

«EVO Reset Service» ROM for ZX Evolution

Contents

ROM «EVO Reset Service» v 0.56c user manual.....	2
Requirement.....	2
WARNING.....	2
Control buttons.....	2
Keyboards keys+RESET.....	2
Main menu control.....	2
Hot keys of “Main” menu	3
Hot keys of «Service» menu.....	4
Setup.....	4
File browser.....	5
Review SCL or TRD images content and running Basic files.....	5
Auto-mount TRD images.....	5
Loading IS-DOS from hard drive.....	6
Update/flash ROM.....	6
Select font for textmode.....	6
Select alternate ROM-pages for start EVO.....	6
Выбор TRD, SCL и FDI образов.....	6
Tape load and loading emulation.....	7
Original Basic 48 and Basic 128.....	7
Run HOBETA and SPG files.....	7
Memories state on running HOBETA files.....	7
CMOS cells editor.....	7
Test PS/2 keyboard.....	7
NMI button (like Magic button on ZX).....	8
ATM CP/M mode.....	8
RTC and CMOS.....	8
Structure of RAM-disk.....	9
Using EVO-DOS v0.41.....	10
Removed features.....	10
Added features.....	10
External utility for flashing ROM.....	12
RST 8 functions.....	13
Main functions.....	13
Device manager functions.....	18
FAT driver functions.....	20
Draw window.....	23
Print text string.....	24

4. Feb. 2016

<http://www.nedopc.com>

nedopc@mail.ru

Please tell us about errors.

ROM «EVO Reset Service» v 0.56c user manual

Requirement

PentEvo 4Mb with FPGA Baseconf configuration only;

TR-DOS controller based on EVO-DOS;

Support run ZX programs from:

diskette on floppy-drive (formatted on TR-DOS);

SD memory card by NeoGS (formatted FAT);

SD memory card (integrated on PentEvo by Z-controller i/o standard) (formatted FAT);

hard drive (IDE integrated on PentEvo by Nemo i/o standard) (formatted FAT, only primary partition).

WARNING

For return to main menu hold key “0” and press/unpress F12 key of PS/2 keyboard.

TR-DOS 6.12E included to Gluk Reset Service from EVO Reset Service 0.54 had access to RAM-disk (but not accessed to mounted images).

Execution files must had HOBETA format (had 17-bytes header) for run direct from FAT. Address and size of execution file is checked before running. Accepted running execution file to address #6000...#FFFF and with size up to RAM top (total address and size must not bigger then #FFFF). Loading made exactly to byte. Files not viewed in menu if it address and size not valid. Screen not clear before running execution file!!! Screen attributes area filled by 0 for hiding loader (it placed in screen area).

File browser view founded directories and files with extension TRD, SCL, FDI, TAP, \$C, SPG, FNT up to 2048 records. If directory content more records then browser view only first 2048.

TRD image may be loaded to RAM-disk or mounted, FDI and SCL images always loaded to RAM-disk only.

After flashing ROM recommends to reset computer!

Control buttons

Keyboards keys+RESET

Any non-functional button (except following) — show help message on screen.

CS-restart to ROM Basic 128.

SS-restert to ROM Basic 48.

D-start DEMO “GRASS” by PSB^Halloween.

C-color table (SPACE or ENTER change color of border).

H-show help message on screen .

S-CMOS Setup.

0-return to main menu of EVO Reset Service.

Any two keys or more start checking pressed keys procedure.

Main menu control

Accepted control from keyboard and mouse (if mouse plugged and found). Cursor keys let to move in menu. Enter key start function in menu. You can use Hot keys for direct start required function in menu. Hot

keys name is first letter of each menu name.

