ØRIC USER MONTHLY

with Alternative Micros

Number 80

April 1994

Keeping the Oric alive

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

T H E   E D I T O R I A L

BONJOUR,
and welcome to yet another bumper issue of O.U.M.
I am putting this issue to bed early (still a week left of March as I finish it).
It's great to see so many articles popping through the letter box,and all before the deadline.
I am in the embarassing position of having to hold certain items over until next time.
I am busy working on the backlog (as usual).

DON'T FORGET - APRIL IS MINE!

Please do not expect any new queries answered during April. I am going to concentrate on what I want to do!

And so to the INDEX for this issue:

==

BAD DEBTS

If you haven't yet paid for OUMDISC#4,then please send your money now. Otherwise you will not recieve further credit.

SUSPENDED

It has been my unhappy duty to suspend membership of OUM to one of our readers. Last Summer I sent some software and a printer lead to the person. Ever since I have recieved excuses like: "I've been busy"," I haven't forgot you", and "I'll send it off this week". Enough is enough. The person will recieve no further issues of OUM until the debt is cleared. He has been informed.

==

N E W S . . . N E W S . . . N E W S

THE CONTACT LIST

With this issue you will finally recieve the complete,un-abridged,uncensored,unexpurgated version of the ORIC CONTACT LIST. Find out where those other weirdos live, what their tackle consists of, what they have for breakfast,and what plonkers also have a SPECCY!

Sincere thanks to Richard (he's 26 you know!) Farrell for the thankless task of typing it all up.
It will be up to YOU to keep it up to date (no addresses in future OUM's - look 'em up yourself).
For the well organised amongst you, the EASYTEXT text file of the Contact List will be on OUMDISC #5.

==

SONIX - Version 3.5

The updated SONIX files have been sent out either individually to users or in the form of a 'chain' disc/letter. If you have not recieved the update,then please contact me. I'll then find out who has broken the chain and break his knee caps!

COLUMNS

Every COLUMNS user should now be using the updated version (amend sound). If you are not then please do the following: A) Smack your Postman in the mouth for failing to deliver, B) Send a photo to me of your blood stained Postman to recieve an update!

THE ORIC MEET

If you haven't already bought your tickets for the Aylesbury ORIC MEET, then NOW is the time. Already comitted to attend and hopeful are: Peter Bragg,Jonathan Bristow,Matthew Coates,Arthur Crawford, Ron Evans,Richard Farrell,James Groom,Jon and Nick Haworth,Chris Hearn,Steve Hopps,John Hughes,Brian Kidd,Rob Kimberley,David Leibniz,Henry and Rene Marke,Steve Marshall and Allison,Ray McLaughlin,Allan Moore,John Peach,Bob Terry,Pete Thornburn,David Utting,David Wilkin,Steve Wright, and some French friends.
Get your 2 pounds off to me now, and if your not coming,then don't forget the raffle tickets at 1 pound each.

==

MAY O.U.M

ARTICLES FOR INCLUSION IN THE MAY ISSUE SHOULD REACH ME BY APRIL 24th AT THE LATEST - PLEASE.

THE GREAT SPRING CLEAR OUT

SPRING IS IN THE AIR,AND I HAVE BEEN BUSY SORTING THROUGH MY REAMS OF ORIC-RELATED PAPERWORK. NOTES TO MYSELF,NOTES FROM OTHERS,AND LITTLE TIT-BITS I HAVE DUG OUT WHICH CAME MY WAY WHEN SOME ORICIANS SOLD UP.
HERE THEN IS A MISHMASH OF ORIC ITEMS - SOME USEFUL AND PERHAPS OTHERS NOT; SOME NEW TO YOU AND SOME PERHAPS OLD NEWS.
LOCHNESS MONSTER (ROMIK) will work with the ALTAI (PASE) joystick interface.
REVERSI (IJK) will only work under Sedoric on a master disc,else an 'out of memory' error occurs.
3D BATTLESTAR (TOPAZ) - some disc versions are corrupt in such a way that you cannot read which order to put in your own choice of keys. The order is:Down,Up,Left,Right,Fire.
ZEBBIE (IJK) - infinite lives: POKE 132000,6 GASTRONON - keys are: Z,X and SPACE
GRENDEL (MIRAGE) - to use the Pokes on the disc version: load the file "PROG.COM" with ,N to stop auto-run. Then enter the Pokes as lines at the start of the program, resave the file,then load the whole game as normal.
GAMEINIT - on OUMDISC #4 we gave you the file so that you could format a Sedoric Gameinit disc to 82 tracks and 17 sectors. If you want to risk it then you may format it to 18 or even 19 sectors.
BBC 'B' - Richard Farrell now owns one
. DRAGON - James Groom tells me that he has a DRAGON - his mother sounded alright on the telephone!
BERING - on this French Game you get one life. You must go up for air,miss the harpoons,and eat the plankton.
A really interesting (yawn,yawn) game!
THE RANDOM CHALLENGE - A Note from Colin Cook.
" There is an entirely different way to generate 'random' numbers,using so-called Chaos Theory. Using the equation:

Xnew = r.x.(1 - x)

and starting values for r between 3.6 and 3.83, (and some higher values work as well),and starting value for x between 0 and 1, a series of values between 0 and 1 is generated. Assigning the variable q temporarily to these values,they can be turned into +1 and -1 using the expression:

Value = 1 - 2 * INT(Q * 2)

which is something like example 5 from John Hughes in the March '94 OUM. For some reason, though,this series is more likely to generate -1 for r in the range given,although it should be possible to cahnge this ratio by altering the constants in the 2nd equation"

SCUBA DIVE - a tip from Steve Marshall: Select level 5. On the first screen there is a baddy on virtually every line (and only one to each line). Now you know where it is safe to go. Surface each time you collect all the oysters (don't collect the far left or far right ones - too dangerous). Level 5 also has more treasures to collect.
MCP40 - Some MCP40 owners may not have the manual,and so here is how to go through the test routines:
Turn on once to get the 4 squares.
Turn OFF and then ON,whilst holding down the Line Feed, to get the character set test.
ZEBULON - The cheat word on ZEBULON is KERGUELIN. You will not lose energy,but will not be congratulated on completion of the game. All you have to do is figure out when to enter the word!

LORIGRAPH - From Steve Marshall comes a menu,which allows you to look at the directory before loading the main program. Very handy for those who can't remember the name they gave to a picture i.e your Editor.

```
  5 PRINTCHR$(17)
 10 CLS
 20 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
 30 PRINT:PRINTCHR$(129)CHR$(142)"LORIGRAPH"
 40 PRINTCHR$(142)"LORIGRAPH"
 45PRINT:PRINT
 50 PRINT:PRINTCHR$(133)"TYPE IN YOUR CHOICE"
 60 PRINT:PRINTCHR$(130)"1 - DIRECTORY"
 70 PRINTCHR$(129)"2 - LORIGRAPH"
 80 GETA$
 90 IFA$="1"THEN CLS:!DIR:GOTO120
100 IFA$="2"THEN CLS:!LORIGRAPH.COM
110 IFA$<>"1"ANDA$<>"2"THEN80
120 PRINT:PRINT:PRINT"PRESS ANY KEY FOR MENU":PRINT:PRINT"OR 'Q' TO QUIT"
200 IFZ$="Q" THEN STOP
220 GOTO10
```

READERS LETTERS

DEAR DAVE,

I have recently acquired an ORIC ATMOS,which my neighbour was about to bin - incidentally he bought it from a GRATTAN shop for a pound a couple of years ago.

Being an avid fan of old 'technology' computers, I also own a Commodore 64 and my favourite - a Dragon 64K c/w disc drive..... I belong to the great National Dragon User Group....

- STUART PARNELL (BENFLEET).

DEAR STUART,

the Dragon user group is so great that it can no longer organise a MEET. From information I have recieved from other members regarding lack of input from readers etc - it is not much to write home about. Perhaps they will prove us all wrong and send someone along to our Aylesbury meet - YOU are more than welcome!

- DAVE

==

DEAR DAVE,

the OUM Disc continues to be great value for money, as TETRIS GB is worth the asking price alone! It's certainly been worth the wait

I'm also impressed with COLUMNS,particularly the smooth colourful graphics (if a little small),and responsive gameplay. The sound is good too,but it is a pity there are no sound effects when the music is switched off. Nevertheless it's obvious a lot of time and effort has been spent producing this excellent game.

- PAUL HUTTON (Worcester).

DEAR PAUL,

nice to get response as it gives myself and our software writers the drive to do more.

- DAVE

==

DEAR DAVE,

could do with a translation of HYPERSPACE off OUMDISC #4 - can you help?

- STEVE 'Muso' (Edinburgh).

DEAR STEVE,

Yes - I'll lend you a French dictionary!

Alternately,perhaps someone could take on the task - how about it Norma?

- DAVE

===

DEAR DAVE,

I think Dr.Ray's ASSEMBLER is very good. I have to put my hands up and say that I am not a 6502 programmer (yet), but as far as I can see it's all an Assembler should be. Plus you have the other goodies on the disc if you want to play around a bit. The manual is good too,all in all it's very professional. I prefer it to the ORION assembler I used to have.

As for MIND MADNEZ - first impressions were a little dissapointing,the graphics were simplistic and the actual gameplay seemed a bit easy too. Now I have played it over the week-end I have been pleasantly surprised. It is addictive,frustrating,at times good fun. So I'm glad I got it to be honest. I know you said it was expensive for an ORIC title,but let's get it in perspective.

.... CONTINUED

A cartridge for the SNES costs 45 quid and you are asking a fiver for Mind Madnez. Fair enough - no comparison on graphics and sound, but I have always felt that it was the game itself which was the key element. Mind Madnez is playable. I know of several SNES cartridges which are complete rubbish. So I don't feel hard done by and I don't think anyone else should either.
- Richard Farrell (Darlington).

DEAR RICHARD,
sorry it took so long to publish your letter (late November). I did ask for your opinion on the software and am pleased to get a response.
One thing bugs me about your typed letters. I hope you don't mind me pointing it (you are not the only one who does it). When using the one letter word 'I', you have a habit of using lower case i.e. 'i'. Technically you are correct (within a sentence), but it looks really naff. Richard - are you still talking to me?

