I.O.U.

The voice of the Independent Oric Users' Group.

ISSUE No. 2 - early June 1985

EDITORIAL

I've just noticed that I put the wrong year at the top of issue 1 and that nobody told me. Obviously too much time at the computer isn't good for you!.

Response to issue 1 was fair and, surprisingly enough, I am still getting an occassional new reply to my letter in PCN. I am now certain that there are enough interested users to make the I.O.U.G. a viable proposition provided that you participate actively.

I have tried to get some more publicity but letters sent to Popular Computing Weekly and Home Computing Weekly some weeks ago seem to have been ignored. The latter surprised me since this magazine provides good support for the Oric. I have since written to Your Computer and Personal Computing Today. I was almost certainly too late to make the current issue of Your Computer so check out next month's issues of both mags.Perhaps some of you could write to the weeklies mentioned above just to let them know that I am not a con-man. If you do then tell them my phone number - (061) 431 4160. If any of you have ideas as to how to get publicity on the cheap I would appreciate some help.

Since issue 1 Oric, as you doubtless all know, has been purchased by the French company SPID. However this does not seem likely to have any short-term beneficial effect for British users so let's stick together. Your contributions are urgently needed be they software reviews, book reviews, programming hints/tips, programs that you've written or any imformation at all that you think will be of interest to other users. Contributions typed/printed on A5 paper make my life a much easier one (ie direct photocopying) but this not essential. Don't worry if you're not a literary genious, send me the facts and if you request then I'll rewrite the article for you.

I hope for your continued support. Until next time,

G.Ramsay

NEWSFLASH

Cheap Software

I've recently purchased several items of software at ultra low prices - 99p and £ 1.99. I got these fro two independent record shops in Manchester's Arndale Centre. Check out similar shops in your area to see if there are any bargains available and if you see shops with knockdown softwar for other micros see if you can persuade them to get some Oric tapes.

Cheap Books

Seen in a discount bookshop in Manchester city centre :-

The Oric Programmer - by S.M.Gee & Mike James £ 1.25
The Oric Book of Games - by the above + K.Ewbank £ 1.25
~~The Oric 1 Companion - by Bob Maunder £ 1.00~~
The Oric 1 and How to Get the Most From It - Ian Sinclair £ 1.25

Last time I checked stocks were high and so I'm willing to get these books for you. Send me a cheque/P.O. crossed and made payable to me for the price of the book that you want plus a stamp for 69p plus an envelope at least 16cm X 24cm or sufficient brown wrapping paper for a book of this size. Do no address the envelope or put the stamp on it just in case I can not obtain the book in which case all money will be promptly refunded less the cost of return postage ie 13p.

New Service

Due to popular demand issue 3 will contain a page that I should have thought of myself. Namely a page for users to advertise software, hardware, books etc. or to get in contact with other users. Since the swopping of software is probably illegal I cannot advertise this but (almost) anything else goes.

STOP PRESS: Letter appears in HCW

RESPONSE - your views, hints and problems

Mr A.H. Meeson from Walsall would appreciate help in converting the 3-D noughts and crosses program in issue 4 of Oric Computing to the Atmos.

Mr A.D Edwards of Widnes is having trouble with programs that contain lots of LPRINT statements - apparently they wont work without a printer attached. Any answers ?.

Mr J. Woods has some Atmos compatible games that he no longer wants. If you have Atmos compatible software of any type and feel interested then contact Mr Woods on (061) 224 0009.

Brian Miles from Leeds would like the answers to the following
1. Is there anywhere in the north of England where a broken Oric can be repaired?.
2. Is it possible to buy/fit an Atmos keyboard to an Oric-1?.
3. What is the availability of French software ?.
4. Is there a 'back-up' copy tape for the Oric-1 ?.

Geoff Clifford of North Petherton, Somerset informs me that Oric appointed an official repairers and that they provided an excellent service when they fixed his Atmos. They are :-
Lorian Computer Services, P.O. Box 129, Weybridge, Surrey. KT13 9SL
He would also like to know whether it is possible to control the cassette recorder via the remote control independently of the CLOAD and CSAVE commands.

Nick Rees sends the following:-
1. Anyone with the game 'Hopper' that loads with 'errors found' try typing CALL#400, as this gets the game going.
2. Does anyone have a spare Atmos manual?. His Oric 1 had the V. 1.1 Rom but was supplied with the old manual.
3. Anyone in the Brighton/South-East area who would like to exchange ideas write to Nick Rees, 12 Hayes Close, Ringmer Lewes, East Sussex. BN8 5HN

ALGORITHMIC ANGLES

As some of you will know if you want a career in computer programming then knowing BASIC isn't a great help. Programs are usually written in other, much more powerful languages. COBOL is by far the most used but due to an attempt by its designers to make programs easy to read it is very longwinded. A rising star amongst high level languages (HLL) is Pascal, a highly structured language which is now the most popular with the British universities and thus increasingly attractive in commercial programming. Another good reason for would-be programmers to learn Pascal is that it forms the foundation for ADA. This new HLL was commissioned by the U.S. Dept. of Defence and is widely tipped as the language of the '90's. Another widely used language is FORTRAN.