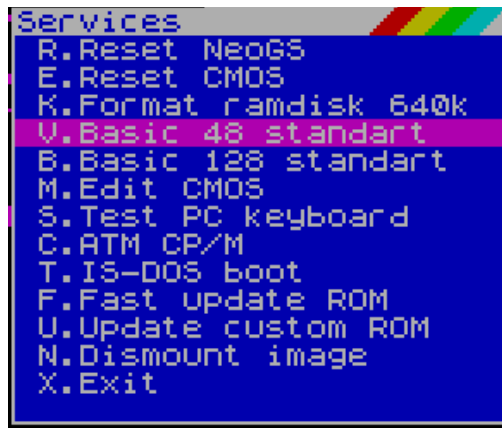
Hot keys of "Main" menu



- 1...4-select floppy drive;
- V-select by loop all real floppy drives;
- Y-select by loop all virtual drives;
- W-select CPU frequency (3,5-7-14 MHz);
- L-switch loading from tape in or loading emulation;
- G-select starting ROM-page after RESET;
- E-reload font from ROM for textmode;
- J-select codepage of textmode font;
- M-select memory manager mode (ALL/128/48);
- Z-load from current floppy drive;
- F-select file from FAT drive;
- T-load from tape (like command LOAD "");
- B-boot from hard drive;
- D-boot from CD (like GLUK function);
- X-load Perfect commander;
- S-reset to TR-DOS;
- I-reset to Basic 48;
- U-reset to Basic 128;
- R-view Service Menu;
- H-view help message;
- 8-switch on/off mouse cursor (if mouse plugged);
- 9-switch on/off viewing date and time.

You can change CPU frequency or memory manager mode in any moment by pressing NMI button and select Setup item from Magic mode menu.

Hot keys of «Service» menu



- R-reset NeoGS (out 128 to 51 i/o port);
- E-reset CMOS;
- K-format virtual drive (to 640 Kb);
- V-reset Basic 48 (to original ROM);
- B-reset Basic 128 (to original ROM);
- I-format virtual drive (to 896 Kb);
- M-CMOS editor;
- S-PS/2 keyboard test;
- C-enter to CP/M mode;
- T-load IS-DOS from hard drive;
- F-update/flash full ROM (512 Kb);
- U-update/flash user ROM pages (64Kb);
- N-unmount TRD images.
- X-exit to main menu.



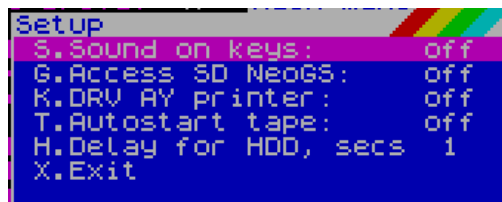
You can select menu items by mouse. And select floppy drive by clicking on text string «**TR-DOS Drive: A**» (letters A-D selected by loop);

Memory manager mode selected by clicking on text string «**Memory Lock: off**» (48/128/off);

CPU frequency selected by clicking on text string «**Turbo Mode: 7.0**» (3,5/7,0/14,0 Mhz);

Run execution files from TR-DOS or FAT drives.

Setup



Accepted next settings:

on/off sound on keys press;

on/off access to SD memory card on NeoGS. (Work after reset or now if autodetect of memory card selected);

on/off autoload TAP files if it selected in File browser;

set pause for detecting hard drive after power up (0..9 seconds). Pause have affect on next times: after power up, on reset hard drive, on reset to main menu (0+F12).

File browser

Starting with version 0.5 option FAT boot renamed to File browser. File browser view all directories and files with extensions TRD, SCL, FDI, \$C, SPG, TAP, BMP. Directories and files sorted by name (only by first letter). Files load dependently its extension but for TRD view submenu.

Starting with version 0.52b added show pictures in BMP format (only pictures 320x200 points 16 colors). Colors of BMP adapt to BMP palette. Any key return to File browser after showing picture.

FAT drives initiate on each reset. Letters sets from letter E. On first sets letters for hard drive, number of letters depend to number of partition (maximum 4 partitions). On next sets letters for SD memory card on NeoGS (if permit access to it in menu Setup). On last sets letters for on-board SD memory card, it select by default on first request to FAT drives. Mandatory computer must to reset after changing or removing SD memory card.

«File browser» always view data from volume what has last request, it is on-board SD memory card after reset. Key «D» change current FAT drive, key «ENTER» load file dependently file extension. «CS+1» or «BREAK» - return to main menu.

