

# List of bugs in Oric

compiled from newsgroups <comp.sys.oric> by André C.

*Question de Olivier J.:* Does it exist a file that summarizes all the bugs of Rom Oric Atmos?

*Question from 'Euan':* Could someone post with the Oric-1 bug list, or point me to a website where they are documented?

*Fabrice F.:* Have a look at Pascal Leclerc's page in the Oric Webring: he has a debugged and improved Rom that should fix all the bugs (and you can read the modifications in the bundled documentation)...

*Geoff P.:* Some of the bugs in Oric 1.0 Roms are most exciting! My favourite, for sheer perversity is number 11 (see below). I vaguely recall bugs from the earlier prototype too. My memory is playing tricks, I thought that entering certain six digit line numbers caused a complete crash, but I cannot see a reference to this in my book, so it must have been cured prior to the official release of the Oric. 320000 I think was one example. Note that this list does not include other software weaknesses, such as in the cassette interface. You can find them in my old book, available from web site <www.geffers.demon.co.uk>. Here are the relevant bits:

## 2.16 Bugs in Basic

Most people will be aware of one or two problems with version 1.0 Basic, the most notable example being the TAB function, which is quite useless (although the previous section should help with the problem). In this section, we look at all the bugs and, where relevant, how they can be overcome. First of all, here are the quirks found in **version 1.0 machines**.

1. TAB and COMMA do not work correctly. It is best to use either SPC or, alternatively, POKE #269 with the TAB position.
2. STR\$, when packing a positive number, puts the attribute '2' at the front instead of a space. This often results in green numbers! The cure is to use MID\$ to take off the unwanted character or to define a new STR\$ function using the & function.
3. ELSE does not work under several conditions, for different reasons, so it is best to simply avoid the command altogether.
4. HIMEM is not set correctly on power-up. The solution is to always put in a HIMEM command at the start of the program, e.g., HIMEM #97FF.
5. When in high-resolution mode, the message 'SAVING' is still output to address #BB80 - putting one line of junk onto the screen. There is no easy cure for this problem, apart from writing your own save-to-tape routine. If you are saving a high-resolution screen, then first copy it to a free area of memory and save that part of memory.
6. When the printer is in the middle of either an LLIST or a series of LPRINTs, characters are often corrupted into 'squiggles'. This is because the interrupt routines,

which read the keyboard frequently, conflict with the use of the printer. The solution is to stop the clock (CALL #ED01) before printing and to start it again after printing is complete (CALL #ECC7). If you are using LLIST, then you can type: CALL #ED01: LLIST and then use the Reset button underneath.

7. When you use CLOAD from within a program, BASIC unkindly ends the program once the load is complete. To get around this, you could do a series of CALL instructions instead of CLOAD. Chapter 4 contains all the necessary information.

8. The function HEX\$ has an unfortunate tendency to print just the hash sign for zero. This condition should be specially tested for in your program.

9. The GET command refuses to believe that you have pressed the single quote key and instead returns an empty string (""). It is important that you test for this condition before using one of the functions such as ASC.

10. If a print line starts with control characters - e.g., ESC N, etc. - then the protected columns 0 and 1 are used, overwriting any PAPER and INK attributes. Always start the line with a non-attribute character, such as space.

11. The alternate character set is exactly one bit out of place! The purpose of the alternate character set, when not modified for a special use, is to provide a 'chunky' graphics capability. The format of such characters is identical to that used in the BBC's CEEFAX system, allowing a resolution of 80 chunks across by 84 chunks down. Each character cell contains six such chunks, which means that 64 graphics definitions are required to allow for all possibilities. The Oric's character set has in fact been set up for this. Characters between 32 and 95 contain all variations between a totally blank cell and a filled cell. However, in version 1.0 the entire character set must first be divided by 2 (and therefore shifted to the right) before it can be used. This can be done either with a simple Basic loop: FORI = #B900TO #BAFF: POKEI,PEEK(I) /2: NEXTI or by using a short machine code routine:

```
LDY 000
LOOP: LSR B900, Y
      LSR BA00, Y
      DEY
      BNE LOOP
      RTS
```

12. If the single quote character is found at the start of a DATA item, then because of confusion with the REM facility, the rest of the DATA line is ignored. Use double quotes around any DATA items containing single quotes.

13. When loading in a machine code program, be warned that the 'end of Basic' pointer at #9C,D is altered to reflect the end address of the machine code. To overcome this you could either reset the value at #9C to #9D after the load or make it a rule to always load the machine code routines first.

14. In the instruction POKE N, #8, the hexadecimal sign upsets BASIC, and zero will be POKEd. Always use a decimal value or a variable instead. This fault is the reason why you will often see decimal numbers mixed with hexadecimal numbers in this book. The DOKE command does not suffer from this fault.

