



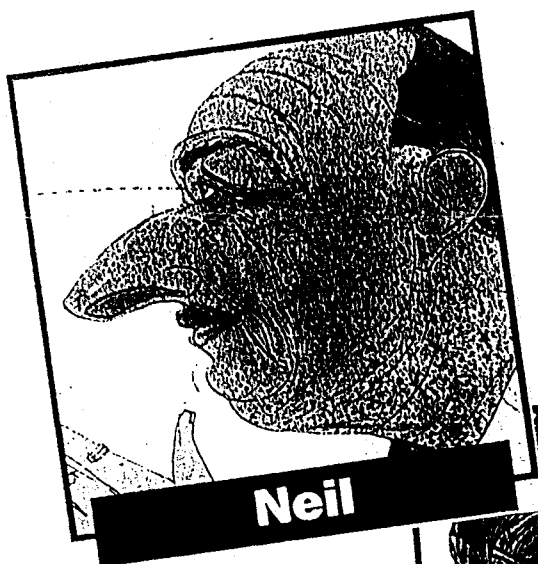
**USER
MONTHLY**

with Oric Enthusiasts

*Europe's longest running
Oric Magazine*

Number 56

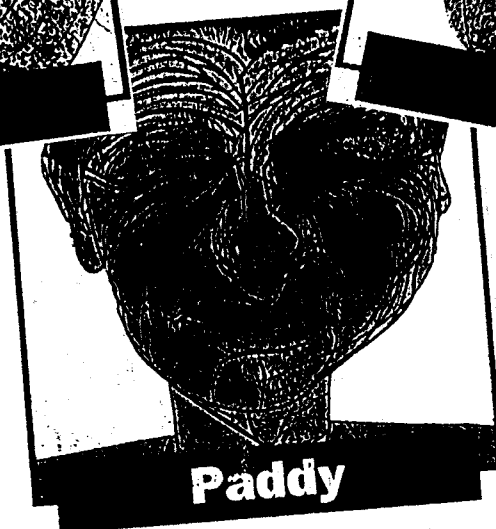
April 1992



Neil



John



Paddy

ELECTION SPECIAL

Edited and Distributed by Dave Dick, 65 Barnard Crescent, Aylesbury, Bucks HP21 9PW

THE EDITORIAL

PAGE 2

HELLO AND WELCOME,

TO ANOTHER PACKED ISSUE OF 'ORIC USER MONTHLY'.
SPRING IS IN THE AIR AND THE LONG WINTER NIGHTS ARE DRAWING TO AN
END. ORICS WILL BE CONSIGNED TO LIE IDLE AS PEOPLE'S THOUGHTS TO
OUTDOOR PURSUITS. READERS WILL BE TAKING THEIR O.U.M TO READ IT IN
THE GARDENS OF A LOCAL PUB, WHILST THE DREADED MCP40 WILL BE USED TO
WEDGE OPEN DOORS TO LET SOME COOL AIR IN.
GO ON FOLKS - ENJOY LIFE!

FOR THE FIRST TIME THAT I CAN REMEMBER; I HAVE ENOUGH ARTICLES TO
FILL 2 ISSUES. WE HAVE HAD TO HOLD SOME ITEMS OVER. I HOPE THAT I
HAVE FOUND THE RIGHT MIX BETWEEN THE SERIOUS AND THE GAMES SIDE OF
COMPUTING. IF I HAVEN'T, THEN I'M SURE THAT YOU WILL LET ME KNOW.

STARTING NEXT ISSUE, WE HAVE PART ONE OF BERNHARD GRONE'S ARTICLE ON
'ORIC READS AND WRITES MSDOS FILES'. IT IS FORMULATED IN 'FORTH'.
(DON'T FORGET THAT DISC USERS CAN GET THE EXCELLENT 'FORTH F83' FROM
THE CEO VIA JON HAWORTH. NOW TO THIS 27 PAGE ISSUE:

THE INDEX

PAGE 1 THE FRONT COVER - JON HAWORTH STRIKES AGAIN! - IN
'PARTY' MOOD.

PAGE 2 EDITORIAL AND INDEX

PAGE 3 NEWS....NEWS.....NEWS.

PAGE 4 READERS LETTERS

PAGE 5 BITS AND BOBS

PAGES 6/7/8 RAMBLING IN THE ROM with Jon Haworth.

PAGES 9/10/11 MACHINE CODE FOR THE ATMOS (Pt.14) - Peter N. Bragg

PAGE 12 THE GAMESTER - YOUR EDITOR DELVES

PAGE 13 THE CHEAT - Paul Baker gets into games.

Page 14 'CAPTAINS LOG' map - information supplied by Graeme
Burton; Artwork by Paul Baker. Next issue we will print where to find
certain objects.

Page 15 BACK ISSUES - deals with nos. 41 to 55.

Page 16 SOFTWARE HOUSES - Brian Kidd delves into who released
what!

Page 17 Listing of 'RADAR LANDING'

Page 18 MORE LISTINGS - 'ANAGRAM', 'ARTIST' and 'KALEIDOSCOPE'.

Page 19/20 65802 ASSEMBLER (Pt. 1) - Bernhard Grone.

Page 21/22 'THE ORIC SERIAL PORT' - part one from Trevor Shaw.

Page 23-27 ORIC ENTHUSIASTS with Alan Whitaker, WHO LOOKS AT
ANOTHER BOOK + MORE ON GRAPHICS AND MACHINE CODE TECHNIQUES, PLUS
MORE ON DISC FILE HANDLING.

=====

MAY DEADLINE

Articles for inclusion in the May issue of O.U.M should be with me
by April 23RD. PLEASE !!!

O.U.M READERSHIP

Issue 49 - 100 readers, Issue 50 - 100 readers, Issue 51 - 100
readers, Issue 52 - 99 readers, Issue 53 - 98 readers, Issue 54 - 97
readers, Issue 55 - 102 readers.

Last year I set a target of 100, which we are just about seeing
regularly. Next target is 120, by the end of June - with YOUR help -
See next item

O.U.M MAILSHOT

With the March issue, 33 members were asked to send out 2 mailshot
letters. Thus if you all did it, then 66 people were contacted. As we
go to press, I have had replies from 12 people, which is about 19%.
Well done to everyone.

WITH THIS ISSUE, SOME MORE READERS WILL BE ASKED TO DO THEIR STUFF.

PARIS MEET

THE NEXT FRENCH ORIC MEET WILL TAKE PLACE IN PARIS ON SATURDAY JUNE 13th.

IF ENOUGH BRITISH READERS WANT TO GO , THEN WE WILL CHARTER A BICYCLE.

TEN YEARS OF THE ORIC

IN THE LATEST ISSUE OF THE 'CEOMAG' THE SAME THOUGHT CROSSED THE MINDS OF THE FRENCH AS THAT OF JON HAWORTH'S.

NEXT YEAR IS THE TENTH ANNIVERSARY OF THE ORIC. JON SUGGESTS WE SHOULD TRY AND GET SOME OF THE ORIGINAL 'ORIC GUYS' TO A BIG MEET/CELEBRATION.

CEO SOFTWARE

VINCENT TALVAS ASKS ME TO POINT OUT THAT CEO SOFTWARE IS AVAILABLE TO BOTH CEO AND OUM READERS AT THE SAME PRICE RATE, BUT UNFORTUNATELY THE BACK ISSUES OF QUARTERLY DISCS (JEO'S) ARE ONLY FOR CEO MEMBERS.

APOLOGIES IF THIS FACT WAS NOT MADE CLEAR PREVIOUSLY.

THE BACKLOG

I AM GRADUALLY CLEARING THE BACKLOG OF ORIC RELATED MATTERS. AS WE GO TO PRINT, ALL SOFTWARE ORDERS SHOULD OF BEEN DESPATCHED. CORRESPONDENCE IS ONLY ABOUT 2 WEEKS BEHIND. THEREFORE THE OUM PHONE IS NOW FULLY OPERATIONAL.

THE ONE RULE IS : PLEASE DON'T RING ON A THURSDAY OR WHEN LIVERPOOL HAVE A SOCCER MATCH ON T.V

PROTEK

I NOW HAVE A FEW PROTEK PROGRAMMABLE JOYSTICK INTERFACES KINDLY DONATED TO THE CLUB BY STEVE HOPPS (OF "GET YOUR HOLIDAYS AND DRIVES FROM ME" FAME).

BEFORE I TELL YOU THE PRICE, I AM GOING TO FIRST TEST THEM. ORIGINALLY THEY SOLD FOR NEARLY 30 POUNDS.

I HAVE HAD ONE FOR MANY YEARS. THEY ARE HANDY FOR GAMES THAT ARE NOT NORMALLY JOYSTICK COMPATIBLE. THEY ARE A LITTLE FIDDLY TO SET. IF FOR INSTANCE YOU WANT TO UTILISE THE 'A' KEY FOR LEFT MOVEMENT, THEN WHILST HOLDING THE JOYSTICK WITH ONE HAND, YOU MOVE THE JOYSTICK TO THE LEFT POSITION WITH THE OTHER HAND AND WITH YOUR BIG TOE YOU HAVE TO HOLD DOWN THE 'A' KEY. IT IS DAMN GOOD EXERCISE. IT IS WORTH THE BOTHER JUST TO PLAY 'THE ULTRA' WITH A JOYSTICK.

JOHN POPS IN

MAKING A VISIT TO OUM THE OTHER SATURDAY WAS JOHN PEACH FROM NORTH LONDON.

JOHN HAS JUST CONVERTED TO A DISC DRIVE.

AS WELL AS OWNING AN ATMOS, HE HAS A SPECTRUM AND A DRAGON 32. ANY OTHER DRAGON OWNERS OUT THERE?

A DELIGHTFUL AFTERNOON WAS SPENT DISCUSSING MANY TOPICS, INCLUDING: ALAN SUGAR, SIR CLIVE SINCLAIR AND MUSIC.

A PLEASURE TO MEET WITH YOU JOHN AND I HOPE TO SEE YOU AGAIN.

=====

DID YOU KNOW?

DID YOU KNOW THAT THE 'PEACHBYTES' CATALOGUE OF MAY 1984 ADVERTISED THE WORD PROCESSOR 'WRITER' AT 30 POUND + V.A.T. ?

ALSO LISTED WERE THE 'BYTE DRIVE 500' DISC SYSTEM FOR 260 POUNDS AND A SEIKOSHA GP100A PRINTER FOR 215 POUNDS (PLUS V.AT OF COURSE).

MY, HOW TIMES CHANGE!

=====

HI DAVE,

YES, I still have my old Oric Atmos on a top shelf covered in a fine dust. Perhaps I'll put a lead on it and give it a RUN!!
I enclose 5 x 24p stamps for the OUM pack.

P.S. I had a bit of a surprise with your letter/ad. dropping through the letter box.

- D.R. HUGHES (Prestatyn).

Dear Mr. Hughes,

nice to hear that you have been spurred into action.

