

ORIC

**USER
MONTHLY**

with Alternative Micros

Number **89**

January 1995

*Keeping the
Oric alive*



THE EDITORIAL

HELLO,

AND A HAPPY NEW YEAR TO ONE AND ALL.

INTEREST IN THE ORIC IS STILL VERY STRONG, AND I ENVISAGE US STILL GOING STRONG IN 1996.

I WILL SOON BE SETTING A DATE FOR THE 1995 AYLESBURY MEET. IT WILL PROBABLY BE IN JULY.

THANK YOU ALL FOR YOUR CONTINUED SUPPORT.

NOW TO THIS ISSUE:-

THE INDEX

Page 1 THE COVER - courtesy of Jon Haworth.

Page 2 THE EDITORIAL Page 3 NEWS

Pages 4/5 .. MACHINE CODE FOR THE ORIC ATMOS (Pt.42) - PETER BRAGG

Page 6 LOOKING AT CLASSIC GAMES - in part 3, Arnt Erik Isaksen looks at IJK Software.

Page 7 MARSHALL'S MUSIC - Steve gets to part 15.

Pages 8/9 .. READERS LETTERS - a look at some of the enormous postbag.

Pages 10/11/12/13 - RAMBLING IN THE ROM - in part 66 Jon is back to Rambling rather than Hobbling.

Page 14 ... BITS 'n' BOBS - a mixed bag!

Page 15 BRIAN'S PAGE - The Welsh Wizzard has another poser.

Well - a smaller issue than I wanted to publish, but if I didn't stop now - IT WOULD BE FEBRUARY!

=====

FEBRUARY OUM

Articles for inclusion in the February issue should be with me by January 26th - PLEASE.

Thanks to ARNT and THE MUSO for getting in advance with their articles.

=====

WELCOME

Welcome to a new reader from EIRE. ✓

David Clifford is at: 10 SLIP PARK, BANTRY, Co.CORK, EIRE.

David is cassette based and enjoys novelty games such as INSECT INSANITY, shoot 'em ups, and adventures such as the HAUNT.

It's nice to start the New Year with a new member.

OFF THE PHONE

Ian Hutchins is no longer on the telephone.

Letters are still accepted - the CSA can't block up the letter box!



N E W S N E W S N E W S
 CLUB EUROPE DISCS

The Autumn and Winter CEO discs have duly arrived.

The Autumn disc contains MEMORIC (a French 'matching pairs' game), DOGGY (classic arcade from Loriciciels), FOOTBALL MANAGER (de-bugged version of the Addictive title), some Music in the form of PRELUDE FUGUE & HUBBARD, and a ROGUES GALLERY (includes Jonathan Bristow).

The Winter disc has JUMP FOX (a crazy arcade), VIDEO FLIPPER (classic pinball from LORICIELS), A.T.M (great arcade), BACH music, COPFORM (Sedoric utility), DISC EXPLORER (an interesting utility), PCX CONVERT (for transferring PCX files to the Oric), BIT 120 (a HIRES screen editor from Jonathan Bristow), and a couple of HIRES screens.

CEO NEWS

Vincent Talvas is standing down as CEO President. We wish his successor all the best and thank Vincent for his work in the past.

The CEO have recently sold a couple of TELESTRATS to youngsters (8-10 year olds). Atari and Amiga have recently closed their local branches and thus the Oric Telestrat fills a niche, as youngsters are realising that you can't program a Nintendo, and PC's are not affordable.

CHANGE OF ADDRESS

Frank Bolton has moved again!
 He and Luis now reside at : 13 WITHENS CLOSE, LEICESTER. LE3 6QN
 Tel: 0533 877128

NEW TELEPHONE NO.

Colin Cook has a new number - 0268 472336

MAGNETIX

Latest news from Jonathan Bristow is that he is still beavering away on MAGNETIX, and completion is still a few weeks away.

THE NATIONAL LOTTERY - AN APOLOGY

In the December issue of OUM, I stated that I would enter 5 tickets in the draw for Sat. December 17th. I completely forgot about this until I recieved a letter from Peter Bragg just before Xmas (see Letters Page.) My sincere apologies. After sending out the December issue of OUM, I became snowed under with work and just lost touch with ORIC-ING for a month. I will now enter 5 tickets for each of the draws taking place on January 7th, 14th & 21st. I know that this tome won't reach you by Jan. 7th, but as there is a guaranteed Jackpot of 10 million pounds for that week, then it's worth a go. Keep glued to the T.V to see how we go.

The numbers will be those that were published in the December OUM.

The Story so far

We have been looking at machine code/assembler programming technique and the last two issues have concentrated on producing a short and simple utility routine that can swap the contents of two areas of memory.

