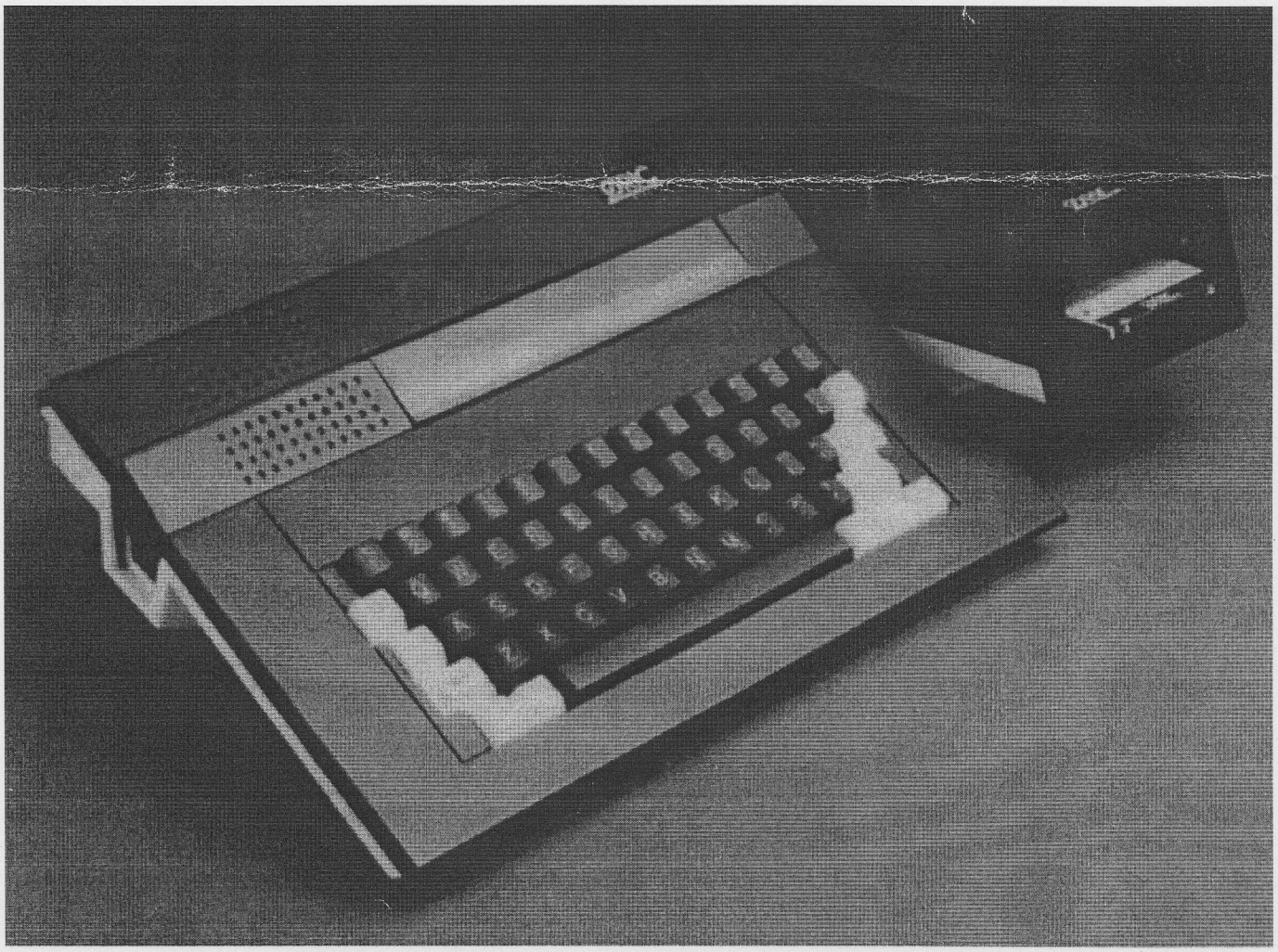
ORIC USER MONTHLY

with Alternative Micros

Number 81

May 1994

Keeping the Oric alive



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

THE EDITORIAL

HELLO AND WELCOME ONCE MORE,
TO AN EXTREMELY LATE ISSUE OF O.U.M.

Over the last 6 weeks I have hardly used the ORIC - not through lack of incentive,but through lack of time.
Yet more changes at work,numerous Discos,travelling and enjoying the sunshine,and of course the Bank Holidays. These have been the main reasons. Over the next 4 weeks I intend to clear the backlog - just in time for the MEET.
As this issue will not go out until the middle of May, and will be lacking in content, then I have had to re-schedule future issues. See below for further details.

I hope you all have an extremely pleasant Summer and look forward to seeing you at the MEET.

==========

THE INDEX

The Index will appear on the back page.

THE AYLESBURY MEET

The latest Oric MEET takes place on Saturday June 18th (Commencing 10 a.m) at RIVETS SPORTS AND SOCIAL CLUB, MANDEVILLE ROAD, AYLESBURY. It is on the AYLESBURY - STOKE MANDEVILLE - PRINCES RISBOROUGH - HIGH WYCOMBE Road.
It is approx. 1 mile from the town centre/bus station/railway station and about half a mile North of the Stoke Mandeville Hospital. The actual room in the Sports Club is called the Terrace Room.
You should remember from last year and hopefully have your maps. New attendees will be sent further details.
If you need any further directions then please telephone me on 0296 26050. DO NOT WRITE as I cannot guarantee to reply in time. I will be away for some of the week leading up to the MEET - so don't leave it until the last minute.
If you can't reach me ,then you can phone BOB TERRY on 0296 436280.

