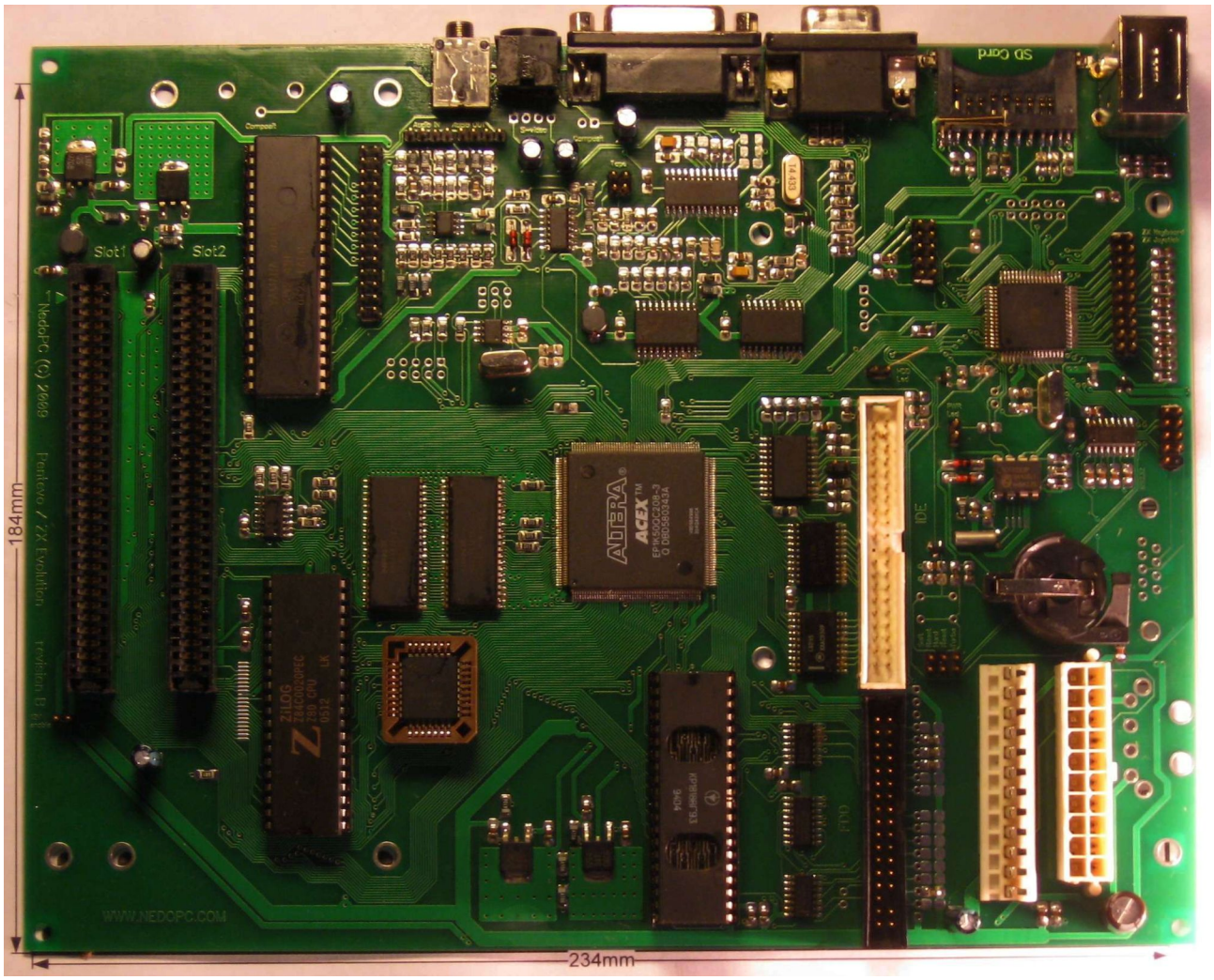
ZX Evolution
Reference manual.



(Version 02/08/2026)

www.nedopc.com

Table of contents

1 Introduction

Under the basic configuration of the computer ZX Evolution means a configuration that is supplied and supported by the manufacturer. The producer support and development of this configuration and fix the found errors in it. Group NedoPC does not support the configuration of third-party developers, in case of problems with such configurations, please contact the developers directly.

The basic configuration consists of the following components:
Configuration for the FPGA EP1K50QC208;
The program for the microcontroller ATMEGA128.

Configuration is available as a binary file (zxevo_fw.bin), which can be downloaded into the microcontroller ATMEGA128 with BOOTLOADER1. Updating configurations, also produced by BOOTLOADER.

FPGA configuration is integrated into the program the microcontroller. Configuring the FPGA is performed by turning on or iron reset the computer.
Modified version of the basic configuration involves changing any or both components.

2 Switch functions

These two switches provide the ZX Evolution with the following functions.

2.1 Hard reset (Full reset)

Switch group Hard reset carries out a full reset of the computer (restarting the firmware of the microcontroller and the reprogramming of FPGA) for the closure of contacts.
It is recommended when using a standard (AT or ATX) case connected to the button «Reset» Corps.
Attention! Hard Reset does not ensure the safety of the contents of computer memory (eg, RAM-drives or pages with source code in).

2.2 Soft reset

Switch contacts Soft reset depends on the configuration.
The basic configuration of this switch group is charged with two functions:
soft reset the computer (CPU and reset the internal registers fpga);
ON / OFF ATX power supply.
It is recommended to use a standard ATX case connected to the Power button. In this case, a single push of the button will turn on the computer. To shutdown the ZX Evolution this switch must be pressed for 5 seconds.
Momentary contact closure will perform a soft reset the computer and install the zero page ROM.
Soft reset ensures the safety of the contents of computer memory.