The first example produced in Part 40 of the series was aimed at providing the means to preserve the entire contents of Zero Page (0000 - 00FF area). This worked by swapping it's contents with those of another area of memory, so that the Zero Page facilities would be available for our own software. Afterwards, the same swap routine could be called again, to restore the original Zero Page contents, ready for return to Basic. The routine shown, was simple, but somewhat limited, in that it could only handle a maximum of 100 bytes. OK for Zero Page, but no good, for anything larger. The main thing at that stage was that it worked and did what it was supposed to do.

Always concentrate on getting your software working first. Fine tuning, improvements and efficiency can come later. They are easier to achieve when you have the essential elements of your software working correctly. If you try to make your software super-efficient from the start, it will take you ten times as long to write and you will be fed up, long before you finish.

However, once software is working correctly and has been put to use, it is time to look with a critical eye and see if it is possible to make a significant improvement. Even if it is satisfactory for the current project, it may be possible to extend it for other software projects.

Try again, for something better ?

This brings us to the second version of the "swap operation" in Part 41 of the series. This is intended to be an improvement on the first version. It is a bit larger and now consists of two routines. However, it is more useful, because it is no longer restricted to a maximum "swap" area of 100 bytes.

If speed had been a consideration, I would have dealt with the operation in a different way. Speed only becomes important where an operation needs to be called many times in a brief period of time. It has a lower priority in an operation that has longer periods of time, between each call as in this case.

Now let's have another look at those routines (see the listing in Part 41). Hopefully, the description in the listing is sufficient to show how the "swap" operation works. However, there are one or two points about the actual technique used to write them, that are worth looking at.

Bear in mind that we all have different ways of working and I can only really speak for my own. Note that all the variable items are located, as usual in a Parameter Block at the beginning of Page10. I always try to locate these Parameter Blocks at the beginning of a Page, because it makes the variable items easier to find, if I need them at a later date. It does not make a lot of difference in a small operation like this, but in a large piece of software containing hundreds of different operations, reserving the same area on each Page for these items, is a valuable time saver.

The routine, "Fetch 2 Bytes and Swap them over" (JSR 1010) is very short. More than half of it consists of variable addresses anyway, so it is not that much out of place, if it is considered as being located in the Parameter Block. There is a reason for this which we will come to shortly.

Now let's have a look at the other routine, "Swap Contents of Two RAM Areas" (CALL#1020). The first part of the routine (instructions 1020 to 1041) ensure that the operation always uses the addresses in the Parameter Block.

The rest of the routine is a loop which goes through the two memory areas, swapping bytes over, one pair at a time, until the end address is reached.

Instructions 1044 to 1052, just test to see if the "swap" end address has been reached. As this is a "built in" operation and not one requiring input from the keyboard, a straight test for a single address is all that is needed. When that address is reached, the "swap" operation ends.

Instruction 1054 does the actual "swap" using JSR 1010. This only swaps a single pair of bytes over from the two addresses as set in the four LDA/STA instructions in the JSR 1010 subroutine. The four instruction addresses must be updated after each "swap" by incrementing (adding 01) to each address. Updating is done by instructions 1057 - 1068 and instructions 106B - 107C. The two updating operations are basically the same, so let's look at one.

Oric addresses are two bytes long and the value 01 is added to the low byte first and any carry over from that result is then added to the address high byte. The instruction "Add with Carry" (ADC) is used for the operation. The first instruction is "CLC", to ensure that the Carry flag is clear, before using the "ADC" instruction, because the Carry flag is included in the addition. If it happened to be "set", it would add another "1" to the result.

The Carry flag is also affected by the result of the "ADC" instruction. If the result is 00 to FF, the Carry flag will be cleared to "0". If the result is higher than that (ie. FF+01=100), the lowest two digits are put into the result byte and the extra digit "1" is put into the Carry flag for use in the next part of the addition operation. This is how addition is handled for large values, starting with the lowest byte and working up to the highest.

So in theory, to update the four addresses we should add 01 to the low byte of each address and then add the Carry to each high byte. That's one way. However there is a better way, that is shorter and probably quicker.

The operation in 1057 - 1068 updates two indential addresses. Instruction 1058 fetches the low byte from the first address and instruction 105B adds 01 to it. That action also affects the Carry flag, so the next instruction which is "Branch if Carry Clear" ("BCC") can be used to control what happens to the two address high bytes. If there is a carry from the addition, the Carry flag will be set to "1", the Branch will be disabled and the two address high bytes will be incremented (+01 in each) by instructions 105F and 1062. However, if there is no carry from the addition, the Carry flag will be cleared to "0" and the Branch will operate so that the two increment instructions are skipped. Finally, the two instructions 1065 and 1068 copy the result of the addition back into the two address low bytes. The operation in 106B - 107C works in the same way, on the other two addresses.

