

# Firmware

brendanalford edited this page 20 days ago · 46 revisions

## ZX Spectrum Diagnostics

By Brendan Alford ([brendanalford@eircom.net](mailto:brendanalford@eircom.net))

Original version by Dylan Smith.

## Introduction

These diagnostics are intended to assist in the repairing of the range of ZX Spectrum computers, from the 16K right through to the +2A and +3 models.

They are designed for use with Dylan Smith's diagnostic board (described in the link above) but can be utilised with any device that can replace the Spectrum's built in ROM with this image.

The [Retroleum SMART card](#) is also supported from v0.30 onwards. If using such a device, install it in Slot B, replacing the preloaded diagnostic firmware. See the device's user manual for full details.

The [ZXC3/ZXC4](#) cartridges are supported from v0.32 onwards.

### ▼ Pages 6

[Home](#)

[Firmware](#)

[Flash Utility](#)

[Hardware](#)

[Spectranet](#)

[Tape Tests](#)

### Clone this wiki locally

<https://github.com/brendana>



📄 Clone in Desktop

If testing a machine that will not boot, ensure that the testrom.bin image is the only ROM image on the cartridge, such that it will automatically start upon power on without invoking the ROM selection menu.

Also ensure that when writing the image to the cartridge, the 'Paging Locked' option is set to 'None' for this image, otherwise the diagnostic tests will not work fully.

Please consult [Paul Farrow's ZXC4 pages](#) for more information.

The Diagnostic Board hardware's original layout and gerber files are available at the above (archived) link, however these are in the process of being updated for a more modern layout and component availability. Those with the ability to construct their own may make use of the schematic and layout in the 'hardware' subdirectory to do this.

## How to Build

---

The test tool is structured such that it can be built using the sjasmpplus assembler. Assuming sjasmpplus is on your path, all that needs to be done is to run the build.bat batch file which will build the diagnostic firmware, which will be named 'testrom.bin'. Testrom.sjasm is the project file for the assembler if modifications need to be made, in this case some familiarity with the tools must be assumed.

Alternatively, binary and Spectrum .TAP file versions of the firmware can be obtained from the release package [here](#), as testrom.bin and testrom.tap.

## How to Use

---

If you simply want to use the firmware as is, a ready to program version of the diagnostic firmware is provided in the release package, named 'testrom.bin'.

Since there are many different hardware devices capable of providing the ability to replace the Spectrum's ROM image with the diagnostic firmware, that process will not be outlined and is left as an exercise for the reader. To be fair, if you own any of these devices and are attempting a Spectrum repair, it's assumed you know what you are doing :)

Plug in the device containing the diagnostic firmware and power on the machine.

You should hear two distinct tones (and see a blue, then green border if the display is active) - this confirms that the machine is running, at least the CPU and ULA are working and that there are no issues with the address or data buses on the machine under test that prevent code execution. If running with a Diagnostic Board (henceforth referred to as a DiagBoard), the eight LED's will light briefly then go out - this confirms that the DiagBoard I/O is working correctly.

The firmware will then execute some memory tests on the lower 16K of RAM which are described below.

## Lower RAM testing

---

As the lower RAM of a Spectrum is used for display, we cannot assume that the machine can produce a correct display nor rely on it for output. Therefore lower RAM testing will produce some test patterns on the display that exercise the RAM in various ways, and if a failure is found, will report the result by way of a series of stripes in the border.

There will be eight of these stripes, representing bits 0 to 7 in downward order. These correspond to memory IC's IC6 to IC13 (which are 4116 IC's) in that order. A green stripe indicates a healthy IC, and a red one that a fault was encountered.

In the screen area, the firmware will attempt to display a message 'RAM FAIL' along with the failing bit positions. Depending on the overall health of the lower RAM in the system, this display may be quite clear or barely legible, but should be visible to some extent on machines in which there is at least one working lower RAM IC.

The computer will then halt.