If selected drive hasn't directories and files with supported extensions then view window with info message for selected another drive. If FAT drive only one then menu for select drive not showed.

Review SCL or TRD images content and running Basic files

Starting with version 0.55 added review SCL or TRD images content and running Basic files. If press «V» key instead «Enter» for review image content. After pressing view window like TR-DOS boot what let to select any file (view only Basic files) and start it by press «Enter». After pressing SCL image load to RAM-disk or view dialog for loading or mounting method of TRD image. Press «X» key to return to File browser for selecting another file.

SCL images always load to RAM-disk. If load SCL image than letter assign same as letter of virtual floppy drive and record of drive assignment change in CMOS for next use. If virtual drive on this letter already used (image mounted) then current image unmount and all new requests will use RAM-disk.

TRD image can mount on any letter used for floppy drives. Record of drive assignment change in CMOS for next use after mount.

Примеры:

просмотр и запуск из SCL файла (также и для TRD образа при выборе загрузки в рамдиск) - виртуальный диск A:, реальный диск C:. После запуска Basic файла и реальный и виртуальный будут A: что будет видно после сброса.

просмотр и запуск из TRD с выбором монтирования — выбрано монтирование на букву D: реальный и виртуальный диск станет D: что также будет видно после сброса.

Auto-mount TRD images

«File browser» view content of last founded volume after reset. Also «File browser» search file IMAGE.MNT which describe TRD images for auto-mount (if option «Automount TRD» is on). This functionality need for setting up mounted images after reset (reset clear settings of images assignment). If select any directory in «File browser» and for example mount image and on next come to EVO-DOS command line (except reset) than selected path will accepted in command “dir”.

Information about mounted images not store after reset but you can prepare text file IMAGE.MNT which describe TRD images for auto-mount.

Letter A...D reserved for floppy drives, FAT drive letters start from E. For example content of file IMAGE.MNT:

[e:/dir1/dir2/filename.ext](#) a:

[f:/dir3/filename.ext](#) b:

[g:/filename.ext](#) c:

[f:/dir4/filename.ext](#) d:

Strings of auto-mount description may be in any order. Unnecessary use all floppy drives letters for mount.

Maximal deep of directories levels in path is 16. Name of files and directories must be short (8.3). Size of IMAGE.MNT file is not more than 512 byte. If file not found than auto mount not proceed (previous mounted files is not unmount). Also images can mount in EVO-DOS (see EVO-DOS description) same as in "File browser".

Mounted images has priority to RAM-disk. If image mount to letter of RAM-disk than all requests to this letter use mounted image and RAM-disk off. You can unmount any disk via menu «Service->Dismount image».

Loading IS-DOS from hard drive

Starting with version 0.54b added loader IS-DOS to Service menu. Loader code copied from ROM of russian clone KAY without any changes. Loader require only harddrive plugged via Nemo IDE interface and not depend from architecture of computer. Loaded data from harddrives IS-DOS bootsector must correspond to architecture of computer.

Update/flash ROM

Starting with version 0.22 added ROM flasher (for updating data in integrated ROM chip). Updating process indicated on screen (cleaning and flashing).

Updating all data of ROM chip is 512 Kb (64 Kb for user ROM pages). You can selected files with 512 Kb size (or 64Kb for user ROM pages) and extension *.ROM.

From version 0.46 and newer added flasher for user ROM pages (64 Kb). User ROM pages must content bytes #00,#C9 on #3D2F address in users service page. User pages must be in next order: Basic48, TR-DOS, Basic128, Service.

Users ROM pages is clean if you flash full ROM. Re-flash users ROM pages if you need after full ROM update.

Select font for textmode

Starting with version 0.39a added load font for textmode from any FAT drive. Select font files with extension *.FNT for loading.

Starting with version 0.4 added restore default font from ROM after reset, depend to option "**E.Reload FONT**" **on** or **off**. And can select codepage of ROM font: ATM (KOI8) or CP866, depend to option "**J.Type FONT**" **ATM** or **CP866**. Font not reload after selecting fonts codepage .