Your description of Mind Madnez sounds like the old woman (addictive,frustrating,simplistic). Perhaps you should do the review for the CEO!
Regarding the rubbish SNES games you have - ARNT wants them back. He didn't realise that he hadn't kept a copy himself after he had wrote them.
OOPS! Just remembered - it's the CEOMAG that Arnt has stopped subscribing to, not OUM! DO YOU STILL LOVE ME ARNTIE BABY!
- DAVE
==

DEAR DAVE,
a number of things to cover in this letter:
OUMDISK #4 - a great effort on your part and excellent value for money. I've only skimmed over the surface yet,but I like the sound effects in TETRIS GB. A pity the score is wiped at the end of the game.
SEDORIC AND OPELCO DISK SYSTEMS - occasionally I have trouble with drive B,which appears to react slower to commands than drive A. With ORIC DOS present,this does not matter,because the drive whirrs for several seconds before the head starts moving. In Sedoric there appears to be a time limit set before the system comes up with a Track/Sector error,and this is nowhere near long enough for the drive to respond. Is there any way of increasing the time that the DOS will wait for a drive to respond?
At present,the only way of 'freeing up' drive B is to re-boot with RANDOS and do a !DIR 1,so that the head can move into a position where it can respond faster when Sedoric is re-booted.....
- COLIN COOK (Pitsea).

DEAR COLIN,
glad you liked the disc.
Regarding your Sedoric problem. It's not a problem that I have come accross. Perhaps Dr.Ray could respond with regards to drive response time.
Failing this - why not bring your set-up along to the MEET as it is always best to see a problem first-hand. After all - it may be your OPELCO and not the DOS!
Other items from your letter are to be found on other pages.
Sincere thanks for your donation to the raffle. Also a big thank you from Frank Bolton for donating an Atmos,MCP40,cassette player and other things for his Romanian appeal.
Also just recieved for Romania is an Oric- 1 from Arthur Crawford.
- Dave Dick
==

B I T S 'n' B O B S

WANTED FOR CASH

Steve Marshall is looking to complete his collection of ORIC OWNER magazines.
Issue 8 is the missing one - contact Steve direct. Even though Steve lives in Scotland, he does understand English!

ORIC MEET RAFFLE

Get your tickets for the GRAND RAFFLE to be held at the MEET. Only 1 pound a ticket. Latest additions to the mega prize list include a box of Continuous paper (nicely perforated Frank!), and a packet of A4 paper from Allan Moore.
And from Colin Cook - 4 books, an ORIC-1 car sticker and six blank cassettes.

AND TALKING OF COLIN!

Peter Bragg's recent discussion in OUM of masking, has given Colin Cook an idea for a way of using the RND(1) expression twice to make the outcome more random. The idea is to make a template number of 8 bits (i.e 0 to 255) using RND(1)*256, and then create a single-bit mask using (in effect) 2 to the power of P to check whether the bit in that position is set or not. Colin says that it is a bit clumsy in Basic, but here goes:

```
10 TEMPLATE=INT(RND(1)*256)
20 POWER=INT(RND(1)*8)+1
25 MASK=.5
30 FOR I=1 TO POWER:MASK=MA*2:NEXT
40 R=1-2*(MA AND TE)/MA
50 PRINT R,
55 COUNT=COUNT+R
60 IF KEY$=" " THEN PRINT:PRINT CHR$(27)"A";COUNT
70 GOTO 10
```

Lines from 55 are to enable printing in red of the running total,which for even amounts of +1 and -1 should oscillate around zero.

===

THE NEXT ARTICLE FROM M.C

Matthew Coates informs me that his next article for OUM will probably be on Speed and Efficiency!
Does drugs does he?
I've heard of Speed,but am not sure about Efficiency. Sounds like an Alka Seltzer derivative!

===

CASSETTE SOFTWARE

I AM CURRENTLY HALFWAY THROUGH PRODUCING AN UP TO DATE CASSETTE SOFTWARE MAIL ORDER LIST.
I MUST CLEAR THEM ALL OUT. I HAVE BOXES OF THEM EVERYWHERE. ALL PRICES (EXCEPT CURRENT 'MIRAGE' TITLES) WILL BE DRAMATICALLY REDUCED. THIS WILL BE YOUR LAST CHANCE TO OBTAIN THEM AS I WILL THEN PROBABLY DITCH THE MAJORITY AND JUST KEEP A FEW. THE FEW REMAINING WILL HAVE THEIR PRICES JACKED BACK UP - SUPPLY AND DEMAND!
SAVE YOUR CASH AND WAIT FOR THE LIST. REDUCED PRICES WILL ALSO BE OFFERED TO DISC USERS.

MARSHALL'S MUSIC - The sixth bit

Hello again folks. Due to the recent volume of music related articles, this month's session is for the tappers. Yes it's time for a bit of BASIC. The main source is Stuart Wilson's 'West Lothian Oric User Group Newsletter'. Some are 'original' whilst others are taken from PCN magazine. Have some fun and tap them in !

'FOOTSTEPS'

```
10 FOR N=1 TO 20
20 PLAY0,1,1,300
30 P=(N AND 1)*5 + 10
40 SOUND4,P,5
50 PLAY0,0,0,0
60 WAIT 60
70 NEXT N
```

SOME NOISES FOR YOU TO TRY IN YOUR PROGRAMS

```
PLAY 40,40,60,4000   - WAVES
PLAY 30,40,20,9000   - MORE WAVES !
PLAY 48,90,100,60    - BUZZER
PLAY 200,100,100,20 - PLANE
PLAY 58,80,100,60    - ROTOR
PLAY 12,91,555,3     - JET
PLAY 200,100,100,200- TRAIN
PLAY 17,80,100,200   - ALERT
```

Here's a program from PCN's 'Microwaves' for ORIC-1 owners. It doesn't work on thr ATMOS.It allows you to type in two values to produce some sounds. Some examples of X and Y registers to try are to the left of the listing.

```
10 FOR I=#A000 TO #A007
20 READ A$: A$="#"+A$:A=VAL(A$)
30 POKE I,A
40 NEXT I
50 DATA A0,00,A2,00,20,6C,FA,60
55 CLS
60 INPUT "Value for X ";X
65 INPUT "Value for Y ";Y
70 POKE #A001,X:POKE #A003,Y
80 CALL #A000
90 GOTO 55
```

X REGISTER	Y REGISTER
#FC	#0F
#FC	#2A
#FC	#82
#FC	#A9
#FC	#B9
#FC	#EC
#FC	#F7
#FD	#0B

This ones for all the budding drummers out there !

```
10 PLAY 20,40,65,200
20 WAIT 40
30 PLAY 80,23,65,300
40 WAIT 20
50 PLAY 20,40,65,200
60 WAIT 20
70 PLAY 20,40,65,200
80 WAIT 40
90 PLAY 80,23,65,300
100 WAIT 40: GOTO 10
```

SOME MORE NOISES I FOUND USING A PROGRAM THAT GENERATED RANDOM PARAMETERS. GIVE THEM A TRY !

```
PLAY 59,32,60,92
PLAY 87,42,68,8.4
PLAY 1.5,29,22,41
PLAY 76,30.6,54,441
PLAY 7,46,80,180
PLAY 0.2,2.7,4.2,15
PLAY 9,5,9,886

    PLAY 3.5,6,0.2,997           PLAY 3.2,7.7,4.9,8.4
    PLAY 6.6,4.2,6.8,37          PLAY 0.9,4.8,3.7,131
    PLAY 135,146,466,560         PLAY 14.3,109,157,44
    PLAY 133,17,100,51           PLAY 178,29,6.8,282
    PLAY 146,60,126,335          PLAY 74,193,174,90
    PLAY 195,153,486,5979        PLAY 37,115,360,533
```

Finally, the funniest music demo I have heard - 'The Mating Call of 200 Gannets'.

```
10 SOUND1,65535/(RND(1)*50+500),0
20 SOUND2,65535/(RND(1)*50+500),0
30 SOUND3,65535/(RND(1)*50+500),0
40 PLAY7,0,5,10 :GOTO10
```

MUSO

Programmer's Corner 9

Structured Programming (part 2)

Last time, if you can remember that far back (January), I started to discuss the subject of structured programming. Between all the digressions, I hope I got across the general idea of modular programming and the importance of good program design. I now propose to expand upon some of the principles of what might be called "good programming practice".

The first of these, which most of us like to ignore if we can get away with it, is that of adequate documentation.

GIVE US A CLUE

At work the other week I came across a masterpiece of brevity in program documentation. This consisted of just four words scribbled on a piece of paper slipped into the disk sleeve. These were, "JUST TYPE PROG NAME". Now as long as a prospective user (in this case me) knew what the program was for, full instructions were provided on screen and no modifications would ever be necessary, I guess this is just about adequate. Unfortunately this is rarely the case.

There are two basic levels of documentation. The first is some form of user guide. This doesn't have to be extensive, but it should describe the basic function(s) of the program, explain in general terms how these are performed and what is required of the user in terms of input. Each part of the system should be discussed and all available options and valid key-presses revealed. It is amazing how some programmers seem to expect their users to be blessed with powers of telepathy. "But I thought that was obvious so I didn't bother to mention it."

While I'm on the subject, what about those on-screen instructions? In my experience there are two main problems with these; they are usually too brief and they are rarely available in the part of the program in which you want to consult them. It's true that it may not be convenient to distribute a paper manual with the software (for example, with PD programs) but how about including a routine allowing the user to print out a copy of the manual on request?

KEEPING NOTES

The second form of documentation is the development notes. These are kept primarily for the benefit of the programmer, but are invaluable for anyone else unfortunate or foolish enough to want to understand the workings of the program at a later date. They will start with the notes made at the planning stage, as discussed in the last article, but should not stop there. Some specific suggestions are given below but the general idea is to provide sufficient information to enable you or someone else to understand how the program works and what each part does.

o Make use of REM statements to introduce sections of code and to explain the function of variables, particularly those set at initialisation. I usually reserve line numbers ending in 9 for REMs, so that if there is a routine beginning on line 100, line 99 will give a brief description of what it does.

o Keep a separate description (in a project book?) of each section, giving the line numbers, a list of the variables used, details of any lines which need special explanation and a list of any subroutines accessed. It's also useful to have a master list of all of the variable names used. For subroutines, keep a list of the sections which use the routine. This allows you to chain backwards to see the effect of any changes.

o You have probably already discovered, by bitter experience, the advantages of saving a program frequently during long programming sessions. While you're at it, it's a good idea to save each version under a different name (eg PROGNAME.001. .002, etc) and to keep a record of the changes made at each stage. This way you can go back to an earlier version if anything disasterous happens or if you change your mind about a particular development. You also end up with a useful "audit trail" which helps in tracing any errors that might creep inexplicably into your otherwise flawless code.

You may have noticed by now that I am quite keen on providing as much assistance as possible if and when I have to return to a project at a later date. The reason for this is that I have never yet written a program which has worked first time and has never needed to be modified or extended. (Unless you count the 10 PRINT "KEVIN WOZ ERE":GOTO 10 sort that we all used to annoy Dixons' sales assistants with in the early eighties.) Maybe I'm just slow but I find I need all the help I can get in these situations.

WRITE-ONLY PROGRAMMING

The main arguments against the use of lots of REM statements in programs are that they take up space and slow down execution. This is true, but not necessarily a problem. This philosophy is largely a legacy from the days when we were all struggling along in 4K or less (unless you bought the massive 16K RAM pack) and interpreted BASIC was the only available option. Another common practice, and one of my pet hates, is that of compacting programs in order to increase the execution speed. For those of you fortunate enough not to have come across this, it involves removing all"unnecessary"spacesbetween words and cramming as many commands onto a single program line as possible.