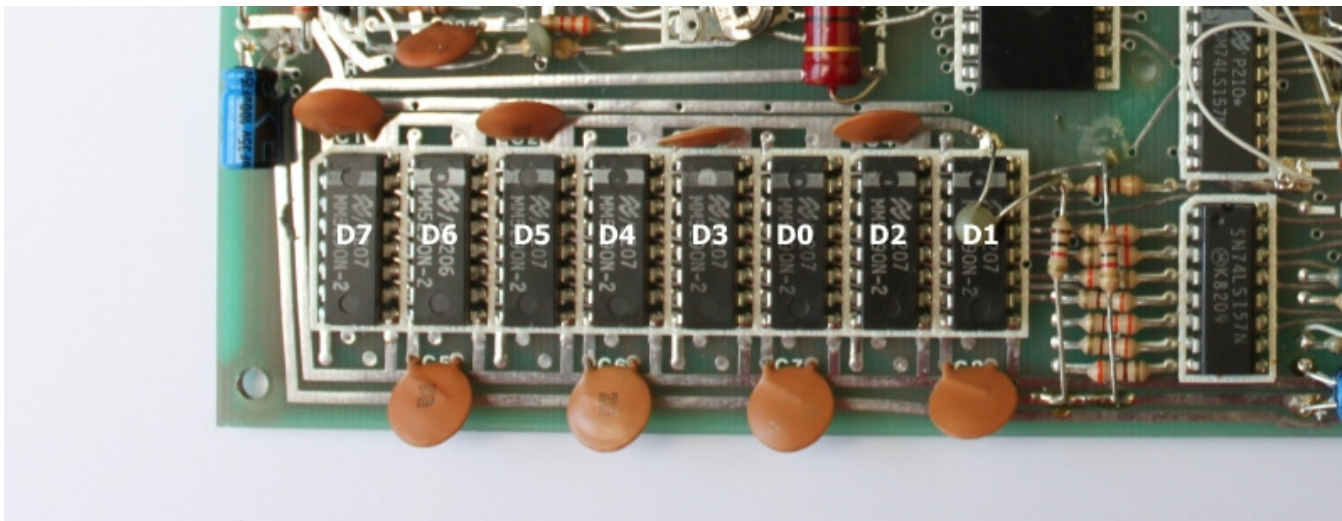
If a DiagBoard is being used, then the LEDs will be lit to reflect failed memory IC's in the same pattern.

**NOTE 1:** If testing a 128K machine, these tests will by default exercise page 5 of RAM. In this case, the memory IC's corresponding to bits 0-7 are as follows:

Bit	IC (128K)	IC (+2)	IC (+2A/+3)
0	IC6	IC22	IC2

U	IC6	IC32	IC3
1	IC7	IC31	IC3
2	IC8	IC30	IC3
3	IC9	IC29	IC3
4	IC10	IC28	IC4
5	IC11	IC27	IC4
6	IC12	IC26	IC4
7	IC13	IC25	IC4

**NOTE 2:** If testing an Issue 1 Spectrum, the lower RAM IC's do not have IC designations on the PCB, nor are they in the same layout as in later issues. The layout is as follows:



## Further tests

If the lower RAM tests pass, then we have usable RAM in which to produce a display and store

some workspace. The machine will then clear to a white screen where results of further tests will be displayed.

If running on a DiagBoard, SMART card or ZXC3/ZXC4 cartridge, the firmware will try to identify the ROM on the computer by checksumming the content. If this is successful, the firmware confirms the checksum and type on the screen, then proceeds with testing appropriate to that machine.

If the ROM is corrupt or missing, then this test will fail with the message 'Unknown or corrupt ROM' and the user will be prompted to select the type of machine being tested.

If another non-supported ROM device is being used, or the system ROM has been replaced by an EEPROM containing the diagnostic image, the message 'Diagnostic hardware not found' will be displayed and the user will be prompted to select the machine type.

**Note:** This autodetection currently does not work on +2A/+3 machines due to the /ROMOE1 and /ROMOE2 lines on these machines not being handled by the DiagBoard hardware - one will need to choose the +2A/+3 test option manually. This is not an issue when using the SMART card or a +2A/+3 fixer device.

On 128 machines, if the first ROM checksum passes but subsequent checksums do not match, then this will be indicated along with the IC designation to check.

## 48K Tests

The upper 32K of memory is then tested using the same test routines that were used for the lower 16K. Any failures for a particular test are noted, and the IC's responsible for the failure are displayed.

If the machine has been identified via the ROM check as a clone, then the failing bit positions are displayed instead.