In fact within ten days of me sending out mailshot letters to OUM readers for onward posting, there have been 4 such replies dropping on my mat.

It's nice to see that our readers are responding to my request, and more of them will be asked to participate this and next month.

- DAVE

=====

DAVE

Do you have a circuit diagram for the 16k ORIC 1 ?

- Michael Gynane (Huyton)

Hello Michael,

I'm afraid you've caught me out there. From looking at an old 16k machine some while ago; I seem to remember that : a) it had only 2 memory chips (Drams) and b) the ULA was different.

I am also informed that the ROM is the same as in the 48k.

If anyone has more info. or indeed a circuit diagram, then perhaps they could let us know.

- DAVE

=====

HI DAVE,

It's me again (Judy).

Firstly, regarding the pokes printed for GRENDAL in the last issue. For disc users, you need to do the following: Enter them at the start of the file 'PROG.COM' at the start of the listing, then resave and then reload the disc. (Note from the editor - it would be wise to back up your original first).

Secondly - do you know of a short program to redefine just 2 letters in the print set?

- JUDY SIMMS (Edgbaston).

HELLO JUDY,

ta for the info. on where to put the pokes.

As for your second item, perhaps our readers can help out.

=====

=====

Z O O L Y M P I C S

Just in are some cassette versions of ZOOLYMPICS that marvellous animal sports game from No Mans Land for the Oric 1/Atmos. GET YOURS NOW from OUM for only 3 pound incl. postage. For each one sold; a pound will go into OUM funds.

PICTURE BOOK and STORY BOOK

The above 2 titles on cassette from SOFTBACKS are now back in stock. They are excellent for kiddies. Price is 2.75 each, with 75 pence going to OUM funds.

ARNT ARRIVES !

On February 29th. it was my pleasure to meet up with Arnt Erik Isaksen our Norwegian reader and author of NHL ICE HOCKEY MANAGER.

Arnt took me through the Calcutta cup stage of said game, which he plays on average about 2 hours per day. He takes the role of manager of Edmonton. Arnt also enjoys soccer simulations. He has stocked up on pens and paper for an MCP40, only to find that the printer he bought does not work. So if anyone has a working MCP40 for sale, then Arnt is in the market for one.

After about 5 hours on the Oric, Arnt said his farewells, but not before he thrashed me at FOOTBALL (the arcade soccer game from Alistair).

As we said cheerio at the station, Arnt said that if funds permitted that both he and Staale Eikbraaten of Sweden would like to attend our July Oric meet.

GRANDAD

GRANDAD, the Quilled adventure from Paul Baker is now ready for release. Paul reviewed his own game in the last OUM.

It is ONLY available on cassette. All you have to do is to send 3 pound to OUM.

SAUSAGES

Paul Baker is currently working on a game called SAUSAGES, which is a sci-fi/comedy adventure. It comes with an accompanying short story, which he has written in his spare time.

KEEP UP THE GOOD WORK PAUL

MUSED

'MUSED', the musical editor is now available. It comes with demos and manual.

It is ONLY available on disc.

Details of this work from Jonathan Bristow have been published in previous issues.

PRICE on 3" disc is 5 pound. On 5.25" disc, it is yours for 4 pound. Orders to OUM at usual address.

4 th ORIC MEET

The fourth Oric meet takes place in AYLESBURY on SAT, July 18th.

PLEASE get your tickets early as ; A) I have to pay for the hire of the venue in advance and B) I will not be available to sort out your travel and other queries between JUNE 27th and JULY 13th and again on the 16/17th of JULY.

Tickets are 2 pound each. If you want to demonstrate anything, please let me know. I will publish a map of how to get to the venue in the next issue.

ALL FORMATS FAIRS

More dates for your filofaxes. All Formats Computer Fairs are open from 10 a.m until 4 p.m. and admission is 4 pound. The HOT LINE is 0926 613047.

Proprietor is Bruce Everiss (where do we ORICANS know him from?? - I just know that you are going to tell me)

APRIL 12 - NORTH EAST - Northumbria Centre, Washington, A194 (M)

APRIL 26 - WEST MIDLANDS - National Motorcycle Museum, J6, M42

Club Europe Oric

Dave's piece last month about the drop in cassette sales foreshadowed a similar experience with C.E.O. renewals. Because many are upgrading to disc (and more power to thier elbows) cassette renewals for 1992 numbered only three. That simply does not justify the effort I put into transferring the CEO discs to tape, time which could be much better spent on other things. As those who renewed know, they have been offered fair alternatives to their cassette membership of C.E.O. To the world at large, C.E.O. subs are now restricted to two variations: monthly magazine only at £11 per year, disc membership at £22. Can I stress that this decision does not affect either CEO software on cassette (Willy, Tetris, etc) nor the P.D. library, which continues on both disc and cassette.

Bumbling on....

This month RESTORE, STOP, END, CONT and RUN....note the amazing muddle at C92E/F...

'RESTORE' (COMMAND)

Entry: Nothing

Exit: AY=#B0-#B1=Start of BASIC-1

C91F	SEC	C952	SEC	
C920	LDA 9A	C953	LDA 9A	
C922	SBC #01	C955	SBC #01	
C924	LDY 9B	C957	LDY 9B	
C926	BCS C929	C959	BCS C95C	Calculate the start of BASIC text
C928	DEY	C95B	DEY	in AY
C929	STA B0	C95C	STA B0	
C92B	STY B1	C95E	STY B1	and place it as the data pointer
C92D	RTS	C960	RTS	
C92E	NOP	?
C92F	RTS	C961	RTS	??

TEST FOR CTRL-C

Remark: this is the only routine that treats Ctrl-C

C930	LDA 02DF	C962	LDA 02DF	Take keyboard buffer
C933	BPL C92E	C965	BPL C960	if empty, exit
C935	AND #7F	C967	AND #7F	if not, eliminate bit 7 (pointless)
C937	LDX #08	C969	LDX #08	(pointless)
C939	CMP #03	C96B	CMP #03	Test for Ctrl-C
C93B	BNE C92E	C96D	BNE C961	no, exit (yes, C=1)

C93D CMP #03 C96F CMP #03 Twice is better than once? Useless

'STOP' (COMMAND)

C93F BCS C942 C971 BCS C974 Ensure C=1 (if no parameter)

'END' (COMMAND)

Bug: On V1.1 the printer is not correctly turned off line. This can cause problems with the line length if the printer has been put on line using the routine at #C816.

C941 CLC	C973 CLC	Indicate END, not STOP
C942 BNE C987	C974 BNE C9B9	Exit if parameters entered
C944 LDA E9	C976 LDA E9	
C946 LDY EA	C978 LDY EA	Take TXTPTR
C948 BEQ C956	C97A BEQ C988	jump if direct mode
C94A STA AC	C97C STA AC	and save for eventual recovery
C94C STY AD	C97E STY AD	
C94E LDA A8	C980 LDA A8	and save the line number as well
C950 LDY A9	C982 LDY A9	
C952 STA AA	C984 STA AA	
C954 STY AB	C986 STY AB	also for eventual recovery
C956 PLA	C988 PLA	Remove the return address
C957 PLA	C989 PLA	to adjust the stack pointer
C958 LDA #C1	C98A LDA #BD	
C95A LDY #C3	C98C LDY #C3	AY points to 'BREAK'
C95C LDX #00	C98E LDX #00	
C95E STX 02F1	C990 STX 02F1	Indicate printer OFF (wrong for V1.1)
C961 STX 02DF	C993 STX 02DF	empty keyboard buffer
C964 STX 2E	C996 STX 2E	no Ctrl-O
C966 BCC C96B	C998 BCC C99D	Jump if END
C968 JMP \$C4AA	C99A JMP \$C49D	if Ctrl-C or STOP, display BREAK and return
C96B JMP \$C4B5	C99D JMP \$C4A8	if END, jump directly to the interpreter

'CONT' (COMMAND)

Principal:

If the restart is authorised, it sets TXTPTR to its old value (pointing to a 00 or ':', unless there is a syntax error. The line number is also adjusted.

C96E BNE C987	C9A0 BNE C9B9	Exit if parameters entered
C970 LDX #D7	C9A2 LDX #D7	Prepare X for 'CAN'T CONTINUE'
C972 LDY AD	C9A4 LDY AD	test 'CONT' flag
C974 BNE C979	C9A6 BNE C9AB	jump if <>0, it's O.K.
C976 JMP \$C485	C9A8 JMP \$C47E	display the error
C979 LDA AC	C9AB LDA AC	and take the low byte
C97B STA E9	C9AD STA E9	
C97D STY EA	C9AF STY EA	and replace TXTPTR
C97F LDA AA	C9B1 LDA AA	

C981	LDY AB	C9B3	LDY AB	Take the line number
C983	STA A8	C9B5	STA A8	
C985	STY A9	C9B7	STY A9	and adjust it as well
C987	RTS	C9B9	RTS	
C989	JMP \$D2A0	C9BA	JMP \$D336	'ILLEGAL QUANTITY ERROR'

'RUN' (COMMAND)

C98B	BNE C990	C9BD	BNE C9C2	Exit if parameters entered
C98D	JMP \$C733	C9BF	JMP \$C708	if not, set TXTPTR and do a CLEAR
C990	JSR \$C73A	C9C2	JSR \$C70F	do a CLEAR
C993	JMP \$C9AA	C9C5	JMP \$C9DC	then GOTO

'GOSUB' (COMMAND)

Principal:

The routine saves TXTPTR and the current line number on the stack so that it can return from whence it came.

TXTPTR is saved before the evaluation of the line number following the GOSUB so that the routine can go neatly into GOTO. The RETURN must jump the number before restarting execution of the program.

The return address to the interpreter remains on the stack, unlike the FOR routine. This is because RETURN exits with a simple RTS. But it is no longer at the top of the stack, which is why you jump directly to the interpreter.

C996	LDA #03	C9C8	LDA #03	
C998	JSR \$C43B	C9CA	JSR \$C437	Request 6 bytes on the stack
C99B	LDA EA	C9CD	LDA EA	Put TXTPTR high byte on the stack
C99D	PHA	C9CF	PHA	
C99E	LDA E9	C9D0	LDA E9	and low byte
C9A0	PHA	C9D2	PHA	
C9A1	LDA A9	C9D3	LDA A9	Put line number high byte on stack
C9A3	PHA	C9D5	PHA	
C9A4	LDA A8	C9D6	LDA A8	and low byte
C9A6	PHA	C9D8	PHA	
C9A7	LDA #9B	C9D9	LDA #9B	and the token for GOSUB
C9A9	PHA	C9DB	PHA	
C9AA	JSR \$00E8	C9DC	JSR \$00E8	Take current char.
C9AD	JSR \$C9B3	C9DF	JSR \$C9E5	do a GOTO
C9B0	JMP \$C8AD	C9E2	JMP \$C8C1	and return to the interpreter

 So what do you think of it so far ?