Now our Kevin is a much more sophisticated programmer these days. When he goes into Dixons he might come up with something like this:

```
  9 REM   Loop to print inane message
          ========================
 10 REPEAT
 20 PRINT "KEVIN WOZ ERE"
 30 A$=KEY$
 40 IF A$>=" " THEN PING
 50 UNTIL A$=CHR$(13) OR A$=CHR$(27)
 60 CLS:ZAP
 70 PRINT "ARSENAL!!!"
```

If he were tempted to compact his program, it would end up as:

```
 10 REPEAT:PRINT"KEVIN WOZ ERE":A$=KEY
$:IFA$>=" "THENPING
 20 UNTILA$=CHR$(13)ORA$=CHR$(27):CLS:
ZAP:PRINT"ARSENAL!!!"
```

I don't know about you, but when I look at a screenful of this kind of stuff, my brain goes out to lunch. It's so much harder to follow, that to my mind it's just not worth the aggro. The increase in speed is usually marginal at best, especially when PRINT statements are involved, and there are a number of more respectable ways to achieve this end (see below). If speed is really critical, use a machine code routine or buy the compiler.

Incidentally, I happen to think that the BASIC compiler is the greatest advance in the Oric world since SEDORIC came along. One computer professional of my acquaintance assured me a couple of years ago that such a thing was impossible. I'm not after a free copy because I've already bought one and I can't vouch for its reliability since I haven't used it much yet, but three cheers for Dr. Ray! Pause while I climb down off my soap box.

GOTO AGAIN

And finally, as I promised last time, about those controversial GOTO statements. GOTO is not a tool of the devil designed to lead innocent programmers to their downfall, but it should be treated with respect and used sparingly. Where possible, replace GOTOs with the GOSUB...RETURN, FOR... NEXT and REPEAT...UNTIL constructs. We are fortunate to have the last of these, since it was not available in early versions of BASIC, so it seems a shame not to make the most of it.

One reason for prefering these alternatives is that they can increase the speed of execution considerably. When the BASIC interpreter comes across a GOTO (or, indeed, a GOSUB) it has to search for the specified line number by examining every line in turn from the start of the program, which is very time consuming. No record is kept of where the program was before the GOTO was invoked. By contrast, when a GOSUB, FOR or REPEAT statement is encountered, the current position in memory (not the line number) is pushed onto the stack (a temporary memory area). When the corresponding RETURN, NEXT or UNTIL statement is reached, the previous memory position is pulled off the stack and execution immediately continues from the original position.

The other main reason for using these constructs is that (and I'm sorry to keep harping on about this) they help to make the program more readable. By introducing a section with a FOR or a REPEAT, it is immediately obvious that this is the start of a loop. Similarly, A GOSUB tells the reader, as well as the BASIC interpreter, that execution will eventually return to this position.

Designing computer programs, like almost every other productive enterprise, is often a process of making compromises. We sometimes have to trade off structure (or readability) against efficiency (or speed). The hard part is knowing where to do it. Good hunting.

Matthew Coates

Machine Code for the Oric Atmos (Part 34)   Peter N. Bragg

The Story so far

---------------- We have looked at essentials for machine code programming and a small selection of useful 6502 Instructions appeared in Part 22 of the series. The last few issues have concentrated on the subject of binary code and how it can be of use, on the Oric.

Essentially, there are two types of useful binary operations available to us on the Oric. One of them, the masking operation, has already been covered in the last couple of issues. The other useful binary operation, is the Shift operation and this is what we are going to look at now.

Shift about a bit

----------------- So what is a shift ? and what does it do ? Well, if you look at any single byte anywhere in the memory, you probably know by now, that while it is usually written as a two digit hex number, it is really an abreviation for an eight digit binary number or eight bits for short.

Consider those eight bits for a moment, as if the byte itself was open at both ends and capable of holding just eight bits. Think of the byte as a tube with eight bits like eight "ping pong" balls with "0" or "1" written on each of them. Now suppose you take a ninth "ping pong" ball with "0" written on it and push it into the "tube" at one end. What would happen ? Well, obviously all the "ping pong" bits would shift along the tube and the last one at the other end would fall out and be discarded.

That is just what happens in a shift operation. The shift instruction pushes a single bit into one end of the byte and causes the eight bits already in there, to shift sideways one place. As the byte can only hold eight bits, the sideways shift, causes the last bit at the other end to "fall out" and be discarded.

The shift can be either direction, left or right. An extra bit is just pushed into one end of the byte by the shift instruction and the last bit at the other end is simply "lost" from the byte, which has gained the bit pushed in, instead.

Shift Work

---------- There are four instructions for shift operations on the single bytes, which can be held in the Accumulator, or stored in memory. Two of them are plain shifts, one for shift left and one for shift right. The remaining two are slightly extended versions of the first two and can make the shift into a complete "rotation" when shifting either left or right.

Let's look at the two plain shift instructions, first. These are ASL for "shift left" and LSR for "shift right". Their full titles are Arithmetic Shift Left (ASL) and Logical Shift Right (LSR). They can be used as Absolute instructions on the contents of locations in memory, or they can be made to operate on the contents of the Accumulator.

The two Accumulator instructions are the easiest to use. They are single byte instructions and simply shift the Accumulator contents one bit to the left or one bit to the right, depending on which one of them you use.

To give an example, let's imagine that the Accumulator contents have been set to the value 01 hex (which is 0000 0001 binary). The instruction to shift left is ASL, which is a single byte instruction (hex code 0A).

If we used the ASL to shift the Accumulator left once only, we would find that it would change the Accumulator contents to the value 02. Shift it twice to the left and the value in the Accumulator would be 04. Shift it three times left and the value becomes 08. So what is happening ? The answer lies in the binary values of those hex results. See list on the left, below.

Start value	01 hex	=	0000 0001	binary
1st shift	02 hex	=	0000 0010	binary
2nd shift	04 hex	=	0000 0100	binary
3rd shift	08 hex	=	0000 1000	binary

Notice that each shift left, moves the binary "1" one bit to the left. This would continue into the second half of the byte, so the next shift left would produce 10 hex and the shift after that, 20 hex and so on. This shift effect of course works on all eight bits, so the eighth bit (furthest left) is discarded by each shift. This means that if you do eight shifts to the left, you will end up with zero (00), because you will have shifted and discarded all eight bits of the byte, which then becomes binary value 0000 0000.

That was the plain shift left operation as done by the instruction ASL. The shift right operation is exactly the same except that it goes in the oposite direction. So if you started off with 01, the very first shift right, would discard the bit on the right, making the value zero (00) from then on. Of course if you started with the value 80 hex, which is 1000 0000, the first shift right would change that to 40, which is 0100 0000 in binary. These two instructions have a number of uses, the obvious one is the ability to halve or double values. It can also be used, for example, together with masking to split up a value consisting of several hex digits, in order to use those digits for display purposes.

A shift around
-------------- Shifts could be used to move a "sprite" across the screen, but of course here we hit a snag. Like most computers, the Oric's display screen is many bytes wide. We need some way to pass on the bits which are discarded, into the next screen byte in line. This brings us to the other two shift instructions which can provide a "rotate" operation.

These two instructions are ROL, which is short for "ROtate Left" and ROR which is short for "ROtate Right". Although these are called "rotate" instructions their operation is exactly the same as that described for the shift instructions, above. However, there is one small addition which makes all the difference. Remember the bit pushed into the plain shifted byte and also the bit "lost" from the other end of it ? Well, the difference in these two "rotate" instructions is that they use the Carry flag for those two bits. (You may remember that the Carry flag was used by addition and subtraction instructions to carry the overflow from one sum to another.)

The "rotate" instructions shift in the same way as plain shift instructions, but instead of pushing a zero bit into one end of the shifted byte, the Carry is pushed in and the bit, normally "lost" off the other end of the shifted byte is copied into the Carry flag. So in effect, the Carry flag provides a ninth bit for the shift and turns the operation into a complete rotation. Nothing is "lost" because the bits that "fall off" one end of the shifted byte are picked up by the Carry flag and pushed back in at the other end, by the next rotate instruction. All this is done automatically by the instruction. Rotating a single byte like this has some use, but "Rotate" instructions really come into their own, when you open the operation out, to include more than one byte and use the Carry flag to pass the contents of one byte to another smoothly, bit by bit, as you would when moving "sprites" around the screen.......................More shifty business, next time !!

THE ULTIMATE HI-SCORE TABLE

HELLO AND WELCOME AGAIN TO THE HISCORE CHARTS. Send scores to:-
STEVE MARSHALL, 149 WARRENDER PARK ROAD, EDINBURGH, SCOTLAND. EH9 1DT.
OR YOU CAN PHONE ME ON 031 228 6496