If all 8 upper RAM IC's are found to be faulty, the machine is either a 16K model or has a problem

with the upper RAM multiplexer IC's. Information to this effect will be displayed if this occurs.

```
ZX Spectrum Diagnostics
Lower 16K RAM tests...          PASS
Spectrum 16/48K ROM...         PASS
Upper RAM Walk test...         FAIL
Upper RAM Inversion test...    FAIL
Upper RAM March test...        FAIL
Upper RAM Random test...       FAIL

48K tests FAILED
Check the following IC's:
IC19

Testing interrupts...          PASS

Failures found, system halted

v0.32 D. Smith, B. Alford
http://git.io/vkf1e
```

**Note:** If testing an Issue 1 with Sinclair 32K daughterboard, this board again has no IC designations. The diagnostics will identify IC15-IC22 as being faulty, representing bits 0-7 in the same order. These bits map to the physical IC's on the daughterboard as marked below:





When testing is complete, if running on a DiagBoard or SMART Card the machine will attempt to page in the Spectrum's own ROM following a brief countdown. If successful, the machine should reset to BASIC as if the machine had been just powered on.

## Other Features

---

### Testcard

---

If the SPACE key is held down when the machine is powered on, following the initialisation tones a colour test card will be displayed, accompanied by a repeating tone. This is intended to facilitate tuning of VR1/VR2 pots on Issue 1 or 2 machines.

Holding down the Break key (or Caps Shift/Space) will reset to BASIC (on supported hardware) or restart the firmware (other devices).

Holding down the 'A' key will switch tone generation to the AY chip if present. All three channels are exercised by this test (which is based on [BLITS](#)).



## Soak Test

---

If the 'S' key is held down when the machine is powered on, the firmware enters soak test mode (signified by a third high pitched tone immediately after the initialisation tones). This performs repeated memory testing, looping back to the start after each successful test.

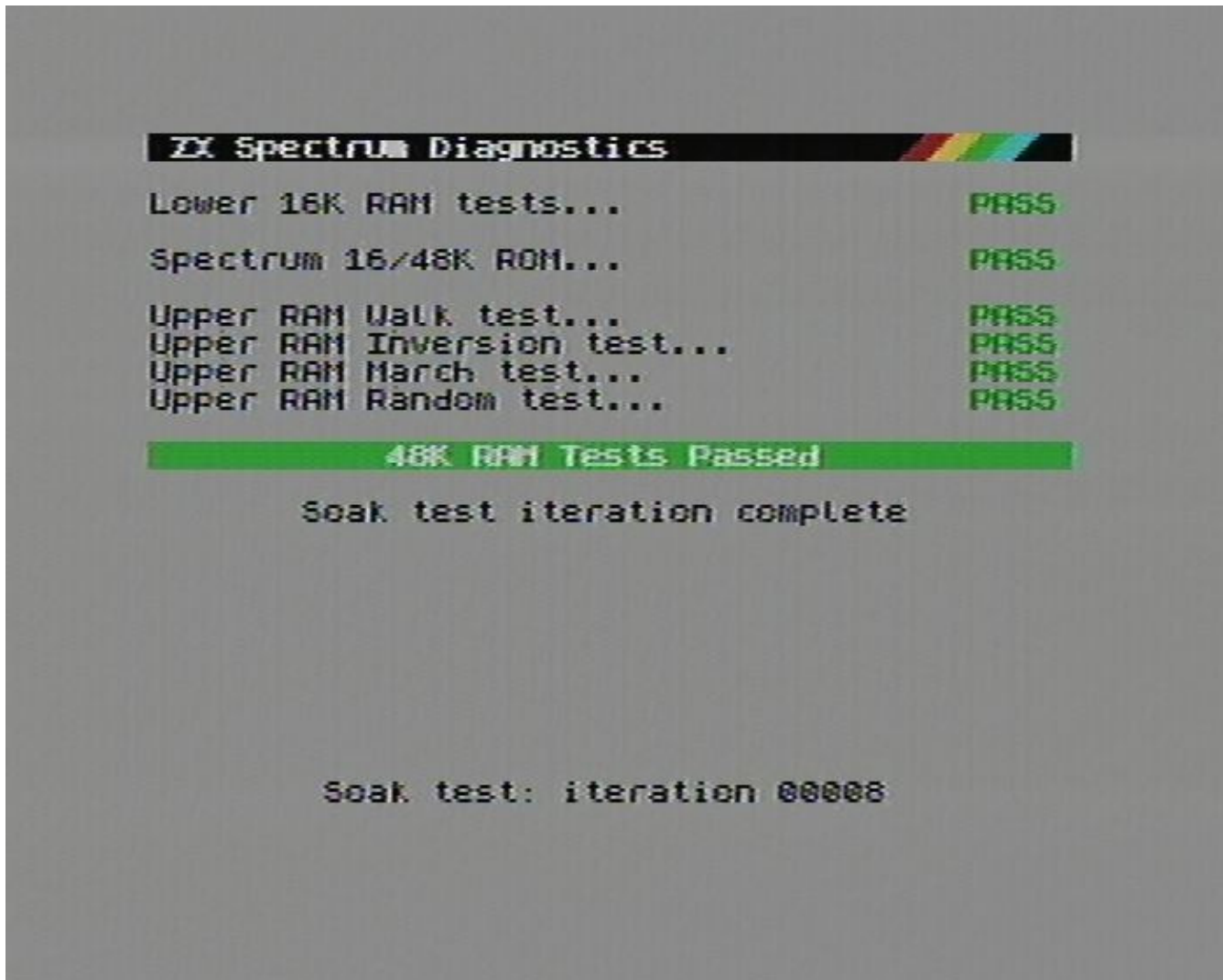
The soak test can also be launched by holding down the FIRE button of a joystick attached to

