RHETORIC

issue #7
march 2000

BRITAINS BEST (AND WORST) SELLING ORIC MAGAZINE OF 2000

LIFE IN THE FAST LANE...

PLUS ARTICLES NEWS
LETTERS SOFTWARE
AND MUCH MUCH MORE

I COULDN'T FIND A PICTURE OF A FERRARI AT SHORT NOTICE, SO YOU'VE GOT TO MAKE DO WITH THE NEXT BEST THING...

CONTENTS

We're in issue #7 now, and we're rolling. New subscribers are coming all the time, and I would like to offer a warm welcome to new readers that have come our way, either from Dave, or Steve, or the website, or any other way. I hope you find the RHETORIC group helpful in giving your Oric a new lease of life.

Things have been a bit better this issue. Last issue was a bit of a panic. I thought I'd save myself a few quid and get one of these 'remanufactured' inkjet cartridges for my HP deskjet printer. Halfway through printing out the last issue, the cartridge clogged up, and I could not get the bloody thing flowing again. If I'd have known that you couldn't buy decent new generic cartridges, I wouldn't have bought the printer. Anyway – I've sent the thing back and got a replacement – and it's just as bad as the first. Anyway – to cut a long story short, I went and bought a new color cartridge and its fine.

Talking of rip-offs, Windows 2000 should be out very soon. I was offered a copy of it a few weeks ago for a fiver (naughty naughty). What I would like to know is… will it crash 2000 times a day instead of 1998 times a day?

In the last issue, I made a request to get hold of some of the old Your Computer listings for the Oric. Steve has sent me some software on disk, which I'm mowing through at the moment – though I'm not sure if they are the full machine code games that were printed. There are some worthwhile titles though, and I don't know whether or not I should print them, or get them onto the next Rhetoric disk… Any ideas anyone?

Here's a quick request. Fabrice is looking for an inlay for Zorgon's Revenge. Can anyone help?

OK – Heres the rundown of this issue….

THE MAILBAG

With The Mail Bag Lady

The Muso Cometh....

Hi Ed,

Got Rhetoric this morning. I must say the cover is a bit dodgy. I find pictures of men wearing Y-fronts somewhat disturbing, especially when they have a matching vest tucked in.

Have you played that record ? You'll hear it don't sound too feminine. That's because she is one of the first legally recognised sex change .. er... people.

Sort of agree with Fabrices comments. It is hard to cover such an argument though and I think it does need covering. I don't think anything was rephrased though (?)

John Hurley wrote to me (I sent you bits of his letter). Apparently he had sent disks of translated programs for Dave to print in OUM. I've fished them out.

The progs were on Randos and had no file extension. This is OK in Randos, but when I converted the programs to Sedoric they just wouldn't load. I had to go back to the Randos disk and rename the files to have .Com extensions and then convert them to Sedoric again. The disks worked fine on the Oric but the PC wouldn't read them (using Readdsk). So I had to recopy the disk until I found a disk the PC liked. Strange !

Anyway, attached is a disk image with the progs. I haven't checked them fully, so check them out before printing the listings. Should fill a few pages in the mag.

Muso

Hi Muso

Mmm... last month's cover... The less said about it the better really. You'll be glad to see that normal service has been resumed this issue. I haven't had a listen to that LP yet... not too sure that I want to hear a man turned into a woman and sounding like Barry White... I remember Alison Moyet in the 80s...that was bad enough...

I've had a look at the programs that you sent me, and I'm sure there's some stuff we can use. Yhatzee is a pretty good version, and one or two others are OK. I may try compiling them into machine code if I get some time. I come across a machine code Breakout game in Your Computer magazine which I may print too – It's not too long, and look quite good.

Let me know what you think...

Simon

Hi Chaos,
Got a letter from (w) John Hurley. I'm replying to him myself but some stuff might be fit for the mag.

[...] I also believe to have read that you may know something about the 'Amstrad' computer ? I acquired such a machine a little while back and find it very clever. The system consisted of CPC 6128, DMP 2000 printer, CTM 644 monitor ands many discs and tapes. Now one of my HELPS is, have you ever come across the word processor program called 'PROTEXT' from ARNOR Ltd?. I do not have any instructions with this disc but have worked out 90% of the commands but am lacking in being able to move/delete/set markers for passages of text.'

{Muso says} Thanks for your letter John. Feel free to write to Simon who is in charge of the mag now, (I'm just the software guy). My advice for Amstrads is to chuck the horrible plastic bit with the keyboard and plug the Oric in to the monitor, (after making ht necessary adjustments). Oh, and keep the disk drive for the Atmos. Seriously though, the 6128 is OK but try and get your hands on the CPC + which are better and nicer looking !

I have this program for the Atari ST but never really used it as there are much better program for the ST. I emailed Brian Watson (of the IEBA). He owns Protext/ Prodata and gave this reply when I asked if he could help.

'Yes. I have Protext manuals at 5 pounds each, including postage. send cheque/ PO to Brain Watson, 39 High Street, Sutton, ELY, Cambs,. CB6 2RA'

Brian also has PC versions of Prodata @ £22. Not much of a bargain but if you need to transfer programs from an ST or something then it could save a lot of bother.

[More from the letter]
'I did think the TOOLKIT' instruction sheet in issue No. 5 was a good idea. I was thinking along these lines myself.The problem as I see it is, although the person who wrote the Toolkit knew

at he or she was talking about, the likes of me eed more information in how to make the most out of it in straight forward English common sense. I love this kind of thing more than arcade games. Any program that serves a purpose or allows you to think a way out of a situation is tops for me.'

{Muso says} Yes it's a good idea. Personally I thought that everyone that bought a copy got the instruction sheet for this. Of course now you get so many programs available on the internet with no help what-so-ever. Once again proves that we are the ones trying to support the Oric user !

If you have ANY idea for contributions PLEASE send them in. We are struggling to fill the mag because only a small group are contributing. we don't get the regular articles of Peter Bragg and Jon Haworth (although they did say they would like to carry on - any chance Peter & Jon ?)
Obed is particularly useful for designing characters for games so might not quite be your cup of tea, but it is well worth trying all Mr. Bristow's 'ED' range (I always said he was an edcase !)

'During 1999 I completely rehashed, 'BACKGAMMON' from IJK because of the three versions I had seen, not one played true. Version 1 played the same dice sequence every time and did not finish correctly. Version 2. you always won as the Oric would never bear off displaying 'I can not go'. Version 3. The Oric cheated during bearing off by moving counters no matter what dice were thrown. The version I now use

THE MAIL BAG CONTINUED.....

has all of these anomalies corrected.'

[Muso says] Great stuff - send it in please ! I think the original is V1 which is the old problem of the 'random' sequence being the same on boot up. Brian Kidd (I think) did some amendments to get change this but when I tried it, it made no difference. Perhaps he is responsible for the other versions. I must say the best Oric version is Dormeres 'Orical Backgammon'. The graphics are much, much better and the game plays properly. I've never managed to convert this prog to disk - can anyone help ???