3D BATTLESTAR	58,500 (level 5.5)	Brian Kidd
3D FONGUS	150,850	Stephane Rezette
3D STARTER	75,400	Robert Cook (Founder of OUM)
ANTICS/SINGERIE	131,372	Peter Thornburn
ATLANTID	13,990 (Duree 62)	Matthew Dick
A.T.M	67,990	Robert Cook -
ARENA 3000	2,953,750 (level 200 - 13 lives)	James Groom
ATTACK OF THE CYBERMEN	4,730	Henry Marke
BERING	168 DAYS	Graeme Burton. Still the Arcade King
BOMBYX	28,530	Robert Cook
THE BOTTLE	81	Steve Marshall
BOZY BOA	4,270	Steve Marshall
CENTIPEDE	59,240	Henry Marke
CHUCKFORD	185,050	Robert Cook
CHOPPER	69,950	Vincent Talvas
COBRA	2,699,993	Rene Marke The highest score on the chart !
COCK'IN	133,057	Steve Marshall
COCORIC	3,620	Stephane Rezette
Columns	see separate entry at the end!	
COSMORIC	908	Steve Marshall
CROCKY	142,735	Steve Marshall
DAMSEL IN DISTRESS	4,860	Graeme Burton
DEFENCE FORCE	1,268,020	Tim Colgate
DELTA FOUR	9,810	Steve Marshall
DOGGY	16,250	Stephane Rezette
DON'T PANIC	25,490	Henry Marke with an excellent new score
DON'T PRESS THE LETTER Q	1,229,620	Bruno Dossier
DRACULAS REVENGE	13,600	Graeme Burton
DRIVER	66,500	J-Yves Brun
ELEKTROSTORM	25,600	Tim Colgate
FIREFLASH	69,010	Romain Dasse
FLY FOR YOUR LIFE	172	Graeme Burton
FORMULE 1	27,487	Arnt Isaksen
FRELON	17,095	Stephane Rezette
FRIGATE COMMANDER	504	Keith Thompson
GALACTOSMASH	50	Matthew Coates
GALAXIANS	69,600	Robert Cook & Nicholas Menoux
GASTRONON	11,050	Dave Dick. The Editor has a go.
GHOST GOBBLER	32,505	Steffan Jacobsson
GHOSTMAN	70,000	Elise Dasse
GOLDMINE	60,900 (GAME COMPLETED)	Henry Marke
GRAVITATOR	5,204	Arnt Isaksen
GRID WARRIORS	55,494	Graeme burton
GUBBIE	339,360	Staale Eikbraaten
HARRIER ATTACK	105,700	Staale Eikbraaten
HELLION	257,550	Matthew Green
HONEY KONG	11,436 (level 11)	Peter Thornburn
HOPPER	40,170	Tim Colgate
HU*BERT	3,120	Steve Marshall
HUNCHBACK	750,200	Benedicte Gareau
HYPERBALL	15,330	Stephane Rezette
ICE GIANT	16,170	James Groom
INSECT INSANITY	94,400	Dennis Bonfield
INTERTRON	10,800 (level 15)	Brian Kidd
INVADERS (ARCADIA)	1,850	Peter Thornburn
INVADERS (IJK)	23,650	Peter Thornburn
INVADERS (PSS)	5,470	Peter the Invader Champion Thornburn
IMAGO	8,010	Stephane Rezette
JEUX OLYMPIQUES	50,147	Arnt Isaksen
JIMMY POUBELLE	11,440	Peter Thornburn
KARATE	23,800	Arnt Isaksen
KINGDOM	109	Graeme Burton
KRILLYS	28,290	Graeme Burton
KROKATILE WALTZ	10,025	Graeme Burton. G.B SCORES A HAT-TRICK
LIGHT CYCLES	4,530	Steve Marshall
LOCHNESS MONSTERS	14,683	Graeme Burton
LODE RUNNER	16,738	Arnt Isaksen
LOKI	62,675	Tim Colgate
LONE RAIDER	80,500	Espen Andersen
LUNAR MISSION	13,129	Graeme Burton

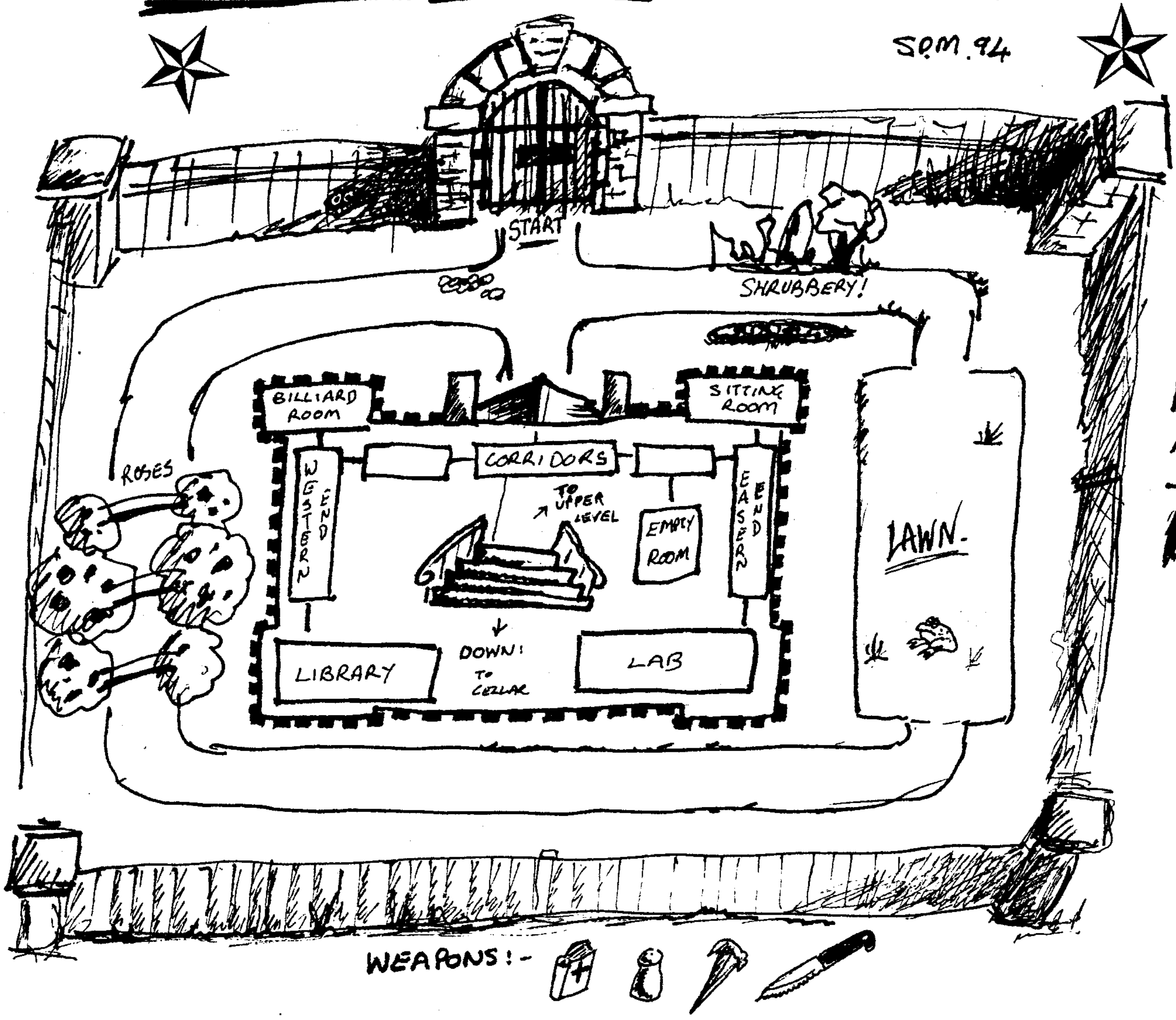
MACADAM BUMBPER - 178,700 - Stephane Rezette
MANIC MINER - 38,156 (AT THE CENTRE OF THE EARTH) - Graeme Burton
MANIC MINER with infinite lives - 115,583 - Robert Cook
MAHJONG - 16,200 (Cleared level 5 twice to give 6 levels) - Henry Marke
M.A.R.C - 1,560 - Graeme Burton
MARIO BROS - 396 - Steve Marshall
MAZE RALLY - 88,920 - Graeme Burton
MIDNIGHT FEAST - 1,500,120 - Henry Marke
MINED OUT - 4,100 - Graeme Burton
MLUCH - 22,000 (All 18 levels completed and 7 lives left) - Henry Marke
MR.WIMPY - 16,549 - Espen Andersen
MUSHROOM MANIA - 471,420 - Tim Colgate
OLIVE AND POPEYE - 69,570 - Rene Marke
OPERATION GREMLIN - 22,617 - Graeme Burton
ORION - 61,200 - Stephane Rezette
ORIC MUNCH - 895,439 - Michel Leclerc
PAINTER - 103,850 - J-Phillipe Merc
PAINTER (with 255 lives) - 143,310 - Peter Thornburn
PANIC - 823 - Peter Thornburn
PASTA BLASTA - 34,480 - Matthew Coates
PLAYGROUND 21 - 92,000 - Tim Colgate
PSYCHIATRIC - 41,070 - Henry Marke
PROBE 3 - 2,450 - Robert Cook
PROTECTOR - 99,594 - Thierry Avannier
Q*BERT - 15,470 - Dave Dick
QUACK A JACK - 95,671 - Colin Cook
QUARKFLIGHT - 709 - Graeme Burton
RABBIT - 169,760 (level 29) - Peter Thornburn
RATSPLAT - 20,150 - Staale Eikbraaten
ROCK RUN - 2,264 - Paul Hutton
SCUBA DIVE - 9,000 - James Groom
SNAKE VENOM - 102,822 - Staale Eikraaten
SORVIVOR - 1,155 - Romain Dasse
SPACE WALL - 3,248 - Brian Kidd
SPOOKY MANSION - 2,100- Steve Marshall
STANLEY - 43,480 - Romain Dasse
STRESS - 1,688 - Peter Thornburn
STOCKMARKET - 82,936 - Graeme Burton
STYX - 194,600 (wave 18) - Graeme Burton
SUPER JEEP - 138,250 - Stephane Rezette
SUPER METEORS - 364,700 - Graeme Burton
SUPER ADVANCED BREAKOUT - 17,050 - Arnt Isaksen
TALISMAN - 8,068 - Elise Dasse
TETRIS - 2,418 - Denis Bonfield
TETRIS GB - 11,568- Peter Thornburn
TETRIX - 9,983 - Jon Haworth
THEM - 1,550 - Steve Marshall
TRIATHLON - 5,270 - Stephane Rezette
TRICKSHOT - 4,128 (screen 14) - James Groom
TRIDENT NEPTUNE - 7,200 - Dave Dick
TROUBLE IN STORE - 1,060,758 - Graeme Burton
TWO GUN TURTLE - 5,890 - Graeme Burton
ULTIMA ZONE - 148,860 - Staale Eikbraaten
ULTRA - 35,780 (level 32) - Peter Thornburn
VIDEO FLIPPER - 55,350 - Graeme Burton
VISION - 285 - Brian Kidd
WILLY - 624 - P.Hutton
XENON I - 117,230 - Eric Eduezi
XENON III - 9,927 - Staale Eikbraaten
YAHTZEE - 306 - Dave Dick
ZEBBIE - 945,560 - Staale Eikbraaten
ZEBULON - All screene completed in 8 minutes - Henry Marke strikes again.
ZOOLYMPICS - 13,677 - Graeme Burton
ZORGONS REVENGE - 155,830 - E.Tollemer

