the 20mA current loop facility as well as the RS232 inputs/outputs, see table below.

N.B. LK 2 on Tanex must be left open.

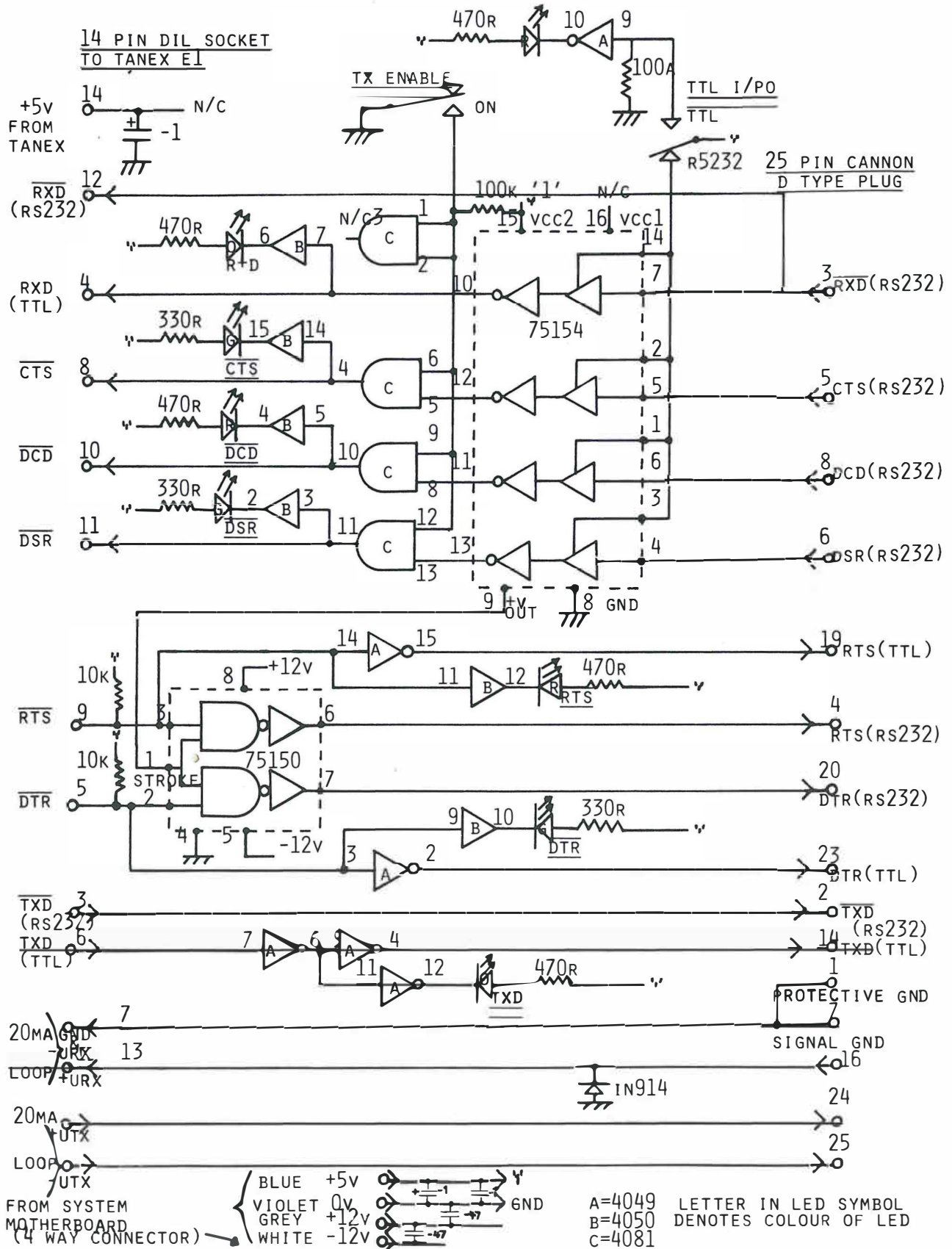
### 25W D-TYPE PLUG CONNECTIONS

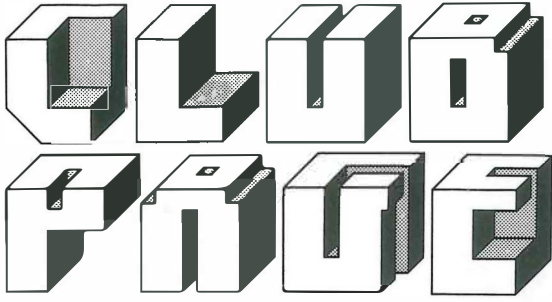
PIN No		USE IN THIS CASE	STANDARD USE
1		PROTECTIVE GROUND	PROTECTIVE GROUND
2		TXD (RS232)	TXD (RS232)
3		RXD (RS232)	RXD "
4		RTS (RS232)	RTS "
5		CTS (RS232)	CTS "
6		DSR (RS232)	DSR "
7		SIGNAL GROUND	SIGNAL GROUND
8		DCD (RS232)	DCD (RS232)
14	++	TXD (TTL)	SEC TXD
16	++	+VE RX (20MA LOOP)	SEC RXD
19	++	RTS (TTL)	SEC RTS
20	++	DTR (RS232)	DTR (RS232)
23	++	DTR (TTL)	DATA RATE SELECT
24	++	+VE TX (20MA LOOP)	EXT TX CLOCK
25	++	-VE TX "	BUSY

++ = DIFFERENCES

= TO MODEM/PERIPHERAL  
= FROM MODEM/PERIPHERAL