Finally, we come to the "instruction" at 107F/80, which provides the loop, back to "test address" at 1044 again. This could have been done with an ordinary JMP 1044 instruction. However, if you look at the routine, you will see that it is completely relocatable, providing that you use the artificial jump produced by the CLC/BCC Branch combination. The Parameter Block is the only fixed item. The main routine can be relocated and called from anywhere you wish, because all the "absolute" addresses used by the routine, are located outside it. Happy New Year to all Oric Users.....See you then.

LOOKING AT CLASSIC GAMES

with Arnt Erik Isaksen

Part 3

IJK Software

The following games from IJK Software have already been reviewed in "Looking at Classic Games" :

- * "Ghost Gobbler", "Invaders" and "Green X Toad" (Part 1.1).
- * "Xenon 1", "Zorgon's Revenge", "Xenon III - The Genesis Probe", "Don't Press the Letter Q" and "Damsel in Distress" (Part 2.1).

ZEBBIE. IJK Software 1985.

In this very original scrolling platform game, you are in control of Zebbie. Increase his energy by holding down a key. When his energy has increased to an acceptable level, releasing the button makes Zebbie jump high or low according to the accumulated energy. Jump from one structure to the other while picking up oil and bonus points. Five bottles allow you to play a bonus level where you eat peanuts to jump around. The scrolling screen makes the game fun and Zebbie's life very hard. Zebbie must rely on your skills!

The idea of this game is simple but extremely well thought up due to its originality. Sounds and graphics are simple but do the job as the game in itself is addictive. The most terrible thing about this game is that it seems to be incompatible when using a disc-drive. Zebbie is, however, recommended for everyone who uses a tape system or drivers who still think it is acceptable to wait 3-4 minutes to load a game once in a while.

O:***** G:*** S:** A:*****

PROBE 3. IJK Software 1983.

In Probe 3 you control a space-craft and must shoot the enemy's crafts that move toward you in some sort of 3D display. Missiles that are fired at you by the enemy must be avoided. That is all this game is about.

Probe 3 is not original and is getting boring already the first time it is played. However, the graphics are brilliant and must be seen by every Orician. Anyway, the not-so-famous author, Barrie Cridland, should take a lesson when it comes to music and sounds. Some terrible music is played before the start of a game - possibly to damage the player's ears. B. Cridland is not as evil as one could think, because the sounds can be turned off.

O:** G:***** S:* A:*

ATTACK OF THE CYBERMEN.

IJK Software 1984. Stephen Haigh.

Control Percy and shoot the cybermen with you autofire. You will get to a bonus level when all of the cybermen have been terminated.

A classic game that allows the player to use a joystick (PASE compatible). Certainly not a strategy game. This is a pure shoot'em-up

game. Attack of the Cybermen is similar to Arena 3000 (Microdeal) but better. I have not much more to say about this game except for the fact that I have not wasted too much of my time for playing it.

O:** G:*** S:*** A:**

TRICKSHOT. IJK Software 1984. Stephen Haigh.

This game simulates a type of snooker with six numbered balls and a points system. You have three balls with three shots each. Shoot the correct balls out of the board at the right time to score high.

Trickshot is just as Attack of the Cybermen written by Stephen Haigh. This is better and as far as I know the only snooker game for the Oric. The simulation is good and realistic. The graphics and sounds are not impressive but do the job.

O:2* G:*** S:**** A:*****

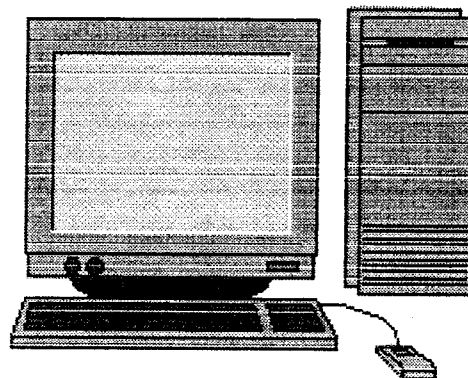
The Oric was famous, or actually infamous, for having very few programs made for it during its early years. IJK Software did, however, do their best for making quality software for our little computer. They succeeded by giving us the programs that have been reviewed in "Looking at Classic Games" and many more.

To show how much influence IJK Software has had in the Oric world, I will give a summary of Your Oric's last top 30 charts (June/July 1987):

- 1.Zorgon's Revenge
- 2.Damsel in Distress
- 4.Xenon III - Genesis Probe
- 8.Xenon 1
- 16.Gubbie
- 17.Don't Press the Letter Q
- 21.Zebbie
- 24.Ghost Gobbler
- 29.Playground 21

This speaks for itself. What should we have done if IJK Software had never existed?

I will look at some French games from Loriciciels next month. Loriciciels has had the same role in France as IJK Software has had in England and elsewhere in Europe.