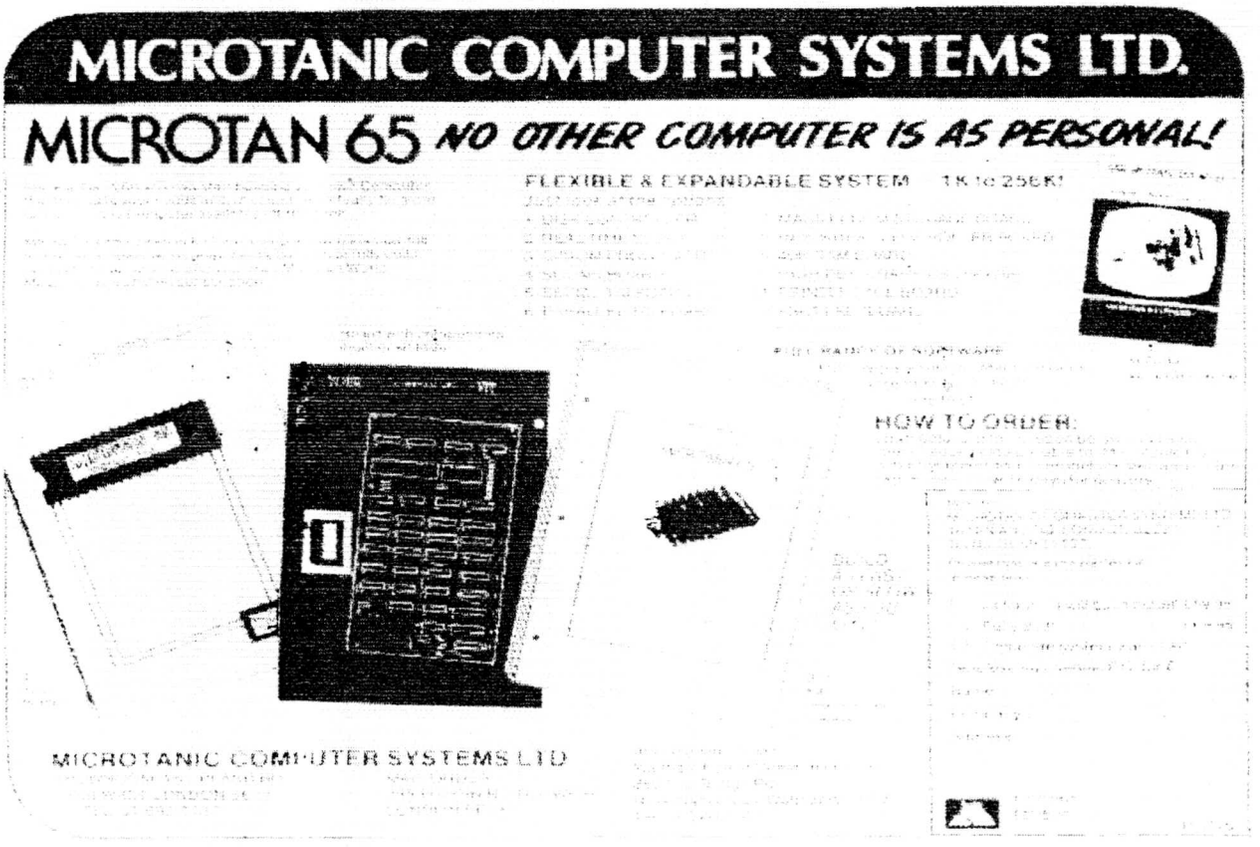
John goes on to say he has done similar work to the PD astrology and Masterpaint.

[Muso says} Sounds excellent ! please send stuff like this so we can get it on the disk.

Knackered after typing all that!
Steve Muso

Chaos replies...

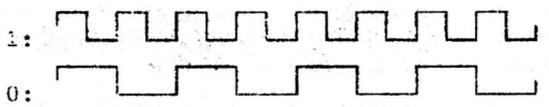
Thanks Steve. I echo Steve's words to send in any programs any of you may have.



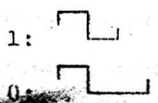
Thanks to Steve for this great nostalgia advertisement. Anyone got any more? Hope it printed out OK.

Life in the fast lane
... or speeding tape I/O like a Ferrari

Some of you have heard about my attempt to reach high speeds when loading programs with Oric's tape input. Well, I'm happy to announce I've finally succeeded to reach the incredible rate of 22050 baud (well, at some time, I was beginning to think it is impossible, I hope you will excuse this enthusiastic adjective). More importantly perhaps, now I know why my previous attempts (like the first Oric-CD) led to failures, so I would like to share these experiences with you.
First of all, this text is mostly dealing with encoding schemes and transfer speeds, so it might be nice to recall some definitions : what a speed expressed in Baud means and what a speed expressed in bps (bits per second) is. Any dictionary will tell you that the Baud speed is derived from the smallest bit time, that is to say, the smallest time needed to emit a bit. Let's take familiar examples to see how this metric can be misleading. As you know, when the Oric is saving a program in slow mode, its speed is said to be 300 baud. If you have connected an oscilloscope to your Oric or if you disassembled the tape routine in ROM, you also know that '1' bits and '0' bits are encoded with waves on the tape : a '1' bit is emitted as a series of eight periods of a square-wave 2400 Hz signal, while a 0 bit is emitted as a series of four periods of a square-wave 1200 Hz signal.



Thus a '1' bit and a '0' bit last the same : 8/2400 = 4/1200 = 1/300th of a second, i.e. 300 baud.
Now, if we look at the FAST mode on the Oric, we see that a '1' bit is emitted as a single period of a 2400Hz signal, while a '0' bit is emitted as a half a period of a 2400Hz signal, followed by another half-period of a 1200Hz signal, like on the following picture:

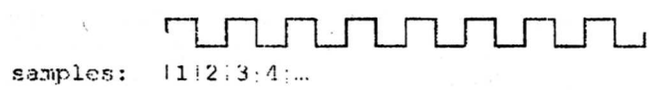


So, a '0' bit lasts 50% more than a '1' bit, but with the definition of the Baud, only the smallest bit time (1/2400th of a second) is used to compute the speed, so it is rightly 2400 baud, even if this number sounds like a commercial argument...
Now, sometimes you will hear about speeds expressed in bps (bits/s or b/s) or in Bps (Bytes/s). These numbers are intended to measure real throughputs but they might also be far from the truth : if we talk about a 10 Mb/s Ethernet or a 56.6 Kb/s modem, don't assume you will transfer a file at these nominal speeds. If you try to express the Oric speed with this unit, you might say that if N is the number of bits per second and half of them (N/2) are '1' bits while the other half are '0' bits, then the '1' bits will last (N/2)/2400 seconds and the '0' bits will last 50% longer : (N/2)/2400x1.5. Adding the two lengths gives one second so

(N/2+1.5xN/2)/2400=1 or N=4800/2.5=1920 bits.