Select alternate ROM-pages for start EVO

Starting with version 0.39 support to use alternate ROM-page for start EVO (EVO start to selected ROM-page after reset). You can select by «G» key any of alternate ROM-page (it selected by name). Evo will start form this ROM-page on next reset.

Starting with version 0.46 added using users ROM-pages (Custom ROM) for start Evo. Size of custom ROM is 64Kb, custom ROM must have pages on next order: Basic48, TR-DOS, Basic128, Service. Reset to custom ROM by calling address #3D2F ,because custom ROM-page must content NOP:RET (codes #00,#C9) on this address (after jump to 0 address).

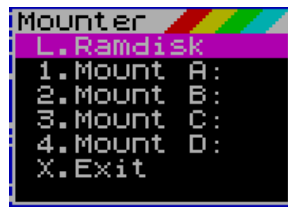
For return to default ROM-page hold key "0" and press/unpress F12 key of PS/2 keyboard.

Выбор TRD, SCL и FDI образов

Starting with version 0.21 added loading TRD and SCL images to memory for a virtual drive of the EVO-DOS. Starting with version 0.27 added FDI images loading. Pre-formatting RAM-disk for loading image are not required.

Starting with version 0.5 adds the ability to mount TRD images directly from FAT drives.

When you select a file with the extension FDI and SCL always are downloading the image to the RAM-disk (the ability to mount is not provided), selecting a file with the extension TRD will be offered the opportunity to mount an image to a drive letter, or upload the image to the RAM-disk. After loading or mount will return to the main menu. Select the menu files downloaded or mount the disc, through the main menu «Z.TR-DOS boot»



Structure of RAM-disk described in end of documentation.

Tape load and loading emulation

Starting with version 0.18 added load tape. This loader is full analog of BASIC command LOAD "". Before tape loading sets Basic environment variables (except Basic128 variables) and CPU frequency and memory manager mode. You can return to EVO service by BREAK till tape data not set own Basic variables and program stack.

Starting with version 0.38 added tape loading emulation through catching enter-point on address #0556 and loading TAP files from FAT drives. Option «Emu tape load» must set to «on» for tape loading emulation.

Selected file load to memory and start automatically. Also you can start repeatedly loaded file from Basic by command LOAD "" or from Basic128 by «Tape loader».

Original Basic 48 and Basic 128

Starting with version 0.56c added reset to original Basic 48 and Basic 128 pages for working in standard ZX-Spectrum mode (switching to standard ZX-Spectrum mode described in zxevo_base_configuration documentation). Reset to selected page right away after selecting. Original Basic 48 and Basic 128 selected in Service menu.

Select options for loading tape in main menu (L.Emu tape load: on/off) and submenu Setup (T.Autostart tape: on/off). Recommend option Autostart set to off before reset to original Basic. Tape loading emulation work always and not depend of Basic page.

Run HOBETA and SPG files

Execution files with extension \$(HOBETA) and SPG run after it selection.

Minimal address for loading is 24576 (#6000), maximal length of code 40960(#A000) bytes (maximal HOBETA file length 40977 bytes). Program load and start from same address (load address is equal start address). Program stack set on address 24575 (#5FFF).

From version 0.47 added load SPG files.

Memories state on running HOBETA files

When you run the code blocks of HOBETA file Basic variables sets in a standard way, the TR-DOS variables initialized. Stack is set to the address # 5FFF. The loader code block works in the display area. Loading is performed exactly up to a byte. Pre-cleaning the memory for downloading is not performed.

Cleaning the screen is not performed !!! The attributes area is filled by byte 0 to mask the loader running in the display area.

CMOS cells editor

Starting with version 0.34 added CMOS cells editor. The corresponding item added to the menu Service. To increase the storage space editor is displayed in the text mode 80x25. Navigating the editing area is performed with cursor keys. Editing is carried out at the cursor position by using the number keys 0 ... 9 and alphabetic A ... F. Cells tips are issued only for the first 17 cells, while other cells are not used. When you edit the cell content of edited cell is stored in the CMOS right now after you enter.

Test PS/2 keyboard

Starting with version 0.46 added PS/2 keyboards tester. On the tester mode displays a standard keyboard.