----- Eric and Ernie fans have an answer to that. But seriously, the aim of this series is to show how it is possible to get more out of our favourite machine. However over the past year or so, the scene has changed quite a lot and the Oric which was supposed to have "died" many years ago, is still a very lively and rather noisy "corpse". Much of the blame for this situation can be laid at the doors of the DUM and CED magazine editors. The result has been an influx of new readers, one or two of whom might even read a little of this and thereby treble my readership figures.

A series of this type that doesn't cater for a constantly changing readership, is about as useful as a chocolate teapot. With this in mind, it is worth pausing occasionally, for a brief overview of what we are doing and why.

This series is not intended to be a complete and detailed course on machine code programming. That is something which has already been well covered many times before in other publications. Instead, I have tried to break the series into a number of separate topics, which have been roughly linked by an initial "Story so far" heading. The idea behind this is to enable the reader to gain something, even if they have only read a couple of the articles.

However, at this stage there may be some new readers who are baffled by some of the references to items such as the "Accumulator" or the "Flags". Obviously it would be a bit pointless to go back and repeat everything again. It is probably better to list the main subjects covered so far, together with the DUM issue number so that if you do lack a particular issue that might help, you can contact our illustrious editor, Dave, who can no doubt be bribed to provide you with the necessary copy. Below, you will see a brief list the articles so far -

Part	Subject	Issue
----	-----	-----
1)	Machine code programming and Oric described.	(43)
2)	"Hexloader" for reading and writing programs (updated later in Part 7).	(44)
3)	Instructions & Registers - The Accumulator and the Program Counter.	(45)
4)	Decisions and how the Oric makes them - "Flags" and Status Register.	(46)
5)	A small Instruction Set described.	(47)
6)	A simple "Screen Fill" demo routine using small Instruction Set.	(48)
7)	Simple programming aids and updated "Hexloader" program (see above).	(49)
8)	Data/Parameter Blocks and demo "Copy" routine.	(50)
9)	More on the demo "Copy" routine.	(51)
10)	Jump instructions and updated small Instruction Set.	(52)
11)	Keyboard input, using Oric's operating system.	(53)
12)	Demo routine for "Keyboard Control".	(54)
13)	More on "Keyboard control" routine.	(55)

To keep things as simple as possible, all the articles so far, have used the minimum number of instructions possible in the hope that it will enable the reader to get a better understanding of how machine code programming works, without having to cope with dozens of different instructions at the start. All the above demo routines used a simplified Instruction Set of just four types, a total of 16 instructions in all. The aim is to eventually cover the whole 6502 Instruction Set in easy stages.

Many people think that programming, particularly in machine code, is very difficult and some are also terrified of "crashing" the computer, so let's have brief look at these points in this overview.

Carry on Crashing

----- First, let's look at the dreaded "crash". A crash does no harm to the computer at all, so don't worry about it. Anyway, it is all part of the programming process. Very few programs run first time and any that do, are often looked at with some suspicion, until they have been thoroughly tested. What usually happens when the Oric "crashes" is this. Although you may not be aware of it, the Oric's Operating System also does a number of "housekeeping" jobs in the background as well as running your programs. These include keeping the display going and checking the keyboard to see if any keys are being pressed. If a programming error interferes with this process, the operating system may stop making its regular check on the keyboard, so any key you press will then be ignored and you will need to RESET or even switch off, to regain control of the keyboard. You may also see some rubbish on the screen or other effects, depending on the type of program error. Providing your program has been saved on tape or disk, this is no problem, just reload and re-check the program again. Note, you should always save new programs before testing them.

Easy Programming

----- Programming may seem difficult at first. The secret is to start small and be prepared to experiment. You will learn far quicker and easier that way and hopefully it should be more interesting. Long routines can be hard to produce and debug. As I like to enjoy my computing, I write my own programs as a series of small routines and build up longer programs from them. A small routine is easier to write and get working. You can then link up the successful routines to produce large programs, confident in the knowledge that at least the individual parts work, so it should be fairly easy to get the whole thing operating correctly.

However, if you have already written a longish program that will not work, break it up into small pieces and check each piece separately, to make sure that it does what it is supposed to do and nothing else. Build up from there by putting several pieces together and testing again. Eventually you will find the cause of the problem. However, it really is best to do your programming in small bits, checking that each part works as you go along.

Why bother to write software ?

----- This is a question that always surprises me. There is a large software selection available for the Oric, but even the most popular computers do not have a complete range, sufficient to satisfy everyone. The software available for any microcomputer is only a starter. If you don't like the way it works, you can change it, by writing your own. That is the main advantage of a computer. It is very different when you buy a new car, washing machine or camera for example, there you are limited, to what is available in your price range and you are stuck with it, whether you like it, or not. If you have a computer, you can add a lot to it, using little more than pencil and paper. For example, if you have another interest/hobby, it's a sure thing that there is some application in it where your computer could be very useful. That is just one example, the only real limits are your own imagination.

While we are having this general overview, let's explode a couple of myths apparently widely held by many self acclaimed "experts". One favourite myth is that computers do not use machine code any more. Frankly this is rubbish. Strictly speaking machine code is hex code which is the only language that is common to all microcomputers, unless you are into binary or have eight legs !!

Another old favourite is "machine code is limited in capability compared with high level languages". Wrong, it is really the other way round. A good well written high level language can be nearly as effective as machine code.

Essentially it is a trade-off between ease of use against the inevitable restrictions in speed and operation. You pay your money and take your choice. Putting it very simply, any language is made up of prefabricated chunks of machine code program. These "chunks" are given labels by which they can be called up and used. The labels PRINT or ZAP are examples used in Oric Basic. The label is used by Oric Basic to call up the appropriate machine code routine when asked to do so, either by your Basic program or command line. The same principle applies to all languages, however high level they may be.

Machine Code ?

----- Although machine code is binary, hex code is exactly the same, thing, but much easier to use. It is a numbering system that uses a base of 16 instead of the more familiar decimal base 10. To keep the base set as single digits, hex code uses the first 6 letters of the alphabet (A to F) instead of the numbers 11 to 15. It may seem a little strange at first, but then so was the decimal money system, when it was first brought in. At least we don't have to use hex code to calculate. If you are working out screen graphics, it can make life a lot easier, because it is easy to convert to binary if required. Hex code also makes memory addressing far easier as we will see later on in the series.

You can use an assembler for writing your program into the Oric. I write my own programs in plain English and assembly language. Being lazy, I then enter the program as hex code and use a dis-assembler to list out and check that I have entered it all correctly.

The Great Reset Button Basher

----- How do you write your programs ? So far there seems to be two schools of thought on this one. One is the keen hacker armed with an exercise book and a biro. Six months later, he is up to the armpits in biro ink and adrift in a sea of scribbled alterations and bugs. His is a rare breed, but I have known them.

The "techie" type on the other hand puts everything on the computer straight away, which is not too bad for fairly short and simple programs, but can result in alphanumeric indigestion, when larger projects are attempted. He needs to be very efficient with REM information statements and text formatting.

I find it easier to use a combination of techniques. I use a wordprocessor to write notes and a disassembler to print listings. However, these do have limitations in display and printout. It is true to say that a picture is worth a thousand words and a small pencil sketch is often useful. For a lot of my designing, planning and programming over the years, I have used a notepad and a pencil, specifically one of those 8" x 5" reporter pads and an Ø5mm "F" lead pencil plus a "Pentel" pencil shaped eraser, ideal for making small, neat alterations to listings. The note pad fits into an ordinary small writing case which also holds a set of "crib cards" which contain essential information, such as instruction sets and computer system addresses. This enables me to write programs anywhere. I never use pens or biro for this job, they are difficult to erase, when you want to make changes and modifications. No design or program is ever final. There is always some modification or improvement that comes to mind, often while still working on the current project. Computer programming should be fun, if it isn't, you may be taking it too seriously.

Political Manifesto

----- How about fitting an Oric and electronic camera to the front door together with a heavy calibre machine gun and software guaranteed to pick out any political lapel badge !!Next month, where to put your "X".

FISHY BUSINESS

FISHY BUSINESS from SALAMANDER could be a collectors item. Only 100 copies were issued

MINI-REVIEW

An old game that I had a bit of fun with recently is QUACK A JACK. The aim of the game is to collect 5 eggs before they hatch and unleash a prehistoric bird at you. The game is played on a grid, with squares disappearing as you cross them. So as you don't get trapped, there is the availability of 'sliding tiles'. Grab the cash, watch for the traps and don't run out of time. Many screens. Not just an arcade, as there is a fair amount of strategy involved. Unashamed fun.

INFINITE LIVES

GHOST GOBBLER - POKE#16EE,16:POKE#10,127

INSECT INSANITY - DOKE#4B57,#EAEA

MR.WIMPY - POKE#4A3D,x (where x is less than 126)

PAINTER - POKE#8FC,#7E

THEM - POKE#24B9,#9E

MAZE RALLY - CLOAD the game until the maze map is shown. Stop the tape. Switch off youe Oric. Switch it back on, type CLOAD"" and re-start tape. When loaded type one of the following:

POKE5093,12 (for 6 lives)

POKE7784,200 (for infinite lives)

To run the game, ATMOS users should type CALL2693:CALL5856, whilst ORIC 1 survivors should go for just CALL5856

ORIC CHESS - How to beat the Oric in just 4 moves at level one:

E2-E4, D1-F3, F1-C4, F3-F7

=====

COMPOSER

Back in 1984, SECTOR 7 ran a competition based on their COMPOSER software.

Sharing the first prize of 100 pounds with a tune called POPCORN was a T. SHAW.

Come on TREVOR, own up, was that you!!

HELLS TEMPLE tip - if you want to avoid getting killed so easily by falling down trapdoors and deep holes - - - USE your spell power to help you detect whether treasure in a room is real or an illusion. The illusionary treasure is often the bait into a trap.

SNOWBALL tip - having trouble with the Nightingales killing you or can't get anywhere on the spaceship? - - - AS soon as you hear a noise, wait until it fades. If you say silly things, you will end up in a padded cell, which is another way to get out of the circle of mortuaries. But the best thing to do is to learn the colour code which can help you find your way about.

FANTASY QUEST - to get past the feather monster - make sure that you are carrying the FEATHER.

=====

THE TOP TEN

----- ADVENTURE PROGRAMS FROM THE AUTUMN ISSUE OF 'YOUR ORIC' -

1) THE HOBBIT, 2) COLOSSAL ADVENTURE, 3) L'AIGLE D'OR, 4) VIEW TO A KILL, 5) ADVENTURE QUEST, 6) LOST IN SPACE, 7) SNOWBALL, 8) FRANKLINS TOMB, 9) LORDS OF TIME, 10) HELLS TEMPLE.

WANTED

----- IF ANYONE HAS TRANSFERRED 'LORDS OF TIME' TO DISC, PLEASE CONTACT O.U.M

WANTED

----- WE WANT YOUR CURRENT LIST OF YOUR FAVOURITE 5 ARCADE AND 5 ADVENTURE GAMES. LET US FIND OUT IF THE NEWER TITLES HAVE LIVED UP TO OUR EXPECTATIONS. GET THOSE LISTS INTO O.U.M NOW!