Note that you wouldn't correctly compute this 1920 b/s speed if you take the sum of the speed of a series of '1' bits (2400 b/s) and of the speed of a series of '0' bits (1600 b/s), and then divide by 2. (This is exactly the same thing when you climb hills at 10 km/h, run the easy direction at 30 km/h and finally find you only get an average of 15 km/h).
Last but not least, when you save a byte on tape, forget a byte is 8 bits : there are additional bits providing synchronisation and error detection. The Oric routines are generous with these extra bits: a start bit, a parity bit and 3 stop bits (in fact 3.5). All this considered, the real average throughput is around 1200 "useful" bits per second, that is, only half of the 2400 baud figure.

Anyway, with the first Oric-CD, I was trying to raise the speed from 2400 baud to 22050 baud. Why this number? Well, you know an Audio-CD is a digital media, storing the amplitude level of the sound as samples : 44100 values (samples) of the amplitude (for each of the left and right channels) are stored per second. With such a digital format, the highest frequency you may store on a CD is 22050 Hz, because you will need at least two samples to represent a single 22050 Hz period:

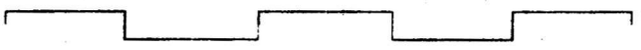


Thus the highest possible rate for an Oric program, encoded like we have seen above, is 22050 baud, and this naturally became my **Quest**.
So, with the first Oric-CD, I tried to use the Oric encoding scheme with a '1' bit being represented by a 22050Hz period (i.e. two CD samples) and a '0' bit being represented by half a 22050Hz period and half a 11025Hz period (i.e. three CD samples). The ROM routines are too slow to deal with such short periods, so I wrote faster routines. Unfortunately, this didn't work... this is because the VIA of the Oric is not directly connected to the tape connector: there's an input circuitry that filters high frequencies. I will try to explain this phenomenon...

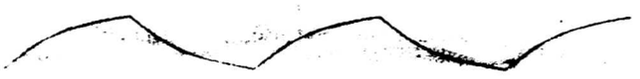
The aim of the tape input circuitry of the Oric is to suppress high frequencies : since the Oric has to read 2400 Hz and 1200 Hz periods, the hardware designers of Tangerine decided to enhance the reliability by suppressing eventual noises (these noises are typically high frequencies). But this circuitry has a side effect: the sharpness of the edges of the signal is eaten, so that if the input signal is perfectly square like this:

Then the signal at the signal at the end of the circuitry is something like this :

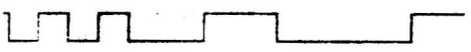
So, if we try to make the Oric read higher frequency signals and feed it this :



Then it will have to decipher something like :

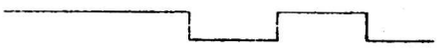


And if we further raise the input frequency like this :



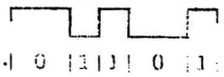
Then the resulting signal won't have a full amplitude :

(the more you raise the frequency, the more flat the resulting signal will be).
How the Oric will interpret this ? The VIA's pin CB1 detects edge transitions: I'm not sure about the exact implementation of the detection process but I guess it is based on a threshold value. So, with such a mangled signal, if the threshold value were the intermediate 2.5V (it is not, more on this later), then you can see many transitions disappear, and the Oric will think the input signal is this :



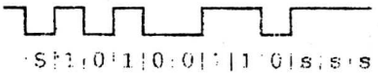
A related experience which you may remember is the connection the video SYNC output to the TAPE IN. This was suggested some time ago by Alexios Chouchoulas and I tried it with the hope to give the Oric a way to count video lines and so to produce some interesting video effects. I was surprised to detect only one positive edge every frame, instead of one per line. Of course, this comes from the same reason: as the horizontal retrace pulse is only 4 µs long (equivalent to a 125 kHz signal), it is completely absorbed by the tape input circuitry. Anyway, detecting the vertical refresh pulse is still interesting: this allows to produce flicker-free animations and I hope future Oric developers will use similar techniques.

Back to the main subject... as raising frequency is quickly limited by the tape input filter, I decided to use a different encoding, more efficient than the Oric's one. This is mandatory if I want to achieve the same speed with lower frequencies. There are several ways to get bits coded with less samples. Stephen Meachen for example has used both high and low levels of the signal to encode bits: thus a '1' bit is half a 2400Hz period, and a '0' bit is half a 1200Hz period. This way, 01101 becomes



Such a scheme brings a 4800 baud speed, or 3200 b/s.

Indeed, why should we use two levels (a complete period) to code a single bit ? Like Stephen, I was at that time sure that better speeds would be achieved by using only one level per bit, but why should '0' bits be twice as long as '1' bits ? We are in a digital world, why not simply coding 0 with a 0V level and 1 with a 5V level ? This way, without raising the frequency, we are able to send the same information in 2/5 the time (one half for 1 bits, one third for 0 bits). In fact, what we re-invent now is digital serial transmission: remember the Oric uses one start bit, 8 data bits, a parity bit and 3 or 4 stop bits ? This means that the now-digital signal for 'A' would be

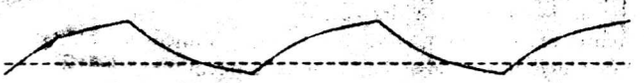


This is really similar to RS232 transmission: the only difference being the levels used. RS232 standard codes 0 with a +12V, and 1 with a -12V. The 6502 uses TTL levels instead : 0V for a 0, 5V for a 1 (as we have seen, these levels are further converted to 0-1V for the tape connector). But when we think that specialized serial chips like the ACIA 6551 or the UART 8250 also use TTL levels, what we are trying to do now is "emulating a serial chip" with a simple I/O chip and software. Of course, this sounds so appealing that I jumped in the bandwagon. Producing such a serial wave with the 6522 is easy: the PB7 output can be controlled with timer 1, so that we just have to compute the cumulative lengths of levels in advance. But reading it is another story: we can either measure the lengths of levels or sample the signal at regular intervals. The first approach seemed rather difficult to achieve in real time because one measured length gives a variable number of bits and also, if last data bits and parity are 1's, they are not easily differentiated from the stop bits. So, I chose the second method; moreover, sampling is the method used in hardware serial chips. Sampling has its own problems, though: as the only synchronization information we have is provided through the start bit, we need to sample the signal with a good timing. Shannon says we need a sampling frequency of at least twice the signal's frequency in order to properly rebuild the original signal. In our case, this is not enough. If we get two successive samples with different values, where is the truth ? Hardware chips like the 6551 or the 8250 use a x16 frequency so that it is easy to devise the original signal with a simple average of the samples. Using such a technique in software with a poor 1 MHz 6502 is not possible, except for very low baud rates, so I tried another technique. One advantage we have is the synchronization brought by the leading start bit. So, if we can be enough timely precise so as to sample the signal in the expected middle of each bit, the game is won. The 6522 capabilities are not used at best in Oric's firmware: this VIA is able to produce interrupts on a variety of circumstances. For example, we can program it so that it raises an interrupt when the leading negative edge of the start bit arrives. This is our synchronization point. Then we can program timer 1 with about one half the theoretical length of a bit so that we get an interrupt at the middle of the start bit, allowing us to sample the signal at that time and to verify it is still at level 0 (otherwise, we have just experienced a "false start bit"). After having programmed the initial "half-bit" temporization, we will also have loaded the timer latches with a "full bit length" so that it will be automatically loaded again and again after the intial countdown. This way, we have one timer interrupt in the middle of every bit, so we can sample the signal with a good timing. One last detail so that you have the complete picture... Sampling the tape input signal is not straightforward: the CB1 pin of the VIA only allows to detect a programmed level transition (i.e. either a front/positive edge or a trailing/negative edge). So, once the leading negative edge has occurred, we have to program the VIA so that it detects the next positive edge, and each time a timer interrupt occurs, we have to check whether the programmed transition has occurred and in this case, we reprogram the VIA so that it detects a transition in the opposite direction.