(This is a picture of an Atmos - NOT)

Here we go again. This time round I've nicked a Speccy program and converted it to proper BASIC for the fab and groovy ORIC. Being a Speccy prog, it is stopped by a syntax error. What does this program do? Read on

This is a program to show how waves combine. The way two waves of the same frequency combine can be constructive, or destructive. This depends on whether the waves are in or out of phase.

If we draw a line through the centre of the wave, the top can be considered positive and the bottom negative. You then just add the waves at different points along the centre line. Waves in phase will combine to produce a wave twice the amplitude. Exactly out of phase waves will cancel each other out. This can be shown by the program. Enter '0' for waves in phase, and '9.4' for almost out of phase waves.

The program can however be adapted to produce much more interesting results. The second wave can be changed so that the wave produced is that of a 'beat', as described in 'Marshall's Music 1'.

We can make other alterations to the program to simulate harmonics being combined. I've done this using the first three harmonics. The source waves have been removed to show the resultant wave more clearly. This is getting very close to a square wave which could be achieved by adding more waves.

Jiggle about with the values and see what sort of shapes you can produce.

```
10 REM Wave Addition
20 HIRES
30 INPUT"INPUT PHASE CHANGE";P$:PH=VAL(P$)
40 FOR F=0TO4:CURSET3,10+(30*F),1:DRAW230,0,1:NEXTF
50 CURSET20,0,1:DRAW0,150,1
55 CURSET0,10,0:CHAR50,0,1:CURSET0,40,0:CHAR49,0,1
60 CURSET0,70,0:CHAR48,0,1:CURSET0,100,0:CHAR45,0,1
65 CURSET7,100,0:CHAR49,0,1
70 CURSET0,130,0:CHAR45,0,1:CURSET7,130,0:CHAR50,0,1
90 X4=0
100 FOR F=1 TO PI*65
110 X1=70+SIN(F/15)*30
120 X2=70+SIN(F/15+PH)*30
130 X3=X2+X1-70:IFX3>X4THENX4=X3
140 CURSETF+20,X1,1:CURSETF+20,X2,1:CURSETF+20,X3,1
150 IFF/10=INT(F/10)THENGOTO200
160 NEXTF
170 REM
200 FORG=-1TO1:FORH=-1TO1
210 CURSETF+20+G,X3+H,1:NEXTH:NEXTG
220 GOTO160
```

'BEAT-WAVE'. Here we change the program to give different wavelengths. When run, type in '3'.

```
110 X1=70+SIN(F/6)*30      120 X2=70+SIN(F/5)*30
...For better viewing also type 140 CURSET F+20,X3,1
```

For the harmonics simulation insert these lines

```
110 X1=70+SIN(F/20)*30      120 X2=70+SIN(F/5)*15
125 X5=70+SIN(F/10)*7.5     130 X3=X2+X1+X5-140:IFX3>X4THENX4=X3
140CURSET F+20,X3,1
```

That's all folks.

Muso



DEAR DAVE,

I see that we didn't win the Lottery Jackpot. Never mind, it's a great idea for an occasional flutter and no doubt had most OUM readers glued to the TV screen on Saturday. The other USER GROUPS would have been green with envy, if OUM had won a few million.

- PETER BRAGG

DEAR PETER,

OOPS! - see the 'NEWS' page.

Hope you recieved the message to tune in on Sat. Jan. 7th.

- DAVE

=====

DEAR DAVE,

ta for December OUM. It was good to see some listings appearing in OUM again. Great to see others contributing. Any more articles on the way from Matthew Coates?

Enjoying OUMDISC #5. Colin's version of MASTERMIND is excellent. MONTANA PATIENCE is also great, but could do with:- a) A better shuffle routine & b) A means of re-playing without having to re-boot every time.

The oyer programs were good, but tended to be aimed at the younger folk. I had fun with TELL ME though, which brought back some memories. The music tutors are highly recommended for those learning music. I found GRAVITOR very tricky to control. So all in all there is plenty to keep one busy on the old Oric.

Work on my game, CYBOJUDGE, has included improving the program structure to allow for additional palying screens. I have got a couple of ideas that should improve the game a lot if they work OK.

Regarding Dr. Ray's ideas: I am very interested in any addons/extensions to the Oric. I wouldn't use 'C' or Pascal myself, but would be interested in an Assembler or WORD-SPEED on EPROM. I wouldn't mind if the device was ready built or otherwise.

- STEVE MARSHALL

(Edinburgh)

DEAR STEVE,

contributions to the magazine recently have been really pleasing. Matthew Coates told me some while ago that he intends to do another article.

Gald you like OUMDISC #5. I decided to put a few games for the youngsters on it, as it was near the festive season, and i knew that many of our readers had children; and it was a way to get them involved. Perhaps Nick Haworth can put your ideas into action regarding MONTANA PATIENCE.

Keep working away on CYBOJUDGE. It really could be a good game from what I've seen so far.