3 Additional features of the PS2 keyboard

If you use the PS2 keyboard on your ZX Evolution you can use advanced features for managing your computer:

- Soft reset;
- Hard reset;
- Switch video output modes (TV/VGA);
- Switch audio modes (beeper / tapeout);
- Issue a NMI to the Z80 cpu.

3.1 Soft reset the computer

Soft reset the computer by pressing «F12» for less than 5seconds.

3.2 Full reset of the computer (hard reset)

A hard reset from the computer keyboard can be done in two ways:

- By pressing «F12» (duration 5sec), You cannot turn off the power.

- By pressing a combination of «Ctrl-Alt-Del», You cannot turn off the power.

3.3 Switching the video output (TV / VGA)

- TV – Output is to a television via composite video.
- VGA – Output is to a VGA compatible monitor.

Switching the video output (TV / VGA) is carried out by pressing the «Scroll Lock» button. Indication of this is shown by means of the «Scroll Lock» LED on the keyboard. The LED on corresponds to VGA PC mode, Off - TV mode.

The timing of a VGA signal generated by scandoubler is not standard, and some monitors can't display correctly. The new mode with modified timing («60Hz») is added especially for such cases. Use a shortcut key «Shift+ScrollLock» for switching between modes - TV60/VGA60.

The «60Hz» mode is nonconventional for ZX Spectrum compatible software and can cause its incorrect operation (speed up, speed down, freezes). NedoPC do not recommend a «60Hz» mode for daily using, and the software development especially for it.

Attention! For the state to be retained when you shut down your computer, you must install the battery.

3.4 Switching the audio mode (beeper / tapeout)

- beeper – Audio to the beeper;
- tapeout – Audio to the tapeout.

Switching Mode Audio (beeper / tapeout) by using the key «Num Lock». Indication of this is shown by means of the «Num Lock» keyboard LED. On corresponds to tapeout mode of the audio output, Off - beeper mode.

Attention! For the state to be retained when you shut down your computer, you must install the battery.

3.5 Issue a NMI to the Z80 cpu

Submission of a NMI to the processor Z80 is by using the key «Print Screen» on the keyboard.

3.6 Adjust the resolution of the PS/2 mouse

Adjusting the Resolution PS / 2 mouse is available only for mice that support this feature. Usually this is optical and laser mice.

The range of supported resolutions PS / 2 mouse:

Value	***Resolution***
0	1 counts per мм
1	2 counts per мм
2	4 counts per мм
3	8 counts per мм

Adjustment takes place by simultaneously pressing two keys PS / 2 mouse (left and right) and control keys on the PS / 2 keyboard:

- keypad '*' - Reset the permissions to 0 (default);
- keypad '+' - Increase the resolution by 1;
- keypad '-' - Decrease the resolution by 1.

Attention! For the state to be retained when you shut down your computer, you must install the battery.

4 Architecture: General hardware.

ZX Evolution has on-board 4 MB RAM and 512 kilobytes of ROM are available through the following models of memory management: ZX Spectrum 48k, ZX Spectrum 128k (port # 7FFD), pentagon 1024k (addressing port # 7FFD), ATM2 (addressing megabytes of RAM) And a third addressing mode of 4 megabytes of RAM. All 512 kilobytes of ROM addressable by the model ATM2, while additionally it is possible to write into the ROM, which is realized on-chip flash-memory.

The architecture of the controllers drive beta-disk appears the concept of shadow ports are available only under certain conditions. The computer ATM2 there is an additional set of shadow port (used, in particular, for memory management) and additional ways to incorporate these ports (in particular, hold the shadow mode when executed from RAM). However, the ATM2 is a unique connection between the shadow and the introduction ROM TR-DOS. The architecture of the ZX Evolution with the full support regimes ATM2 is possible to hold the shadow mode without the inclusion of ports ROM TR-DOS.

The concept of input-output ports for the ZX Evolution differs significantly from the other thereof, in the Spectrum-compatible computers. First of all, decryption always takes place in at least eight minor bits. Significantly different logic block port cards ZX-Bus: if the other Spectrum-compatible computers cards block the ports of the motherboard (and each other, when there are multiple slots), the ZX Evolution, on the contrary, all the ports that are present on the motherboard, not 'Income 'to cards ZX-Bus. The mechanism used IORQGE cards only to block each other (in order of location in the slots). This implies an important conclusion - any cards that duplicate any functionality of the motherboard (with ports, coinciding with the existing ZX Evolution), such as keyboard controllers, mouse, IDE, SD-cards, etc., will not work.

Thereafter, for each port indicate the mode of its scope (the shadow is always available: «**shadow**», «**noshad**», «**always**»), access to it for reading or writing (read-only, write only, read and write: «**RO**», «**WO** »,« **RW** »). Also, the address of the port or in the form of «**# xxFE**» (decoding only the lower 8 bits), or as «**# 7FFD**» (decoding not only the lower 8 bits). The first version of the address port is also used in the case, if any address lines A8 .. A15 are used as information inputs (for example, a set of ports for switching pages in the 16k processor windows: **# 3FF7**, **# 7FF7**, **# BFF7**, **# FFF7**) , With a description of the port shall specify the value of the high byte address, which alone should be used. If listed, 16-bit address port, a senior specific bits of address byte used for decoding, deliberately not specified and may vary from firmware to firmware.

5 Architecture: Memory management.

Memory management in the ZX Evolution in many respects similar to memory management in the ATM2. The principal changes relate to the possibility of addressing a megabyte of RAM via port # **7FFD** (model pentagon 1024k) and the addressing of 4 megabytes of RAM.