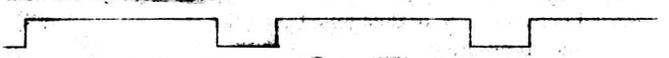
Anyway, I wrote such an interrupt intensive routine and was able to read RS232-like serial data on Oric's tape port. Moreover, by removing extra bits in the serial format, I was able to achieve more than 3 times the real throughput of the standard Oric routines without raising the frequency: as I recalled at the beginning of this article, the real throughput of the Oric is about 150 bytes per second; without raising the 2400 Hz frequency (note that one 1/2400 second period now allows to code 2 bits instead of one) and by using only 10 bits per byte

(1 start, 8 data, 1 stop, no parity), serial transmission achieves 480 bytes per second. You may say it is not fair because I have no way to detect bit errors with this format, but how useful is a parity bit ? A parity bit can detect a single bit error but it can not help in correcting it, so a good checksum algorithm can be almost as efficient with much less bandwidth.

4800 baud achieved through software is not bad, but then of course I was interested in achieving higher rates: not 44100 bauds since we have seen that a frequency of 22050 Hz is absorbed by the tape input filter, but I had the hope to reach 22050 bauds (2 CD samples per bit) or at least 14700 bauds (3 CD samples per bit), the later having the additional advantage to be very close to 14400 bauds which is commonly used by modems. Unfortunately, the interrupt routine I had developed was not able to cope with a 22050 baud rate: too much overhead is carried by the timer operation. So I wrote a new routine based on the number of cycles used by 6502 instructions. Again, it didn't work. So I decided to be more progressive and start with lower rates. Doubling the original Oric frequency didn't work so easily: I had to tune the exact point at which the sampling was done (that is to say, it didn't work when I tried to sample at the expected middle of the bits, but it worked when the sampling point was at about 3/4). This means a rate of 9600 bauds, which is common in the RS232 world, and this already is more than 6 times Oric speed). Then I tried 11025 bauds but did not succeed in reading it in a reliable way (what I mean with reliable is the ability to read large amount of exhaustive data without any error. All my tests were conducted with a 16KB regular sequence of every byte pattern). So, at this rate, I always had errors with some patterns and examining which patterns were erroneous quickly lead me to the logical reason: the length of low levels (0V) and high levels (5V) are not those expected... More precisely, a sequence of n '1' bits is always longer than a sequence of n '0' bits. This brings us back to the threshold value I was mentionning when talking about the edge detection process of the VIA. My guess is that the threshold value is around 0.6 V (this is a usual threshold value for diods). As the tape input circuitry converts the 0-1V input range signal to a 0-5V range with a (surely) linear law, the threshold value is not centered. So, if after the filtering/range-converting stage the signal looks like this (the horizontal dotted line is supposed to be located at 0.6V):



then the signal read is:



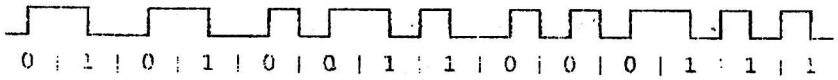
With these conditions, you understand that raising the frequency tends to diminish the low level lengths until they completely disappear ! Also, if we combine this effect with the first one (the levels aren't enough long to stabilize at 0 or 5V), it becomes nearly impossible to predict where the middle of the bits lie.

That's too bad, I was quite happy with this "software serial device" but unfortunately I can't raise the speed as much as I would like... Anyway, understanding the reasons of a failure is always of benefit, and now the initial leading idea (moving from a wave-based format to a digital format) shows its implication in a clearer way: the format used by the Oric carries its own "clock", that is to say, it is easy to synchronize with every bit since we can consider a bit always starts with a positive edge. Au contraire, the RS232 format carries less information since the clock start is only given once for a sequence of 10 bits. This is perfectly acceptable when the edges of the signal are not damaged by a circuitry such as the Oric's. So, I finally decided to give up with the RS232-like transmission on the tape connector and to investigate ways that carry a clock with the signal. By the way, the work done with the "software serial device" is not lost at all , I have a great use for it: the technique and the routines developed can be easily adapted to other pins of the VIA (which don't suffer of a filtering circuitry), and the best candidates for this "software serial device" are the STROBE and ACK pins of the parallel port because they can be used without interfering with the keyboard operation (and vice-versa). This means that you should soon hear about a way to connect a modem to your Oric parallel port with just two transistors for the level adaptation.

This problem of carrying clock with the signal was ringing a bell to me, so I first had a look at popular ways to handle this clock, in the world of communication and in the world of magnetic media. Let's see for example how Ethernet transmit data using the Manchester encoding scheme: there is always a level transition in the middle of each bit, so it is easy to keep synchronised when receiving the signal. A '0' bit is coded like this :

and a '1' bit like this:

For example, here is how the sequence 0101001100011 would be encoded:

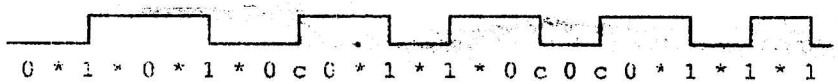


This encoding scheme is not better than Tangerine's one: the only advantage is that '0' bits have the same length as '1' bits, so it has a better throughput than Oric's fast mode. But reading data is harder than with Oric's format: both positive and negative edge have to be detected and there's not a one-to-one relation between level transitions and bits, so the time between transitions has to be measured. Last but not least, the maximum frequency of the signal is equal to the baud rate (this can be seen with sequences of '0' bits, or with sequences of '1' bits), so raising the baud rate is rapidly limited by the frequency.

Then, I had a look at how data is encoded on magnetic media. This is interesting because the problematic is similar: we are limited by frequency and magnetic media (e.g. floppies) are limited by their density, which means bits cannot be too close each together. When you know that magnetic information is a matter of polarisation (only two spins are used, let's call them 'up' and 'down'), this density limitation translates to how small a polarisation can be, which is equivalent to how much frequently you can reverse the spin.