Through these pages I'd like to thank you and Ally for all the effort you've put into your music articles, Hi-score tables, and in the other ways that you have helped with OUM.

Nice to see yet another viewpoint for Dr. Ray.

- DAVE

=====

DEAR DAVE,

I have been watching a recording of PINK FLOYD in concert. I must say it was pretty good. Excellent effects.

Is there anyone in Oricland who can put some Floyd music in a program along with special effects (flashing lights, lasers etc.)?

- KEN DUDDLE (Leicester)

LETTERS (Contd.)

DEAR KEN,

perhaps someone out there will respond. They should note, however, that you are cassette based.

- DAVE

=====

DEAR DAVE,

I hope that you have rested after your hectic week before Christmas and that life is getting back to normal again.

Please find enclosed cheque for the game COLUMNS. A good idea to be able to try a limited edition out before buying (as per OUMDISC #5).

- ALLAN MOORE

(Sheffield).

DEAR ALLAN,

unfortunately life is still very hectic. Work at the dairy was hectic all the way through December and continues to be that way due to another change in working procedures. Add to this the fact that I am now regularly doing three Discos a week, and you will understand why the ORIC is suffering somewhat at present. I am at least a month behind with letters/orders. Apologies to all. I am taking steps to rectify this - I've bought a clock with 40 hours in a day!

Glad you liked the idea of view before buy on COLUMNS. Hopefully I'll do the same on future OUM Discs.

- DAVE

=====

DEAR DAVE,

thank you for another year of great work with the magazine. A Happy New Year to all.

- RAUL HAKLI (Helsinki)

DEAR RAUL,

glad to hear that you enjoy the magazine. How about telling us what you do with your Oric, and we would be interested to know whay computers are popular in Finland.

- DAVE

=====

DEAR DAVE,

the OUM seems to be getting better all of the time. I cannot put it down when I recieve it. I am finding the music sections of interest and the Hi-Score Table. Thus during the holidays I will probably have a go at a few games to try and beat some scores.

I have not bought a new interface yet, as I have been trying to get mine up and running; as yet without success.

Please let me know if you hear of any second-hand ones for sale. I can't wait to get onto disc. If no second-hand ones, can you let me know the current price of a new interface.

- GARRY EGERTON (Leek)

DEAR GARRY,

I'm sure that the 'MUSD' will be pleased to here that his efforts are appreciated.

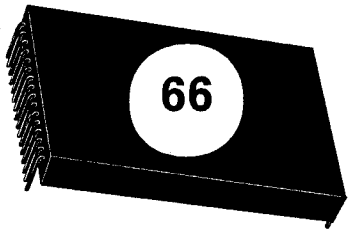
I do not know of any second-hand interfaces currently up for grabs.

New disc interfaces are available from Steve Hopps at 45 pounds - I think postage is extra. Steve can be contacted on 0734 328251 (evenings/weekends).

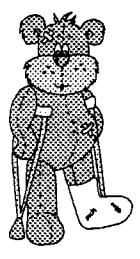
- DAVE

=====

RAMBLING



IN THE ROM



MICROWAVES

Repeating paradox

There is a simple way of speeding up either Oric Basic or the keyboard auto-repeat. Type POKE 775, 10. The auto-repeat will be much faster but programs will run more slowly.

Alternatively, put a big number in 775 (for example, POKE 775, 255) and the auto-repeat will switch off but programs will execute noticeably faster.

To get back to normal POKE 775, 39.

*Andrew Mellanby,
Aberdeen*

PCN 21
4 Aug 83

And so another year fades into the memory... And one I shan't be too sad to see the back of! I'm back at work full-time from the New Year, so my personal wish is that 1995 is very normal! That includes of course the continued success of OUM, - time to thank Dave once again for all his enthusiasm and hard work.

I'm already taken by the idea of Paris in 3 hours from London - might we put together a small expedition for the June CEO meeting? Drop me a line if you're interested.

And so...

Rambling on...

D847 JSR \$00E8	D902 JSR \$00E8	take first character of string
D84A JSR \$DFCF	D905 JSR \$DFE7	and evaluate a number in ACC1
D84D PLA	D908 PLA	recover the character
D84E LDY #00	D909 LDY #00	adjust index
D850 STA (93), Y	D90B STA (93), Y	and put it back in place

RECOVER TEXTPTR IN #E0-E1

D852 LDX E0	D90D LDX E0	
D854 LDY E1	D90F LDY E1	take saved
D856 STX E9	D911 STX E9	
D858 STY EA	D913 STY EA	and replace in TXTPTR
D85A RTS	D915 RTS	

TAKE TWO INTEGER PARAMETERS

Entry: TXTPTR correctly positioned

Exit: #33-#34 contains the first parameter, X the second. The parameters must be separated by a comma. The exit is via a JMP #00E8, so A, Z and C are positioned for the next character.