# MICROTAN I/P/O/P OPTION AND INDICATOR PANEL





If you have formed a club or user group in your area for the Oric-1 please keep us informed of your activities and meetings so that we can print the information on this page. We would be delighted to receive any newsletters or circulars that you produce. If you feel like dropping us a line to tell us of your aims or projects and membership rates we will try to give you a mention.

All correspondence to: Club Page, Oric Owner Magazine, 3, Club Mews, Ely, Cambs. CB7 4NW.



---

## Coming Next Issue...

- We talk to Peter Harding, Sales Director of Oric.
- How to get more out of your Oric.
- Array saving on cassette.
- Plus all the regulars: News Brief, Captain Tanex, And Forthly, and your programs.

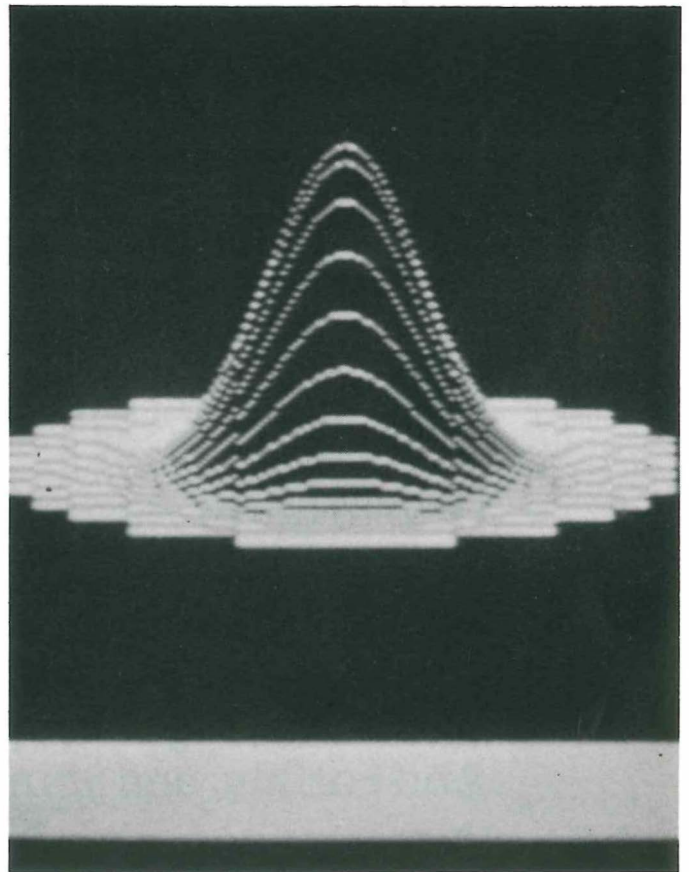
# and Much Much More!