either a Kempston compatible interface, or an Interface 2 (Port 1) compatible interface.

When testing upper RAM (48K) or 128K memory, soak test mode is indicated as a counter at the bottom of the screen of the form 'Soak test: iteration xxxxx'. This may be used as a guide to how long soak testing has been running for.

All tests are executed identically and in the same order as in normal test mode, except for the interrupt count test.

Following each successful test iteration, a message 'Soak test iteration complete' is displayed and the firmware commences the next test iteration after a short delay.



If a lower RAM failure is detected, soak test mode is indicated by a narrow yellow border stripe at the bottom of the screen.

A failure detected during soak testing will halt testing (except where no upper memory is found working; this will be the case if testing a 16K Spectrum).

If the ROM checksum test fails (and therefore the machine type cannot be determined), the firmware assumes 48K hardware and continues testing as such.

# ULA Test

---

If the 'U' key is held down when the machine is powered on, following the initialisation tones the ULA Test screen will be shown. This is a rudimentary facility to allow basic ULA functionality to be verified.

The values read from Port 0xFE are displayed following the 'ULA Read' message, set bits are rendered as white ink on black paper, reset bits as the inverse. Pressing keys on the keyboard influences the values of bits 0-4 as you would expect, and allow the full keyboard to be verified.

Holding down the 1 key will output a tone via the MIC port of the ULA, accompanied by red/cyan stripes. This can be used to verify tape saving output.

Holding down the 2 key will output a tone via the EAR port of the ULA, accompanied by blue/yellow stripes. This can be used to verify speaker output (a little redundant perhaps)

Holding down the 3 key will output all possible colours as a rainbow border pattern, and to verify that this functionality is operating successfully.

Holding down the 4 key will, on 128K machines, test the shadow screen switching functionality by overlaying the screen with green bars on the screen and in the border. A problem with screen switching would result in the entire screen becoming green, or stripes in the border only (which is what 48K users will see if selecting this test). If the stripes continue to move a couple of seconds after the 4 key is held, then this indicates an issue with paging port contention (the stripes should move a little but then settle into a non-moving display).

A moving multicolour stripe is displayed to allow verification of interrupt functionality and frequency. If the firmware detects missed interrupts or too many interrupts, then the words 'FAIL FAIL FAIL' will be displayed in the middle of the line that the stripe occupies. If this occurs, you may also observe a thin red stripe in the border (this would ordinarily appear immediately after vertical sync and is only visible by adjusting the vertical hold control, if you have a suitably aged TV).

When holding down keys 1,2 or 3, interrupts are disabled and the multicolour stripe will be paused until the key is released.

If a tape is played back during this test, then black and white loading stripes will appear in the border. This will allow the azimuth angle of the tape deck to be adjusted to achieve more uniform stripes, and therefore more reliable loading,

As with the test card, holding down the Break key (or Caps Shift/Space) will reset to BASIC (on DiagBoard/SMART hardware) or restart the firmware (other devices).