I don't know about you folks but after about five minutes of playing a game I start to wonder if there is a cheat mode or a poke to stop the little man from getting killed, or make him jump to another section of the game. Well just when you were about to give up hope, here are a few tips on Zebbie and Xenon-1. The games are available from Dave Dick by mail-order, if anyone has any cheats of their own or has found any Pokes to type in please drop me a line, get your name in print! Or you can write to Dave at O.U.M, also if you have a request or are stuck on a particular game write, we may be able to help.

ZEBBIE is an excellent game, one of the best I have ever played. I managed to find these cheats myself, the first one gets rid of the border at the top of the screen so you can sit on top of the score, it also takes you straight to the bonus stage. The second Poke gives you infinite lives.

POKE 32000,6:CALL 8189 infinite lives.

POKE 8434,173:CALL 8189 no top border.

XENON-1 this great game from IJK was one the first games I bought for my Oric. I didn't have a cheat for this until I joined O.U.M so I, developed a pretty effective method over the years of blasting the Zorgons to space dust.

On the first screen with the birds just keep your finger on the fire button as soon or even before the game starts, this will get rid of three aliens almost straight away, as soon as the third bird is shot, move a tiny bit to the right and shoot the last bird but beware, you will have to be quick.

The second level with the "wierd looking round nasty things" is a bit harder. When you shoot them a piece falls off, although you can shoot it if you are fast enough. Press fire before the screen comes on so as to be firing straight away, the first beastly should move into the line of fire and be killed, as soon as the first is killed stop firing, move a touch to the right and fire again (keeping your finger on space to shoot the smaller bit). Then go all the way to the right and sneak out a bit every now and then, and shoot one that comes near, don't forget the smaller versions of alien that fall off.

By now if you havn't been killed it will be time to refuel, if the coloured bar at the top of the screen is on one of the first two bands (red or yellow) then you need to refuel. Press the UP ARROW key to go onto the refuel screen. Now press the RIGHT ARROW until you nearly hit the centre of the screen, then press the LEFT ARROW key to centre up and at the same time press SPACE and keeping your finger on SPACE until you reach a safe distance from the moving electricity bolts, which move across the screen. If you do it right you steer your ship into the gap at the base of the Mothership and refuel.

The Turnip stage is easy, just don't get hit by a meteor.

On the paratrooper level try to follow the big space ship while taking periodic pot shots at it, this way you can usually hit the baddies before they get a change to come down the screen, if you do miss one, let it get as close to you as possible (about 1½ cm) before you shoot it, you stand a better chance of hitting them if you can see the 'whites of their eyes'.

If you want to cheat on XENON-1 type :-

POKE 24617,173:RUN

(The game uses a basic program to execute all the machine code routines)

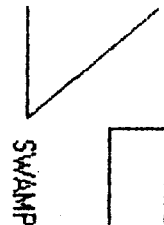
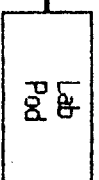
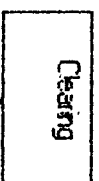
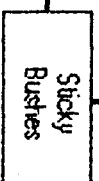
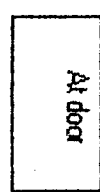
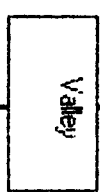
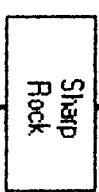
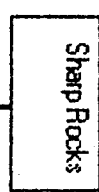
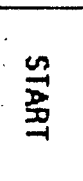
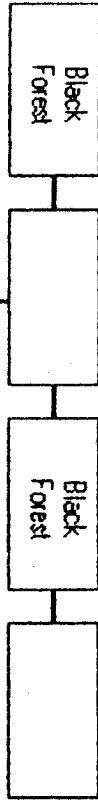
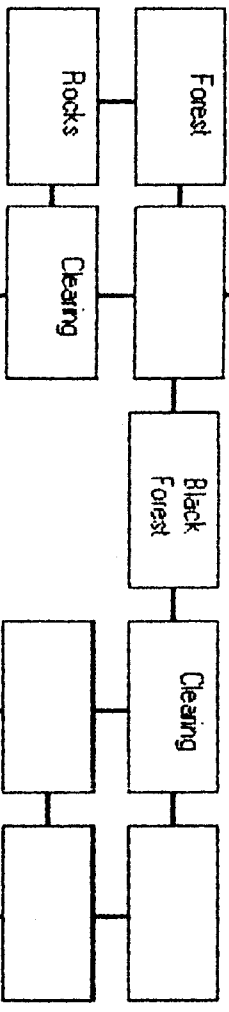
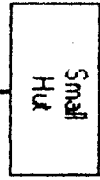
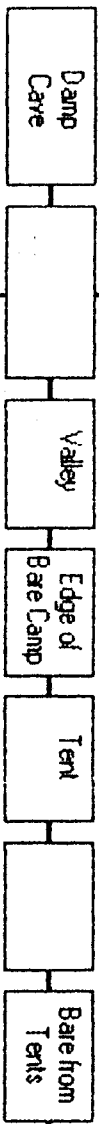
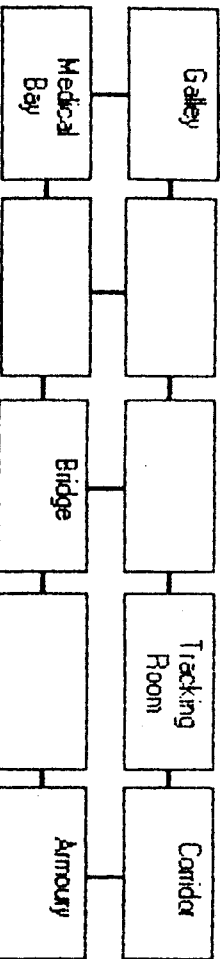
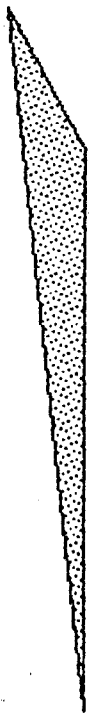
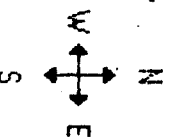
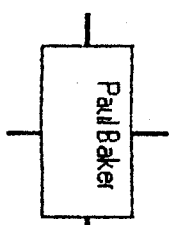
MORE TO FOLLOW

Captains Log

by Chymesoft

Not a complete map

0.U.M 1992



NOW TO GET RIGHT UP TO DATE;WE LOOK AT ISSUES 41 TO 55 OF 'OUM'. TO ORDER BACK ISSUES OR CERTAIN ARTICLES,JUST DROP ME A LINE AND YOU WILL BE QUOTED A PRICE - DAVE DICK

ISSUE 41 (January '91) - 16 pages - solution to LAST WARRIOR,review of TETRIX,BASIC program structures,ROM Dissassembly,ORIC data transfer/linking,and Printers..... ISSUE 42 - 16 pages - Questionnaire, Letters,News,ROM dissassembly,conclusion of interview with Geoff Phillips, German listing,adventure helpline..... ISSUE 43 - 18 pages - 2nd ORIC MEET report and photos,the charts, listing,RAMROM,more peeks at the ed's software,HIRES,letters, more on Oric data transfer linking , and cassette to disc transfer.and the First episode of Peter Braggs teach-in on Machine Code for the Atmos..... ISSUE 44 - 13 pages - News,Questionnaire results,RAMROM (The basic interpreter),Machine Code part 2 (Peter Bragg), Letters and News..... ISSUE 45 - 18 pages - completion of questionnaire results, listing for decimals,RAMROM, a look at Public Domain titles from CPD 52 -67,Disc access,games pages,Auto-stop listing, Machine Code (part 3),review of STRIP 21..... ISSUE 46 - 22 pages - readership of 83, Letters,listings,news,Ramrom,Packet Radio,Fractals, Machine Code part 4,project to replace the Eprom in the OLD version Cumana interface,Pokes,reviews of FLIGHT SIMULATOR/Mr. PRESIDENT (CEOSOFT) and the Contact list.....ISSUE 47 - 21 pages - letters,news,contact list, Ramrom looks at the step speed of a drive & more PD and more Dissassembly, The editor visits Paris for a French Oric meet,Eproms,Shareware,Disc file handling techniques,Graphics and machine code techniques,M/code part 5, Packet Radio part 2 ISSUE 48 - 21 pages - the Editor shows those dreaded knees on the cover,software charts,news,3rd Oric Meet report and pictures,Machine Code part 6,Disc file handling,Graphics & machine code techniques (Oric Enthusiasts - Allan Whitaker),Contact list , Packet radio, Letters and a games page..... ISSUE 49 - 24 pages - News, letters,65SC802,Dissassembly,Machine code part 7,Differences between basics,dos's,programs and data,graphics & machine code techniques,THE CASTLE, THREE listings,contact list,reflections - 97 readers..... ISSUE 50 (October 1991)- 20 pages - Wordspeed, 80 Track double sided,fitting a Printer,The STRATOS/TELESTRAT,Ganes hints and pokes,Machine code part 8, HEX coding form,Pick of the floppies,Reflections..... ISSUE 51 - 21 pages - Machine code part 9, RAMROM,PD,DELTA FOUR,BDDISK,ORIC ENTHUSIASTS,News, THE ORIC - a future!,Modem Matters, Krystal Worlds part one..... ISSUE 52 - 21 pages - SEDORIC Dos , WORDSPEED, RAMROM, THE LOST YEARS, POTATOMANIA listing,LPRINT,MUSED,letters,news,SUPER JEEP, Machine code part 10 and a FREEFLOW FLOW CHART..... ISSUE 53 (Jan '92) - 19 pages - Machine code part 11, PD charts, FIND A BASIC LINE,Interview with Paul Baker,Adventure Map design sheet,TREVOR and DAVE talk ORIC,ORIC ENTHUSIASTS price list,MANIC MINER screens.... ISSUE 54 - 23 pages - Machine code pt.12,RAMROM pt.35,VISIORIC and HUMIDITY listings,HI ! - I AM ADAM, letters,software charts,news,a look at back issues 1 to 20,the games pages,ORIC ENTHUSIASTS,and a couple of short listings..... ISSUE 55 - 24 pages - news,ads, a look at back issues 21 - 40,RAMROM,MACHINE CODE part 13,games pages,KRYSTAL WORLDS map,making a disc controller,VISIORIC and PISTON listings,letters,OUM finances, and ORIC ENTHUSIASTS articles continue.....

WELL, THAT'S IT , RIGHT SMACK UP TO DATE. AS I LOOK BACK, I AM AMAZED BY THE AMOUNT OF DIFFERENT TOPICS COVERED - VARIETY IS THE SPICE OF LIFE. AS I TYPE THIS WE HAVE A READERSHIP OF 101. BY THE TIME I'VE FINISHED THE MAG. IT COULD BE MORE - SEE ELSE WHERE.