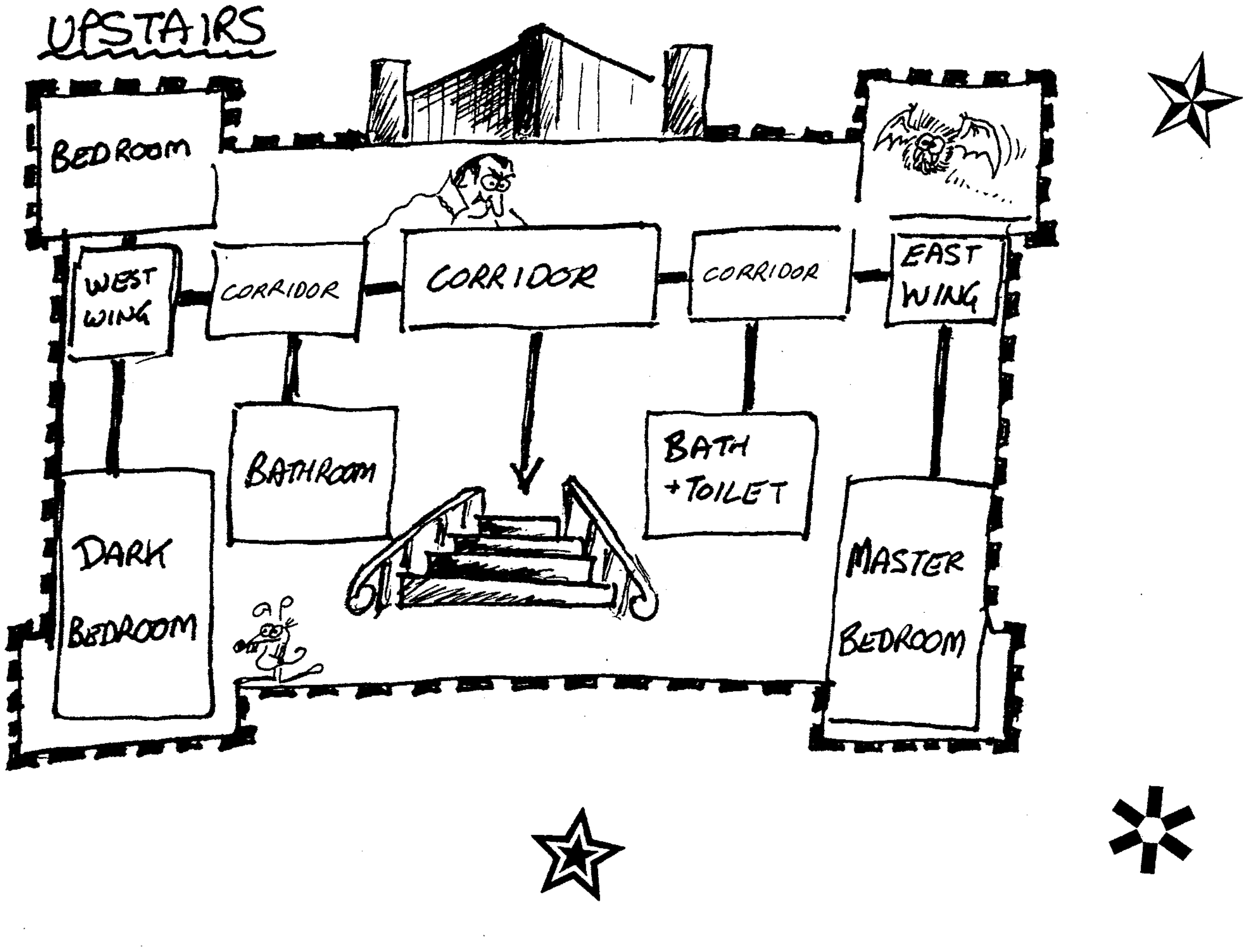
COLUMNS -
O/E- 235,650 - Brian Kidd   O/N- 12,378 - Steve Marshall
O/H- 4,725 - Liz Coates   F/E- 3 secs - Brian Kidd *
F/N- 5 secs - Brian Kidd   F/H- 10 secs - Brian Kidd *

* These scores have been equalled, but Brian was first !

TREAT YOURSELF TO SOME NEW SOFTWARE - GET YOUR NAME ON THE CHARTS

"PALLIDA MORS" - the Map





WACCI

WACCI purports to be the UK's only serious CPC magazine. The March issue (No.64) costs 1.50 incl.post and can be obtained from: WACCI, 7 Brunswood Green,Hawarden,Deeside,Clwyd. CH5 3JA.
It contains 28 pages on: Basic,Machine Code,Letters,Competitions,Discounts,Help-line,Dis(c)organization,Libraries,Reviews,Colur Printing,PD Software etc.
Professionally put together and reaching some 300 readers this seems to have something for everybody. The club seems well organised. A healthy readership and advertisers is probably what keeps the cost down.
MY HUMBLE OPINION - worth a try if you want to know what's happening!

==

APPLE IIe

Does anyone have any games or a wordprocessor on 5.25"disc disk for the APPLE IIe that they can sell or lend etc. Please contact Colin Cook direct.

==

SUB EFFECT

Issue 2 of SUB EFFECT for owners of other micros is now available from Simon Ullyatt. Issue 1 brought a good response from OUM readers and has now sold out. Simon even got an enquiry from the Lithium/Megabyte User Group in Australia.
Price of issue 2 is 1 pound (incl.post). Due to technical problems, the cassette covertape is dropped. However, 3.5" and 5.25" discs for the C64,Amiga,and IBM PC are still available. Disc are 1 pound each. The IBM disc contains 7 games,whilst the Amiga one contains 2 great! Defender variants.
Also available are an Amiga Spectrum Emulator (V1.7), and an Amiga Spectrum Games disc - both at 1.25.
Cheques should be made payable to Simon Ullyatt.

==

AMSTRAD NOTEPAD

Latest to succumb to the cheapo prices on the AMSTRAD NC100 is Chris Hearn. After reading my report on my 'crashing' experience,he still bought one , but of course wanted to know where to get the extra memory cards.
According to an old Amstrad magazine, there were many such crashes. It crashed before the memory was full and therefore no warning given. Trevor Shaw had the same problem, but overcame this by writing a routine to transfer th efiles to the trusty Oric.

SEE BACK PAGE FOR LATE NEWS ON THIS

FOR SALE £

ACORN ELECTRONS (2 in quantity), c/w Cassette software and books.
48K SPECTRUMS (2 in quantity) - these are the 'rubber' key type and come with over 50 cassette games.

Offers to: RICHARD FARRELL

BREAK ON BYTE - Steve Marshall
----- -- ----

After I recieved Sedoric V2.01 and V1.07 discs from Allan Whitaker, I tried formatting a disc, and got a 'BREAK ON BYTE' error on one and the other just crashed. I finally traced this to one of the IDC sockets. I removed this and placed this a bit further on the cable and the problem was cured! I mention this as I've read many reports of some disc drives not functioning 100%. Mine would load/save etc. with the fault,and only failed on formatting. So the message is - CHECK YOUR WIRING if a fault occurs.

NOTE FROM THE EDITOR: that's one to add to the list. Often the 'BREAK ON BYTE' is caused when trying to load software that needs the QUIT first.

SEDORIC FOR PLONKERS! - Dave Dick
------- --- ---------

After noting Frank Bolton's comments in the last issue of OUM with regard to Sedoric manual explanations, and then being caught out myself whilst using the SEEK command; I finally decided that it was time to delve further into certain aspects of SEDORIC.

Perhaps I am the plonker for not reading between the lines. But if I help a few more plonkers along the way,then I'll be happy.

What I set out to do was to change the PAPER/INK in parts of NHL ICE HOCKEY MANAGER. Certain colours do not go together,and Arnt certainly found them.

I loaded in a huge file and scanned it for PAPER/INK. With the size of the program this would take me ages to scan the listing. Seek and ye shall find and so to the Sedoric command: SEEK.

The Sedoric manual informed me that: SEEK AE lists all program lines in which the string AE exists. It also states that to search for Basic keywords, AE must contain the token value rather than the text equivalent (e.g. #80 for END).

O.K that sounds easy enough!!!

Right here we go - look up page 153 of the Atmos manual to ascertain that the Basic token for PAPER is 177. AH! But the Sedoric manual has an example in HEX rather than Decimal. So to the Atmos manual to find that 177 is in fact #B1. Right I've loaded the file with the ' ,N ' extension to stop auto-run. I then type in : SEEK #B1 and ..... Yes you guessed it - an error! Okay back to basics (as John Major would say). Try using the token: SEEK 177......... AH! a message to Insert my Master Disc (I wonder why?). And then.....and then .. - ?DISP MISMATCH ERROR or something similar.

Swear,swear,swear.

Okay, enough is enough. I dig out my full translation of the Sedoric manual.

Now according to this I should be typing: SEEK CHR$(#B1). Where the bloody hell did the CHR$ come from?

I try it and it works!

I also try SEEK CHR$ (177) and that also works.

I also find that I can SEEK for text within quotes e.g to find all lines of a program containing "BONJOUR".

In my make believe world, SEEK "BONJOUR" would list line 20 -

20 PRINT"BONJOUR PLONKERS"

Wildcards are allowed. Now this is very handy when checking Arnt's Ice Hockey game as I can seek with F and a wildcard rather than use the full expletive that he uses within the game!

Well - am I the plonker or not? I'll ring Frank to see if he mastered SEEK. Frank can't remember using it,but admits to having problems with CHANGE.

Funny that, as the Sedoric manual actually mentions the CHR# on this one.

Hang on a minute while I come out of this - temptation got the better of me!

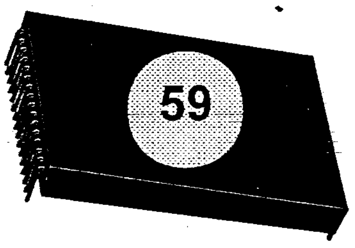
Oops! That was close - the drive hung up while I was saving this. A few tricks of the trade (tapping ,shaking,ejecting,swearing,praying and thumping), and I got lucky!

Right - I'm back. Hang on a minute. I'm going to pour a lager and change the Compact Disc. The music is very relaxing when I'm typing - I even sing along at times. Currently on the CD player is one Bobby Womack,who is a Soul singer/writer/producer who has been around for over 30 years.

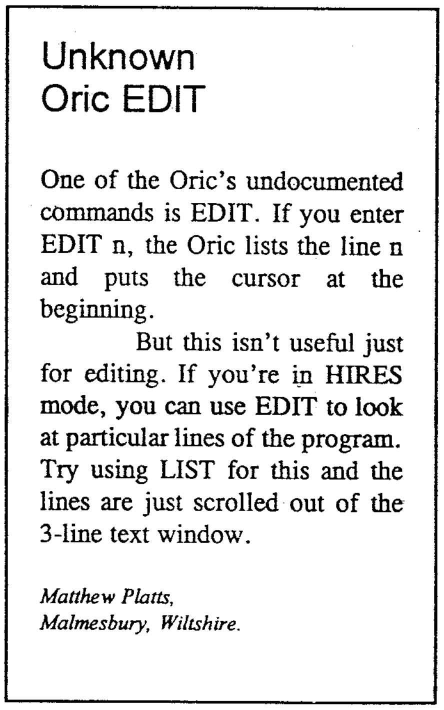
Meanwhile CHANGE worked a treat for me. How was SEEK for you Frank? - did the earth move for you or just the ORIC!

YAWN! YAWN! By the way - I spent so much time sorting out SEEK and typing this up, that i've currently given up on changing the colours on Arnt's game; but I really must CHANGE the 'F' words!

RAMBLING IN THE ROM



MICROWAVES



Unknown Oric EDIT

One of the Oric's undocumented commands is EDIT. If you enter EDIT n, the Oric lists the line n and puts the cursor at the beginning.