In each 16k window processor can arbitrarily include any of the 16k pages of RAM or ROM. There are 2 independent memory cards, which are switched by bit 4 of port # **7FFD**. In compatibility mode ZX Spectrum 128k or pentagon 1024k these two memory cards are programmed identically except for the included page numbers in the zero-ROM window processor to this bit worked like a ZX Spectrum 128k. The mapping of memory pages in each window processor is programmed independently via special shadow ports. In addition, for each such window, there is an option substitution under 3 or 6-bit numbers included page RAM data from the corresponding bit port # **7FFD**: thus ensuring compatibility modes ZX Spectrum 128k and pentagon 1024k. When displaying the same page there is an option ROM substitution LSB page number on the signal DOSEN (enable signal TRDOS). The inclusion of this signal (the "transition in TR-DOS») occurs in the execution of code from ROM to any window of the processor at offset # **3Dxx** in this window during a bit D4 in the port # **7FFD**. Shutdown occurs when executing code from RAM (out of TR-DOS »). In addition, using an appropriate shadow port, you can disable shutdown signal DOS, thus leaving a shadow ports are available in the performance of RAM. In addition, there is the possibility of including the shadow ports regardless of signal DOS.

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#xxBF
WO
always</td><td>Enable shadow mode ports write permission in ROM.
<table>
<tr><td>7..2: set to 0 for compatibility</td><td>1: if 1, then write the ROM is enabled. 0 after reset.</td><td>0: if 1 then enable shadow ports. 0 after reset.</td></tr>
</table>
Bit 0 of the port has priority over other methods including the shadow ports: shadow mode is enabled unconditionally for writing 1 in this bit. When writing 0 inclusion of the shadow regime is controlled by other means, such as port # xx77 or transition in TR-DOS. This bit does not include the TR-DOS.
This port is not in the ATM2.</td></tr>
</table>

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#xx77
WO
shadow</td><td>Enable shadow mode ports of the memory manager's permission.
<table>
<tr><td>A8: if 0, then disable the memory manager. In each window processor is installed the last page of ROM. 0 after reset.</td><td>A9: If 0 then "force" the inclusion of TR-DOS and the shadow ports. 0 after reset.</td></tr>
</table>
Attention! The remaining bits (as the data bits, and bits of high byte address) of this write-only ports are used for other purposes and are described elsewhere in this documentation.
The inclusion of TR-DOS and shadow port bit A9 takes precedence over inclusion of TR-DOS and the shadow ports execution of the offset # 3Dxx page ROM during a bit D4 port # 7FFD («transition in TR-DOS »). When A9 = 0, then the shadow ports and TR-DOS are included, regardless of place of performance of the processor (including to and from RAM).
Allowed to use the following addresses for this port:
#FD77 — the inclusion of the shadow of ports and TR-DOS,
#BD77 — the same, plus permission to overwrite the palette (A14 = 0);
#BF77 – permit rewriting the palette without the inclusion of the shadow of ports and TR-DOS
#FF77 – off the shadow of ports and TR-DOS.
To install the A8 to 0 from any of the above addresses subtract # 100.</td></tr>
</table>

Port	*Description*
#xFF7 **WO** **shadow**	The memory manager pages. 5..0: inverted bits of the page number included in the window of the processor. Available in 16k pages numbered 0 .. 63 RAM (from 1 megabyte), and page 0 .. 31 ROM (a total of 512 kilobytes, bit 5 is ignored.) 6: if 0, then in the window of the processor include ROM, if 1 - RAM 7: if 1, then: for RAM - in the window there is a substitution under 3 or 6 bits (depending on the mode of ZX Spectrum 128k or pentagon 1024k) page numbers are not inverse bits from port **# 7FFD**. For ROM - there is a substitution LSB page numbers signal the inclusion of TR-DOS (1 if the TR-DOS included). In addition, there is the inclusion of the shadow of ports and TR-DOS («log in TR-DOS »), if in this box will code execution with the offset **# 3Dxx**. Bits A14, A15 determine the window of the processor, for which there is inclusion of the page. Should use the following addresses for this port: **#3FF7** – Window #0000..#3FFF, **#7FF7** – Window #4000..#7FFF, **#BFF7** – Window #8000..#BFFF, **#FFF7** – Window #C000..#FFFF. Writing to this port changes the page in which either the current window (active) memory cards. In this case, the inactive card remains unchanged. Which card is activated, sets bit 4 port **# 7FFD**.

Port	*Description*
#x7F7 **WO** **shadow**	The memory manager pages. 7..0: inverted bits of the page number of RAM, included in the window of the processor. Available in 16k pages numbered 0 .. 255 (only 4 megabytes). Bits A14, A15 determine the window of the processor, for which there is inclusion of the page. Should use the following addresses for this port: **#37F7** – window #0000..#3FFF, **#77F7** – window #4000..#7FFF, **#B7F7** – window #8000..#BFFF, **#F7F7** – window #C000..#FFFF. Writing to this port changes the page in which either the current window (active) memory cards. In this case, the inactive card remains unchanged. Which card is activated, sets bit 4 port # 7FFD. Attention! When writing to this port remains the same mode of substitution of lower-bit page number of the RAM specified by writing to bit 7 of port # xFF7. However, the entry in # x7F7 certainly includes the RAM in the box. This port is not in the ATM2.

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#7FFD
WO
always</td><td>In the ZX Spectrum 128k:
<table>
<tr><td>6..7: no fix.</td><td>5: when writing a block further entry into the port # 7FFD, before moving to the regime pentagon 1024k or prior to discharge. 0 after reset.</td><td>4: choice of memory card (see the description of port # xxF7). 0 after reset.</td><td>3: selection screen. 0 after reset.</td><td>2..0: replace the lower 3 bits of a page number, if such substitution is permitted for any window (see the description of port # xxF7). 0 after reset.</td></tr>
</table>
In the pentagon 1024k:
<table>
<tr><td>7..5, 2..0: replace if necessary, lower 6 bits of the page number.</td><td>4: choice of memory card.</td><td>3: selection screen.</td></tr>
</table>
</td></tr>
</table>

Initial program loader stored in ROM ZX Evolution (reset service), you configure the card so that bit 4 port # 7FFD works as a ROM page selection box **# 0000** .. **# 3FFF** (basic128 / basic48), bits 2 .. 0 (mode ZX Spectrum 128k) or 7 .. 5.2 .. 0 (mode pentagon 1024k) - replace the corresponding bits of a page number in box **# C000** .. **# FFFF** (older bits of the page number at the same time – 0) in the other two windows display the page corresponding to the configuration memory ZX Spectrum 128k.