AND SO FOREVER ONWARD !!!

INTRODUCTION FROM BRIAN KIDD (THE WELSH WIZZARD)

Once upon a month ago, many moons before our intrepid editor even hit the front page; he decided to list all known software ever produced for our beloved ORIC. Alas, for reasons still unknown, the list stopped after just one issue.

Now I have attempted the near impossible, and over the next few issues of OUM, will list (I hope!), all known BRITISH software houses and their software releases; leaving only Dave to complete the Mirage section, as this is his own baby.

It has been compiled from known info. and from info. gained by examining held copies of Oric Owner and Oric Computing. It is hoped that it will provide some interesting reading. How many titles do you recognise, or even own?

Perhaps Jon H could find some time to list all known French releases, Dave could do the same for the Scandinavian side, and Andre the German side?

As a side line of this I have tried to contact companies from the list. A few have replied (e.g. R & R) to let me know what happened to them, whilst others - not to mention those who are still trading (e.g.: OCEAN, PSS, DURELL, MICRODEAL etc) did not bother to reply, EVEN WHEN AN S.A.E was enclosed. This is a shame, not to mention - damn right rude.

Anyway, back to the list, which reads like this.....

HANG ON A MO !!! - IT IS I, DAVE THE EDITOR. Firstly - oh! hang on, sorry Brian I've missed a paragraph of your intro. This wine is good stuff. Anyway here is the missing bit... "The list comprises of 108 companies, and over 340 titles. No doubt there are some omissions, so please inform me or Dave of any, so that they can be added." RIGHT BRIAN, before we get stuck into this, I'd like to reply to some of the points that you have raised.

The reason I stopped printing my A to Z was lack of time to collate it all. However, I still have my notes and thus will input to your list. I think the list will probably double in size. I think we will deal with British software first and then deal with overseas titles. I am sure that readers would be interested in replies that you recieved from software houses.

O.K here we go - Software house and title as per Brian with additional notes in brackets from the Editor, who will also mention those missed as and when.

```
=====
A & F
----- - DEATH SATTELLITE, PAINTER ( a nice game that works with Altai
joy/iface), ZODIAC (same as Tansoft !)
ADD ON
----- - ACE IN THE HOLE, ARCAINE QUEST, HORROR ATOLL, LEOPARD LORD,
ROUNDSBY INCIDENT, TERROR FROM THE DEEP
APOCALYPSE
----- - DREADNOUGHT DISASTER
ARCADIA
-----
- INVADERS, MUSHROOM MANIA, PASTA BLASTA (works with Altai)
ARGUS SOFTWARE PRESS
-----
- STOCKMARKET, THE VALLEY (Adventure), WHITE BARROWS (Adventure)
ARTIC ASL AZTEC
-----
ASTEROIDS DUNGEONS OF INTRIGUE BATTLESHIPS, SQUIRM
BAJ SERVICES BAMBY
-----
BETABOMB T-COMP
BASIC CONCEPTS
----- - HAPPY LANDINGS, TEACH YOURSELF BASIC, TEACH YOURSELF
GRAPHICS
BRAINBOX BUGBYTE BYTELAND
-----
ELECTRONIC CALENDAR THE CASTLE BAR BLASTER, MEMORY MAP, ROUTE RIDER
CABLE
----- - QUEST FOR POWER, TEVROGS KINGDOM
CASCADE CCS
----- - CASSETTE 50 --- - AIRLINE, DALLAS (both strategy)
TO BE CONTINUED...
```



```

0 CLS:PAPER0:INK3:PRINT
1 PRINTCHR$(6);CHR$(17);CHR$(138)"          RADAR LANDING"
2 PRINTCHR$(138)"          RADAR LANDING"
3 PRINT:PRINT:PRINT:PRINTCHR$(134)"ASSEMBLE A BRIDGE FROM YOUR AIRCRAFT"
4 PRINTCHR$(134)"TO ENABLE ALBERT TO CROSS THE RIVER."
5 PRINT:PRINTCHR$(135)"PRESS SPACE BAR TO LAND THE BIRDERS."
6 PRINT:PRINT:PRINTCHR$(133);CHR$(140)"          PRESS ANY KEY TO START"
7 PRINT:PRINT:PRINT
10 FORT=46856TD46919:READU:POKET,U:NEXT
20 DATA63,33,18,12,63,0,0,0,12,30,12,30,45,12,18,33,62,62,62,0,55,55,55,0
30 DATA63,63,63,63,63,63,63,31,0,1,2,4,4,2,1,63,16,63,31,14,12,24,48
40 DATA48,1,62,33,0,0,0,0
50 DATA33,18,12,45,30,12,30,12
60 GOTO 2120
100 CLS
101 POKE48035,0
102 POKE618,10
103 S=0
105 PLOT2,0,"SCORE:0"
110 PAPER6:INK0

```

```

120 A$="dddddddddddddddddddddddddddddddddddd"
130 FORT=21T026:PLOT0,T,4:PLOT1,T,A$:NEXT
140 FORT=13T020:PLOT1,T,"cccc":PLOT35,T,"cccc":NEXT
150 X=37
160 W=32
170 PLOT33,13,"aa"
180 F=1
200 FORT=35T01STEP-1
210 PLOTT,6,"efg "
211 PLAY0,1,2,10
212 SOUND4,100,13
213 SOUND4,100,0
220 X=X-0.025
225 O=INT(X)
226 IF O=1 THEN 4000
230 IF SCRN(0,13)=32 THEN 2000
250 PLOTX,12,"b "
300 X$=KEY$
310 IF X$=" " AND F=1 THEN D=T+1:GOSUB1000
700 NEXT
710 PLOT1,6," "
715 F=1
720 GOTO200
999 END
1000 F=0
1040 FORE=8T013
1041 PLAY2,0,2,10
1042 MUSIC2,2,INT(E/2),9
1050 PLOTD,E,"a":PLOTD,E-1," "
1060 IF SCRN(D,E+1)=97 THEN 3000
1065 IF SCRN(D,E+1)=99 THEN PLOTD,E," ":RETURN
1800 NEXT
1810 IF D=WHENW=W-1:S=S+1:PLOT8,0,MID$(STR$(S),2):RETURN
1820 FORE=14T020
1830 PLOTD,E,"a":PLOTD,E-1," "
1840 NEXT
1850 PLOTD,20," "
1860 RETURN
2000 PLOT1,12," "

```

O.U.M

1st.

for

Tappers

```

2005 FORT=13T020
2010 PLOT0,T,"h":PLOT0,T-1," "
2020 WAIT10
2030 NEXT
2040 PLOT0,20," "
2050 MUSIC2,4,6,0:PLAY2,0,1,2000
2060 WAIT200
2070 CLS
2080 PRINT"OH DEAR ALBERT IS DEAD!"
2090 PRINT"BUT YOU SCORED ";S
2100 PRINT
2110 PRINT"PRESS RETURN TO PLAY AGAIN"
2120 IF S>HSTHENH$=S
2130 PRINT"THE HI-SCORE IS ";HS
2140 GETS$
2150 IF S$="" THEN 2140
2160 GOTO100
2999 END
3000 PLOTD,12," ":PLOTD,13," "
3010 FORT=14T020
3020 PLOTD,I,"a":PLOTD,I-1," "
3030 WAIT10
3040 NEXT
3050 PLOTD,20," "
3055 W=W+1
3060 RETURN
4000 A$="010203040506070809009080706050403020"
4010 FORT=1T036
4020 N=VAL(MID$(A$,T,1))
4030 MUSIC1,3,N+1,0:PLAY1,0,1,2000
4040 WAIT15
4060 NEXT
4070 CLS
4080 PRINT"YOU SAVED POOR ALBERT!"
4090 PRINT"VERY WELL DONE!"
4100 GOTO2110

```

```

100 REM
110 REM "ANAGRAM"
120 REM
130 CLS
140 PRINT:PRINT
150 DIM WD$(100),M$(12)
160 J=1
170 REM"READ IN DATA"
180 READ WD$(J)
190 IF WD$(J)="END" THEN 220
200 J=J+1
210 GOTO 180
220 J=J-1
230 REM"SELECT WORD"
240 R=INT(RND(1)*J)+1
250 A$=WD$(R)
260 FOR I=1 TO LEN(A$)
270 R=INT(RND(1)*12)+1
280 IF M$(R)<>" " THEN 270
290 M$(R)=MID$(A$,I,1)
300 NEXT I

310 FOR I=1 TO 12
320 IF M$(I)=" " THEN 340
330 PRINT M$(I)
340 NEXT I
350 PRINT:PRINT
360 INPUT "YOUR GUESS";G$
370 PRINT
380 IF G$=A$ THEN 430
390 IF G$="QUIT" THEN PRINT A$:PRINT:GOTO 450
400 IF G$="STOP" THEN END
410 PRINT "TRY AGAIN!":PRINT
420 GOTO 360
430 PRINT"CORRECT!"
440 NC=NC+1
450 A$=""
460 FOR I=1 TO 12
470 M$(I)=" "
480 NEXT I
490 PRINT
500 GOTO 240
520 DATA POND,WOOD,MOUSE,TIGER,HOUSE,APPLE,TRAIN,WORD,PARTY,CHIMNEY
530 DATA DIGIT,IDEA,ANAGRAM,ORBIT,PENNY,PEAR,BINARY,PUPIL
540 DATA VIDEO,RECORD,MODULAR,LEAF,BOOK,FILM,CRICKET,PICTURE
550 DATA END

100 REM
105 REM ARTIST
110 REM
120 CLS:PRINT:PRINT
121 REM PRINT INSTRUCTIONS
122 PRINT"USE KEYS 1 TO 7 TO CHANGE COLOUR":PRINT
124 PRINT"ARROW KEYS AND UIJK TO DRAW":PRINT
126 PRINT"B KEY TO TURN OFF":PRINT
128 PRINT"D KEY TO TURN ON":PRINT
130 PRINT"S KEY TO STOP":PRINT
132 PRINT"PRESS SPACEBAR":GET N$
134 X=120:Y=100
136 HIRES
138 REM LOOK AT KEYBOARD
140 GET K$
152 IF K$="B" THEN B=1
154 IF K$="D" THEN B=0
156 IF K$="S" THEN TEXT:END
160 IF K$=CHR$(11) AND Y>2 THEN Y=Y-1
170 IF K$=CHR$(10) AND Y<197 THEN Y=Y+1
180 IF K$=CHR$(8) AND X>2 THEN X=X-1
190 IF K$=CHR$(9) AND X<237 THEN X=X+1
192 IF K$="U" AND X>2 AND Y>2 THEN X=X-1:Y=Y-1
194 IF K$="I" AND X<237 AND Y>2 THEN X=X+1:Y=Y-1
196 IF K$="J" AND X>2 AND Y<197 THEN X=X-1:Y=Y+1
198 IF K$="K" AND X<237 AND Y<197 THEN X=X+1:Y=Y+1
199 REM CHANGE COLOUR
200 IF ASC(K$)>47 AND ASC(K$)<56 THEN INK VAL(K$)
210 IF B=1 THEN 230

220 CURSET X,Y,1
224 GOTO 140
230 CURSET X,Y,0
234 GOSUB 500
240 GOTO 140
500 DRAW 1,1,1
505 CURMOV -1,-1,3
510 DRAW 1,1,0
515 CURMOV -1,-1,3
520 RETURN