The pressed key is displayed in white, the non-pressed key is displayed in red and released keys is displayed in green. System keys F12, Print Screen, Scroll Lock, Pause is displayed in red purple (not recommended to press). Exit tester ALT + F4 or CTRL + X.

NMI button (like Magic button on ZX)

Starting with version 0.49 adds support for NMI button (like Magic button) with its menu. At the press of Magic saved the configuration of ports and processor registers to return to the currently executing program. The display output after pressing the Magic displays the contents of the saved registers (display register R is not equal) and contents of the readable ports in the form in which they are written to the ports.

In the Setup menu duplicated the ability to change the memory and processor speed.

In the upper right corner of screen displays the current time and date.

ATM CP/M mode

Starting with version 0.54 added CP/M implemented on a computer ATM Turbo 2+. The difference is only in the implementation of access to the hard drive, accessed through the controller ports Nemo. For those who want to partition/repartition the hard drive - FDISK program will not work in this case because it uses direct access to the HDD to determine its configuration. Out of this situation 2: knowing the parameters of the hard drive can then run FDISK (for the version of the standard floppy ATM Turbo 2+) to enter the desired settings in the dialog box, or use the FDISK modified as part of the standard boot floppy, which considers the necessary parameters.

Implementation ramdisk has not changed. Loss of information on the RAM-disk on reset occurs due to the fact that the start of ramdisk begins on page 6, that is, within 128 kilobytes (cleared by reset). Needs to take this when working with the RAM-disk.

RTC and CMOS

When the EVO-service firmware works, in the upper-right corner of the screen displays the time and date read from the chip clock attached scheme Gluk. In addition to displaying the time and date used by the cells contained in the CMOS memory to store some settings.

Used cells of CMOS:

- 0x00...
 - ...0x0D used by RTC (can't store parameters and settings).
- 0x0E... | used for storing
- ...0x3F | GLUK Reset Service settings
- 0x40... 64 bytes
- ...0x7F used for storing ProfROM settings.
- 0x80... | free
- ...0xE9 |
- 0xEA pause for detecting hard drive after power up in seconds (0..9, default value is 0)
- 0xEB floppy settings, bits:
 - 7 -
 - 6 -
 - 5 -
 - 4 -
 - 3 - | number of real
 - 2 - | drive
 - 1 - | number of virtual
 - 0 - | drivedefault value is %00000000
- 0xEC EVO modes, bits:
 - 7 - 1=TURBO OFF (3,5MHz), 0=TURBO ON (7MHz)
 - 6 - 1=access to SD NeoGS on, 0=off
 - 5 - 1=auto-mount enable, 0=disable

- 4 - 1=not display date/time, 0= display date/time
- 3 - 1=key sound off, 0=on
- 2 -
- 1 - | 2=128Kb, 1=48Kb,
- 0 - | 0=1Mb (by Pentagon clone standard)
- default value is %00000010
- 0xED EVO modes, bits:
 - 7 - 1=TURBO 14MHz, 0=TURBO 3,5/7MHz (depend of bit7 of cell 0xEC)
 - 6 - 1=tape load emulation on, 0=load from tape-in connector
 - 5 - 1=AY printer mode, 0=standard LPT Basic 48
 - 4 - 1=load font from ROM on reset off, 0=load font on
 - 3 - 1=codepage ATM, 0=codepage CP866
 - 2 - 1=auto-load tap files enable, 0=disable
 - 1 - | start page on reset: 0 - EVO Service, 1-GLUK Service
 - 0 - | 2 - ProfROM, 3 - Custom ROM
 - default value is %00000100
- 0xEE... |Control sum (CRC16) of cell 0xEB...0xED. If CRC16 is not equal then cells reset to
- ...0xEF |If CRC16 is not equal then cells reset to default value

Structure of RAM-disk

Structure of RAM-disk is same like TRD image structure, i.e. a sequence of sectors located in upper memory. RAM-disk begins with the 10th page from the end of the memory (the page number of the beginning of the RAM-disk can be changed at any later version) and grows down.