If you haven't got your tickets,then please send off for them now - we have to pay for the hire in advance and thus we expect that you help out. Tickets are 2 pound each, with raffle tickets available to all at 1 each. If you are not coming then please buy some raffle tickets. Latest addition to the prize list is a year's subscription to O.U.M, courtesy of Peter Bragg.
We are extremely hopeful of having members of both the DRAGON and AMSTRAD user groups at the MEET.

AFTER THE MEET

In the past I have held a BAR-B-Q etc. at my home after the MEET has finished.
This year I am unable to do so,as I will be hosting a 60's/70's DISCO at a local pub that evening. The music starts at 8 p.m and I have arranged with the landlord of said drinking establishment to have a BAR-B-Q at the pub. You may,if you wish, listen to the music or sit in the pleasant gardens.

If you wish to stay the night in Aylesbury,and haven't yet contacted me, then please do so. Currently I have 7 people sharing the settee and lounge floor!

RAFFLE TICKETS are being sent out with this issue.
Entry tickets for the MEET can be picked up on the day - I haven't run them off yet.

If you want a lift to the MEET, then please contact me immediately!

THE CONTACT LIST

The Contact List did not go out last month as promised - you will find it with this issue.

THE JUNE O.U.M

The June issue will also be late. It will be sent out on or about the 14th June to those not attending the MEET. Those attending can pick their copy up on the day - this will save postage on a hopefully enormous issue.
Articles for inclusion in the June issue should reach me by June 3rd. at the latest.

GOLF

I have now had chance to have a better look at a new golf game for the Atmos from Kieron Smith. It contains numerous features more generally seen on PC versions. A hell of a lot of work has been put into it. I am now in the process of transferring it from tape to disc and will then look into speeding the game up. I will keep you posted.

CYBOJUDGE

I have also had a chance to view CYBOJUDGE - an arcade game from Steve Marshall, which is still some way from completion. It has some novel features,which I will not divulge at this stage. More news as and when.

It is really good news that we are getting new programs being worked on.

ORIC ENTHUSIASTS

I recently recieved the following letter from Allan Whitaker of ORic ENTHUSIASTS:-

".. I am slightly said to say that I will not be re-newing my subscription for OUM this year. (Not that it influenced me in any way, it was interesting to note that the subscription renewal charge went up by 2 pound between issues 79 and 80. And there's me thinking that inflation was a thing of the past. Just goes to show that you cannot trust the Tories!)

Issue 80 is a very good issue and you must be congratulated on your efforts.

My job has recently been transferred from British aerospace to Computer Sciences Corporation,although I am still working at the same site, and with the extra responsibilities that I have been given I cannot realistically see myself having time to use the ATMOS anymore.

The 80th issue is a good milestone in which to bow out. I do wish you continued success for as long as you wish to continue with the magazine. Please convey my best wishes to all current subscribers and I will try to reach the Aylesbury Meeting in order to say a personal goodbye.

I will be making arrangements with Jon to take over the distribution of SEDORIC DOS,since he has the manual....."

BEST WISHES

Allan Whitaker

THE EDITOR REPLIES

I am sure that each and every one of our readers,both past and present,wish Allan and his family all the best for the future. I have also conveyed to him our sincere thanks for his tireless efforts over the years.

FOR SALE

Not an Oric user,but with an MCP40 in Oric-1 colours to sell is Frank James. Complete with pens and paper at an asking price of 30 pounds incl. post or 25 pounds if buyer collects.
Frank is at : 92 Manor Road, Brackley, Northants. NN13 6EE
Telephone number is: 0280 703694 - after 8 P.M

ORIC ACORN!

In the March issue of the "Acorn User" magazine is reference to a Demo on the cover disc. The software was titled "ArmOric'. The Acorns use the famous "ARM" chip, but where does Oric come into it?

THANKS TO ALI

Alistair Way has kindly agreed to let me put his FOOTBALL game on the next OUMDISC. He has also stated that royalties from all his games can go to OUM funds or a charity. Finances are tight at OUM and therefore I humbly accept. OUM will benefit by about 15 pounds when it goes out on the OUMDISC, and I will re-price all other titles by Alistair to make them even more tempting,though real gamesters will probably already have his GALACTOSMASH,GRENDEL and GRAND PRIX.
Alistair hopes to make the long haul from Scotland to the Aylesbury MEET.
Alistair is an IPSWICH TOWN fan - SAD!!

DR.RAY

DR.RAY will be bringing along his COMPLIER update to the MEET. He will also be 'sounding out' our views with regard to a Rom extension.

WHERE ARE THEY NOW?

James Groom has tracked down the author of such Oric favourites as: GRAVITOR and GREEN X TOAD. An interview is imminent - watch this space!

ANOTHER ONE BITES THE DUST!