So, let's have a look at the most basic encoding scheme : FM (Frequency Modulation) was ruling in the seventies (I'm not talking music !). FM encodes '0' bits as 'no transition' and '1' bits as 'transition'. Additionally, a clock transition takes place between two successive data bits. For example, the previous bit sequence (010100110001 11) is coded like this:

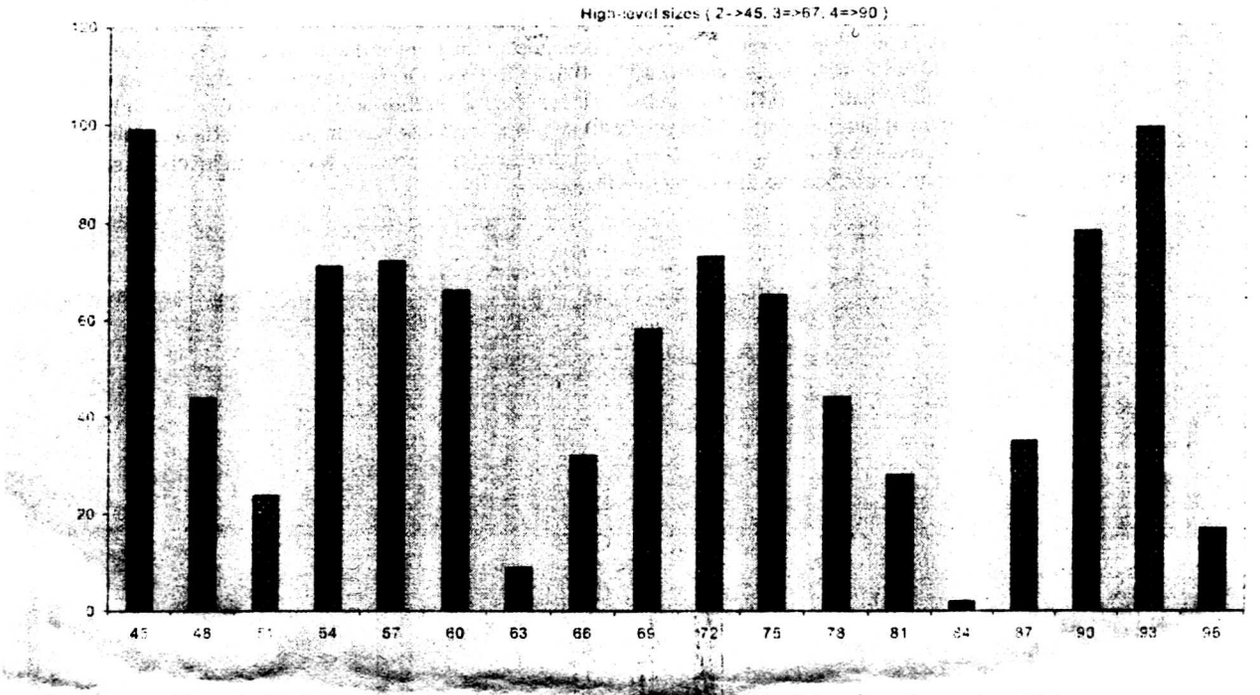
Well, as you can see, this encoding scheme brings the same advantages and drawbacks as the Manchester coding. So, let's go a step further and have a look at an improvement of the FM encoding that allowed to double the density: MFM. Like FM, MFM translates a '1' bit in a level transition and a '0' bit in no level transition. A clock bit is also inserted between two data bits but this clock bit is not always translated to a level transition: in fact, it is only translated to a level transition if it is inserted between two '0' data bits. Using the same bit sequence example, this gives (clock bits are denoted 'c' if they translate in a transition and '*' otherwise):



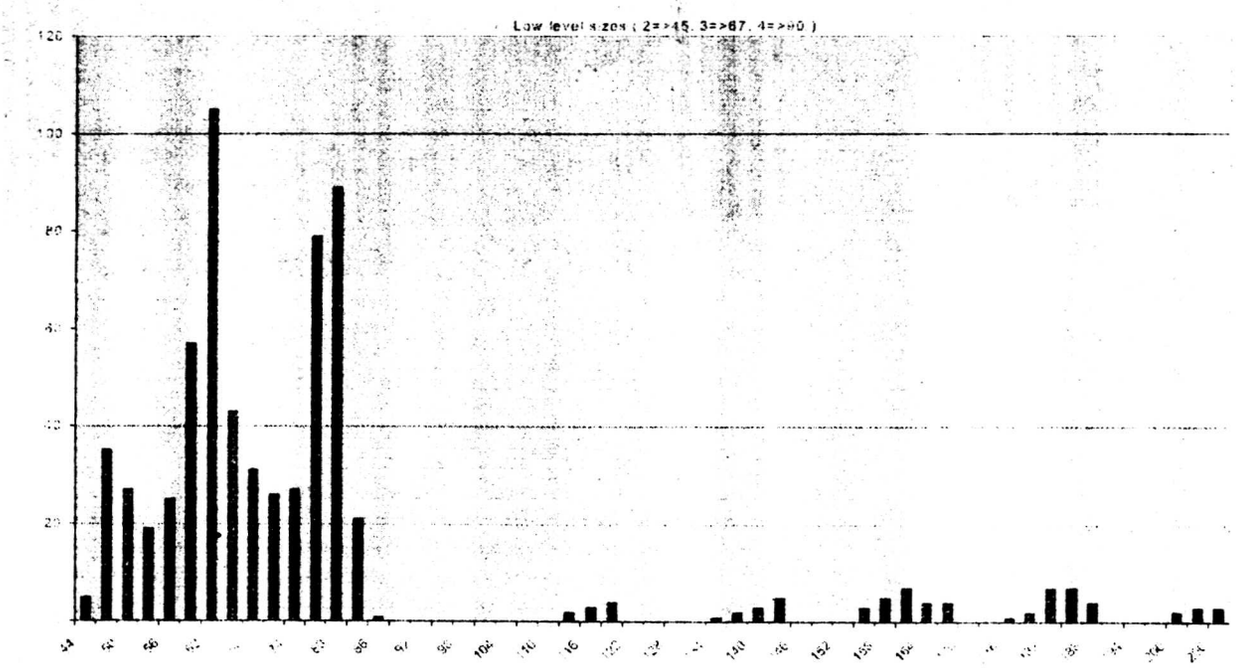
Wow ! Do you see the difference between this and FM ? The smallest levels are now double the size, this is why MFM brings a x2 improvement. In terms of floppies, this allowed to double the floppy capacity by multiplying the writing speed by 2. What about using it for Oric tapes ? Once again, the idea to emulate a dedicated floppy controller chip like the one found in Oric's Microdisc (WD1793), using just a 6502 and a simple I/O chip, that was very appealing...