For universal support for SCL, TRD and FDI images the first page from the beginning of the RAM-disk (full 16Kb page) includes a handle location of sectors/tracks in memory pages. Determine the start page of the RAM-disk can be by byte «RD» located in the first page from the beginning of the RAM-disk in the address #3FFF (bytes «R») and #3EFF (bytes «D»), to these bytes is determined whether the RAM-disk is formatted. In the absence of these bytes is made to force formatting of the RAM-disk to the size of a standard TR-DOS floppy disk, which content 2544 sectors.

Header of FDI image converted when loaded into memory, the header for the SCL and TRD images force created after loading. When formatting headers are also mandatory.

Structure of RAM-disk sectors/tracks descriptors:

Offset from beginning of page	Description
+#000	Offset in blocks (by 256 byte) of track from beginning of page
+#100	Offset in pages (by 16Kb) from beginning of RAM-disk +1. If RAM-disk begin from 3th page then count begin from 4 th page.
+#200	Number of sector on track
+#300	High byte of sectors size*2 (for 128 bytes sectors stored 1, for 256 bytes sectors stored 2, for 512 bytes sectors stored 4 and for 1024 bytes sectors stored 8)
+#400	Number of sector on track
+#500	High byte of sectors size*2
	And so on until you finish all the sectors of the track
+#001	Description of next track

Using EVO-DOS v0.41

Due to the fact that the basis of EVO-DOS served firmware TR-DOS 5.03, you can read a full description of it on the TR-DOS. This article will describe the differences and additional features.

Removed features

Removed support "Magic" button and the ability gives to download the stored memory dump by the magic button (command GOTO).

Removed support for the one-side drives and 40-tracks drives, removed the command for selecting the number of tracks (40 and 80).

Added features

Added several new commands, new commands enter by character. Commands begin to enter the character "." to change the cursor type «K» to «L» and to be able to input by character. Inputs register does not matter. After the execution of the new commands (regardless of the result of execution) in the command line will display the letter of currently selected drive, until will execute a standard command TR-DOS. The brackets [] indicate an optional parameter. A separate error handling now is absent, always given the standard error message.

Full list of added commands:

- .CD**
- .CLRCMOS**
- .DIR**
- .DRIVE**
- .HELP**
- .MOUNT**
- .UMOUNT**
- .VER**
- .VIRT**

Command **.VER** output name of DOS, version and build date. Information taken from the last 8 bytes of DOS page.

Command **.VIRT [X:]** output the drive letter to which is currently mounted RAM-disk. The same command can assign RAM-disk on any letter, change immediately saved to CMOS. Command format **.VIRT X:** where X any drive letter A, B, C, D. Presence ":" after the drive letter is mandatory.

In addition to the possibility of a standard TR-DOS drive select command **"X:"**, where X is letters A ... D, can change the drive by command **.X:**, where of X any letter A ... D to change the drive, and the letter E and further change the current FAT drive. The presence of the symbol ":" is mandatory. Maximum letter of founded drives you can take by command **.DRIVE**, when you try to switch to unassigned drive error will be issued.

Command **.DRIVE** output founded volumes of hard drive and SD memory cards. First letter always is **E**, last letter depend to number of founded volumes. Commands outputting content is **«Drives: E-Z»**.

Command **.DIR [/L]** displays the contents of the currently selected directory of the current volume. Outputting content performed in the order as recorded in a directory. The output is short names. Displays long names produced by the **/L** option.

Command **.CD** allows you to go into the directory or return to parent directory. For one command can go only one level. For example:

- .CD dirname** – go to dirname directory
- .CD ..** - return to parent directory

Command **.MOUNT** allow to mount the TRD image for read/write instead of the real floppy drive. The parameters specify the name and file extension, and the drive letter on which you are mounting. For example:

.MOUNT FILENAME.TRD B:

The extension and “:” symbol after the letter of mounted drive must necessarily be specified.

Command **.UMOUNT** unmount image from drive specified by letter. Command have one mandatory parameter – drive letter, symbol “:” after drive is mandatory too.

Command **.HELP** output list of addition commands.

Command **.CLRCMOS** set CMOS cells to default state.