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#EFF7
WO
noshad</td><td>
<table>
<tr><td>2: off for a 1 - mode ZX Spectrum 128k, otherwise - mode pentagon 1024k.
Value after reset - 0.</td><td>3: When placing a 1 in box # 0000 .. # 3FFF forces the zero page RAM. This bit has priority over all other ways to switch the memory page in the window.
Value after reset - 0.</td></tr>
</table>
Attention! The remaining bits of the write-only ports are used for other purposes and are described elsewhere in this documentation.
</td></tr>
</table>

6 Architecture: Video modes.

The ZX Evolution supports the following video modes:

- Standard ZX-mode 6912 bytes
- ZX hardware multicolor, which differs from the standard ZX-mode presence of a single attribute byte for each byte of pixel
- 256x192 16 color standard Alone Coder'a
- 320x200 16 colors on the standard ATM
- 640x200 hardware multicolor standard ATM
- Text mode 80x25 standard ATM

In addition, supported by a palette of standard ATM.

6.1 Selecting a video mode.

Selecting the video mode by a combination of bits in ports # EFF7 (noshad, WO), and # xx77 (shadow, WO), in accordance with the table:

Port **#EFF7**, bit 0	Port **#EFF7**, bit 5	Port **#xx77**, bits 2..0	Video mode
0	0	011	Standard ZX-mode
0	1	011	ZX hardware multicolor
1	0	011	256x192 16 colour
0	0	010	640x200 hardware multicolor
0	0	000	320x200 16 colour
0	0	110	Text mode 80x25

All other combinations of these bits are invalid, the behavior of a computer during the installation of such combinations is not defined and may change in future versions of the firmware FPGA.

6.2 Palette.

The palette gives a map of the 16 available for each video mode, and border colors in the 64 hardware colors (each component of R, G and B is 2-bit gradations of brightness from black to maximum saturation). When you reset the computer installs the ROM palette in compatibility mode with the ZX-Spectum, ie, the colors in the standard ZX-mode, match the colors of ZX-Spectrum.

Program change panel is as follows. The current display color palette defines the element to be changed. For permission to change the palette to record the port # xx77 (WO, shadow) with purified address bit A14 (Warning - the other bits (such as data bits and bits of high byte address) of this port is used for other purposes and are described elsewhere in this document). The new value of the palette entry is set to port # FF (WO, shadow), which, when purified bit A14 Port # xx77 starts simultaneously perform the functions of the controller drive port and the port setting panel.

Reading from port # FF.

Bit bytes to write	Value
0	B
1	R
2	Set it to 1 for the correct operation of the controller drives
3	Set it to 1 for the correct operation of the controller drives
4	G
5	b
6	r
7	g

Gradations of brightness component R, G and B to set the color: ~ {Rr}. ~ {Gg}, ~ {Bb}, where ~ denotes the bitwise inversion.

Note: it is convenient to change the palette when displaying border: in this case the current display color is the color of the curb. To make the process of changing the palette is not visible on the screen, it should be in the process of treatment personnel interrupt (option - after command HALT).

6.3 Border.

Border - the visible screen area, located outside the pixel, determined by a video mode. Border color set bits 2 .. 0 data points in the port **# FE** (RW, always) or **# F6** (WO, always): when writing to the port **# FE** can set the colors 0 .. 7, when writing to the **# F6** - 8. .15.

6.4 Addressing of video modes.

For each video mode, there are two sets of pages from which the sample video. Choosing a set of pages determines bit 3 port # 7FFD (WO, always). Pages that match the identity of this bit are shown in the descriptions of modes in parentheses.

6.4.1 Standard ZX-mode.

Excerpt from page 5 (7). Pixels in each line are composed of bits of continuous byte string (the procedure for issuing a single bit of bytes - from the highest bit of the youngest).
The initial displacement byte strings are calculated from the number line Y (Y = 0 .. 191, Y = 0 for the upper row) as follows: pixoffset = {Y [7:6], Y [2:0], Y [5:3] } <<5, where Y [m: n] - a sample corresponding bit sequence from the value of Y, {} - the concatenation of bit sequencesin a number <<-shift left for a specified number of bits to fill LSBs with zeros .

Bitmap mode is divided into squares of 8x8 pixels, each such square has its attribute byte located at

offset attroffset = # 1800 + Y * 32 + X, where X, Y - coordinates of the square of 8x8 pixels (0,0 - upper left quadrant, 1 , 0 - his right-hand man, etc.).

Pixel color corresponding to the unit bit is defined as an attribute byte attr {attr [6]. attr [2:0]}. Pixel color corresponding to a zero bit - like {attr [6], attr [5:3]}. Bit set attr [7] leads to the hardware periodically (with period of ~ 1 Hz), inversion of all bits read from the memory of pixels within a square of 8x8 pixels corresponding attribute.

6.4.2 ZX hardware multicolor.

This regime coincides with the standard ZX-mode in terms of sample bytes pixels. Attribute, in contrast to the standard ZX-mode, individual pixels in each byte. The offset attribute attroffset = pixoffset + # 2000, where pixoffset - displacement of the corresponding byte pixels. Interpretation of the attribute byte is the same as in the standard ZX-mode.

6.4.3 256x192 16 colours.