What this all means is that since a programmer is likely to use several languages during his/her career the most important skill needed is that of program design.It is essential to construct a good design before converting into a HLL so that a well structured, easily understandable program can be obtained. Furthermore a good design is language independent i. can be coded into any HLL with reference only to the design. The design method that I've been learning is called design by progressive/stepwise refinement and one of its big pluses is that it avoids tedious flowcharting. I'm going to try to pass on some of what I've learnt and show how to code the designs into Oric BASIC. This issue I shall start with the looping structures.

Loops are the basic element of computing since the strongpoi of computers is their ability to perform operations over and over again to the point where humans would find it tedious and thus make errors.

There are three fundamental looping constructs:-
1) WHILE loop
2) REPEAT-UNTIL loop
3) FOR loop

continued on next page...

1) WHILE loop

```
loop while (condition is true)
     (block of loop)
end of while loop
```

The instructions in the block of the loop are repeated WHILE the while condition is true. When this becomes false the program continues from the point immediately following the end of the while loop. Note the condition must have a value prior to encountering the loop so that it can be tested. This makes it possible not to execute the instructions within the block of the loop at all. Although not available in Orio BASIC the while loop can be neatly coded using GOTO statements.

```
e.g.  0  REM  * WHILE loop *
      10 LET X=0
      20  IF X<10 THEN 30 ELSE 60: REM loop while X<10
      30  PRINT"A"
      40  X=X+1
      50  GOTO 20 :REM end of while loop
      60  REM * REST OF PROGRAM *
```

This small code fragment simply prints out ten ' A's '. If line 10 is altered to LET X=10 then lines 20 to 50 are not executed.

2) REPEAT-UNTIL loop

```
repeat loop
       (block of loop)
 until (condition is true)
```

The Repeat loop is of course included in Orio BASIC as is the other fundamental loop:-

3) FOR loop

```
loop for index:= x to y
     (block of loop)
end of for loop
```

Note: a) 'index' represents a variable known as the control variable. x and y are constants or expressions.

b) ':=' indicates an assignment of a value to a variable as opposed to '=' which indicates the evaluation of an expression.

c) The NEXT statement is not used since it is peculiar to BASIC.

G.Ramsay

```
10 REM*****************************************************************
11 REM                        'DATA-MAKER'                              *
12 REM This program constructs data statements from assembler.         *
13 REM It is totally relocatable, and uses just under 200 bytes.       *
14 REM                                                                 *
15 REM Requires:                                                       *
16 REM          DOKE 0,start address                                   *
17 REM          DOKE 2,end address+1                                   *
18 REM          DOKE 4,line-no                                         *
19 REM                                                                 *
20 REM The data statements will be added to the end of any program     *
21 REM already present, so line-no must be greater than the last       *
22 REM line of the original program.                                   *
23 REM The data statements will be numbered in increments of 5,        *
24 REM starting with the given line.                                   *
25 REM                                                                 *
26 REM To use, enter the above DOKEs and then CALL #9700               *
27 REM                                                                 *
28 REM                        Steve Brunton. March 85                  *
29 REM*****************************************************************
100 HIMEM#9700:I=#9700
110 READ DTA
120 REPEAT
130 : POKE I,DTA
140 : I=I+1
150 : READ DTA
160 UNTIL DTA=-1
1000 DATA#A5,#9C,#38,#E9,#02,#85,#06,#A5,#9D,#E9,#00,#85,#07,#A2,#00,#A0
1005 DATA#02,#A5,#04,#91,#06,#A5,#05,#C8,#91,#06,#A9,#91,#C8,#91,#06,#A9
1010 DATA#23,#C8,#91,#06,#A1,#00,#48,#29,#F0,#4A,#4A,#4A,#4A,#C9,#0A,#90
1015 DATA#04,#69,#36,#90,#02,#09,#30,#C8,#91,#06,#68,#29,#0F,#C9,#0A,#90
1020 DATA#04,#69,#36,#90,#02,#09,#30,#C8,#91,#06,#E6,#00,#D0,#02,#E6,#01
1025 DATA#A5,#01,#C5,#03,#90,#06,#A5,#00,#C5,#02,#B0,#36,#C0,#40,#B0,#07
1030 DATA#A9,#2C,#C8,#91,#06,#D0,#B6,#C8,#A9,#00,#91,#06,#98,#A0,#00,#A6
1035 DATA#07,#38,#65,#06,#90,#01,#E8,#EA,#91,#06,#C8,#48,#8A,#91,#06,#68
1040 DATA#85,#06,#86,#07,#18,#A5,#04,#69,#05,#85,#04,#90,#80,#E6,#05,#18
1045 DATA#90,#F9,#A9,#00,#C8,#91,#06,#C8,#91,#06,#C8,#91,#06,#88,#98,#A6
1050 DATA#07,#18,#65,#06,#90,#01,#E8,#EA,#A0,#00,#91,#06,#8A,#C8,#91,#06
1055 DATA#88,#B1,#06,#18,#69,#02,#85,#9C,#C8,#B1,#06,#69,#00,#85,#9D,#60
1060 DATA -1
```

Assembly Language Programming Techniques

1. Multiplication

There are several methods of multiplication in assembler language. The easiest of these to understand is probably the 'repeated addition' method. To multiply two numbers n and m say, simply add n to itself m times. Because of its simplicity, this method is commonly used. However, when the numbers are large the number of additions needed can make this method comparatively slow. For instance, to multiply 174 by 203, the minimum number of additions required is 173. If the numbers are not checked for size, then 202 additions may be needed.