D85B JSR \$CE77	D916 JSR \$CF03	Evaluate numeric expression
D85E JSR \$D867	D919 JSR \$D922	ACC1--> #D4-#D3 (unsigned)
D861 JSR \$CFD9	D91C JSR \$D065	seek ','

D864 JMP \$D80D D91F JMP \$D8C8 then take an integer in X

ACC1--> YA UNSIGNED

Entry: ACC1 contains the desired value

Exit: YA, #D4-#D3, #33-#34 contain the value as an integer. Z and N are set according to the low byte.

D867	LDA D5	D922	LDA D5	take sign
D869	BMI D807	D924	BMI D8C2	if negative, 'ILLEGAL QUANTITY'
D86B	LDA D0	D926	LDA D0	take exponent
D86D	CMP #91	D928	CMP #91	if above 16
D86F	BCS D807	D92A	BCSD8C2	also 'ILLEGAL QUANTITY'
D871	JSR \$DF74	D92C	JSR \$DF8C	ACC1--> #D4-#D3-#D2-#D1 signed
D874	LDA D3	D92F	LDA D3	
D876	LDA D4	D931	LDA D4	recover value in YA
D878	STY 33	D933	STY 33	
D87A	STA 34	D935	STA 34	and save
D87C	RTS	D937	RTS	

'PEEK' (FUNCTION)

D87D	LDA 34	D938	LDA 34	
D87FF	PHA	D93A	PHA	
D880	LDA 33	D93B	LDA 33	Has to be saved because of POKE
D882	PHA	D93D	PHA	
D883	JSR \$D867	D93E	JSR \$D922	ACC1--> #33-#34
D886	LDY #00	D941	LDY #00	
D888	LDA (33), Y	D943	LDA (33), Y	take the value
D88A	TAY	D945	TAY	and save it in Y
D88B	PLA	D946	PLA	
D88C	STA 33	D947	STA 33	and recover #33
D88E	PLA	D949	PLA	
D88F	STA 34	D94A	STA 34	and #34
D891	JMP \$D3FD	D94C	JMP \$D4B6	Y--> ACC1 (unsigned)

'POKE' (COMMAND)

Remark:

Given the routine's configuration (place the address in #33-#34, then fetch the value), it is essential that no function affects #33-#34, even though that may mean saving it elsewhere temporarily (cf. PEEK). It's a curious complication, the same addresses as for DOKE (#1D-#1E) could have as easily been used.

Bug:

In V1.0 POKE with a hexadecimal number utterly unpredictable. The POKE itself is not the cause, or at least not directly. Simply, the routine for evaluating a hexadecimal number uses... #33-#34, which contains the value to poke at the end instead of the address to poke it to! The bug has been corrected by using other system variables than #33-#34 (see #E848/#E981).

D894	JSR \$D85B	D94F	JSR \$D916	take address in #33-#34 and value in X
D897	TXA	D952	TXA	value in A
D898	LDY #00	D953	LDY #00	index
D89A	STA (33), Y	D955	STA (33), Y	and store the value
D89C	RTS	D957	RTS	

'WAIT' (COMMAND)

Principal:

Initialise Timer 2 to the value given, and wait until it reduces to 0. The timer is decremented on each interruption, i.e. every 1/100th of a second normally. Beware, though, if you inhibit the IRQ, WAIT hangs...

Programming:

A more rational routine would have been, for V1.1 for example:

```
JSR $E853
TAX
LDA #02
JMP $EEC9
```

D89D JSR \$CE77	D958 JSR \$CF03	evaluate a number
D8A0 JSR \$D867	D95B JSR \$D922	ACC1--> #33-#34
D8A3 LDY 33	D95E LDY 33	
D8A5 LDX 34	D960 LDX 34	take the number
D8A7 LDA #02	D962 LDA #02	indicate Timer 2
D8A9 JMP \$EBDC	D964 JMP \$EEC9	YX--> timer 2 and wait until Timer 3 =0

'DOKE' (COMMAND)

D8AC JSR \$E79D	D967 JSR \$E853	evaluate the address
D8AF LDA 33	D96A LDA 33	
D8B1 LDY 34	D96C LDY 34	and take the address and save it
D8B3 STA 1D	D96E STA 1D	(useless, it's in YA)
D8B5 STY 1E	D970 STY 1E	
D8B7 JSR \$CFD9	D972 JSR \$D065	look for ','
D8BA JSR \$E79D	D975 JSR \$E853	take the value to position
D8BD LDY #01	D978 LDY #01	prepare to move down two bytes
D8BF LDA 0033, Y	D97A LDA 0033, Y	take the value
D8C2 STA (1D), Y	D97D STA (1D), Y	and position it
D8C4 DEY	D97F DEY	
D8C5 BPL D8BF	D980 BPL D97A	and the next...
D8C7 RTS	D982 RTS	

'DEEK' (FUNCTION)