In this mode, each block of pixels of width 8 and height 1 pixel aligned bytes pixels standard ZX-regime, which has an offset zxoffset. To display a block of pixels is read 4 bytes:

- byte0 — Page 4 (6), offset zxoffset + #0000
- byte1 — Page 5 (7), offset zxoffset + #0000
- byte2 — Page 4 (6), offset zxoffset + #2000
- byte3 — Page 5 (7), offset zxoffset + #2000

Each pixel block of pixels (left) has a color:
{byte0[6], byte0[2:0]}, {byte0[7], byte0[5:3]}, {byte1[6], byte1[2:0]}, {byte1[7], byte1[5:3]}, {byte2[6], byte2[2:0]}, {byte2[7], byte2[5:3]}, {byte3[6], byte3[2:0]}, {byte3[7], byte3[5:3]}.

6.4.4 640x200 hardware multicolor.

We divide the 640x200 pixel array into blocks, each 1 pixel height and width 16 (blocks 16x1). We introduce coordinates such blocks X, Y (top left block - 0,0; its neighbor to the right - 1,0; neighbor below - 0.1, etc., X = 0 .. 39, Y == 0 .. 199). Enter offset offset = X + Y * 40. To display a block of pixels is read 4 bytes at the following addresses:

- byte0 — Page 1 (3), offset offset + #0000
- byte1 — Page 5 (7), offset offset + #0000
- byte2 — Page 1 (3), displacement offset + #2000
- byte3 — Page 5 (7), offset offset + #2000

Within each block is two bytes and two bytes of pixel attributes. For the left 8 pixels block bytes pixels - byte1, byte attributes - byte0; for right 8 pixels, respectively, byte3 and byte 2.

Interpretation of bytes of attributes: {attr [6], attr [2:0]} - color corresponding to the identity of the bit bytes pixels, {attr [7], attr [5:3]} - color corresponding to a zero bit.

6.4.5 320x200 16 colours.

320x200 pixel array is divided into blocks of 8x1 and enter X, Y coordinates similar mode 640x200 hardware multicolor. Required to display a 4-byte block is read to the same addresses as the mode 640x200 hardware multicolor. Drawing pixels of the 4 bytes is the same as for the 256x192 16 colors.

6.4.6 Text mode 80x25.

Introduce a partition of an array of characters into blocks of 2x1, we introduce the coordinates of these blocks X, Y (X = 0 .. 39, Y = 0 .. 24). Within a block of code left and right characters and lsym rsym, as well as the corresponding attribute byte lattr rattr and read the following addresses:

- lsym — Page 5 (7), offset #01C0 + Y*64 + X
- lattr — Page 1 (3), displacement #21C0 + Y*64 + X
- rsym — Page 5 (7), offset #21C0 + Y*64 + X
- rattr — Page 1 (3), displacement #01C1 + Y*64 + X

Character code defines the displayed character according to KOI-8. Bytes for each character attribute is used similar mode 640x200 hardware multicolor.

7 Architecture: I / O ports

7.1 Port #FE.

Port read the state of the keyboard and a tape input port setting bits beeper, bits of tape output, color border.

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#xxFE
RW
always</td><td>Reading:
<table>
<tr><td>7: 1</td><td>6: tape_in</td><td>5: 0</td><td>4..0: keyboard</td></tr>
</table>
Writing:
<table>
<tr><td>7..5: No fix.</td><td>4: beeper</td><td>3: tape_out</td><td>2..0: Border color</td></tr>
</table>
</td></tr>
</table>

Reading the keyboard is standard for the ZX-Spectrum way - setting to 0 of address bits A15 .. A8, the corresponding desired rows, all the selected data series consist of AND, the result is returned in bits 4 .. 0. Reading occurs without wait'ov. Tape input - Separate entrance ZX Evolution.
A separate tape output is missing, so the bits and tape_out beeper output through a single channel biperny adding them to XOR.

7.2 COVOX port (#xxFB, WO always).

The port is designed to output 8-bit sound. Sound is output on biperny output by PWM with a frequency of about 218 kHz. Recorded values - unsigned counts from 0 to 255. After recording the port # xxFB biperny output begins to operate in COVOX'a until then writing to the port # FE, then biperny output begins to work normally.

7.3 Kempston Joystick port.

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#xx1F
RO
noshad</td><td>Reading the value of the port gives the state joystick.
Format byte value obtained:
<table>
<tr><td>7: 0</td><td>6: 0</td><td>5: 0</td><td>4: Fire</td><td>3: Up</td><td>2: Down</td><td>1: Left</td><td>0: Right</td></tr>
</table>

Fire – 1: pressed "Fire " on the manipulator, 0: not depressed;
Up – 1: pressed the "Up" on the manipulator, 0: not depressed;
Down – 1: pressed the down button on the manipulator, 0: not depressed;
Left – 1: pressed the Left on the manipulator, 0: not depressed;
Right – 1: pressed "Right on the manipulator, 0: not pressed.

</td></tr>
</table>

7.4 Kempston Mouse ports.

Supported by three-button mouse, as well as the mouse wheel.

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#FADF
RO
always</td><td>Reading the port to determine the state of buttons and a counter rotation of the wheel.
Format byte value obtained:
<table><tr><td>7..4: Wheel counter</td><td>3: 1</td><td>2: MMB</td><td>1: RMB</td><td>0: LMB</td></tr></table>
Wheel counter – counter rotation of the wheel;
MMB – 0: Click middle mouse button, 1: not depressed;
RMB – 0: pressed the right mouse button, 1: not pressed;
LMB – 0: pressed the left mouse button, 1: not depressed;</td></tr>
<tr><td>#FBDF
RO
always</td><td>Reading the port to determine the value of counter X coordinate of the mouse.</td></tr>
<tr><td>#FFDF
RO
always</td><td>Reading the port to determine the value of counter Y coordinate of the mouse.</td></tr>
</table>

If the mouse is not detected, the ports reads the value 0xFF.

7.5 Clock and NVRAM ports

Access and control the clock and NVRAM software compatible with the prevalent in the former Soviet Union scheme Gluk Clock (based on a chip or K512VI1 DALLAS 12877, 12877A).