An alternative method uses an 'add and shift right' algorithm. This makes use of the property that each partial product will be equal to either the multiplicand or to zero (see fig. 1).

```
multiplicand      1101     13
multiplier        1010     10
                  ----
          ¦       0000
partial   ¦      1101
products  ¦     0000
          ¦    1101
               --------     ---
product       10000010     130
```

fig. 1. Example binary multiplication

The algorithm can be expressed as

```
repeat
  take least significant bit of multiplier
  if =1 then add multiplicand to most
      significant bits of product
  rotate product right
  shift multiplier right
until all bits done
```

Note that the least significant bit of a binary number is that on the right as it is written down - ie the bit representing 1. The example in figure 1 shows two 4-bit numbers multiplied to produce an 8-bit result. Multiplying two 8-bit numbers would give a 16-bit result.

Consider decimal multiplication, say 100 x 50. 100 can be expressed as 1x10^2 and 50 as 5x10^1. When multiplying, the exponents are added, giving 5x10^3 or 5000. The same holds for binary. 8 bits can hold 2^8 different values (0-255). The maximum product will be 2^8x2^8 = 2^16 so a 16-bit number is required to hold the product of two 8-bit numbers. Now - back to the multiplication algorithm. Working through it shows:-

```
multiplier  add c¦product
  101¦0      0    0¦0000
                  0¦00000 rot.right
 . 10¦1    1101   0¦11010
                  0¦011010 rot.
   1¦0      0    0¦011010
                  0¦0011010 rot.
    ¦1     1101   1¦0000010
           .      0¦10000010 rot.
```

The answer of 10000010 is obtained, which is of course correct. A study of this method shows it is the same as standard binary multiplication as you may do on paper, but the partial products are added in as they are produced. the number of significant bits in the product increases by one from the left every step. The routine is straightforward to implement, and may be as follows.

```
      LDA #$00     ; load product with zero
      STA PRDLOW   ;
      LDX #$08     ; x used to count 8 bits
LOOP  LSR MPLIER   ; test LSB of multiplier
      BCC SKIP     ; if 0 then skip add
      CLC          ;
      ADC MPLCND   ; if 1 then acc=acc+MPLCND
SKIP  ROR          ; shift product right 1 place
      ROR PRDLOW   ; save carry in PRDLOW
      DEX          ; if all bits done then end
      BNE LOOP     ; else loop back
```

where MPLIER is multiplier
MPLCND is multiplicand
PRDLOW is low byte of product
accumulator holds high byte.

A similar 'shift left' algorithm also exists, and the routine can be coded in a variety of different ways. Happy programming.

Steve Brunton

ATMOS COMPATIBILITY

The following games are Atmos compatible:-

All IJK
Pasta Blasta by Arcadia
Elektrstorm, Light Cycle, Moon Base Alpha(M.A.R.C.), The Ultra all from PSS.
Reversi - CDS Microsystems
Ghostman, Grail, Lone Raider, Gravitor, Moria - Severn
Orio Trek, Lost in Space, Classic Racing - Salamander
Scuba Diver - Durrel

This list is just the tip of the iceberg. I would like to compile an extensive list of Orio software. I would like to know which of this software is Atmos compatible and which software has appeared in both non-compatible and Atmos compatible formats. This is something that you can all help with so write in.

Issue 3 will be out in the second week of July ,price 35p. Please send crossed cheques/P.O. made payable to me. No stamps please. If you send cash then wrap it well.

To summarise the procedure for getting the next issue of I.O.U. is:- send an S.S.A.E. plus payment in advance. This shall continue to be the procedure until otherwise notified. In the event of the group failing you will be promptly refunded using your S.A.E. .

n.b. The I.O.U.G. is non-profitmaking.

Definition of computing terms

INTERFACE - "He called me a wally so I 'it 'im interface

ASSEMBLER - Worker on a production line

SOFTWARE - Washing done in fabric conditioner

FIRMWARE - Too much starch in the collar

HARDWARE - Far too much starch in the collar

PROGRAM - Computer show on television

RS232 - A type of Ford Escort

CURSOR - Person uttering expletives

BYTE - A piece of an apple

BIT - A small amount of memory

INTERPRETER - Parlez vous Francaise ?

BASIC INTERPRETER - Non !

(editorial comment: I didn't make these up!! .)