Let's compare MFM with Tangerine's simple encoding: at the same highest frequency of 2400 Hz, MFM transmits data with a speed of 4800 bps vs 1920 bps. If we don't use start, stop and parity bits, this can translate in a real throughput of 600 bytes per second vs 150 with Tangerine's routines... Same frequency, 4 times the speed ! Even better than the RS232-like experience and we still carry a clock. This also means that the initial goal of 22050 baud is reached with a maximum frequency of only 11025 Hz. That's for the advantages, now to be fair, there's also a small weakness : the encoding scheme is more complicated than the previous ones, it is not so much a problem when emitting data but mostly of interest when reading. As I have already said, the CB1 pin is good at detecting positive and negative edges, so the best way to read data seems to measure the lengths of levels between edges. As you can see from the previous wave example, there are three possible lengths, respectively 2, 3 and 4 units long. Unfortunately, there is not a one-to-one relation between levels and bits, a level transition means we have found either a '1' bit or a clock ('c') transition. '0' bits and additional clocks ('*') are not detected. So, we can decode the incoming data using two pieces of information: what sort of transition was encountered last time ('1' bit or 'c' bit) and how long is the next level. When we have encountered a '1' bit, it can be checked from the example that measuring a 2-units long level means we have received '*1', a 3-units long level means '*0c' and 4-units long level means '*0*1'. When we have encountered a 'c' transition (this is the case when measuring a 3-units long level after a '1' bit transition), measuring a 2-units long level means '0c', a 3-units long level means '0*1' and a 4-units long level is impossible (this never happens). So, I wrote a decoder and then I wanted to know how fast this encoding scheme could go. Reaching 22050 baud with this scheme recorded on a CD means that one CD sample is used for one unit, that is to say the lengths of the levels are 2, 3 or 4 samples. At this speed, the Oric only has 45 µs for its decoding work between two successive

transitions. Also, I wanted to know if the filtering circuitry let a 11025 Hz frequency pass through. So, I produced a random wave whose levels were all 2, 3 or 4 samples long, and wrote a small statistic routine on the Oric whose job was to measure the lengths of levels and to build a statistic table of those lengths. The expected result, of course, is an histogram showing three separate sets of value. But as you can see on the following graph showing the histogram for the measured high levels, the three sets of data aren't cleanly separated (lengths are expressed in µs, and the histogram shows how many times a length was measured).

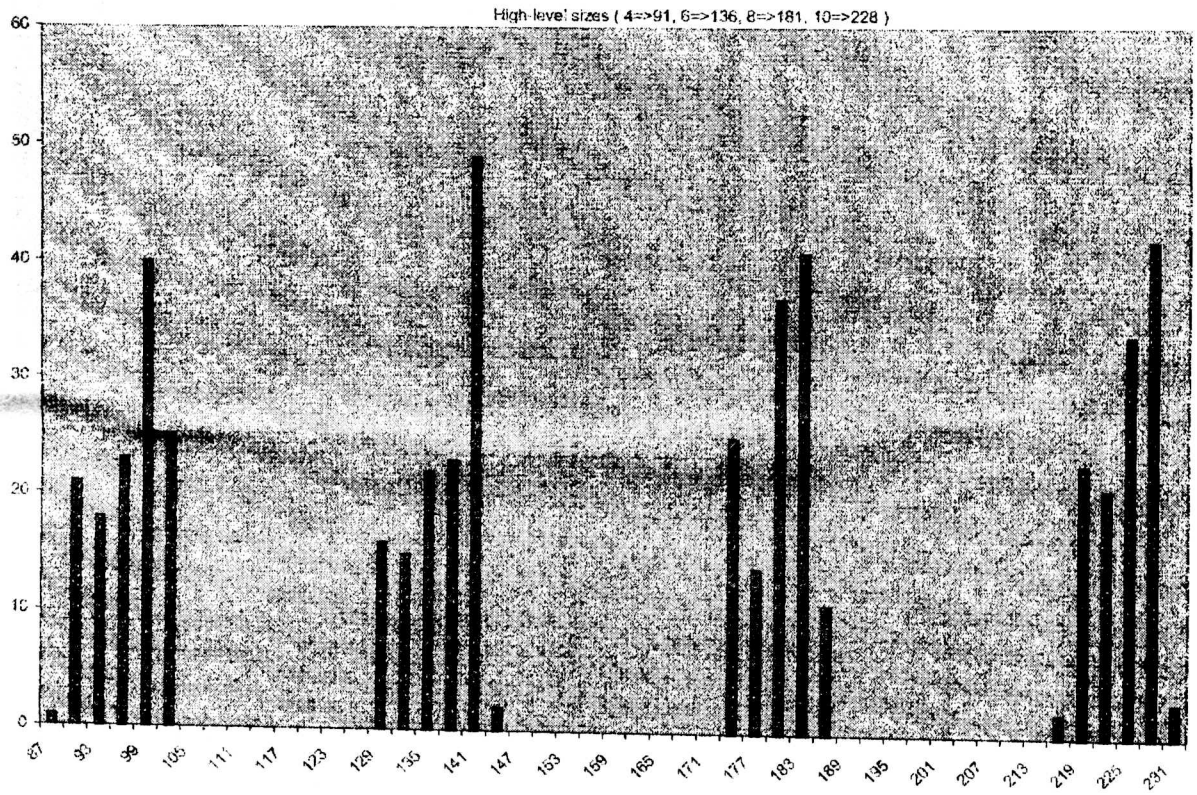


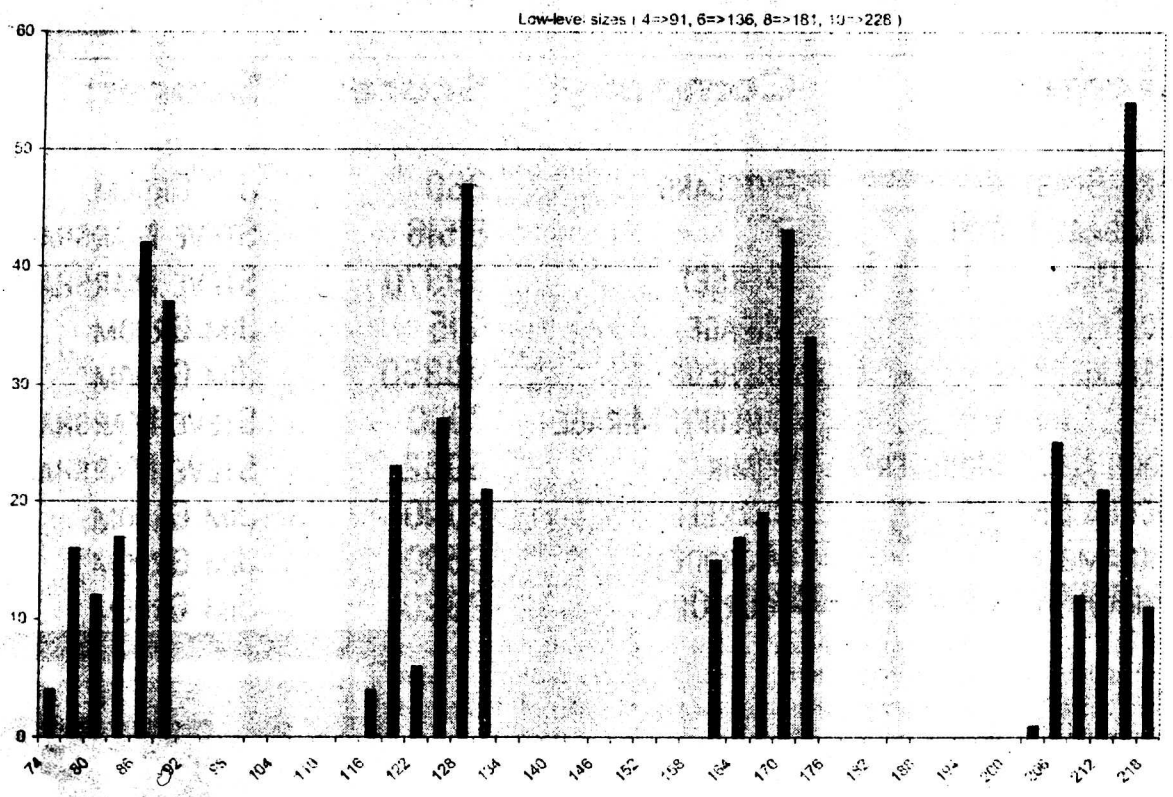
So, how would we decide if a measured length of 63 is 2 or 3 samples long ? Anyway, the results of this test were even worse on the low levels (remember they are shorter), as you can see on the following graph. .