D8C8 JSR \$D867	D983 JSR \$D922	ACC1--> #33-#34
D8CB LDY #01	D986 LDY #01	index high byte
D8CD LDA (33), Y	D988 LDA (33), Y	take high byte of result
D8CF PHA	D98A PHA	and save on the stack
D8D0 DEY	D98B DEY	index low byte
D8D1 LDA (33), Y	D98C LDA (33), Y	and take it
D8D3 TAY	D98E TAY	in Y
D8D4 PLA	D98F PLA	YA contains the value
.....	D990 JMP \$DF40	YA---> ACC1 (unsigned)

YA --> ACC1 (unsigned)

Principal:

One first converts to a signed number, that is to say that the numbers from 37268 to 65535 will become -32767 to -1, the others remain correct. It is then a matter of adding 65536 to recover the correct result.

D8D5 JSR \$D3ED	YA--> ACC1 (signed)
D8D8 BIT D5	is the result positive?
D8DA BPL D8E3	yes, OK
D8DC LDA #E4	no, add #10000 to recover
D8DE LDY #D8	the correct value. AY points to 65536

D8E0	JMP \$DA97	ACC1=ACC1+(AY)
D8E3	RTS	
D8E4	BYT #91, #00, #00, #00, #00	65546 or #10000
D8E9	BYT #82, #49, #0F, #DA, #9E	3.14159265 or π (PI)

'PI' (FUNCTION)

PI is a function - a constant certainly, but a function and not a pseudo-variable as some have described it. A pseudo-variable is used by the routine which finds the address of a variable, and which intercepts if it recognises an internal variable name (the TIME variable of some computers is treated like this). The Oric has no pseudo-variable. Some computers effectively treat PI as a pseudo-variable.

D8EE	LDA #E9	
D8F0	LDY #D8	index the value of PI
D8F2	JMP \$DE73	(AY) --> ACC1

Convert A into hexadecimal

Principal:

Classic, break down into two digits, and then add the ASCII code for 0, then add 7 to find the A,B... if necessary.

D8F5	PHA	D993	PHA	save the number
D8F6	LSR A	D994	LSR A	and move 4 bits of the high byte
D8F7	LSR A	D995	LSR A	into 4 bits of the low byte
D8F8	LSR A	D996	LSR A	b7 b6 b5 b4 x x x x -> 0 0 0 0 b7 b6 b5 b4
D8F9	LSR A	D997	LSR A	
D8FA	JSR \$D8FE	D998	JSR \$D99C	convert to ASCII
D8FD	PLA	D99B	PLA	recover number

Convert into 1 digit ASCII

D8FE	AND #0F	D99C	AND #0F	and isolate the 4 bits of the low byte
D900	ORA #'0'	D99E	ORA #'0'	add the code for '0'
D902	CMP #'.'	D9A0	CMP #'.'	is it a number or a letter?
D904	BCC D908	D9A2	BCC D9A6	it's a number, OK
D906	ADC #06	D9A4	ADC #06	+6+C=+7:#41-#46, it's a letter
D908	CMP #'0'	D9A6	CMP #'0'	is it in fact 0?
D90A	BNE D910	D9A8	BNE D9AE	no, save it (and jump)
D90C	LDY 2F	D9AA	LDY 2F	yes, take indicator for number 0
D90E	BEQ D916	D9AC	BEQ D9B4	that's it, no need for a 0 in front
D910	STA 2F	D9AE	STA 2F	indicate not 0
D912	STA 0100, X	D9B0	STA 0100, X	and recopy into the buffer
D915	INX	D9B3	INX	and index the next number
D916	RTS	D9B4	RTS	

Quite fun, these short routines, aren't they? More next month...

A very Happy New Year to all,

Jon Haworth

EPROM CARD

Paul Hill has sent me an article that was first published in the June 1984 issue of 'ETI'.

Titled 'EPROM CARD FOR THE ORIC/ATMOS', it describes a card that can be used as a way of storing machine code routines, or simply as a computer-controlled EPROM programmer/verifier/duplicator.

The main feature of the project is that the function performed by each 2K block of addresses in the 8K total can be programmed individually to be one of four functions.

These can be:

- a) Read only from any one EPROM socket.
- b) Read from an EPROM socket, write into the RAM.
- c) Read & write to the RAM.
- d) Read EPROM programming socket, program a byte into EPROM in socket.

The article also comes with P.C.B construction & testing etc.

I have passed all to Dr.Ray for his comments.

=====

HELP REQUIRED

Steve (The Muso) Marshall wants to know if anyone can write an article about handling graphics? e.g. when do you get a chunk out of the screen (when using colour attributes). How to avoid it e.g. as in Colin Cook's MASTERMIND - you don't have a chunk of screen missing in front of each different colour peg.