```

```

100 REM
105 REM KALEIDOSCOPE
110 REM
120 CLS
125 PRINT CHR$(17)
130 R1=INT(RND(1)*38)
140 R2=INT(RND(1)*25)+1

```

```
\ 65SC802 Assembler for the F83 - System      written 1991 by B.Gröne
\ This is an assembler for the ORIC F83 Forth system. It supports
\ the 6502, 65SC02 AND 65SC802 processors. It consists of a 65C02
\ assembler written by Peter Glasmacher published in the mag.
\ C't in January 1984. I have extended it with the 16 Bit routines
\ for the 65802 and with the ORG / ASS words.
```

```
ONLY DEFINITIONS FORTH ALSO VOCABULARY (ASSEMBLER)
' (ASSEMBLER) IS ASSEMBLER ( if assembler a deferred voc. )
ALSO ASSEMBLER ALSO DEFINITIONS HEX
DEFER PC ( program counter is deferred for all purposes )
' DP IS PC ( PC is here for coded words )
VARIABLE OFFST 0 OFFST ! : PCC PC @ OFFST @ + ;
: ALLOT PC +! ; : C, PCC C! 1 ALLOT ; : , PCC ! 2 ALLOT ;
: , " ASCII " PARSE TUCK PCC SWAP MOVE ALLOT ;
```

```
\ Backward and forward branches in absolute or relative mode :
: $BE DUP -80 7F BETWEEN NOT ABORT" Zu großer Sprung " ;
: <MARK PC @ ; : >MARK PC @ DUP ; : +OFFSET OFFST @ + ;
: <RESOLVE ; ( if you want more readable sources ... )
: >RESOLVE DUP +OFFSET C@ DUP DUP 1F AND 10 = SWAP 80 = OR IF
  DROP PC @ OVER 2+ - $BE SWAP 1+ +OFFSET C! ELSE 82 = IF PC @
  OVER 3 + - SWAP 1+ ELSE 1+ PC @ SWAP THEN +OFFSET ! THEN ;
```

```
\ Index, Mode (adressierungsarten) , ?WORD
VARIABLE INDEX -2 ALLOT \ Offsettables for Adressing modes
0909 , 1505 , 0115 , 1211 , 1303 , 1D00 , 8019 , 8080 ,
0080 , 1404 , 0114 , 8080 , 8080 , 1C0C , 3C1C , 2C80 ,
6978 , 1404 , 8080 , 8080 , 8080 , 1C0C , 8080 , 8080 ,
8038 , 1404 , 8080 , 8080 , 8080 , 3E3C , 8080 , 8080 ,
```

```
VARIABLE MODE \ contains current adressing mode (Index)
: .A 0 MODE ! ; : # 1 MODE ! ; : MEM 2 MODE ! ;
: ,X 3 MODE ! ; : ,Y 4 MODE ! ; : X) 5 MODE ! ;
: )Y 6 MODE ! ; : Z) 7 MODE ! ; : S 8 MODE ! ;
: S)Y 9 MODE ! ; : ) F MODE ! ; MEM
: ENDIF [COMPILE] THEN ; IMMEDIATE
VARIABLE ORIGIN
```

```
\ Upmode, 1-Byte-Opcodes
```

```
: UPMODE IF \ (adr f -- adr f)
  MODE @ DUP 8 AND 0= SWAP E AND AND \ Tests if adr.mode
  IF 8 MODE +! \ (in Mode) allowed for command
  ENDIF \
  ENDIF \
  1 MODE @ OF AND ?DUP
  IF 0
    DO DUP + LOOP
  ENDIF
  OVER 2+ @ AND 0= ;
```

```
: S/C CREATE C, DOES> C@ C, MEM ;
```

```
\ 1-Byte-commands
```

```
00 S/C BRK, 18 S/C CLC, 08 S/C CLD, 58 S/C CLI, B8 S/C CLV,
CA S/C DEX, 88 S/C DEY, E8 S/C INX, C8 S/C INY, EA S/C NOP,
48 S/C PHA, 08 S/C PHP, 68 S/C PLA, 28 S/C PLP, 40 S/C RTI,
60 S/C RTS, 38 S/C SEC, F8 S/C SED, 78 S/C SEI,
AA S/C TAX, A8 S/C TAY, BA S/C TSX, 8A S/C TXA, 9A S/C TXS,
98 S/C TYA,
```

```

\ New 65C02 - Opcodes
5A S/C PHY, DA S/C PHX, 7A S/C PLY, FA S/C PLX,
\ New 65SC802 - Opcodes
8B S/C PHB, 0B S/C PHD, 4B S/C PHK, AB S/C PLB, 2B S/C PLD,
6B S/C RTL, 5B S/C TCD, 1B S/C TCS, 7B S/C TDC, 3B S/C TSC,
9B S/C TXY, BB S/C TYX, CB S/C WAI, EB S/C XBA, FB S/C XCE,

VARIABLE LONGA VARIABLE LONGI \ Akk. / X&Y 8 od. 16 Bit ?
O DUP LONGA ! LONGI !
VARIABLE COMCODE

: ?WORD OVER FF00 AND DUP IF OVER DUP 1+ C@ NOT SWAP
  C@ OF AND OR IF LONGA ELSE LONGI ENDIF @
  MODE @ 1 = AND NOT SWAP DROP ENDIF ;

\ M/C (Multi-Mode Addressing)
: M/C CREATE , , DOES>
  DUP 1+ C@ ?DUP
    IF 0 DO 10 MODE +! LOOP
  ENDIF ?WORD UPMODE UPMODE
    ABORT" wrong addressing mode"
  C@ MODE C@ INDEX + C@ + DUP 100 >
    IF 100 - DUP 5A =
      IF DROP 1A ENDIF
    ELSE DUP 50 = MODE C@ OF AND OD = AND
      IF DROP FC ENDIF
    ENDIF DUP COMCODE ! C, MODE @ OF AND DUP
      IF DUP 9 > SWAP 1 = IF COMCODE @ OF AND 9 =
        IF LONGA ELSE LONGI ENDIF @ ELSE 0 ENDIF
      OR IF , ELSE C, ENDIF
    ELSE DROP ENDIF MEM ;

\ commands
2400 0114 M/C JSR, 0406 01E0 M/C CPX, 0406 01C0 M/C CPY,
040C 0180 M/C STY, 0414 0182 M/C STX, 0C0E 01A0 M/C LDY,
0C0D 0001 M/C ASL, 0C0D 0041 M/C LSR, 0C0D 0021 M/C ROL,
0C0D 0061 M/C ROR, 1416 01A2 M/C LDX, 1FEC 0080 M/C STA,
1FEE 0060 M/C ADC, 1FEE 0020 M/C AND, 1FEE 00C0 M/C CMP,
1FEE 0040 M/C EOR, 1FEE 00A0 M/C LDA, 1FEE 0000 M/C ORA,
1FEE 00E0 M/C SBC, A400 0140 M/C JMP, 0C0D 02C2 M/C DEC,
0C0D 02E2 M/C INC, 0C0E 0220 M/C BIT, 0C0C 0360 M/C STZ,
0404 00FF M/C TSB, 0404 000F M/C TRB, \ New 65C02 commands

\ 65SC802 - commands:
0004 01FE M/C COP, 0400 0148 M/C MVN, 0400 0138 M/C MVP,
0400 01E8 M/C PEA, 0004 01D0 M/C PEI, 0004 01BE M/C REP,
0004 01CE M/C SEP,

\ 1.Word:addressing modes, 2. MSB:Indextable #, LSB:Offset

\ Relative Addressing : Branches, PER
: BRAN CREATE C, DOES> C@ DUP
  C, SWAP MEM PC @ 1+ -
  SWAP DUP 62 = SWAP 82 = OR
  IF ,
  ELSE $BE C,
  ENDIF ;

10 BRAN BPL, 30 BRAN BMI, 50 BRAN BVC, 70 BRAN BVS,
90 BRAN BCC, 80 BRAN BCS, 00 BRAN BNE, F0 BRAN BEQ,
80 BRAN BRA, 82 BRAN BRL, 62 BRAN PER,

: EQU CONSTANT ;
B0 EQU CS F0 EQU 0= 30 EQU 0< 70 EQU VS CS EQU >=
0= EQU =
: NOT 20 XOR ;

```

Practically all modern computers are provided with both parallel and serial ports. These are considered to be the minimum requirements. The Oric-1 and Atmos were both endowed with a reasonable number of ports for their prices at the time of their production. But the one port they were lacking was a serial port such as the RS232.

Oric put this right when they produced the Oric Modem (which had been promised in the earliest advertisements but did not materialise until just before they went bust); they produced a sealed modem interface which connects between the Oric expansion buss and the Oric Modem. Unfortunately no technical information was provided for the interface, not even a pin-out for the 7-pin DIN connector at the modem end! If Oric had survived then this information may have been published through Tansoft's 'Oric Owner'. But since Oric intended that the complete modem package be purchased (which included Prestel software with User-to-User software supposedly to follow), I assume they didn't want to encourage the use of the interface with other modems nor other devices anyway.

However, the technical details have been known for some time (at least by myself) since John Rushton and I dissected one of the interfaces and I worked out the circuit diagram in order that we could produce the Oricomms software. Dissecting the potted modem interface is not a job I would recommend others try - since the interface is almost certainly to be rendered useless afterwards. For those who are interested, the circuit diagram for the Oric serial (modem) interface is available from either Dave (OUM) or myself. For most people's purposes though the most important information is about how to use and program the interface. Hence that is the subject I intend to deal with in this and subsequent articles. I hope that after reading these articles, which will include example programs in BASIC, more Oric users will venture to experiment with the serial port. Perhaps someone will even be keen enough to write a new communications program?

So what use is a serial interface? Well whereas a parallel interface such as the Oric's Centronics port is primarily of use as a printer port (or perhaps a limited user port) the serial interface can be used to communicate with many devices. Hands up all those who would like to be able to connect a mouse to the Oric, or a serial printer, or transfer data directly to another computer, or connect a graphics pad! Whilst the Oric Modem Interface was obviously intended primarily for use with the Oric Modem there is no reason whatsoever why it should not be used for any of the purposes suggested above. It is possible to communicate directly with other computers and exchange files using the serial port. For example how about transferring BBC Micro 6502 machine code files, or ASCII text files from any other computer with a serial port (e.g. Archimedes, Atari or even an IBM PC). These are just some of the uses that you may wish to put the serial port to. However to achieve any of this you have to know how to use (i.e. connect and program) the interface. I shall cover programming later but first I have to explain a little about serial interfaces and the Oric interface in particular.