After many years of enjoyment from his Oric system, Gavin Williams has sadly decided to sell up. I have contacted users who I thought would be interested in certain items and basically this is what remains:

MEGABASE (Registered) on 3" disc c/w manual - 4 pound (incl.post) or 3 pound if picked up at the MEET.
ACCOUNT BOOK/CALC - both registered and on 3" disc c/w manuals - price as above.
GAMES DISCS (approx. 30) - for re-formatting! - 5 for 4.50 or 3.50 if picked up at MEET.
ORDERS/CHEQUES direct to Dave Dick.

SOFTWARE SOUNDS - 4

The fourth 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

Machine code is to Basic what the oil painting is to the water-colour; you know the saying: it's harder work, but the results are much more beautiful...
It's certainly true if we are to play complex musical scores on our Atmos!

Thanks to our machine code programmers, we know how to create a subroutine that works every 1/100th of a second on an interrupt occurring. We also know that it is all but impossible to play pieces with the correct rhythm with 'several channels' in Basic.

So here we shall explore a new music 'driver', working on the interrupt in question. It is not only possible, but you can play music at the same time as a Basic program is running!

From interpreter to interpretation

We don't intend here to detail the art and practice of writing a Basic interpreter. Nonetheless, it is useful to understand the way in which Basic 'operates' its instructions.

You can imagine the interpreter as a genie: not necessarily Einstein, but certainly as capable as the one in the Thousand and One Nights! This good genie reads the program text like you or me, from beginning to end, line by line in order, from left to right for each line of the program.

On the way the genie seeks to identify 'keywords' which are part of its known vocabulary, and which are like orders that you (Aladdin) give to Basic (the genie of the lamp). You know the magic keywords, from GOSUB to FOR, from PEEK to POKE, from PLAY to MUSIC...

When it comes across one of these known instructions, the Basic genie looks for execution **parameters**; so with MUSIC it searches for four values: channel, octave, note and volume. It's only when this collection of information is available that the MUSIC command will be executed. To do this, the authors of the Basic interpreter have written a subroutine in machine code; complete technical information is in the manual.

As a preliminary four bytes must be set aside in memory for the passing of the parameters. This zone is called PARAMS, and starts at address #2E0. The procedure to interpret a note (in the musical sense of the term) is to place:

the channel number in PARAMS + 1
the octave in PARAMS + 3
the note in PARAMS + 5
the volume in PARAMS + 7

That done, it is only necessary to call the sub-routine MUSIC, at address #FC18. By way of example, here is a little program that plays middle C:

```
10      'Basic Version
20      MUSIC 1,4,1,8
30      PLAY 1,0,0,0
40      WAIT 50
50      PLAY 0,0,0,0
```

and here is how to obtain exactly the same result using POKE and CALL:

```
5       'Sub-routine Version
10      PARAMS=#2E0   'Parameter zone
11      MUSIK=#FC18   'S/routine address
12      POKE PARAMS+1, 1     'channel
13      POKE PARAMS+3, 4     'octave
14      POKE PARAMS+5, 1     'note
15      POKE PARAMS+7, 8     'volume
20      CALL MUSIK
30      PLAY 1,0,0,0
40      WAIT 50
50      PLAY 0,0,0,0
```

To proceed further in Basic is of no real interest. You write more text, and also take more risks. The Basic genie looks very carefully at the accuracy of your instructions!

If you tell him:

```
MUSIC 1,4,25,8
```

he protests, because there is no note '25', as follows:

```
'?ILLEGAL QUANTITY ERROR'
```

The Basic genie speaks English.

We can try to alter line 14 of our program:

```
14      POKE PARAMS+5, 25
```

Execution is a disappointment: the loudspeaker emits a vague click and nothing happens.

More serious is an error in the subroutine address. If line 11 were wrongly entered as:

```
11      MUSIK= FC19
```

you call address zero and the computer is struck by amnesia!

We are not trying to discourage you from using machine code and direct manipulation of the microprocessor. But you do have to keep in mind several responsibilities when you embark on it. And you must, as much as possible, write programs capable of protecting themselves against serious errors.

Interrupts - a metronome

The interrupt generated by the clock beating one hundred times a second inside the Atmos can obviously be used in playing notes. Instead of having MUSIC commands which the Basic interpreter must execute 'with the vagaries of the current', with a rhythm more or less according to the WAIT commands you use, you can easily imagine that several instructions in machine code can be executed at each interrupt 'pip' with one or more calls to the MUSIC subroutine.

The main advantage is the exceptionally precise rhythm that can be achieved; in fact, that of the quartz clock which activates the microprocessor. It's hard to imagine a stricter metronome! The other advantage appears if you try to produce two or three notes at one time, a thing you can do if you execute the MUSIC command on two or three channels in the space of one interrupt. The speed of machine code is such that the delay between, let us say, three calls of the MUSIC sub-routine is not heard by the human ear, even the keenest! In similar cases, the execution speed of the 'true' processor is better by a factor of 10 to 1 compared to the speed of the Basic interpreter. In computers, as in everything, an intermediary costs you (very) dear...

Planning is needed

Even the most frenzied pieces of music do not produce one hundred notes per second! More like ten.

In other words our hypothetical subroutine, executed one hundred times per second, must only **truly** play a note once over several 'pips', the other hundredths of a second constituting the **duration** of the notes the synthesiser is in the process of producing.