But this isn't useful just for editing. If you're in HIRES mode, you can use EDIT to look at particular lines of the program. Try using LIST for this and the lines are just scrolled out of the 3-line text window.

Matthew Platts,
Malmesbury, Wiltshire.

The above clipping is undated, but must have been published in Personal Computer News early in 1983. The first manual included with early Oric-1's omitted any mention of the EDIT command! Can anyone date this clip?

Matters arising...

It was marvellous to read so many pieces from 'new' contributors in last month's O.U.M. - and pieces that either call for a response from me, or with which I can offer some help. So here goes...

SEDORIC FILE NAMES (p.13)

Although you can enter filenames in lower case within Sedoric, you will notice that a DIR always produces a directory with them in capitals. In other words, Sedoric always converts file names to capitals, and therefore expects to find capitals when it looks for a file. Similarly, it always expects an extension (since it adds one by default), and therefore won't recognise a filename in capital letters if it has no extension. I came across this quirk when transferring Oric Dos files which of course do not need an extension.
The simplest answer is to run 'Nibble', select track 14, sector 04, 'R'ead the sector, switch to Ascii, and overtype the filename you will see on screen in capitals, adding a '.COM' if there is no extension. 'W'rite the sector, and when you reboot all should be well.
The sector in question is the first directory sector, Track 20, Sector 4. The 14 you enter in Nibble is simply 20 in hex. Back around issue 29 of O.U.M.(was that really fifty issues ago?) I did a series of articles on how an Oric disc is organised, to enable users to read useful information from disc via 'Nibble'. Might it be worth repeating the Sedoric part for all the new disc users, Dave? And shame on you for not thinking of 'Nibble'!

THE SEDORIC MANUAL (p.15)

I take Frank's point straightaway - it's very easy for an experienced user to describe a command in shorthand, leaving the novice scratching his head. In the authors' defence, though, I should say that the Manual was never intended to be a thorough tutorial - more a useful reference work. It actually contains more information than the original on many of the more important commands, and we did try to include examples of the use of most commands. However, it's not all perfect, as Frank's article shows.

As for LCUR, the original manual reads as follows:

Returns in the variables CX and CY, the horizontal and vertical co-ordinates of the cursor in TEXT mode.

```
Example:     10 CLS: PRINT: PRINT "test LCUR"
             20 LCUR
             30 PRINT CX, CY
             gives: 12   2
```

We omitted the example, which Frank has worked out for himself! I've now added further examples to the Manual where they were omitted before. The moral? - write in and ask if you're stuck!

RECOVERING CASSETTE FILES (p. 18)

I enjoyed reading of your tribulations, John - I've been there as well! There is, however, a very simple way of achieving your aim. If you have the C.E.O. Nibble/BD Disk, you will find that BD Disk has a 'Transfer' option (0C on the main menu). This literally reads a file from cassette (at either speed) and dumps it to disc. If you then do a FILENAME, V you get the start, end and execution addresses of the file and whether it is Basic or machine code (see the Sedoric manual under LOAD). It is then a simple matter to exit, do a QUIT (important!), !LOAD,N the file and CSAVE it to cassette. You can transfer a whole series of files one after the other, though be sure to pause the tape while the transfer file is written to disc if you don't have a relay lead. Once you have re-created the tape, you can then load ROMORIC1 and away you go!

By the way, with the Oric-1 ROM you don't get bogus 'Errors Found' messages. If you get that message, there really are errors. The bug was in the Atmos V1.1 ROM (first edition) only.

Rambling on...

Well I enjoyed that! Now it's ever onwards, however, and into the realms of functions...

'POS' (FUNCTION)

Bug: In V1.1 one instance gives a false result: if you want the screen position and the printer is on-line, you will get the same result for the screen as for the printer.
Since the function is not available for the printer on V1.0, there is no bug. The ROM updates #30 (screen or printer, whichever is current), and then the result is put in #269 (text column). Hence the bug...

```
......... .................  D4A6  JSR $D8CB    ACC1 --> X
......... .................  D4A9  TXA         and X in A !
......... .................  D4AA  BEQ D4B4    if argument is 0, screen
......... .................  D4AC  LDY 0258    if not, screen
......... .................  D4AF  BIT 02F1    are we in fact on the printer?
......... .................  D4B2  BPL D4B6    yes, it's the correct value
D3FA  LDY 0269               D4B4  LDY 30      take current value

                          Y --> ACC1 (unsigned)

D3FD  LDA #00                D4B6  LDA #00     high byte = 0
D3FF  BEQ D3ED               D4B8  BEQ D499    and place result in ACC1
```

'DEF' (COMMAND)

```
D401  CMP #&USR    D4BA  CMP #&USR    is it DEF USR?
D403  BNE D426     D4BC  BNE D4DF     no, jump
```

Treat DEF USR

```
D405  JSR $00E2    D4BE  JSR $00E2    jump USR
D408  LDA #&=      D4C1  LDA #&=      look for an '='
D40A  JSR $CFDB    D4C3  JSR $D067
D40D  JSR $E79D    D4C6  JSR $E853    and evaluate the expression
D410  LDA 33       D4C9  LDA 33       recover the value
D412  LDY 34       D4CB  LDY 34       (pointless, it's in YA)
D414  STA 22       D4CD  STA 22       and modify
D416  STY 23       D4CF  STY 23       the vector
D418  RTS          D4D1  RTS
```

PROHIBIT DIRECT MODE

```
D419  LDX A9       D4D2  LDX A9       take line number
D41B  INX          D4D4  INX          test if #FF
D41C  BNE D3A4     D4D5  BNE D4D1     no, exit
D41E  LDX #95      D4D7  LDX #95      yes, 'ILLEGAL DIRECT'
D420  BYT #2C      D4D9  BYT #2C      jump the next instruction
D421  LDX #E5      D4DA  LDX #E5      'UNDEF'D FUNCTION'
D423  JMP $C485    D4DC  JMP $C47E
```

Treat DEF FN()

Principal:

this routine takes the address of the variable, then stacks the 5 characters (variable address and definition address plus a dummy value. This dummy value is moreover the first character of the definition). These values are then taken off the stack and sent to the function definition.

```
D426  JSR $D454    D4DF  JSR $D50D    take function address
D429  JSR $D419    D4E2  JSR $D4D2    prohibit direct mode
D42C  JSR $CFD6    D4E5  JSR $D062    look for '('
D42F  LDA #80      D4E8  LDA #80      indicate not integer
D431  STA 2B       D4EA  STA 2B
D433  JSR $D0FC    D4EC  JSR $D188    take variable address
D436  JSR $D0FC    D4EF  JSR $CF06    verify not a string variable
D439  JSR $CFD3    D4F2  JSR $D05F    look for ')'
D43C  LDA #&=      D4F5  LDA #&=
D43E  JSR $CFDB    D4F7  JSR $D067    and look for '='
D441  PHA          D4FA  PHA          save first character of the definition
D442  LDA B7       D4FB  LDA B7       take variable address
D444  PHA          D4FD  PHA          and save it
D445  LDA B6       D4FE  LDA B6
D447  PHA          D500  PHA          same for low byte
D448  LDA EA       D501  LDA EA
D44A  PHA          D503  PHA          save TXTPTR
D44B  LDA E9       D504  LDA E9
D44D  PHA          D506  PHA          (it's the definiton address)
D44E  JSR $CA0A    D507  JSR $CA3C    go to the end of instruction, via DATA
D451  JMP $D4C2    D50A  JMP $D57D    and assign the function
```

Take the function address

Entry: TXTPTR poiints to FN
Exit: #BD-#BE contains the function address

```
D454  LDA #&FN      D50D  LDA #$FN      look for FN
D456  JSR $CFDB     D50F  JSR $ D067
D459  ORA #80       D512  ORA #80       adjust the code of the first letter
......... ................. D514  LDX #80       prohibit integers
D45B  STA 2B        D516  STX 2B
D45D  JSR $D103     D518  JSR $D18F     take function address
D460  STA BD        D51B  STA BD
D462  STY BE        D51D  STY BE        and save it
D464  JMP $CE7A     D51F  JMP $CF06     verify not a string
```

Evaluate a function

Principal:

The routine takes the address of the function and saves it on the stack to evaluate the parameter (for the case where the parameter itself would contain calls to a function). Then the value of the variable (which must remain unaltered) is saved on the stack.
The variable then takes the value of the parameter, and the function is evaluated and saved. Finally the original value of the variable is restored, so that it appears unchanged.

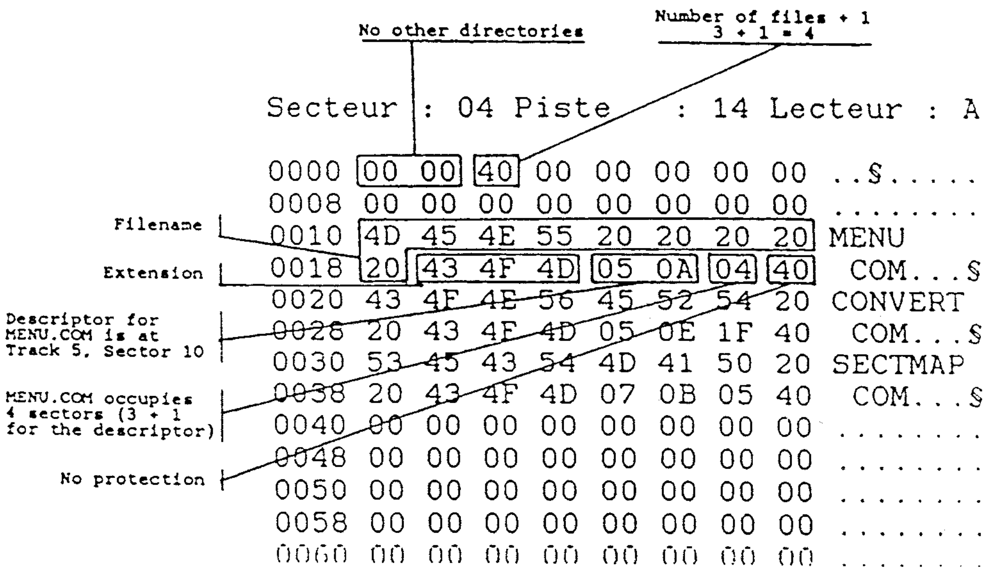
```
D467  JSR $D454     D522  JSR $D50D     find the address of the function
D46A  LDA BE        D525  LDA BE        and save its address
D46C  PHA           D527  PHA
D46D  LDA BD        D528  LDA BD        and low byte
D46F  PHA           D52A  PHA
D470  JSR $CFCD     D52B  JSR $D059     evaluate the variable
D473  JSR $CE7A     D52E  JSR $CF06     verify it's numeric
D476  PLA           D531  PLA           recover function address
D477  STA BD        D532  STA BD
D479  PLA           D534  PLA
D47A  STA BE        D535  STA BE        and high byte
D47C  LDY #02       D537  LDY #02
D47E  LDA (BD), Y   D539  LDA (BD), Y   take address of variable
D480  STA B6        D53B  STA B6        and save it
D482  TAX           D53D  TAX           in X as well
D483  INY           D53E  INY
D484  LDA (BD), Y   D53F  LDA (BD), Y   and the same for the high byte
D486  BEQ D421      D541  BEQ D4DA      if 0, UNDEF'D FUNCTION
D488  STA B7        D543  STA B7        if not, save as well
D48A  INY           D545  INY           Y=4
D48B  LDA (B6), Y   D546  LDA (B6), Y   save value of variable on the stack
D48D  PHA           D548  PHA
D48E  DEY           D549  DEY
D48F  BPL D48B      D54A  BPL D546
D491  LDY B7        D54C  LDY B7        XY = address of the variable
D493  JSR $DEA5     D54E  JSR $DEAD     AACC1 --> (XY) (Y=0 on exit)
D496  LDA EA        D551  LDA EA        save TXTPTR
D498  PHA           D553  PHA
D499  LDA E9        D554  LDA E9
D49B  PHA           D556  PHA           and low byte
D49C  LDA (BD), Y   D557  LDA (BD), Y   take definition address
D49E  STA E9        D559  STA E9        in TXTPTR
D4A0  INY           D55B  INY
D4A1  LDA (BD), Y   D55C  LDA (BD), Y   and high byte
D4A3  STA EA        D55E  STA EA
D4A5  LDA B7        D560  LDA B7        save function address
```

```
D4A7  PHA           D562  PHA
D4A8  LDA B6        D563  LDA B6
D4AA  PHA           D565  PHA            and low byte
D4AB  JSR $CE77     D566  JSR $CF03      evaluate the definition
D4AE  PLA           D569  PLA
D4AF  STA BD        D56A  STA BD         recover the function address
D4B1  PLA           D56C  PLA
D4B2  STA BE        D56D  STA BE         in #BD-#BE this time
D4B4  JSR $00E8     D56F  JSR $00E8      does the definiton terminate correctly?
D4B7  BEQ D4BC      D572  BEQ D577
D4B9  JMP $C4EF     D574  JMP $D070      no, 'SYNTAX ERROR'
D4BC  PLA           D577  PLA            yes, recover it
D4BD  STA E9        D578  STA E9         TXTPTR to return to normal execution
D4BF  PLA           D57A  PLA            of the program
D4C0  STA EA        D57B  STA EA         and low byte
```