A serial interface is a port which sends or receives data one bit at a time in sequence. The speed at which data is transferred is called the 'baud rate'. The Oric interface can be programmed to operate between 50 and 19200 baud (approx. 1900 characters per second). However, the Oric itself will not accurately receive data at such fast rates unless the program is written in very efficient machine code (a BASIC program will just about cope with 300 baud). The most commonly used baud rates for telecommunications over a standard 'phone line are 300, 1200, 2400 and 4800 baud; the main restriction here has always been the poor quality of telephone lines. 9600 baud is now frequently used on private or leased lines. However, for other purposes (e.g. direct data transfer between computers where a modem is not needed) the rate used can be much faster, even 19200 baud (Incidentally, the MIDI interface which is a type of serial interface operates at 31250 baud!).

RS232C (or V24) is a standard for serial interfaces. A full implementation uses a 25-way D-type connector. Very few computers, the Oric included, offer a full implementation though. All that is actually needed for a serial interface link are 3 pins, one for transmit, one for receive and one ground. However, most computers also provide some 'handshaking' connections so that the data flow can be better controlled. These do not have to be used however, and can usually be left unconnected without

major problems.

22

In the Oric interface there are seven pins. Details of these are given in the Oricomms manual so I shall not go into detail here. (Apart from the essential three mentioned above, also included is a handshaking connection, RTS, and a line status connection, DCD. DTR is also there but cannot be used normally without first being inverted. In addition a non-RS232 connection is provided called RxC which enables the use of other baud rates). All that you need to know for now are that pin 2 is transmit data (TxD), pin 5 is receive data (RxD) and pin 3 is ground (GND) since these are all you need for simple data transfer procedures. By connecting pins 2 and 5 together you can test any simple programs that you might write - including those provided in the future articles!

Okay, so let's assume that you want to connect your Oric to another computer's serial port using the simplest method. You would connect the Oric's TxD to the other machine's receive pin, the Oric's RxD pin to the other's transmit pin, and GND to GND. All you need then is some software to drive the port on each computer. This configuration should also work with a printer, a mouse, a graphics pad or other input or output device, but the handshaking pins may also be needed depending upon the peripheral's software.

Alternatively, if you want to connect the Oric to a different modem then you would connect the Oric's TxD pin to the device's transmit pin, the Oric's RxD pin to the device's receive pin, and again GND to GND.

Sounds simple enough doesn't it? Those really are the only two alternatives with the simple three wire link. If a slightly more sophisticated link is required then you might also want to connect the RTS (computer request-to-send data) pin and the DCD (data-carrier detected) pin to the RTS and DCD pins respectively on the other device. This will then allow handshaking to take place which will control the direction of the data flow. If the simple 3-wire link is used then any data flow control has to be provided by the use of 'flow control characters' sent with the data. Obviously this can make the software unnecessarily complicated.

Oh well, that's it for now, I hope this will generate some interest in making more use of the Oric's serial port. Next time I shall begin explaining how to program the interface.

Trevor Shaw.

MODEMS WITH OR WITHOUT RS232's

AVAILABLE DIRECT FROM: O.U.M.

WRITE WITH REQUIREMENTS.

"ORICOMMS" AVAILABLE FROM 'ORIC ENTHUSIASTS'.

ORIC Enthusiasts (OUM 56)

INTRODUCTION

This month sees two more pages, each, on Geoff Phillips' book and how to access disc files from BASIC. In my haste to get last month's article to Dave on time, a few grammatical errors crept into my introduction regarding the development of the ORIC. So I will just go the point one more time.

Although I did not receive any response to an earlier request for continued development of the ORIC, I still think that a few enthusiasts would like to see some changes. It would be interesting to see what could be done that would be of interest to everyone and could form a new standard for the ORIC's hardware and software. As I stated last month I intend to write a specification for what the developments should and could contain but to make it worthwhile I need some feedback of what level of interest there is in such a project. So far I have received one letter in response to last month's article which I thank Tim Colgate for, but in order to proceed there to be more visible interest. I would like to circulate this specification to interested parties in order to gain an overall consensus of what the development should contain, but I am particular looking for experienced users of the ORIC who would be interested in sharing the development work. Write to me if you are interested (SAE would be appreciated if a reply is required).

BOOKS

Continuing with the series on books available from my price list. The next title is "Games For Your ORIC", primarily by Peter Shaw but it also has a section written by Tim Hartnell who was the series editor. Computer enthusiasts who were around in the early mania of the 80s computer boon will remember Tim as a prolific writer on a wide range of home computers.

This is yet another book first published in 1983. In some respects it is a schizophrenic sort of a book in that it addresses topics other than games. Certainly, it is aimed at the budding enthusiast, which is its common theme, but it is split into four sections. The first and largest section covers over 20 listings for games for the ORIC; the second section, written by Tim, deals with advice on how to write better programs; the third section covers a fine glossary of computing terms and is quite comprehensive. The last section is a bibliography written by Tim which is now showing its age but it does provide an insight into the influences at the beginning of the last decade. It is also interesting to note that Tim had met quite a number of the authors either at a PCW show or in a restaurant. I personally found this section to be an enjoyable read.

The book contains 126 pages with games listings from page 9 to 97, section 2 occupies pages 99 to 106, the Glossary appears on pages 107 to 118 and finally the Bibliography concludes the book.

The games listings are written entirely in BASIC and make use of redefined character sets where required. The titles are as follows and each one includes a short description. LEAKY ROOF (collect as much water as possible in your rusty bucket); JIGWORD (a version of Hangman); JUGWORD (guess the anagram); LIGHT RELIEF - DESIGNER (an aid to drawing shapes on the screen) & BUTTERFLY PLOTTING (a graphics demo routine); ROCK-SCISSORS-PAPER (a computer version of that age old hand game designed to cripple you for life - let your ORIC wrap your knuckles); 3D DRIVER (similar to the game included with the ATMOS demonstration tape); LIFE CHANGE UNITS (determine how much the changes in your life are contributing to your stress level); NIM (a counter-based strategy game); BOMB RUN (as the book says, one of the oldest micro games where to have to clear the buildings before you can land your plane safely); THE RIDDLE OF THE BLACK CHATEAU (a 25 kbyte adventure game); MAGIC SQUARE (creates Magic Squares from an integer number between 1 and 9); TRAPPER BIKE (the proverbial TRON game); QUACK ATTACK (the proverbial Duckshoot); M4 (another racing game based on driving along the M4 during rush hour - surely this must be a static screen!); SAVE THE SAUSAGES (move around the screen, collecting as many sausages as possible in the time limit); NOIRE (as in the ATMOS book and demo); PAC-MUNCHER (the proverbial Pacman); DR. WATSON (guess the computer generated number); SQUARE (another graphics demo program) and BOX-BOUNCE (a game based upon Ping-Pong).

All in all, a variety of games to while away the hours but none at a standard that will take your breath away, but then that is not the intent. The book is designed to get the user to see how games are written in order to develop the masterpieces of their own. There is the usual proviso about the applicability of this book to the V1.1 ROM but since all the programs are written in BASIC it is an easy task to convert the PLOT commands. Remember to add 1 to the 'x' position in the PLOT command so that the PAPER attributes are not overwritten unintentionally. The book is available at £1.50, inclusive of postage and packaging, (original price was £2.95).

ORIC ATMOS and ORIC-1 GRAPHICS & MACHINE CODE TECHNIQUESChapter 3 - BASIC (continued)

copyright of Geoff Phillips

3.7 Patching into BASIC (continued)

#24A(#230) is the address of the 'slow' interrupt routine. Control is passed to here at the end of the fast interrupt routine. Although 3 bytes are reserved here, there is only the single-byte instruction instruction RTI present normally.

#247(#22B) contains the jump vector for the NMI (Non-Maskable Interrupt) routine, which on the ORIC connects to the 'Reset button'.

On V1.1 ROM only, there are a few extra jump vectors located in page 2 which are concerned with input/output :

- 1 #238 links to the screen output routine used by BASIC commands like PRINT.
- 2 #23B jumps to the sub-routine which finds which key was last pressed.
- 3 #23E jumps to the printer output sub-routine.
- 4 #241 contains a jump to the sub-routine that prints messages on the top line of the screen. Changing this jump could be useful if you want to stop messages like 'Loading' from showing.

By far the most useful of these patches is the slow interrupt jump which allows you to make the maximum use of the system's interrupts.

3.8 Interrupts The purpose of an interrupt is to stop a program temporarily and to enter a special sub-routine in order to handle a priority condition. An interrupt on a computer will often be caused by a peripheral (such as a card-reader) announcing that it has data to transfer. The ORIC takes its interrupt line from the 6522 VIA device which is capable of causing an interrupt for a variety of reasons. Unless the ORIC is loading or saving to the cassette port, the 6522 is set up to create an interrupt at exact intervals of 10,000 machine cycles - or every 10 milliseconds. In other words, the machine is interrupted every one-hundredth of a second. (You should be warned that some BASIC instructions may cause an interrupt to be missed - e.g. PRINT.) The length of time between interrupts is stored is stored on the 6522's timer-1 latch at #306,7. By altering #306,7 you affect:

- 1 The repeat rate on the keyboard.
- 2 The flash rate of the cursor (but not the automatic flash of the VDU chip).
- 3 The speed of the WAIT command.
- 4 The speed of processing is inversely affected. This happens because the interrupts 'steal' time from the processor; the more time spent in interrupt handling, the less is available for the main task.

When an interrupt occurs, and providing that the 'fast interrupt' jump vector has not be altered, the following events take place:

- 1 The three software timers are decremented by one. These are 16-bit counters located in page 2 of the memory and will be discussed in section 3.9.
- 2 If the first timer has reached zero, after counting down from 3, the keyboard is scanned in a search for any key press.
- 3 If the second timer has reached zero, counting down from 25, the cursor is flashed on or off.

Note that the timers being discussed are merely counters in RAM, and should not be confused with the timer-1 and timer-2 of the 6522. When an interrupt occurs, the 6502 jumps to the address given by locations #FFFE and #FFFF. As was discussed in section 3.7, the vector to this address is in page 2 of RAM, and the jump into ROM can be modified for one's own requirements. If the fast interrupt routine does jump into ROM the last operation is to jump back to the slow interrupt location in page 2, containing the RTI instruction. You would use the fast interrupt patch if you wanted to add some processing before the keyboard is scanned. The slow interrupt link allows you to add some processing after the keyboard has been scanned. If you intend to modify the interrupt routines, remember:

- 1 Save all the registers that you use, and restore them before you finish.

- 2 Save any locations that might be in use by the system. For instance, if your interrupt routine calls the SOUND command you will need to save locations #2E0 to #2EF and #204 (#204 is used when checking your SOUND parameters).