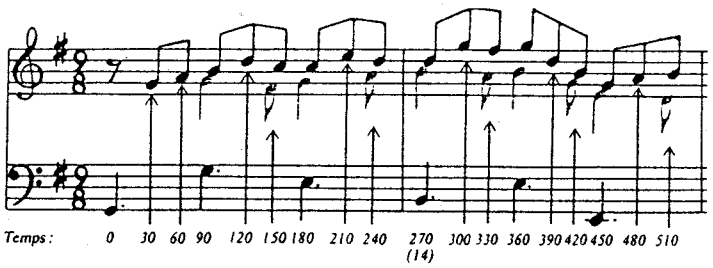
To give the phenomenon a more human scale, imagine a man sitting at his desk, with a sort of alarm clock which sounds every hour to attract his attention (the interrupt). In front of him, this man has a planner containing things to do, hour by hour. Sometimes, he has nothing to do... except cast an eye over the planner to realise it. From time to time a particular action must be carried out: our man will deal with the indicated task, and then return to his other work. Until the next ring of the alarm bell.

From the score to the route plan

No-one would accept such working conditions. But the computer, our docile slave, will, obedient and ready to please. Before programming the computer equivalent of our man working to the rhythm of his alarm clock, let's return for a moment to the music.

We have already used several 'models' to write music in the form of numbers which the computer can digest. Here is another, very simple in its principle: a score can be expressed like a **planner**...

Let's take again the initial two bars of 'Jesu, Joy of man's desiring' by Jean-Sebastian. Suppose (and it's only an example) that each beat lasts 9/10ths of a second: for notes and rests that means that a quaver lasts 30/100ths, a crotchet rest 90/100ths, etc.



AT TIME	YOU MUST
0	Play a lower G on channel 1 'embracing' channels 2 and 3 (1/3 crotchet rest)
30	Play an upper G on channel 3
60	Play an upper A on channel 3
90	Play an upper B on channel 3 **and** play an upper G on channel 2 **and** play a middle G on channel 1
120	Play a top D on channel 3

If one arbitrarily accepts that the piece starts at time 0, and that the time is to be counted in 1/100ths of a second, you can start to write the required sequence - see the table at the top right of this page.

This planning can be structured as a speed table where:

- each line is a separate action
- in each line, you have the 'time' where this action must have a place, and the usual parameters: channel, octave, note and volume.

The table (at the bottom of the page) is arranged in increasing time - take a moment to look at it.

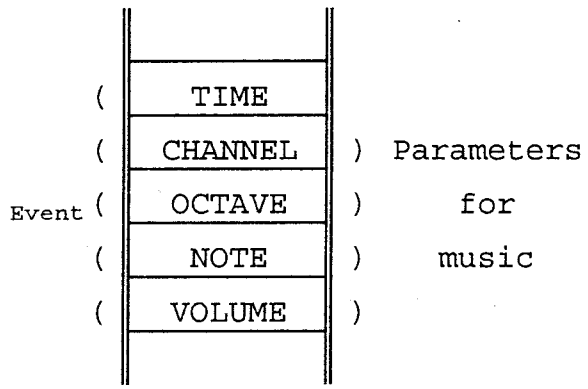
This representation is exactly equivalent to the score: you could take such a table and the rules and faithfully recreate the original score.

The piece of music thus coded will be placed in memory in blocks of 5 'boxes' per event; an event, that is a note or a rest, which must start to play at a certain 'time'; the note is

---

TIME	CH.	OCT	NOTE	VOLUME
0	1	2	8	8
0	2	1	1	0) Volume = 0
0	3	1	1	0) means silence
30	3	4	8	8
60	3	4	10	8
Three (90	3	4	12	8
notes (90	2	4	8	8
together (90	1	3	8	8
120	3	5	3	8
...				

described in the same order as for the MUSIC command:



Installing the piece in an area of memory is very easy in Basic using the DATA command. You could, for example, proceed as in the following subroutine:

```
500     '
510     ' Copy of the score
520     '
530     ' Needs a HIMEM
540     '
550     A=#8800 ' Zone start address
560     READ T   ' "time"
570     IF T=-1 THEN GOTO 700
575     POKE A, T
580     READ C          ' channel
590     POKE A+1, C
600     READ O          ' octave
610     POKE A+2, O
630     READ N          ' note
640     POKE A+3, N
650     READ V          ' volume
660     POKE A+4, V
670     A=A+5
680     GOTO 560
700     POKE A,0: POKE A+1,0 'end marker
710     RETURN
1000    DATA 0,1,2,8,8
1010    DATA 0,2,1,1,0
1020    DATA 0,3,1,1,0
1030    DATA 30,3,4,8,8
1040    DATA 60,3,4,10,8
1050    DATA 90,3,4,12,8
1060    DATA 90,2,4,8,8
1070    DATA 90,1,3,8,8
1080    DATA 120,3,5,3,8
9999    DATA -1
```

Of course the program is far from optimised so as to be clear. It works by simply copying the DATA items 5 by 5 into memory starting at our arbitrary #8800.