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#EFF7
WO
noshad</td><td>7th bit of this port controls access to the cells of hours is not the shadow mode:
setting bit 7 - allows access;
removing bits 7 - denies access.
Value after reset - 0.

Attention! The remaining bits of the write-only ports are used for other purposes and are described elsewhere in this documentation.</td></tr>
<tr><td>#DFF7
WO
noshad

#DEF7
WO
shadow</td><td>Port sets record in a cell address (0x00. .0 xEF).

Note: In the shadow mode port # DEF7 available regardless of bit 7 port # EFF7.</td></tr>
<tr><td>#BFF7
RW
noshad

#BEF7
WO
shadow</td><td>Read / write cell.

Note: In the shadow mode port # BEF7 available regardless of bit 7 port # EFF7.</td></tr>
</table>

Since the computer's clock ZX Evolution is based on the chip PCF8583, then compatibility is

achieved by emulation.

Features of emulation:

Supports read / write all cells NVRAM (0x0E. .. 0x3F), added the ability to access additional user cells (0x40. .0 xEF);

Supports reading all of the cells carrying the extra features (0x00. .0 x0D);

The record of cells that control the time:

1. 0x00 - Register seconds;
2. 0x02 - Register of minutes;
3. 0x04 - Register of hours;
4. 0x06 - register the day of the week;
5. 0x07 - register the day of the month;
6. 0x08 - the register of the month;
7. 0x09 - the register of the year.

Status of the cell returns the following values:

1. 0x0A - returns 0x00;
2. 0x0B - returns 0x02;
3. 0x0C - returns 0x00 (supported by the state update flag);
4. 0x0D - returns 0x80.

1.

- The read cell value is vaytovoy and occupies a small number of cycles (reading comes from the internal registers ATMEGA128, duplicate values in PCF8583);

- Write operation the cell value is vaytovoy and has a sufficiently large number of cycles (write cycle in a cell includes the step of recording the values in PCF8583).

7.6 Reading the versions of the basic configuration / bootloader.

Application software has the ability to read the version / name of the basic configuration and bootloader ZX Evolution. Accessing this functionality is through the ports of control for hours and NVRAM (see p7.5) as if this extension functional emulated m / c, 12877.

Record values in any of the cells with addresses 0xF0 .. 0xFF, you can choose that will be read. Valid values are:

0 - version of the basic configuration;

1 - bootloader version.

Data length version is 16 bytes (respectively cell 0xF0 .. 0xFF) and has the following format:

0xF0 .. 0xFB - 12 bytes, the name of the version encoded in ASCII (if the title is shorter than 12

bytes, padded with zeros);
0xFC .. 0xFD - 2 bytes (little endian), and a bit of release date (if a bit is not set, it is believed that a test version of beta);
0xFE .. 0xFF - 2 bytes (little endian), CRC value of the firmware. CRC is the firmware file for AVR (zxevo_fw.bin) and has no practical meaning for the Z80.

Packaging format dates and bits of release:

- 7 bits 0xFD - 0: test version; 1: The official release;
- 6 .. 1 bits 0xFD - year release cycle, the values 0 .. 63 corresponds to 2000 .. 2063 years;
- 0 bits 0xFD and 7 .. 5 bits 0xFC - month release cycle (valid values: 1 .. 12);
- 4 .. 0 bits 0xFC - day release cycle (valid values: 1 .. 31).
-

Example:

```
50 65 6E 74 31 6D 00 00 00 00 00 00 7B 14 3C B1
```

50 65 6E 74 31 6D 00 00 00 00 00 00 — Line «Pent1m»;

7B 14 - the number # 147B or% 0001_0100_0111_1011: beta, 1910, 03 months, 27 days (27.03.2010);

3C B1 - number # B13C, CRC.

7.7 Accessing the SD-card.

Access is similar to the Z-controller from the CFU. In normal mode:

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td>#xx77
RW
noshad</td><td>Record: control signal CS to SD-card:
<table>
<tr><td>Bits 7 .. 2: set to 0 for compatibility</td><td>Bit 1: signal CS, 1 after the reset, set to 0 to select the SD-card</td><td>Bit 0: Set to 1 for compatibility with Z-controller</td></tr>
</table>
Read: always 0 (the card is inserted in the regime R / W - in accordance with the interpretation of the Z-controller). The actual presence of maps should attempt to verify its initialization time an out.</td></tr>
<tr><td>#xx57
RW
noshad</td><td>Record: sending bytes to the SD-card on SPI, both of bytes can be further considered from the same port.

Reading: Read previously adopted bytes sent # FF in a map. Newly adopted bytes available during the second reading.</td></tr>
</table>

Note: in the cycle of exchange on the SPI, initiated by writing or reading port **# xx57**, occurs simultaneously sending bytes to the SD-card and receive bytes from it. Sending byte is the same as that recorded in the port (if the cycle is initiated by the exchange of records), or **# FF**, if the cycle of exchange initiated by reading the port.
Received bytes is stored in an internal buffer and is available for the reading from the same port. This reading of the re-initiates the cycle of exchange, etc.
Allowed to read / write port **# xx57** teams INIR and OTIR. Example of reading the sector:

```
LD   C,#57
LD   B,0
INIR
INIR
```

SD-card is also available in shadow-mode. In this mode, port **# xx77** used for other purposes, because there remains only the port **# xx57**, and addressing changes as follows.

Port	*Description*
#xx57 RW shadow	Record: if A15 = 0, then sending bytes to SD-card on SPI, both of bytes can be further considered from the same port - the same record in the **# xx57** in noshad-mode. If A15 = 1 - control signal CS, the same record in the **# xx77** in noshad-mode. Reading: Read previously adopted bytes sent # FF in a map. Newly adopted bytes available during the second reading. Reading is completely analogous noshad-regime.