# Oric Quickies

## 3 dimensional function plot

This short routine uses the Hi-Res graphics to plot a 3 dimensional representation of a function. The function is defined in line 5 of the listing and can be altered to give different pictures. The program as it stands takes about 10 minutes to run.

```
3 HIRES
5 DEFFNA(Z)=90*EXP(-Z*Z/100)
10 HIRES:CLS
15 PAPER4:INK1
20 Z=1:X=0
110 FORQ=-30TO30STEP.4
120 L=0
130 Y1=5*INT(SQR(1000-Q*Q)/5)
140 FOR R=Y1TO-Y1STEP-2
150 S=INT(25+FNA(SQR(Q*Q+R*R))-.7*R)
160 IFS<=LTHEN190
170 L=S
180 Y=S:GOSUB1900
190 NEXT R
200 X=X+1
210 NEXTQ
380 GOTO180
1900 IFX>239ORX<0THENRETURN
2000 IFY>199ORY<0THENRETURN
2100 CURSETX+50,199-(Y+50),1
2200 RETURN
```



---

## Subscription

---

# Why you should subscribe to ORIC OWNER

Your Oric 1 is one of the most powerful low-cost computers on the market today. Oric Owner is the only magazine totally dedicated to the Oric and its supporting hardware and software.

From issue to issue Oric Owner is able to bring you in-depth information on the Oric, advance news on the latest add-ons, superb programs to try and 'behind the scenes' interviews with the engineers who designed it.

If you wish to keep ahead of the latest developments and news on the Oric then Oric Owner is essential. At only £10 for a year's subscription of 6 issues (£15 for overseas readers) can you afford to be without Oric Owner?

**SUBSCRIBE TODAY!**

---

## subscription form

Please send me the next 6 issues of **ORIC OWNER**

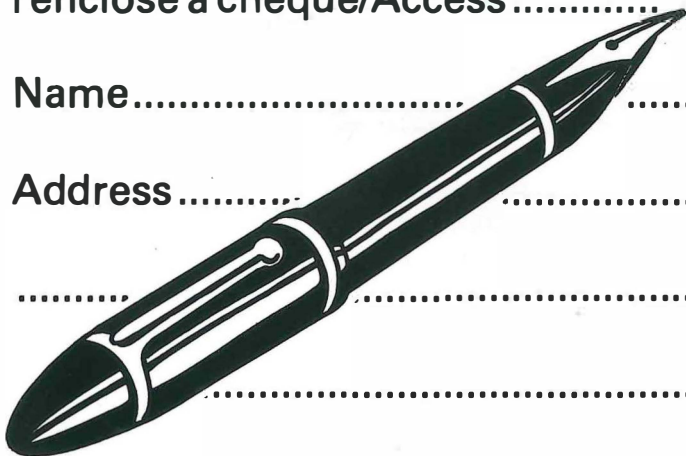
I enclose a cheque/Access ..... Barclaycard ..... for £10

Name .....

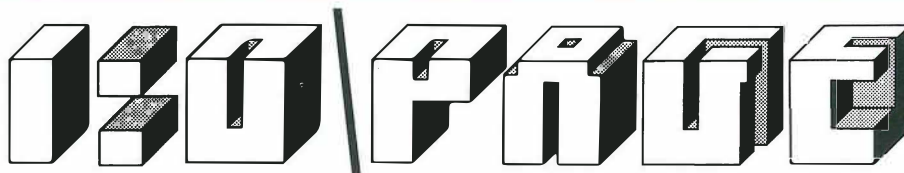
Address .....