The driver at interrupt time

From our machine code programmers, we know how to connect to the Atmos interrupt a subroutine that will execute every 1/100th of a second, leaving the Atmos' own routines (such as reading the keyboard) untouched. To do this, just copy the following hex values into memory starting at address #BFE0:

```
48 8A 48 98 48 20 00 00 68 A8 68 AA
68 40 78 60 58 60
```

For a subroutine at a particular address to execute, we must write the following initialisation sequence:

```
CALL #BFEE
```
stop interrupts

```
DOKE #BFE6, ADDR
```
create the link with the subroutine

```
POKE #24A, #4C
DOKE #24B, #BFE0
```
reroute the interrupt

```
CALL #BFF0
```
enable interrupts

We still have to write a few instructions in machine code to execute the 'events', as set out in our planner table.

The 'driver' subroutine has several pieces of information for its own use, so it knows where it is. The suggested routine needs only 4 bytes - its **context**.

The END flaag enables the driver to know whether it has completed its task, i.e. it has played the last note.

Memory location T is the subroutine's own clock. On each interrupt the driver increments T by one, and compares it to the 'required time' of the next event.

This next event is designated by an address which **points** to the table we have already created. This address occupies two bytes, and is called P for pointer. The 6502 microprocessor of the Atmos makes access easy when the address (pointer) is in page zero, the initial 256 bytes of memory. We have found that we can use #FA and #FB for

this without interfering with Basic.

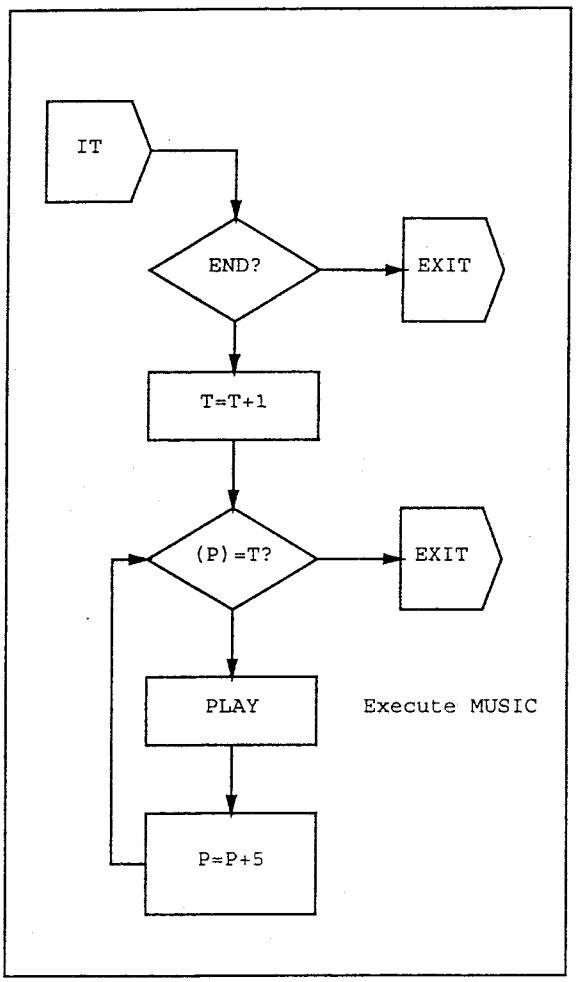


Figure 1

Figure 1 illustrates how the driver functions. Every 1/100th of a second we enter the subroutine (named IT), which tests whether it has something to do by checking the END flaag. If the piece has ended, we exit the subroutine immediately.

If not, the driver increments its own clock (T) by one, and compares its current 'time' to that of the next event. If it is not yet time to execute it, the subroutine terminates here for this hundredth of a second.

On the other hand, if it is time for the event to occur, the driver calls a subroutine which plays the required note by calling, in its turn, MUSIC.

There is one exception: it is this same subroutine which, in the appropriate case, detects in the event table a channel value of 0 which indicates a rest.

Here is the complete driver subroutine, (*presented 'à la Nick Haworth', that is to say as used for the music in 'Columns'...*) It is an Assembler listing:

```
8000   A5    FA          LDA 'END'
8002   C9    01          CMP #1
8004   F0    79          BEQ EXIT
8006   E6    FB          INC 'T'

.RUN
8008   A0    00          LDY #0
800A   A5    FB          LDA 'T'
800C   D1    FC          CMP('P'),Y
800E   D0    6F          BNE EXIT
8010   20    30    80    JSR MUS
8013   18                CLC
8014   A5    FC          LDA 'P'
8016   69    05          ADC #5
8018   85    FC          STA 'P'
801A   A5    FD          LDA 'P'
801C   69    00          ADC #0
801E   85    FD          STA 'P'
8020   4C    08    80    JMP RUN

.MUS
8030   C8                INY
8031   B1    FC          LDA ('P'),Y
8033   C9    00          CMP #0
8035   F0    19          BEQ FMUS
8037   8D    E1    02    STA PARAM+1
803A   C8                INY
803B   B1    FC          LDA ('P'),Y
803D   8D    E3    02    STA PARAM+3
8040   C8                INY
8041   B1    FC          LDA ('P'),Y
8043   8D    E5    02    STA PARAM+5
8046   C8                INY
8047   B1    FC          LDA ('P'),Y
8049   8D    E7    02    STA PARAM+7
804C   20    60    80    JSR MUS1
804F   60                RTS

.FMUS
8050   A9    01          LDA #1
8052   85    FA          STA END
8054   60                RTS

.MUS1
8060   A9    00          LDA #0
8062   8D    E2    02    STA PARAM+2
8065   8D    E4    02    STA PARAM+4
8068   8D    E6    02    STA PARAM+6
806B   8D    E8    02    STA PARAM+8
806E   4C    18    FC    JMP MUSIC

.EXIT
807F   60                RTS
```

Let's complete the links between Basic and the machine code. We have only to put in place the mechanism to start the driver.