Also, sometimes when you use coloured text, you get a block at the start. Steve would be very interested if someone could write about this and clarify the various conditions.

Note from the Editor: First place to start would be with a couple of books, namely: "Getting More from your Oric" by H.E.Hicks and "Oric Atmos & Oric 1 Graphics & Machine Code Techniques" by Geoff Phillips. The latter was partly serialised by Allan Whitaker in previous issues of O.U.M. So check through your OUM INDEX and back issues of OUM.

Meanwhile, perhaps someone out there could do an article specific to your problems, or at least get in touch with you. Surely one of our programmers out there has come across these problems.

=====

CAPABLE CABLE

Robert Crisp has come across a review of a 'CAPABLE CABLE' for the Oric/Atmos. Published in a March 1984 issue of P.C.N, the cable (named the HYBRID) gives the Oric/Atmos disk drive power, cures known bugs in the Rom & improves its Basic. Made by ITL & selling at 100 pounds, it contained miniaturised component mounting in a tiny, almost flat box at the micro's end of the cable. No bigger than the average connector, it contained the disk interface & Basic enhancements. It could be bought with the Byte Drive 500 (Dr.Ray's baby).

Robert has sent a copy of the review to Dr.Ray, and I have sent one on to Jon Haworth.

=====

LYNX 6502

Jonathan Bristow informs me that the Atari Lynx uses a 6502 processor (same as the Oric), but runs at 16 Mhz. Why shouldn't the ORIC run at a faster speed? Apparently hardware can be obtained to run the 6502 on the Commodore 64 at 8 Mhz. An expensive piece of hardware we are informed, and it has been noted that not all 6502's work - some burn out.

Any views?

BRIAN'S PAGE.....

HELLO & A HAPPY NEW YEAR TO YOU ALL !

I WILL START THIS MONTH BY GIVING THE ANSWER TO LAST MONTHS POSER , WHICH IS 908 , WHICH WHEN USED WITH THE GIVEN FORMULAE GIVES A REMAINDER OF 823 FOR EACH NUMBER GIVEN.

NOW FOR THIS MONTHS PUZZLE . THE FOLLOWING CRYPTARITHM WAS DEVISED BY J.E. TREVOR OF CORNELL UNIVERSITY , IN THE 1940's .GIVEN THAT ALL "*"s" REPRESENT PRIME NUMBERS , CAN YOU WORK OUT THE SUM - OH BY THE WAY THE ONLY PRIME NUMBERS YOU CAN USE ARE 2 , 3 , 5 & 7 .

```

      * * *
            X
      * *
-----
    * * * *
    * * * * 0
-----
    * * * * *
-----

```

ANSWER NEXT MONTH

AS AN END PIECE I WILL LEAVE A SMALL BASIC LISTING THAT SHOWS HOW TO DRAW CIRCLES ON THE HIRES SCREEN - IT ALSO ALLOWS FOR WRAP OVER - SOMETHING NOT NORMALLY AVAILABLE . IT IS SLOW TO PLOT THE CIRCLES - SO HOW ABOUT A BUDDING M/C PROGRAMMER OUT THERE CONVERTING IT ? A LITTLE CONVERSION CAN RESULT IN PART CIRCLES BEING DRAWN ANYWHERE ON THE SCREEN .

```

100 REM DRAWING CIRCLES - FULL WRAP      110 :
120 HIRES                                  130 INPUT "START 'X' POSITION ";K
140 CLS                                    150 INPUT "START 'Y' POSITION ";L
160 CLS                                    170 INPUT "WRAP AROUND (Y/N) ";A$
180 BF=1:CLS                               190 IF A$="N" THEN BF=3
200 REM G=RADIUS                           210 FOR G=10 TO 50 STEP 10
220 REM WORK OUT CURSET POSITIONS         230 FOR N=0 TO 2*PI STEP PI/180
240 FB=1                                    250 H=INT(G*COS(N)):I=INT(G*SIN(N))
260 HI=K+H:II=L+1                         270 GOTO 320
280 FB=BF                                  290 REM FIT IT INTO SCREEN WINDOW
300 REM X=20 TO 219                       310 REM Y=20 TO 179
320 IF HI<1 THEN HI=219+HI:GOTO 280       330 IF II<1 THEN II=179+I:GOTO 280
340 IF HI>239 THEN HI=HI-219:GOTO 280    350 IF II>199 THEN II=II-179:GOTO
280                                         370 NEXT N
360 CURSET HI,II,FB
380 NEXT G

```

NEXT MONTH - IT'S BINGO TIME FOR THE ORIC - WELL WORTH WAITING FOR .

IF YOU HAVE ANY SUGGESTIONS FOR TOPICS THEN LET ME KNOW - OR BETTER STILL WRITE AN ARTICLE YOURSELF - NEW CONTRIBUTIONS FOR THE MAG WILL ALWAYS BE WELCOME .