What this graph tells us is that the filtering circuitry swallows the 11025 Hz frequency (the smaller length of 2 samples for a level), so the longer measured lengths correspond to 3, 5, 7 or more concatenated levels... No wonder I couldn't transmit reliably at 11025 baud with the RS232-like scheme !

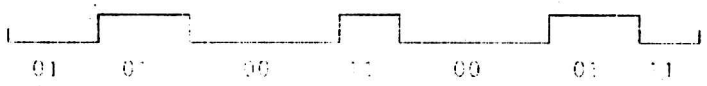
That's too bad, once again, I had a nice encoding scheme, and the filtering circuitry is here to annoy us. So how far can we go with this filter ? We can imagine longer lengths of levels with the same encoding with the hope the signal will pass through the filter. For example, if the levels are 4, 6 and 8 samples long, the maximum frequency will only be 5512.5 Hz but of course the speed will also only be 11025 baud. Or we may use levels of 3, 5 and 7 samples long: these ones are not proportional so the speed depends on the transmitted bit patterns. The baud rate would be 14700 baud, with a throughput ranging from 12600 to 14700 bps. Or, we may use levels of 3, 4 and 5 samples long if we are able to make the difference between these lengths. In this case, the baud rate would be 17640 baud with a throughput ranging from 14700 to 17640 bps. Anyway, that was enough with dreams, and I needed to know what was really possible with this filter. So, I produced new random waves with levels being 4, 6, 8 and 10 samples long and obtained the following two histograms.





This time, the histograms are what I expected: you can see 4 sets of separated values, which means that when you measure some length, you can tell which length it was supposed to be. In other words, waves built with levels of 4, 6, 8 and 10 samples are still readable once they have been smoothed by the filter. But, you can imagine from the previous results it wouldn't be possible to separate closer lengths like 4, 5 and 6. And we want a reliable encoding scheme, not one that will load one time out of 10. So, good bye 22050 baud, the Oric tape circuitry won't allow you to go in...

Hey, wait a minute, it's not finished ! These irregular speed variants of the MFM encoding were in turn raising new ideas: take for example the MFM-like encoding with levels of lengths 4, 5 and 6. A small level carries only 1 bit, so the throughput is only 11025 bps. But a large level (of length 6) carries 2 bits, so the throughput is 14700 bps, and a 5 samples long level carries 1.5 bits, so its throughput is 13230 bps. That was the idea: why should we use identical throughput for every length ? And what about carrying 2 bits with every level ? This should make the decoding easier than MFM, and as the baud rate is taken from the smallest bit time, we can benefit from this definition... So, this new encoding scheme was quickly defined as follows: 2 bits are grouped together and transmitted as one unique level being either 4, 6, 8 or 10 samples long. For example, if we decide that '00' is represented by a 10-samples long level, '10' by a 8-samples long level, '01' by a 6-samples long level and '11' by a 4-samples long level, then our previous bit sequence example (0101001100011) is translated as below:



Count yourself: the smallest bit time happens when levels of 4 samples are used, and as 2 bits are transmitted, this means that at last we have reached 22050 baud, with a maximum frequency of only 5512.5 Hz which is not completely swallowed by the input filter. Of course, you may argue I am cheating because the average throughput is only 12[illegible]0 bps, but hey, just compare with Oric's 1200 bps throughput: that's 10 times faster ! Just imagine your favorite 30KB program loading in only 20s ! Well, you should be able to judge by yourself soon, as I intend to provide tools that will allow people to produce their own program collection on a CD... Cheers to all,

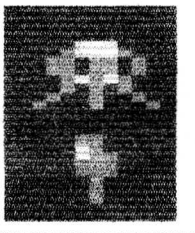
Fabrice

HALL OF FAME

Game:	Company:	Score:	Scorer:
Bar Blaster	Byteland	260	Jim Groom
Cabbage Patch	?	61516	Steve Marshall
Dig Dog	Taskset	39370	Steve Marshall
Don't Panic	Mirage	1215	Jim Groom
Hellion, The	Orpheus	48350	Jim Groom
Insect Insanity	Tansoft/Mirage	7900	Steve Marshall
Loch Ness Monster	Romik	5952	Steve Marshall
Scuba Dive	Durrell	3040	Jim Groom
Skramble	Microdeal	6500	Jim Groom
Tyrant	Your Oric	2820	Jim Groom

The address to send your scores to is:

7 Debdale Avenue,
Lyppard Woodgreen
Warndon Villages,
Worcester.
WR4 0RP or email: james.groom@virgin.net



WHATS NEW ON THE WEB

Your monthly guide to new and updated webpages.

Club Europe Oric Homepage **http://ourworld.compuserve.com/homepages/laurentch/**
Small update giving details of the Paris Oric Meet on 19th February.

The Reluctant Dragon **http://freespace.virgin.net/james.groom/homepage.htm**
Updated on January 15th. Added new sections - "What software was promised but never released?" and "What happened to.....?".

7 Debdale Avenue,
Lyppard Woodgreen
Warndon Villages,
Worcester.
WR4 0RP or email: james.groom@virgin.net

News...news...news...news...news...news...news

Jean Boileau's Crabes Problem.

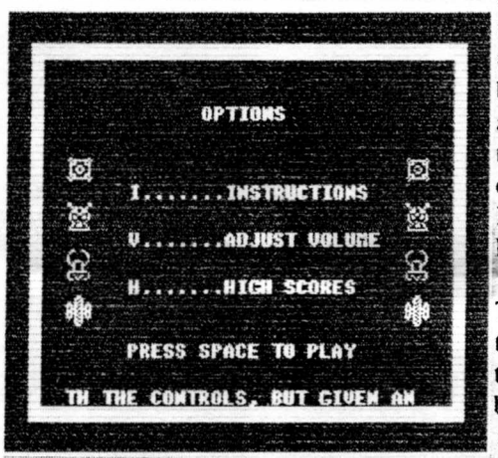
Jean has tracked down the publication in which this game appeared. This program was published in a book called "Oric, jeux d'actions".