Before grafting onto the interrupt, it is necessary (at least) to position the END flaag, for example by:

```
POKE #FA, 1
```

in such a way that the driver is 'neutralised' until one wishes to play the numbered score.

Let us pass over the construction of the score table, an example of which among an endless number we have already given (and why not a program to automatically compose it?).

Let's place ourselves at the moment when execution starts. It is obviously necessary to indicate to the driver the start address of the table, i.e. to initialise the pointer P; if the score starts at #8800, we can use:

```
DOKE #FC, #8800
```

The attentive reader will be asking about our variable T: one byte for our driver clock? In effeect a byte takes a maximum value of 255; put another way, we are unable to go beyond 2.55 seconds. Unable?

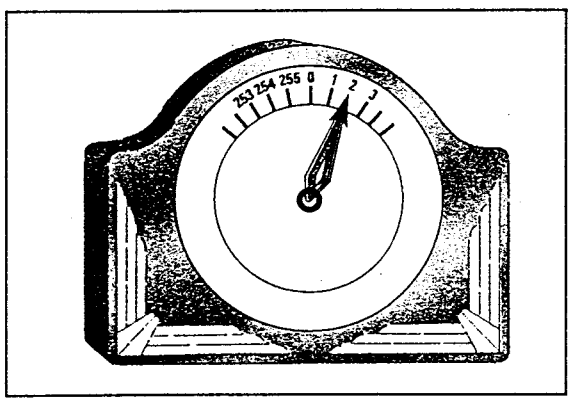
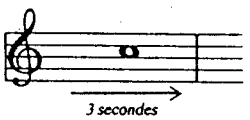


Figure 2 - The clock of our music driver is like a face with 256 gradations

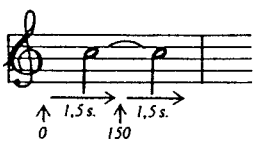
Not at all. Computer programmes often have recourse to arithmetic which is 'modulo so-and-so'; in our case, you can view the clock as a face graduated from 0 to 255. 0 follows 255; those who remained entirely resistant to their maths lessons will still admit that immediately after midnight less one minute there comes zero hour!

This trick makes the machine code programming simpler. It has no impact on the coding save for this - it is necessary to subtract 256 from the successive times of events so that, for example, the initial note of the second bar of the Bach piece will come to 270-256 = 14, etc.

When a note lasts more than 2.55 seconds, as for example with a semibreve lasting 3 seconds:



you simply break it down into two tied minims of 1.5 seconds each:



Continuity is assured by the human ear, which will not pick up the replaying of the same note by the synthesiser.

And now for music

To complete our initialisation of the driver, we just give T the value prior to 0, so

```
POKE #FB, 255
```

so that the driver discovers that the time is 0 on starting to operate.

Here is a complete example sequence to start the driver:

```
1  HIMEM #7FFF
10 POKE #FA, 1          ' END=1
20 CALL #BFEE           ' stop interrupts
30 DOKE #BFE6, #8000 ' subroutine
                            address
40 POKE #24A, #4C       ' JMP interrupts
50 DOKE #24B, #BFE0
60 CALL #BFF0          'restart interrupts
70 GOSUB 500           'initialise table
80 DOKE #FC, #8800     ' P, table address
90 POKE #FB, 255       ' T (minus 1)
95 PLAY 7,0,0,0        ' channels 1+2+3
100 POKE #FA, 0        ' END=0
110 END                                 ...→
```

```
500     '
510     ' Copy of the score
520     '
530     ' Needs a HIMEM    .
540     '
550     A=#8800 ' Zone start address
560     READ T   ' "time"
570     ..etc, as earlier in the article.
```

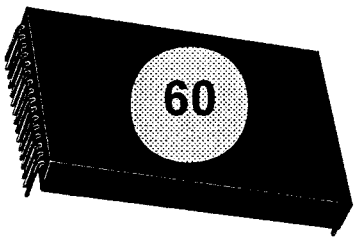
The instruction in line 100 starts the music; the motor is like a horse, you only need to loosen the bridle.

And moreover, a miracle! The music continues even though the Basic program has finnished, and the Ready prompt is displayed...

And Basic really has finnished; if the piece lasts long enough, you could even edit the Basic program whilst it plays.

This illusion of simultaneous music, obtained by using interrupts, is typical of real-time systems where the computer often changes its hat to create the impression that there are several processors all working at the same time.

RAMBLING IN THE ROM



Rambling on...

Or not, as the case may be! After that marathon episode of musical knowledge, the Rom disassembly is suspended for this month (do I hear cries of relief?), and I'm left with two-thirds of a page to fill. Er....