## ULA Test

ULA port 0xFE read..... 7E543210

Interrupt test (movement should be smooth)

Select:

- 1) Output tone to MIC port
- 2) Output tone to EAR port
- 3) Test border generation
- 4) Test screen switching (128K)

Hold BREAK to exit

v0.32 D. Smith, B. Alford  
<http://git.io/vkf1e>

## Keyboard Tester

The firmware includes a simple keyboard test so that the membrane in the machine under test can be verified. It is activated by either holding down the K key on startup, or by triggering an NMI at any point during testing (this can be done using the NMI button on the Retroleum SMART card, or by any supported interface that includes such a button).

A representation of the Spectrum's keyboard is displayed, and when a key is pressed, the corresponding key on the on-screen keyboard will light up, accompanied by a short beep.

Holding down Caps Shift/Space or BREAK will exit the test and return to BASIC (supported devices) or restart testing (unsupported devices).

## Memory Browser

---

Holding down the 'M' key on startup enters memory browser mode, which allows the ROM/RAM space to be examined, browsed and modified. All available ROM and RAM banks can be paged in for inspection, and entering hexadecimal values directly will immediately modify the byte upon which the cursor is positioned.

ROM space is highlighted in red to indicate that it cannot be modified.

Holding down Caps Shift/Space or BREAK will exit the browser and return to BASIC (supported devices) or restart testing (unsupported devices).

## Firmware Details

---

These can be accessed by holding down the Symbol Shift key on startup, and display version information and other build information. In the event that the firmware malfunctions or does something unexpected, please note the information represented here and provide it to me along with your bug report.

## Testing Details

---

There are four distinct types of memory test performed on each block of memory, these are described below.



# Walk Test

---

This is a very simple test - all it does is set each bit and reset each bit in memory, checking the memory holds the desired value on each iteration. If the test fails here, it means the failed RAM chip simply isn't reliably (or at all) able to be set to a given value. This can be caused by a failed chip or a bad solder joint or broken PCB track. (When testing lower RAM, you can often see this once tests halt, by vertical lines 1 pixel wide running down the screen being visible)

This test will also show up failures in logic caused by data bus lines being shorted together.

# Inversion Test

---

Many faulty chips can pass the first test sequence. This test looks for faults where setting or resetting a bit in memory causes another bit to erroneously be set or reset. The tests consist of setting all of the memory bank to zero, and then writing 1s in all even memory addresses, and then checking the pattern is as expected. The test is run again - memory is blanked, then odd addresses are written to, and the pattern checked. Then, all of the ememory bank is set to 1, and even addresses are reset and tested. Finally, memory is all set to 1 again and odd addresses are reset and tested.

If the memory fails the inversion test, you can often get some insight into what is happening by watching the screen if it's lower RAM that has failed (and if your Spectrum won't boot normally at all, this is highly likely). You should see alternating black and white vertical stripes appear on the screen. If a set of stripes is anything other than black or white, setting the bit in the failed memory likely sets the entire contents of the chip to that value. It's likely the row/column select circuitry in the failed chip has a fault in that case. Other failures can be simply one or two adjacent bits getting flipped in the wrong place. This may not be visible by the screen even if it's happened in lower RAM (the portion of RAM that the frame buffer occupies might be working fine).

# March Test

---

This test aims to shake out simple failures caused by addressing a memory location that causes adjacent locations to be written erroneously. The algorithm used works as follows:

- Step 1: Write 0 in ascending addressing order;
- Step 2: Read 0 and write 255, again with ascending addressing order;
- Step 3: Read 255 and write 0 with descending addressing order;
- Step 4: read 0 with descending addressing order.

## Random Fill Test

---

There are some more subtle kinds of failure that the inversion or March test won't pick up, such as setting a memory bit in the faulty chip causing a bit to be set somewhere in the other half of the chip to be set. The random fill test tries to shake these out.

The routine uses a 16 bit pseudo random number generator to fill the memory bank with values, and then restarts the random number generator with the starting seed, and compares the output of the random number generator with what is held in memory. Memory is filled from bottom to top with one pattern, verified, then top to bottom with another pattern and verified again.

When lower RAM is tested, you should see a scrambled pattern of random pixels and attributes on the screen.

Memory failures in this test are most likely problems with the chip's row/column select circuitry. For example, if the most significant bit gets stuck on for column access, setting a bit at location 0 in the chip will also set location 8192 in the case of a 4116.