Thus the full access to the SD-card and stored in the shadow-mode. Record team OTIR 512 bytes looks like (OTIR works with pre-decrement, B):

```
LD   C,#57
LD   B,#80
```

```
OTIR
LD   B,#80
OTIR
LD   B,#80
OTIR
LD   B,#80
OTIR
```

7.8 Access to IDE-devices.

IDE-interface is modeled on the standard nemo-ide with some extensions. Unlike the original (and the Z-Controller), IDE-ports are available and in shadow-mode. However, unlike the original, are supported ports # 10, # 11, # 30, # 50, # 70, # 90, # B0, # D0, # F0 and # C8, at the same time providing a complete decoding (8 bits Jr. address). Advanced data transfer mode from the original is the ability to read and write sector completely through the port # 10 teams and INIR OTIR, in the latter case there is no need to rearrange the major and minor bytes. Advanced mode does not require any special inclusion, ie, works simultaneously with the old (nemo-ide).

Port	*Description*
#xx10 RW always	Port for reading or writing data read / written as the lowest of the 16-bit word (the compatibility mode to nemo-ide), and older too (advanced mode).
#xx11 RW always	Reading and writing the high part of 16-bit word (the compatibility mode to nemo-ide). Reading occurs after reading # 10, the record here - to write at # 10. Appeal to this port does not lead to physical abuse in the IDE-devices.
#xx30 RW always	Port Error / properties. This and further ports, see the documentation on the protocol of ATA. SHA bits Z80 7 .. 5 mapyatsya to address bits ATA bus 2 .. 0 ports # 10, # 30, ..., # F0 mapyatsya on CS0 bus ATA, port # C8 - on CS1.
#xx50 RW always	Sector counter
#xx70 RW always	Number of sectors (CHS) or bits 7 .. 0 LBA-address
#xx90 RW always	Bits of cylinder number 7 .. 0 (CHS) or bits 15 .. 8 LBA-address
#xxB0 RW always	Bits of cylinder number 15 .. 8 (CHS) or bits 23 .. 16 LBA-address
#xxD0 RW always	Number of head (CHS) or bits 27 .. 24 LBA-address choice of devices (master / slave), mode selection (LBA / CHS)
#xxF0 RW always	Register status / command register
#xxC8 RW always	Register status / control

Reading data in nemo-ide looks like this:

```
IN   A,(#10)    ;read low byte
LD   (HL),A
INC  HL
IN   A,(#11)    ;read high byte into #11
LD   (HL),A
INC  HL
```

Writing data in nemo-ide:

```
LD   D,(HL)
```

```
INC  HL
LD   A,(HL)
INC  HL
OUT  (#11),A   ;write to latch high byte
LD   A,D
OUT  (#10),A   ;write low byte at the same time with the
               ;High byte of latches
```

Read and write in advanced mode:

```
LD   C,#10
LD   B,0
INIR ;Read sector from # 10 (2 times 256 bytes)
INIR
...
LD   C,#10
LD   B,0
OTIR ;writing in Sector # 10 (2 × 256 bytes)
OTIR
```

Attention! In Advanced mode, synchronization issue low and high bytes (write first low and then high byte) at the beginning of a read or write sectors carried out at the time of access to ports # 30, # 50, etc. (any IDE-ports except # 10 and # 11).

7.9 floppy drive controller port

Port	*Description*
#xx1F RW shadow	Instruction register / status VG93
#xx3F RW shadow	Registry Tracks VG93
#xx5F RW shadow	Register sector VG93
#xx7F RW shadow	Data register VG93
#xxFF RW shadow	The System Registry. Reading: Bit 7 - state of the signal from INTRQ VG93 Bit 6 - state DRQ signal from VG93 Bits 5 .. 0 - single Record:

	Bits 1 .. 0: selection of one of the 4 drives Bit 2: RESET signal to the VG93, 0 after Reset Bit 3: HRDY signal to VG93 Bit 4: choice of a floppy disk

7.10 RS232 port (communication interface)

Working with communication RS232C interface for ZX Evolution is implemented according to the standard proposed by Kondratyev without interruption. The standard we used a standard PC ISA-compatible modem, were based on the chip 16550, or compatible. Accordingly, the internal registers of interface registers, the above correspond to the chip. Due to the fact that not all features 16,550 used in our implementation of the registers or register bits are not used. Therefore it is recommended to familiarize with the programming constraints implementation (documentation describes only used registers and bits).

Features :

- No supported interrupts;
- supported only by the following additional communication interface signals: CTS, RTS;
- FIFO is always used.

Port	*Description*
#F8EF **RW**	<**LCR**&0x80==0> **DAT** – data register <**LCR**&0x80==1> **DLL** – Low divider register
#F9EF **RW**	<**LCR**&0x80==0> **IER** – Interrupt Enable (not used) <**LCR**&0x80==1> **DLM** – High divider register
#FAEF **RW**	Writing: **FCR** – FIFO control register Reading: **ISR** – status register interrupt (not used)
#FBEF **RW**	**LCR** – line control register
#FCEF **RW**	**MCR** – modem control register
#FDEF **RO**	**LSR** – Line status register
#FEEF **RO**	**MSR** – Modem status register
#FFEF **RW**	**SPR** – User Registry

7.10.1 Data Register (DAT)

Register is used for transmitting / receiving data.

7.10.2 Divider registers (DLL,DLM)

Register is used to set the baud rate ports.

There are two modes divider:

- Standard mode (DLM MSB set to 0);
- native mode (DLM significant bit set to 1);
-

For standard duty rate is calculated as for 16,550 chips, according to the formula: <baud rate = 115200 / ((DLM) * 256) + DLL)

For native mode values DLL and DLM (excluding the top bit) is directly recorded in the registers and UBRRL UBRRH chip ATMEGA128. In this case, the values of the DLL and DLM is calculated as follows:

(uint16) ((DLM & 0x7F) * 256 + DLL) = ((F_CPU/16) / <baud rate>) -1
where F_CPU = 11059200 (for ZXEvo)

Example: Calculate the value of the DLL and DLM for the rate of 31,250 baud (speed MIDI Interface) when using native mode.