MICROWAVES

Don't CALL us, Oric

A non-useful but very entertaining comand on the Oric-1 is CALL 59150

J. Howarth,
Hebden Bridge,
West Yorkshire.

For added satisfaction, power up your Oric, enter CALL 59150 and press Control T three times - Ed.

Firstly, no it's not me who sent in the 'Microwave' on the right. This is another undated clipping I have - can anyone help by dating it? Secondly, anyone writing a game should take a close look at the 'Software Sounds' article. It was this system that Nick used to program the music for Columns, so we know it works. Incidentally, no-one has yet managed to name the title of Tune A on Columns - the prize is a refund of the cost of the game!

Bad news about Maxell stopping making 3" discs. I have sufficient supplies to last the rest of this year so far as C.E.O. discs and software are concerned, but I can't imagine no-one will step in to supply all those Amstrad CPC and PCW users out there. At least we can attach a 3½" drive very easily for about £20! See you next month...

Jon Haworth

Machine Code for the Oric Atmos (Part 35)

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. Recent issues have looked at masking and shifting instructions.

As I have mentioned before, these instructions have a number of uses. They are useful for setting up hardware and can also be used for arithmetic, multiplication and division. They can also be used in graphic displays. It might be useful to look at one way, I have made use of them in the past.

Shifty Graphics

Some years ago I wanted to transfer some graphics from the BBC micro to the Oric. I managed to dump a graphic image from the BBC micro into the Oric's data memory. However, it certainly could not be put straight into the Oric's HIRES display, because it was incompatable with the Oric's display requirements and had to be converted first. For that, you need to know what the differences are between the Oric and BBC graphic displays ?

The BBC display is essentially character based and each byte in the BBC display mode that I was using, represented eight pixels on the display. Now, in the Oric Atmos HIRES screen, each byte represents six pixels only. The other two bits in each screen byte are used to control the Oric's display effects. So, if the BBC graphic was put straight into Oric's HIRES display, the result would be that the first six BBC pixels would set Oric pixels correctly, but the remaining two would create havoc in Oric's display controls. That would apply to each and every byte on the Oric's screen. Not a pretty sight ! Obviously the BBC graphic pixels had to be re-arranged from eight bit groups, into six bit groups for the Oric.

A complete description of how to convert BBC graphic screens to Oric HIRES is beyond the scope of an article like this and has no interest to users of other machines, anyway. However, a quick look at how the problem was tackled, could be useful to most users and may also provide some useful insight into programming technique, as well as the rotation instructions.

So how does one start out to write a piece of software to convert a graphic from one machine to another ? The best way is to reduce the conversion operation to the smallest possible unit and work it out for that unit only. Once that has been achieved, it can be wrapped up into a program routine which can then be repeated for the whole graphic image, reading it in bit by bit and converting it and putting it into Oric's HIRES screen.

The smallest unit common to the Oric and BBC computers is one horizontal line of 24 pixels. The maths experts amongst us, will have worked out that if the Oric display byte contains six pixels and the BBC byte contains eight, three BBC graphic bytes are equal to four Oric HIRES display bytes, because they both produce 24 pixels. The rest of us can now put our shoes and socks back on !! Now, back to the rotate instructions. How can they help ?

You may recall that the rotate instruction pushed the Carry bit into one end of a byte, shifting all the eight bits sideways, causing the bit at the other end to "fall out" and be copied into the Carry. This means that the next rotate instruction will use that bit, saved in the Carry and will push it back into the byte at the other end.

So by repeating the same rotate instruction, you can make the entire byte rotate through the Carry, bit by bit. This rotate operation can go in either direction, using ROL for (ROtate Left), or ROR for (ROtate Right).

However, it isn't necessary to restrict the operation to a single byte. If you rotate two bytes in succession, the bit put into the Carry from the first byte will be transferred to the second byte, by the second rotate instruction. In effect you will have linked the two bytes with the two instructions and by repeating the two instruction operation, you can transfer the contents of one byte to another literally, bit by bit.

Three instructions will link up three bytes in the same way, in fact there really is no limit to the number of bytes that can be rotated as a single group in this way.

Back to our BBC graphic, held in the Oric's data memory. We can link three BBC graphic bytes plus one Oric HIRES byte together in this way and rotate the BBC pixels, bit by bit into the Oric HIRES byte, using a simple routine. Repeat the routine six times to put six BBC pixels into the Oric HIRES byte. Get the next Oric HIRES byte and repeat the same operation six times to transfer the next six pixels into the second HIRES byte. The same can be done to the third and fourth Oric HIRES bytes, by which time the entire collection of twenty four pixels from the three BBC graphic bytes will have been transferred in the correct order to the four Oric HIRES display bytes.

Of course the two Oric control bits will have been rotated out of the Oric display bytes by the above operations, but they can be easily re-installed with a couple of masking instructions. I used an AND #3F (0011 1111) to clear bits 6 and 7 of each HIRES byte, followed by ORA #40 (0100 0000), which set bit 6 to "1" in each HIRES byte (required for pixel data display). The two masking instructions were tacked onto the end of the rotate routine.