External utility for flashing ROM

The utility is designed to update the ROM. The utility does not provide the choice of flash file. The utility can be run from any convenient place (for HOBETA version). If the ROM-file is not found in the root of the SD card you will receive an error message and the offer to press any key to return to the main menu PentEvo.

ROM-file must have name ZXEVO.ROM, size of ROM-file must be 512Kb (524288 bytes), ROM-file must be in root directory of SD card.

RST 8 functions

All functions performed by RST instruction 8 and followed one or more bytes specified in call. All required parameters are passed via registers. When you return registers are not changed, except for those that will return the requested values. All defines are described in the file define.a80

Main functions

All functions has codes up the 0x40.

- 1 AY printer initialize. (Basic 48).

RST 8
DB _AY_PRN_INIT (0x40)

Registers:

in: none
out: none

- 2 Print symbol on AY printer. (Basic 48).

RST 8
DB _AY_PRN_A_ (0x41)

Registers:

in: A=symbol code from 0x20...0x7F
out: AF=set after code LD A,B:CP 3:SBC A,A (if emulation off)
none (if emulation on)

- 3 Print Basic's token on AY printer. (Basic 48).

RST 8
DB _AY_PRN_TOKEN (0x42)

Registers:

in: A=code of Basic's token
out: none

- 4 Print screen on AY printer. (Basic 48).

RST 8
DB _AY_PRN_SCR (0x43)

Registers:

in: none
out: B=0xB0 (if emulation off)
none (if emulation on)

- 5 Tape load emulation initialize. (Basic 48).

RST 8
DB _TAPE_INIT (0x44)

Registers:

in: none
out: none

6 Call the tape emulation. (Basic 48).

RST 8
DB _TAPE_EMUL (0x45)

Registers:

in: **IX**=load address
DE=length in bytes
out: **C=A**
F=sets after CP A (if emulation off)
IX=load address+block length
DE=0
flag **Z**=0
flag **C**=1

7 Draw window on screen. Details [here](#).

RST 8
DB _WINW (0x46)

Registers:

in: **IX**=address window descriptor
out: none

8 Print text string with control codes on screen (ZX mode or text mode of screen).

RST 8
DB _PRINT_MESSAGE (0x47)

Registers:

in: **HL**=address of text string (string must ended by 0)
out: none

9 Print symbol on screen.

RST 8
DB _PRINT_A (0x48)

Registers:

in: **A**=symbols code for print
out: none

10 Move window up (with or without color).

LD A,color mode
RST 8
DB _SCROL_UP (0x49)

Registers:

in: **A=0** – move without color
A=1 – move with color

out: none

11 Move window down (with or without color).

LD A,color mode
RST 8
DB _SCROLL_DOWN (0x4A)

Registers:

in: **A=0** – move without color

A=1 – move with color

out: none

12 Sets CPU frequency, screen mode and memory manager mode.

A=1 – standard ZX screen mode on

A=2 – one-page text mode on

A=0x10 – set CPU frequency to 3.5 MHz

A=0x20 – set CPU frequency to 7.0 MHz

A=0x30 – set CPU frequency to 14.0 MHz

A=0x40 – set memory manager mode to 48 Kb

A=0x80 – set memory manager mode to 128 Kb

A=0xC0 – set memory manager mode to 1024 Kb

LD A,parameters
RST 8
DB _SET_MODE (0x4B)

Registers:

in: **A=parameters**

A=0 – get current parameters

out: **A=current parameters**

13 Call the device manager

RST 8
DB _COM_DEV (0x50)
DB subroutine number

Registers:

in:

out:

14 Call FAT driver

RST 8
DB _COM_FAT (0x51)
DB subroutine number

Registers:

in:

out:

15 Files sort routine

RST 8
DB _SORT_FINDFILES (0x52)

Registers:

in: none
out: none

16 Sub-routines for working with floppy-image files.

RST 8
DB _MOUNTER (0x53)
DB subroutine number

Subroutines:

16.1 Mount floppy-image file.

RST 8
DB _MOUNTER (0x53)
DB _OPEN_MOUNT (0x00)