1. Calculate the value of the divisor ((11059200/16) / 31250) -1 = 21 [0x15];
2. Set the value of DLL = 0x15, DLM = 0x80 (MSB includes native mode).

Attention! ZX Evolution supports additional speed transmission 256,000, it needs to install the DLL and DLM to 0.

7.10.3 FIFO control register (FCR)

Register is used to control FIFO.

- 7 .. 3 bits: not used;
- 2 bits (XMIT FIFO reset): Reset FIFO dispatch, if set to 1;
- 1 bit (RCVR FIFO reset): Reset FIFO reception if set to 1;
- 0 bit (enable FIFO): includes a FIFO, if set to 1. Since the FIFO is always enabled, this bit must be set in case of a command to reset the FIFO. In the case of 0, reset command FIFO are ignored.

7.10.4 Line control register (LCR)

Register is used to set the format of data transmission over a communication interface.

- 7-bit (DL enable): installs a visible record of port # F8EF, # F9EF relevant parts of the divisor register DLL, DLM;

- 6 bits: not used;
- 5 .. 3 bits: specifies the algorithm for calculating the parity data. Supported only by the following values:
  001 - odd parity;
  011 - even parity;
  all other values - the parity is not used.
- 2 bits (stop bits): number of stopbit. Number of stop-bits ": for 0 = 1, with 1 = 1,5 in the 5-bit data, or = 2.
- 1.0 bits (word length): length of data. The possible values are:
  00 - 5 bits;
  01 - 6 bits;
  10 - 7 bits;
  11 - 8 bits.
-

7.10.5 Modem control register (MCR)

Register is used to set the phone interface and additional control signals.

- 7 .. 2 bits: do not use;
- 1 bit (RTS): RTS control signal interface;
- 0 bit: not used.

7.10.6 Status line register (LSR)

Register to receive line status and FIFO.

- 7 bits: not used;
- 6 bits (transmit empty): if set to 1, the transmitter and the FIFO sending empty;
- 5 bits (THR empty): if set to 1, then send the data FIFO is empty;
- 4 bits: not used;
- 3 bit (framing error): if set to 1, then an error frame;
- 2 bits (parity error): if set to 1, then there was a parity error;
- 1 bit (overrun error): If set to 1, the receive data FIFO is full;
- 0 bit (data received): If set to 1, the receive data FIFO is not empty;

7.10.7 Modem status register (MSR)

Register to obtain the status of additional interface signals.

- 7-bit (CD status): always 1;
- 6 bits (RI status): always 0;
- 5 bits (DSR status): always 1;
- 4 bits (CTS status): displays the status of CTS line interface;
- 1 .. 3 bits: not used, always 0;
- 0 bit (CTS change): set to 1 if there has been a state change signal CTS (cleared to 0 after reading the case).

7.10.8 User register (SPR)

Register is used to store user data. Cleared when power is turned off, or completely reset the computer.

8 Version history

8.1 Version 11.05.2011

- Added support for native mode for the divider speed RS232;
- Take into account the control frequency multipliers ICS501M (for PCB revision C).

8.2 Version 26.04.2011

- Added support for a communication interface (RS232);
- Improved reliability of downloading FPGA;
- Improved reliability with the RTC (fixed a bug that appears on some boards).

8.3 Version 30.03.2011

- Corrected description of Kempston mouse ports.
- Added description kovoksa.
- Corrected recommended values of the high byte address port # xx77.
- Added description of the supported video modes.
- Added access to additional cells NVRAM (0x40. .0 xEF).

8.4 Version 07.12.2010

- Added adjustment permits the PS / 2 mouse.
- Mapping of the PS / 2 keyboard in the ZX Evolution implemented without using the RAM of the microcontroller.

8.5 Version 29.11.2010

- Implemented choice («Num Lock») and display (Num Lock led) regime audio (beeper / tapeout).
- Supported by the mapper PS / 2 keyboard in the ZX for all the extra codes keyboard (E0 xx).
- Corrected memory management and visibility of ports in different modes.

8.6 Version 17.10.2010

- Support for booting from a tape input.
- Adjusted control reset with PS / 2 keyboard («F12» and «CTRL-ALT-DEL»).
- Supports the manager for the entire memory (ROM and RAM).
- Optimized to work with PS / 2 devices.
- Supported by the mapper PS / 2 keyboard ZX (yet without editing) ..
- Implemented extension nemo-ide

8.7 Version 30.03.2010

- Fixed an error loading the FPGA.

8.8 Version 28.03.2010

- Fixed error handling PS / 2 mouse. Reinitialize the mouse in case of errors exchange.
- Support to obtain a version baseline configuration and bootloader.
- Fixed error handling timeout exchange with the PS / 2 devices.

8.9 Version 24.03.2010

- Fixed behavior of the LED power (Pwr Led), now turns off when the main power is off.
- Added the signal NMI to ZX by clicking «Print Screen» PS2 keyboard.
- Soft reset the computer (without restarting the fpga) for the closure of the switch «soft reset» on the board.

8.10 Version 10.03.2010

- Added display mode TV / VGA to display «Scroll Lock» PS2 Keyboard (LED on- VGA, Off - TV).
- Fixed display bug NVRAM, used in GLUK, in NVRAM PCF8583.
- Fixed a bug in the emulation clock GLUK, implemented emulation Update flag in register C.
- Implemented saving mode computer in NVRAM.
- Added support for mechanical ZX keyboard.
- Added support for Kempston joystick.

8.11 Version 04.02.2010

Basic Version (1.00 в SVN).