First of all I tried all of this, using the first three addresses of the BBC graphic image stored in the Oric data memory and the first four addresses on the Oric HIRES screen. A line of 24 pixels is quite small, but it was sufficent to check that the operation worked correctly.

Once that was achieved, I was then able to exchange the fixed addresses of the graphic and display bytes for a more flexible set of addresses set up in a parameter block instead. The parameter block made it easy to change the addresses used by the conversion routine to access the graphic image storage area and the whole HIRES display RAM area. The conversion routine could now be called repeatedly in a sequence, that would convert the BBC graphic image to the Oric HIRES display, in sets of 24 pixels at a time.

Yes, I have over simplified the description and left a lot to imagination, for reasons already explained. There are other solutions, some are better !

This series is mainly aimed at those who are not too familiar with machine code/assembly language programmming. Inevitably there will be those who say "Gawd, I didn't understand a word of that !!". Dont worry, it's not essential to be able to convert graphics from the BBC Micro. The main idea was to illustrate how the rotate instructions can be used and to give some idea how to tackle a software project. If you are still unsure about instructions for rotating and masking, have another look at the last few issues. Try a few experiments, using simple values for a start. Doing that on the Oric and seeing results conveys far more than just reading about it. So switch on and go !! .....See you next month.

MARSHALL'S MUSIC numero 7

Hello again folks! I think it's time for a look at MIDI. What it is and how it can be used. Here we are stepping out of my 'field of expertise', as I am involved with acoustic musical instruments, but I hope I can give you some idea of what it is.

MIDI stands for Musical Instrument Digital Interface. That should give you some idea of what MIDI is all about. Like the interface you may use to connect a disc drive to your computer, or your joystick interface, MIDI is something that allows you to connect one device to another. As MIDI is an industry standard, any MIDI devices can be connected together, (using a cable with a 5 pin DIN plug at either end.)
If you have a MIDI keyboard, you can plug it into a box called an expander. This will give you a whole new set of sounds that you can play on your keyboard. You could plug the same box into a MIDI guitar, or MIDI wind synthesier. The wonderful thing is that whatever instrument you play, you can have access to an unlimited number of sounds which previously only keyboard players had access to. Most new keyboards also come in the form of an expander. This is usually called a module. It has all the facilities of the keyboard minus the actual keyboard. You can plug several expanders/modules into one MIDI keyboard to give a wonderful array of sounds and possibilities.

Because MIDI is digital you can use a computer to manipulate the information. How useful the computer is is largely dependant on the software used. The Program will usually have some form of sequencer. This is something that stores the information of sounds, their pitches and duration so that you can play tunes.

Music expansion cards for more powerful computers than the Oric often include a sampler. A sampler will record a sound in digital form. This sound can then be used as one of the sounds on your keyboard. A few years back someone sampled various dog barks and used these sounds to play some famous classical tunes which some of you may remember. The piano is one of the hardest sounds to synthesize, but excellent reproductions of its sound have been achieved by using sampling.

So, is MIDI any good? Well... yes and no. It all depends on what equipment you have and what you're trying to acheive. You need at least two devices to make use of MIDI. Although MIDI devices are compatible, some features are not because of manufacturers including things to it. Pitch bend, for instance, is not always available.
It can help people who aren't musically proficient to produce good results. It can give musicians greater flexibility. The thing to remember is that it is a tool and so depends upon user as to its effectiveness.

If you're into music, give it a try.

THE BACK PAGE

CF2

I have now tracked down a supplier of Amsoft 3" discs. WAVE of 1 Buccleuch Street, Barrow In Furness, Cumbria. LA14 1SR can supply them at 14.50 per box of ten (includes postage). Their telephone number is 0229 870000

WACCI

I recentlty mentioned the March 1994 issue of WACCI (Amstrad magazine). It turns out that this was their April Fool joke and was in fact the April 1993 issue. I have now recieved issue 77 (April '94). Will OUM outlive WACCI!

A GENIUS IDEA!

An idea from James Groom:-

Stand for the local elections representing the Oric Party. After cruising to victory,move on to General Elections - soon Oric Party forms government. Get ORIC,TANSOFT and other Oric related companies resurrected. Generous tax incentives for those owning or buying Orics. Soon everyone owns Orics and Jon Haworth becomes Chancellor. Re-name country Utopia and everyone lives happily ever after!

Who said politics is complicated?

PLASTERED AGAIN!

A get well soon message to Jon Haworth,who recently broke an ankle and is well plastered. The good news is that his wife is getting plenty of practice in mowing the lawn and topping up Jon's Lager glass!

MESSAGE TO ARNT ERIK ISAKSEN

Regarding your letter of April 21st. - is it meant as a serious letter or is it an attempt at humour? An early reply would be appreciated.

RE - TONY CLARK

Aylesbury OUM reader and entrepeneur Tony Clark is no longer trading. Tony can still be contacted at his home address with regard to other 8 bit machines.

THE INDEX

==========

AND FINALLY!

And finally - some late news about the MEET. David Wilkin sends his apologies,but Austria calls. Meanwhile Dr.Ray confirms his attendance,as do Frank Bolton and Luis.

MEATBALLS!

In the next issue an article from Bob Terry concerning our trip to Frank's, nad how 3 tins of meatballs and a tin of spaghetti saved the day in setting up an Oric system.