Assign a function, recover the initial value of the parameter

```
DC42  LDY #00       D57D  LDY #00        index the definition address
D4C4  PLA           D57F  PLA            recover definition, low byte
D4C5  STA (BD), Y   D580  STA (BD), Y    (or the exponent of the paramter) and save
D4C7  PLA           D582  PLA
D4C8  INY           D583  INY
D4C9  STA (BD), Y   D584  STA (BD), Y    same for the high byte (or byte 1)
D4CB  PLA           D586  PLA
D4CC  INY           D587  INY            recover function address (or byte 2)
D4CD  STA (BD), Y   D588  STA (BD), Y    and save it
D4CF  PLA           D58A  PLA
D4D0  INY           D58B  INY
D4D1  STA (BD), Y   D58C  STA (BD), Y    and the same for address high byte (or byte 3)
D4D3  PLA           D58E  PLA            recover dummy byte or byte 4
D4D4  INY           D58F  INY
D4D5  STA (BD), Y   D590  STA (BD), Y    and save it, and that's it
D4D7  RTS           D592  RTS
```

Next time we start a whole new section of the ROM, the treatment of strings. So I hope you'll forgive me, Dave, for running to five pages - blame it on all those articles last month! It also leaves me with a quarter of a page blank, so here's a sample Sedoric directory sector annotated to illustrate what I was saying about Nibbling filenames. Which reminds me, Nibble/BD Disk and instruction sheets costs just £4.99 on 3" disc, £3.99 on 3½" - Byeee.


No other directories
Number of files + 1
3 + 1 = 4
Filename
Extension
Descriptor for MENU.COM is at Track 5, Sector 10
MENU.COM occupies 4 sectors (3 + 1 for the descriptor)
No protection

```
Secteur : 04 Piste   : 14 Lecteur : A

0000 00 00 40 00 00 00 00 00  ..§.....
0008 00 00 00 00 00 00 00 00  ........
0010 4D 45 4E 55 20 20 20 20  MENU
0018 20 43 4F 4D 05 0A 04 40   COM...§
0020 43 4F 4E 56 45 52 54 20  CONVERT
0028 20 43 4F 4D 05 0E 1F 40   COM...§
0030 53 45 43 54 4D 41 50 20  SECTMAP
0038 20 43 4F 4D 07 0B 05 40   COM...§
0040 00 00 00 00 00 00 00 00  ........
0048 00 00 00 00 00 00 00 00  ........
0050 00 00 00 00 00 00 00 00  ........
0058 00 00 00 00 00 00 00 00  ........
0060 00 00 00 00 00 00 00 00  ........
```

SOFTWARE SOUNDS - 3

The third of a series of articles originally appearing
in 'SOFT & MICRO' between October 1984 and June 1985
written by Jean-Marie Cour
translated by Jon Haworth

**In the Middle Ages it was the monks who laid the foundations of our western musical tradition.
Still today their successors sing the melodies of the Gregorian chant and jealously guard the medieval tradition, witnesses to a fantastic invention: polyphony.
Our Atmos computer, with its three channel sound generator, offers us a simple way to trace the main strands of musical history.**

In those times the monks had discovered that several 'songs' sung at the same time were much to be preferred to adding several 'voices'. A rich new sound was born. We are blasé. Choral works, the symphony orchestra, jazz and rock groups have accustomed our ear to chords, arpeggios and other counterpoints...

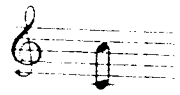
These singing Brothers have, so to speak, sown all that we have reaped. This includes the usual musical notation, the stave, notes and clefs, all of which derive directly from their 'song-guide' books.

The most elementary sound effect on our Atmos is obtained by playing the same note simultaneously on several channels. For example, you could play C3 and C4 at the same time with these instructions:

```
10      'UNISON
20      MUSIC 1,3,1,10
30      MUSIC 2,4,1,10
40      PLAY 3,0,0,0
50      WAIT 100
60      PLAY 0,0,0,0,
```

The parameter '3' in the PLAY command is the result of a binary code explained in the adjoining box. It opens channels 1 and 2 *at the same time*.

On the stave, the musician indicates that several notes start at the same time by placing them one above the other, at the same moment of time. If they are the same note in different octaves, they produce a single sound, a *unison*.

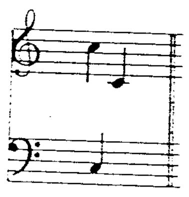


Several effects already...

Our unison of several C's will be brighter with a C2, which we can add on channel 3:

```
10      '3 CHANNEL UNISON
20      MUSIC 1,3,1,10
30      MUSIC 2,4,1,10
40      MUSIC 3,2,1,10
50      PLAY 7,0,0,0
60      WAIT 100
70      PLAY 0,0,0,0,
```

Instead of adding innumerable little extra lines under the G clef, the musician prefers to connect two staves as follows:

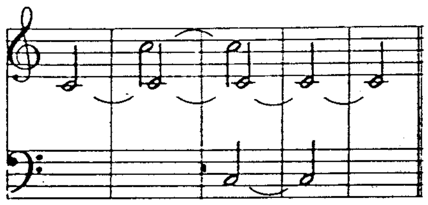


The lower one is the *F clef*. The two dots enclose the line which represents F on this stave. The two together, one stave the G clef, the other the F, have a technical advantage when writing music with notes at the extremes: only C3 is 'in the gap', and the stave lines cover the notes from G1 to F4.

For the pianist, this double stave coincides with the middle of the keyboard. The G clef stave is (mostly) played with the right hand, and the F clef stave (almost invariably) with the right.

... with time

Played together, the same note in octaves gives a richer C than one of them played alone. A brighter effect still is obtained if the following is played:



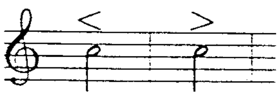
This little piece has a C3 continuing (tied) over five bars. It is accompanied by a C4 during the second and third bars, and a C2 during the third and fourth. In this way the unison progressively builds up and reduces. It can be played in Basic as follows:

```
10   'UNISON EFFECT
20   MUSIC 1,3,1,10          'C3 alone
30   PLAY 1,0,0,0
40   WAIT 100
50   MUSIC 2,4,1,10          'Plus C4
60   PLAY 3,0,0,0
70   WAIT 100
80   MUSIC 3,2,1,10          'Plus C2
90   PLAY 7,0,0,0
100  WAIT 100
110  PLAY 5,0,0,0
120  WAIT 100
130  PLAY 1,0,0,0
140  WAIT 100
150  PLAY 0,0,0,0
```

It can also be fun to play with the *volume* as in this program:

```
10   'VOLUME EFFECTS
20   PLAY 1,0,0,0
30   FOR V=0 TO 15   'crescendo < effect
40   MUSIC 1,4,1,V
45   WAIT 10
50   NEXT
60   FOR V=12 TO 0 STEP -1 'diminuendo
70   MUSIC 1,4,1,V
75   WAIT 10
80   NEXT
90   PLAY 0,0,0,0
```

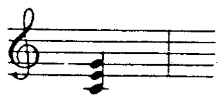
where the volume increases regularly (line 30 ff.) and then decreases (line 60 ff.). The musician writes this as a crescendo followed by a diminuendo:



Chords

Unison is fine to an extent, and one can achieve some marvellous effects; but you get a bit tired of it. There are other 'magic' note combinations which musicians call '*chords*'. They are the basis of *harmony*.

Even if you didn't pay much attention to your music lessons at school, many are aware of the basic 'perfect' chord of C-E-G:



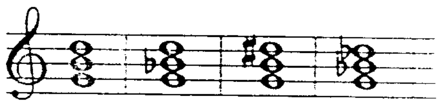
If the chord is made of notes of the same length, they are joined by a 'common tail' (here, for a crotchet). There is no technical difficulty in programming this chord on the Atmos:

```
10   'PERFECT CHORD C-G-E
20   MUSIC 1,3,1,10
30   MUSIC 2,3,5,10
40   MUSIC 3,3,8,10
50   PLAY 7,0,0,0
60   WAIT 100
70   PLAY 0,0,0,0
```

To use musical terminology more precisely, this is a *major* chord of three notes; all other major chords of three notes are obtained by *transposition*.