TYRANT GAME ZONE

Tyrant By Your Oric
Written by Gary Munro, Ian Marshall & John Cumming
Released in 1987

Your Oric was a well produced Oric fanzine of the mid-eighties. It only ran for about eight bi-monthly issues and released its only piece of software, Tyrant, shortly before it closed. If you have an original tape copy, it is therefore very rare, so don't lose it! The game was supplied in an ordinary cassette box with an inlay produced on a home printer. Upon loading, you are presented with a reasonable screen shot of an insectoid creature in a maze. The first thing I noticed was the GARIAN.BAV logo which identified the authors as the same people who wrote Krillys and The Hellion. It therefore seems likely that this program would have been released on a professional label, were it not for the demise of Oric in the UK. After the title screen has been displayed for a few seconds, the main option screen appears. Here you have the choice to read the instructions, adjust the volume, view the High Scores or play the game. The option screen is neat, but sparse. A line of animated characters from the game is displayed on either side of the screen - thats you at the top. A scrolly message at the bottom of the page doesn't reveal much about the game, but is quite lengthy and shows that the authors were rather fond of Heavy Metal, including Manowar, who featured on the cover of Rhetoric Issue 5 (so it was an Oric related cover after all!). There is an old Judas Priest song called 'Tyrant', I wonder if this is where they got the title from?

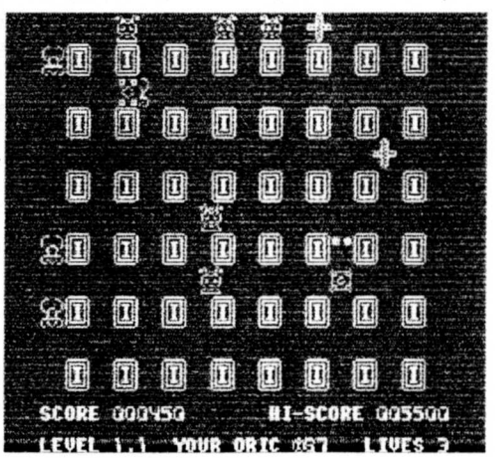
Tyrant – Title Screen



Options Screen

The instructions in the game are lucid, spinning a tale of how you have entered the grounds of the mighty Sadujen, tyrant of the land. You've been detected by his robot guards and must fight them off at the controls of your robot. Your heart will sink when you see that you need eight keys to move the robot and fire its weapons - this is going to be tricky. Fortunately they are bunched logically in two groups. You'll move with your right hand and fire with your left, four keys determining whether you fire up, down, left or right. Sounds difficult? Well, it isn't, its nearly impossible!

As soon as you start the game you are treated to a familiar tune - Flight of the Bumble Bee, which is also used in Insect Insanity from Tansoft/Mirage. This is really quite appropriate as Tyrant is another game guaranteed to drive you mad. The display is quite simple, consisting of a grid of pillars which you and your adversaries can move between. Sixteen Type 1 Defence Robots are displayed around the edge of the screen and these move into the maze in groups of up to four at a time. Collision with one of these or the bullets they occasionally fire results in the loss of one of your three lives. Further lives are awarded at 1000 points and every 1500 points after that. Your only defence is to shoot the Defence Robots, whereupon they will be replaced by one of those waiting in the holding areas at the screens edge. In addition, when a Type 1 Defence Robot is destroyed, a Type 2 Defence Robot appears in one of the holding areas. When a Type 2 Defence Robot is destroyed, it is replaced with a Type 3 Defence Robot. If this is destroyed it is not replaced. The computer randomly chooses which robots to allow into the maze, so in a short space of time you will find yourself facing a mix of all three types of robot. I haven't noticed any difference in the aggresiveness or intelligence of the three different robots, but then again I have been too busy trying to survive than to notice any patterns! When the last Type 3 Defence Robot is destroyed, a bonus is awarded and then its on to the next wave. Its worth noting that if a robot fires at you and hits one of its colleagues instead, then the shot robot will be destroyed and you still get the points for the kill.



TYRANT CONCLUSION....

The games graphics are ok, but there isn't much variety. They are nicely animated (two frames) however and when the game is paused, although the action stops, the characters continue to animate - a nice touch. All the graphics move smoothly during play.

Control of your robot is good and it responds well to keyboard inputs. Firing is a different matter. Because there is a separate key for each of the four directions you can fire in, you will find it very difficult at first to co-ordinate movement and shooting. The fire button response also seems to be a bit dodgy when firing at right angles to your direction of travel, but I suspect this is more to do with my timing - you have to be lined up exactly with a side passage before the robot will fire. Your robot also has limited ammo. When the ammo level gets low a recharge box will appear somewhere in the maze and you'll need to get to it quickly if you don't want to be left without any firepower.

The action is fast and frantic - you need to be both aggresive and defensive at the same time. Having to use eight control keys is, at first, extremely daunting but with practice you'll be able to cope. Once you've mastered the keys and become used to watching everything going on around you, the game becomes very addictive and you end up flying round the maze at high speed, zapping away carefully to conserve ammo. Tyrant is a solid blast - recommended to hardened arcade addicts, but perhaps a little too tricky for the average player.

Final Conclusions:

* Graphics ok, not superb, but well animated.
* Adequate sound effects and good, if unoriginal, music.
* Very difficult to start with but becomes playable with practice.
* Excellent arcade game for the experienced player.

.....RHETORIC DISK UPDATE.....RHETORIC DISK UPDATE.....

STEVE 'RONNIE JAMES' MARSHALL WRITES:

How do,
just got another letter from W.John Hurley. He says Cock In doesn't work and it needs to be on a Gameinit disk. (didn't anyone else notice ??)
Thanks again for the letter John . Sedoric 3 pretty much makes Gameinit redundant so the problem is elsewhere. It's a case of a missing QUIT. Here's what to do.
Type:
REN "COCKIN" TO "CI"
10 !QUIT
20 !LOAD"CI"
SAVEO"COCKIN",AUTO
That'll fix things. Apologies for the mistake. Anyone that has problems send the disk to the Muso.
Hope you can get that in the mag Simon.
John also says that there are rude words to be found in the Arrow of Death adventures and worried that his grandson might view them. This isn't really the case. The game will respond if you type certain words in The keywords are words the game recognises -not words that are put on screen. Some of the fun of adventures was seeing what 'hidden extras' were in the game. Some responded to swearing, others to things like 'records' which would result in the programmers current favourites being listed.
The rude words that come up if you try to hack in the game only appear if you try to hack into the game. All the rude words are not encountered during ordinary playing so I feel there is no problem, though I understand and acknowledge your concerns.
(Thanks again for WJH sending a disk of software. There is still a problem with Backgammon which I'll be in touch about).