.....

.....



# Regulars



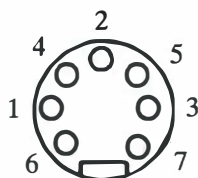
Dear Sir,

I have recently received my Oric and am experiencing great difficulty in loading cassettes in. I have tried a number of cassette recorders and assorted leads with little success. The sound seems to be getting to the Oric because it comes out of the built in speaker quite loudly. Is there anything else you can suggest that I should do?

Yours faithfully,  
D. Turner, Ealing, London.

Editor: From your description of the problem I would immediately suspect your lead. The pins on the socket at the back of the Oric have a number of functions and if they are incorrectly connected up then all sorts of nasty things can happen. The socket connections are as follows:

- 1 = Tape out
- 2 = Ground
- 3 = Tape in
- 4 = Sound (Ext.spkr)
- 5 = Sound (Ext.spkr)
- 6, 7 = Relay Contacts



Many 5 pin DIN leads have pins 4 and 5 shorted together. These will not work correctly on the Oric and should be modified.

If you find that when a program is loaded in half of it is missing or corrupted it is a safe bet that the volume is either too low or too high so try varying it. As a general rule Oric likes the sound to be fairly loud with lots of treble. When a load is taking place you should be able to hear the sound very faintly from the Oric's speaker.

Dear Sir,

I have just received my Oric a fortnight earlier than expected (!). So far I have been delighted with the machine as it seems to do a lot more than stated in the adverts. One thing that concerns me is the mixed comments that some of the reviews in the magazines have had. Can you put my mind at rest?

R. Wilson, Liverpool.

Editor: Due to the keenness of many of the magazines to review the computer, Oric Products were forced to send out pre-production models. These had all sorts of nasties in them such as bad T.V. signals and preliminary ROMs. I think that some of the reviewers did not realise this and were perhaps harder on the machine than necessary. It is probable that they will be taking a second look at the machine when they have more information and full production models of it. Some of the comments were obviously due to lack of information than any fault of the machine. At least two reviews said that you cannot have more than two colours on the screen at any time, this, of course, is complete nonsense as the Demo tape supplied plainly shows that all the colours can be used in all modes.

All letters for the I/O Page should be sent to:

The Editor, Oric Owner Magazine, 3, Club Mews, Ely, Cambs. CB7 4NW.

# ORIC SOFTWARE

ORIC  
SOFTWARE



Oric-  
Chess

## Oric-Chess

A superb chess game written in basic with a machine code move search routine. Features high resolution colour graphics, record of moves. Allows castling and en-pessant 5 levels of play and computer can play black or white.

Only £9.99 inc. V.A.T. 48K only.

ORIC  
SOFTWARE



Zodiac

## Zodiac

A 36K adventure game that will test your intelligence and patience for hours. Find the 12 hidden signs of the zodiac to discover incredible treasures. Can you ward off the angry Yeti, what is the meaning of the radio that plays the 'Archers' over and over again (through your loudspeaker!)?

Only £9.99 inc. V.A.T. 48K only.

ORIC  
SOFTWARE



Oric-Base

## Oric-Base

A full data management and file retrieval package. Allows sorting of data reports to be printed, conditional data retrieval and selective updating.

Only £15 inc. V.A.T. 48K only.

Please send me:

Oric-Chess @ £9.99  
Zodiac @ £9.99  
Oric-Base @ £15  
Postage per cassette is 40p.

Quantity

Total

Name: .....

Total:

Address: .....

.....  
All cheques payable to Tansoft Ltd., 3, Club Mews, Ely, Cambs. CB7 4NW.

Tansoft Ltd., 3 club mews, ely, cambs, cb7 4nw.