At the end of your interrupt routine, you will usually either execute the RTI instruction if all interrupt processing is complete, or jump back into the normal ROM interrupt routine (to read the keyboard, etc.).

Writing interrupt routines is much more difficult than writing a normal sub-routine. For one thing, testing can frequently crash the whole machine, and often a fault will not show up for a long time. Two important points are:

- 1 Remember to save any location that could be used by both your interrupt routine and the main program.
- 2 Do not assume the state of any of the processor flags. Be especially wary of the decimal flag - use CLD or SED if you are doing any addition or subtraction.

Several programs in this book modify the interrupt patches, and by understanding how these work you will be able to create your own routines.

NON-MASKABLE INTERRUPT The Reset button on the ORIC does not, in fact, connect to the RESET line of the 6522. Instead, it activates the Non-Maskable Interrupt (NMI) line of the 6502. Whereas a normal interrupt can be disabled, the NMI causes an unconditional jump to the address contained locations #FFFA, #FFFB. On the ORIC, this is a jump instruction in page 2 of memory which on the ORIC is normally leads to a 'warm-start' routine in ROM. This sets up the 6522, clears the screen, initialises the character sets, and returns to command mode in BASIC. When writing machine code programs it is customary to alter the appropriate address in page 2 (see section 3.7) so that pressing the Reset button restarts the machine code program. The button can be disabled by typing POKE DEEK(#FFFA),64. The 'BRK' instruction causes an interrupt, but sets the BRK flag in the 6502 processor. It is used by some machine code monitors as a terminating command - just as RTS is used to return to BASIC after a CALL instruction. Use RTS instead of BRK if your machine code monitor expects it.

3.9 Software timers This subject was mentioned when interrupts were discussed. There are three 16-bit counters stored in RAM, maintained by the interrupt routine. The first two timers are in permanent use on the ORIC: the first counts three interrupt cycles (normally 30 milliseconds) before each keyboard read while the second counts 25 interrupts (250 milliseconds) before flashing the cursor on or off. The third software timer is only used occasionally by the system - for WAIT, TEXT and (in version 1.0 only) when using the HIRES command. This means that it is available for use in your own program. With very little trouble, you can time events to one-hundredth of a second. Remember that the software timers will only be decremented when interrupts have been enabled. Each of the three timers occupies 2 bytes, in the normal tradition of the low byte first, starting at #272. Therefore, the all important third timer is located at #276,7. The WAIT command can be simulated by a simple use of DOKE and DEEK into location #276, but with the advantage that the program can do further work while the third timer is counting. Although it is a simple matter to set up this timer, there are a number of sub-routines in ROM which handle each of the timers. The A, X and Y registers need to be set up as follows:

A - set to the timer number minus one. For instance, the 3rd timer requires a value of 2.

Y - set the Y register to the low part of the timer value.

X - set the X register to the high part of the timer value.

Here is a table of calls which relate to the software timers:

	V1.1	V1.0
Start 6522 clocks	#EDE0	#ECC7
Stop 6522 clocks	#EE1A	#ED01
Update timers, etc	#EE34	#ED1B
Clear all timers	#EE8C	#ED70
Read a timer into X,Y	#EE9D	#ED81
Write X,Y into timer	#EEAB	#ED8F
Wait for time X,Y	#EEC9	#EDAD

Next month -- continuing Chapter 3 and the start of 3.10

DISC FILE HANDLING TECHNIQUES USING BASICUNSORTED SEQUENTIAL FILES (continued)

Let me first apologise for the poorer quality of last month's article. In the rush to meet the deadline, I forgot to box in my diagrams. However, they weren't too complex so I hope that you could make sense of them. This month sees the completion of the CREATE PFILE program and the access of it for reading and writing.

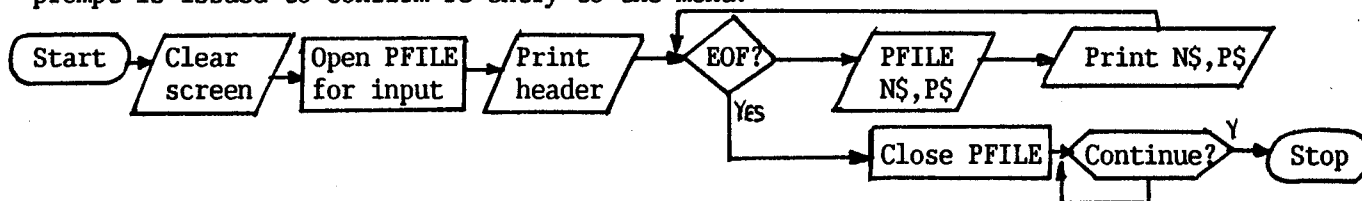
The program lines to complete the creation of PFILE are :-

```

RANDOS      1090 !PUT1,N$;!PUT1,P$
SEADORIC    1090 PUT1,N$,P$
             1100 PRINT:PRINT
             1110 GOTO 1030
             1120 GOSUB 1300:      REM CLOSE PFILE
             1130 RETURN
             1200 REM ----- OPEN PFILE FOR OUTPUT -----
MICROSOFT    1210 OPEN "O",#1,F$
RANDOS      1210 !OPEN 1,F$,W
SEADORIC    1210 OPEN S,F$,1
             1220 RETURN
             1300 REM ----- CLOSE PFILE -----
MICROSOFT    1310 CLOSE #1
RANDOS      1310 !CLOSE 1
SEADORIC    1310 CLOSE 1
             1320 RETURN

```

Read and Printing PFILE. Again, I shall use a flow diagram to illustrate the design of the routine. So as you can see, after clearing the screen the file PFILE is opened for reading. The header is printed and then the file is read until the end is reached. At each record the fields are printed out. On reaching the end of file, PFILE is closed and a prompt is issued to confirm re-entry to the menu.



The program lines are :-

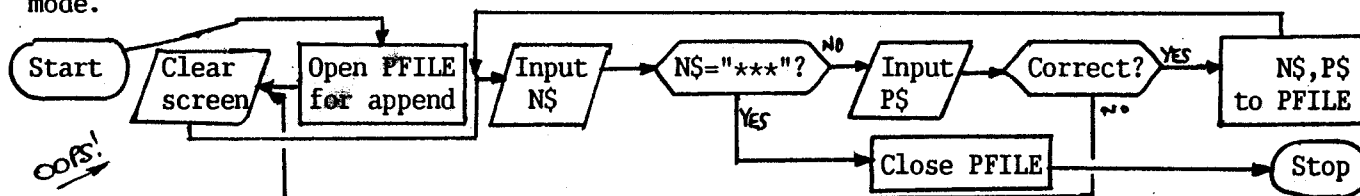
```

2000 REM ----- READ AND PRINT PFILE -----
2010 CLS
2020 GOSUB 2200:      REM OPEN PFILE FOR INPUT
2030 PRINT "LIST OF PHONE NUMBERS"
2040 PRINT "-----"
MICROSOFT    2050 IF EOF(1) THEN 2090
RANDOS      2050 IF PEEK(#4FF)=29 THEN 2090
SEADORIC    2050 IF (-&(1)) THEN 2090
MICROSOFT    2060 INPUT#1,N$,P$
RANDOS      2060 !GET 1,N$;!GET 1,P$
SEADORIC    2060 TAKE 1,N$,P$
2070 PRINT N$;TAB(20);P$
2080 GOTO 2050
2090 GOSUB 1300
2100 PRINT:PRINT
2110 INPUT "CONTINUE (Y/N)";A$
2120 IF A$<>"Y" THEN 2110
2130 RETURN
2200 REM ----- OPEN PFILE FOR INPUT -----

```

MICROSOFT 2210 OPEN "I",#1,F\$
 RANDOS 2210 !OPEN 1,F\$,R
 SEDORIC 2210 OPEN S,F\$,1
 2220 RETURN

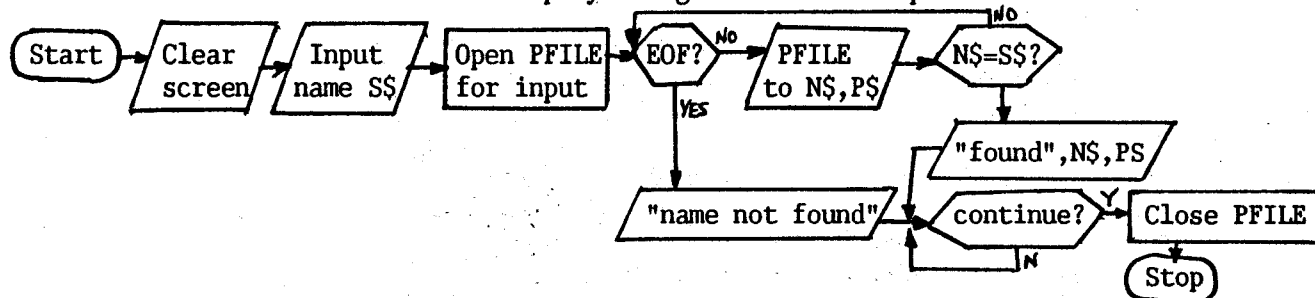
Adding new records. The flow diagram is similar to that for reading and printing PFILE except that the file is opened for appending a record to the end of the file, i.e. APPEND mode.



```

3000 REM ----- ADD NEW NAMES AND NUMBERS -----
3010 GOSUB 3200: REM OPEN PFILE FOR APPEND
3020 CLS
3030 PRINT "TYPE NAME AND PHONE NUMBER (***=STOP)":PRINT
3040 INPUT "NAME";N$
3050 IF N$="***" THEN 3120
3060 INPUT "PHONE NUMBER";P$
3070 INPUT "CORRECT (Y/N)";A$
3080 IF A$<>"Y" THEN 3020
MICROSOFT 3090 PRINT#1,N$:PRINT#1,P$
RANDOS 3090 !PUT 1,N$;!PUT 1,P$
SEDORIC 3090 PUT 1,N$,P$
3100 PRINT:PRINT
3110 GOTO 3030
3120 GOSUB 1300
3130 RETURN
3200 REM ----- OPEN PFILE FOR APPEND -----
MICROSOFT 3210 OPEN "A",#1,F$
RANDOS 3210 !OPEN 1,F$,A
SEDORIC 3210 OPEN S,F$,1:APPEND 1
3220 RETURN
  
```

Finding a record. First lets provide the flow diagram for the search method, where the key is assigned to S\$. All records from PFILE are read to see if there is a match and if none is found then the message NAME NOT FOUND will appear when the end of file is reached. If the name is found then it is displayed together with the phone number.



```

4000 REM ----- SEARCH FOR NAME -----
4010 CLS
4020 INPUT "NAME TO SEARCH FOR";S$
4030 GOSUB 2200: REM OPEN PFILE FOR INPUT
MICROSOFT 4040 IF EOF(1) THEN 4090
RANDOS 4040 IF PEEK(#4FF)=29 THEN 4090
SEDORIC 4040 IF (-&(1)) THEN 4090
  
```