Major, minor, augmented, diminished

In harmony, musicians learn four variants for chords that have the same *root-name*: that of the lowest note. So all of these are G chords:



major minor augmented diminished

The first variant is the *minor* chord, which sounds 'sad' and is used to express sadness and emotion. Check this with the following example:

```
10   'G MINOR CHORD
20   MUSIC 1,3,8,10
30   MUSIC 2,3,11,10
40   MUSIC 3,4,3,10
50   PLAY 7,0,0,0
60   WAIT 100
70   PLAY 0,0,0,0
```

The 'diminished' and 'augmented' variants are frequently used for transition between major and minor chords. But that's enough of harmony (an art which is not really improvised!) for this time.

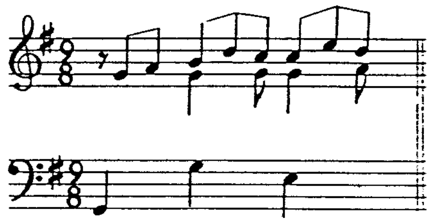
You can, by way of illustration, program a chord generator which will accept any note as the *fundamental*, and play the four chords of three notes: major, minor, augmented and diminished. In principle it is very simple. You just add to the given fundamental the number of semitones required to obtain the two other notes:

+3	and	+5	major chord
+2	and	+5	minor chord
+3	and	+6	augmented chord
+2	and	+5	diminished chord

There are other chords of 4, 5 or more notes. But since our hardware has only three channels, it can remain theoretical!

Questions of length...

It is easy to play three notes of the same length together on our computer.: three MUSIC instructions and one WAIT are enough. The technical problem is of a quite different order when you want to play a score such as this:



It's the first bar of a J.-S. Bach chorale, an extract from Cantata no. 147 known as 'Jesu, Joy of man's desiring'. One can achieve it using the MUSIC and WAIT instructions, but it is not simple, as we shall see.

Firstly we have to assign notes to the different channels. Thus the 'top' line of 8 notes can be played with MUSIC 3..., the four below with MUSIC 2..., and the lower 3 on the F clef with MUSIC 1...

This piece is in triple time, like a waltz; with this rhythm, there are three quavers per crotchet, and three crotchets per bar. The base unit of time can be given in the program by a variable T. So, the length of a quaver is obtained by WAIT T; a bar is 9 x T in length.

Rests...

At the start we have to play only the lower G during the time of the quaver rest on the upper stave; in other words, only channel 1 will be selected. Then (after the time T) channel 3 must be opened to play a G3 and the following A3. Finally, after a delay of a crotchet rest (3 x T) from the start, channel 2 is opened to play the first G3 in the second 'line' of notes.

Now that's quite a complex procedure. You can open and close channels with the PLAY command; however, we prefer to use a different technique. In effect, you can give a MUSIC command a volume parameter of 0: the C of MUSIC 1,3,1,*0* is quite inaudible!

Thus you can start the program with PLAY 7,0,0,0 to activate three channels, and 'play' notes with a nul volume when a channel is not required to be active.

Line by line...

On this basis the first bar of our piece can be programmed 'line by line'.

```
10   ' J-S BACH LINE BY LINE
15   T=30
20   '
40   MUSIC 1,1,1,0: MUSIC 2,1,1,0:
     MUSIC 3,1,1,0
50   PLAY 7,0,0,0
60   ' 1/9th of a bar by 1/9th
70   MUSIC 1,1,8,8: WAIT T
80   MUSIC 3,3,8,8,: WAIT T
90   MUSIC 3,3,10,8: WAIT T
100  MUSIC 1,2,8,8: MUSIC 2,3,8,8:
     MUSIC 3,3,12,8: WAIT T
110  MUSIC 3,4,3,8: WAIT T
120 MUSIC 2,3,8,8: MUSIC 4,3,1,8:WAIT T
130  MUSIC 1,2,5,8: MUSIC 2,3,8,8:
     MUSIC 3,4,1,8: WAIT T
140  MUSIC 3,4,5,8: WAIT T
150  MUSIC 2,3,10,8: MUSIC 3,4,3,8:
     WAIT T
160  PLAY 0,0,0,0
```

It works. But from the computer owner's point of view it is tedious. There are clear regular repetitions to be seen in successive lines, and it is obvious that it would be as absurd to transcribe the sixty or so bars of the piece in this way as it would be to list the first thousand numbers by typing in a thousand lines of Basic:

```
PRINT 1
PRINT 2
...
PRINT 1000
```

One solution is to put the parameters (channel, octave, etc.) in some DATA lines, and to READ the data as needed. However, at certain times you are simply playing one new note, at others two, or three...

It doesn't take long to reach the solution: place an indicator of the number of channels in the DATA lines. So line 120 would be coded as:

```
DATA 2,        2,3,8,8          3,4,1,8
  ↓               ↓                 ↓
2 x MUSIC     parameters       parameters
             of 1st MUSIC     of 2nd MUSIC
```

and executed thus:

```
     READ NB        'number of MUSIC
   → FOR I=1 TO NB
   ↑ READ C,O,V,N: MUSIC C,O,N,V
   ← NEXT
     WAIT T
```

The corresponding program, complete, would be:

```
10   'J-S BACH IN DATA
15   T=30
20   '
40   MUSIC 1,1,1,0: MUSIC 2,1,1,0:
     MUSIC 3,1,1,0
50   PLAY 7,0,0,0
60   '1/9th of a bar by 1/9th
70   READ NB
80   IF NB=0 THEN 999
90   FOR I=1 TO NB
100  READ C,O,N,V: MUSIC C,O,N,V
110  NEXT
120  WAIT T
130  GOTO 70
200  '
210  DATA 1,1,1,8,8
220  DATA 1,3,3,8,8
230  DATA 1,3,3,10,8
240  DATA 3,1,2,8,8,2,3,8,8,3,3,12,8
250  DATA 1,3,4,3,8
260  DATA 2,2,3,8,8,3,4,1,8
270  DATA 3,1,2,5,8,2,3,8,8,3,4,1,8
280  DATA 1,3,4,5,8
290  DATA 2,2,3,10,8,3,4,3,8
500  DATA 0             'end flag
999  PLAY 0,0,0,0       'stop
```

This is undoubtedly more of a computer program. There are more lines than in the earlier program, but it is much easier to set out the DATA. In particular the program and the data have been well separated.

... and

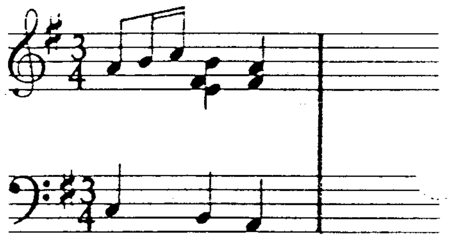
To the listener, this program, which should give the same tune as the previous one, in fact plays it rather more slowly. This is understandable, because the READ instruction and the loops need more time to execute. Could one then reduce the wait time T accordingly?

Unfortunately, no. It's a bit more serious than that.

The rhythm is not as regular as it should be. Indeed, it is sufficiently irregular for a musician to grimace. The reason is simple, and regrettably uncorrectable in Basic; in the loop (line 90 ff.) the program loops one, two or three times. The time taken therefore depends on the number of simultaneous notes.

This is unacceptable; when the pianist plays a chord, he isn't hamstrung by a delay which depends on the number of notes in the chord! There is a remedy, which is to 'equalise' the cycles by always playing three notes, one or two of which might simply be 'repeating' a note already playing. Good, except that the volume of DATA takes a serious leap upwards...

The catastrophe arrives with the sixteenth bar of the chorale:



The rhythm has changed.; we are still in triple time, but now with two quavers per crotchet. (You will have realised that this is the indication of the 3/4 marking at the start of the bar).

Even worse, there is a pair of semiquavers, two notes each lasting for 1/12th of a bar. If we want a common unit of length (variable T)it would be necessary to take the lowest common denominator of 1/9th and 1/12th as our unit of time, which is 1/36th of a bar. If we tried to use our present programming method, it would need a monumental quantity of DATA, most of it empty of any new information!

Try was we might, therefore, it is not possible to play good polyphonic music in Basic alone. The quantities of data become enormous, much of it devoid of real information, and it gets harder and harder to preserve the rhythm of the piece. It's all because Basic simply will not execute instantaneously.

What do we do? The solution will be explained next time, with the aid of interrupts and a little machine code.

T H E   B A C K   P A G E

POWERING A 3.5" DRIVE - Matthew Coates

I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am a fish I am

HA! HA! HA! Im so funny - by Matthew Dick (David's son) - now here's the real thing.

Get off that Oric Matthew! Now I've not enough space for Matthew C's article. Perhaps,anyway,it would be best to give you just an insight into it due to possible ramifications with various set-ups.

Here goes - Matthew has an old style CUMANA system i.e an interface and a stand-alone 3" drive (with built-in power supply). Initially these drives would not support Sedoric. This is overcome by changing a chip on the controller board. Matthew has since also bought a 3.5" drive and has successfully powered this from the 5V power supply on the 3" drive.

It should be noted that the 3.5" drive was just a straight 5V,whereas some are 5V and 12V, though from speaking to Matthew and Steve Hopps,this should not cause a problem,although I believe no more than 1 amp should be drawn from the 5V output.

Let me stress that I have no electrical knowledge,but from talks would summise that a 3.5" could run from a power supply used by the original MICRODISC and also by those who use a BYTE DRIVE power supply to power a 3" drive,but please check with the experts. The Opelco system could be a problem as some had one power supply and others two. You can check the output of the 7805 voltage regulator on these or check with Steve Hopps.

If you are about (or have recently) bought a Cumana interface (only) from Steve,then this will be of no interest as the interface will only power itself and the Atmos.

If you wish to see a parts list,write-up,and diagram then please send an S.A.E to OUM.

Matthew is quite happy with his new system - both drives seem quite happy with his arrangement To cap it all - it was a simple operation.

- Dave Dick

==

MORE ON WACCI AND THE NC100

After reading through the WACCI magazine I came across some Help-Lines. The one that particularly interested me was the guy who had and Amstrad Notepad NC100,which he has in fact now upgraded to the NC200. Chris Green told me that the crashes on the NC100 were caused by a design fault and did not neccessarily occur when the memory was nearly full. At one stage he lost 44K of data. He tells me that the memory cards solve this problem and can be obtained for around 30 pounds from the likes of EVESHAM MICROS and The SILICA SHOP - this is the price of a 64K card. You can get them for less than 30 pounds from MACRO (a cash and carry outlet) in Acton,North London.

Chris is willing to help NC100 users and can be reached on 0895 633641 (a Middlesex number), between 5 and 9 P.M.

Chris informed me that the club plan to have their first convention this year - we will let you know the note. It is also hoped that Chris will attend our ORIC MEET.

==

THE END OF 3" DISCs

I am informed that Maxell have stopped manufacturing 3" discs. My supplier (Dabs Press) has let me down and certain other suppliers have jacked up their prices. Meanwhile I am hopeful of another source of supply. If you use 3" discs then now is the time to stock up.

It is muted that an Italian firm may continue to manufacture them.

THAT'S YER LOT