15. One interesting bug is that POINT will work in text mode!

16. When loading a file, the filename is only printed when it is actually supplied within the CLOAD"" command.

17. Although potentially useful, it is still a fault that makes the screen scroll down when the cursor is moved too high.

#### **The following faults lie in version 1.1 Roms (Atmos)**

1. ELSE fails to work should the colon character occur in quotes after the ELSE. For example: IF A=1 THEN PRINT ELSE PRINT «HELLO:».

2. One very obscure problem arises when:

(a) The cursor has been turned off.

(b) A character is placed at the very spot where the cursor would have been.

(c) That character is 'inverse' - between 128 and 255. When this happens, and providing interrupts are running, that character is forced back to 'normal' mode - losing the top bit of the character. One solution for this problem is to force the current cursor position to a place on the screen (or even off the screen!) where it can do no harm. This is done by poking locations 268 and 269 as described earlier.

3. One very minor bug is that going into HIRES when in control - S mode results in Basic writing to the wrong part of the screen. Make sure that you have enabled the screen before using the HIRES command.

#### *Steve M.:* **Oric-1**

[1] When I tried a listing to demonstrate the comma bug, it did not seem to work for me!

[3] Again, it seems to work for me, but maybe I am not using it in the way it goes wrong. Seem odd that it does not work under only some conditions.

[6] I have not experienced this one either. Only using the MCP40 and I think that is prone to chucking out the odd bogus character.

[8] I do not think I have seen that happen either.

[12] This is V1.1 is not it? The single quote is not recognised as a REM by the Oric-1. Well I have just tried it on Euphoric and it did not recognise it!

**Atmos** [1] Nope, again it seems to work correctly for me. There were at least two versions of the Atmos Rom though. Mind you, there were at least three different V1.0 Rom versions too, but I am not sure if the code was altered on those. For Atmos Roms type ?PEEK (#E4B6) with 162 being the earlier Rom and 142 being the later one. This was said to have the ELSE bug fixed and an improved cassette loading routine.

*Geoff Phillips:* [12] Those first section were all version 1.0 bugs, fixed in 1.1. As far as I remember, the sequence of events was:

1. Pre-release 1.0 Roms with an incredible number of bugs.

2. 1.0 Roms, used in Oric-1 with bugs as listed.

3. 1.1 Roms put in some Oric-1's.

4. 1.1 Roms put in Oric Atmos.

5. Perhaps later releases of 1.1 Roms?

It is possible the bugs were fixed in the emulator? The bug list came about initially because I was doing the bug testing for Tansoft, as part and parcel of everything else that was going on. Perhaps someone with a real Oric-1 can verify...

*Steve M.:* Do you know if there is any difference between the code of the 2xRom version as compared to the single ROM version V1.0? There was definitely a later release of the V1.1 ROM. I have one! I have done some checking on the REM thing and the V1.0 does not allow it at the start of a line like I was trying. That explains why it was not working! You can see some limited info on the Roms here <<http://www.48katmos.freeuk.com/roms.htm>>. And you can read my dodgy review of your book here <<http://www.48katmos.freeuk.com/works.htm>>.

*Geoff P.:* I always liked the cover McGraw Hill did, better than the Dragon one I'd done the year before which was all silver and shiny. I met up with an old school friend earlier this year, who had cleared out his loft and found a copy I'd given him. It was a little worse for wear!

I think I wrote the book as a kind of personal aide-mémoire, inasmuch as needing to know this sort of thing to code for it. Afterwards, when I had spent some years working on other video and audio technology, and writing embedded BIOS code, I felt slightly embarrassed by my lack of deep-down knowledge. That same feeling occurred within a few years when I found out better techniques for sprites, animation, game control, and so forth; but the difference there was I started to work alongside other programmers, compared to working alone, and so one could exchange ideas and techniques. The Dragon book I had done was very high-level in its approach, just really listing applications in BASIC and explaining how they worked. For the Oric, since I'd already pretty much realised that the best could be achieved in machine code, I was trying to help expose the inner workings for other programmers - but at the slightly removed from bare metal level of how you could use the ROM routines to achieve some things, without actually driving the hardware directly. There is another difference in the way I worked - a few years later, for machines like the Spectrum, Amstrad, and Atari, I would scorn the BIOSes altogether and use self-contained routines that as much as possible drove the hardware directly. That all had to change back again once the PC came along, though I continued in this mode with the embedded coding I was doing up until a few years back.

You are right about the graphics section being a little thin on the ground, but as is always the case, I was running out of time to finish, and had other pressures coming from Tansoft at the time.