Registers:

in: A=number of drive to mount (0-3). Mounted image which specified by internal pointer of FAT driver.
out: none

16.2 Read/write to mounted image.

RST 8
DB _MOUNTER (0x53)
DB _RDWR_MOUNT (0x01)

Registers:

in: A=bits 1-0=drives number (0-3)
bit 7=0-read image, 7=1-write image
D=tracks number
E=sectors number
HL=address of read/write
out: none

16.3 Search mounted images from file IMAGE.MNT

RST 8
DB _MOUNTER (0x53)
DB _FIND_MOUNTED (0x02)

Registers:

in: none
out: none

16.4 Get descriptors of mounted images (table of 256 bytes, 64 bytes per descriptor)

RST 8
DB _MOUNTER (0x53)

DB _GET_MOUNTED (0x03)

Registers:

in: HL=address for table (256 bytes)
out: none

Structure of descriptor of mounted image (64 bytes per descriptor):

+0x00... descriptor of file
...+0x1F copied from FAT
+0x20 flags (reserved), bits:
7-
6-
5-
4-
3-
2-
1-
0-
+0x21 image type (now supported only TRD images)
+0x22 size of cluster (if size of files by clusters more then 64)
+0x23 FAT type (0,1,2 correspond for FAT 12/16/32)
+0x24 number of selected volume
+0x25 address of driver mounted images
+0x27 address of device/volume driver
+0x28 number of mounted drive

16.5 Unmount image.

RST 8

DB _MOUNTER (0x53)

DB _CLOSEMOUNT (0x04)

Registers:

in: A=number of mounted drive
out: none

16.6 Load image to RAM-disk.

RST 8

DB _MOUNTER (0x53)

DB _LOADIMAGE (0x05)

Registers:

in:
out:

17 Install loader of Hobeta and SPG files

RST 8

DB _INST_FATBOOT (0x54)

Registers:

in: HL-address for copying FAT variables of currently selected volume

out: none

18 Working with CMOS

RST 8

DB_CMOS_RW (0x55)

DB subroutine number

18.1 Check CMOS CRC and initialize CMOS if not equal.

RST 8

DB_CMOS_RW (0x55)

DB_INIT_CMOS (0x00)

Registers:

in: none

out: none

18.2 Read from cell of CMOS.

RST 8

DB_CMOS_RW (0x55)

DB_READ_CMOS (0x01)

Registers:

in: H=address of cell

out: A=L=value from cell

18.3 Write to cell of CMOS.

RST 8

DB_CMOS_RW (0x55)

DB_WRITE_CMOS (0x02)

Registers:

in: H=address of cell

L=value

out: none

18.4 Reset CMOS to default values.

RST 8

DB_CMOS_RW (0x55)

DB_CLEAR_CMOS (0x03)

Registers:

in: none

out: none

Device manager functions

Device manager is used to initialize the devices through their drivers, switching between them, the return of the number of devices and the currently selected device. When switching device changes the device driver and configured variables table corresponded to this device.

1. Search and initialize device drivers.

RST 8
DB _COM_DEV (0x50)
DB _DEVFIND (0x00)

Register:

in: none
out: **D** – number of current volume
E – number of founded volumes
A – FAT type of selected volume (0-FAT12, 1-FAT16, 2-FAT32)
flag **C=1** – nothing found.

2. Select volume.

Drive names A-D not accept, 0 correspond letter “E” FAT drive. Maximal number of devices/volumes limited to 16.

RST 8
DB _COM_DEV (0x50)
DB _SET_VOL (0x01)

Registers:

in: **A** – number of selected volume. Value can be from 0 to number of devices/volumes -1.
out: flag **C=1** – try to select unknown volume.

3. Get number of founded volumes.

Return same as function “Search and initialize device drivers”.

RST 8
DB _COM_DEV (0x50)
DB _KOL_VOL (0x02)

Registers:

in: none
out: **D** – number of selected volume
E – number of founded volumes
A – FAT type of current volume (0-FAT12, 1-FAT16, 2-FAT32)
flag **C=1** – nothing found.

4. Get volumes table.

RST 8
DB _COM_DEV (0x50)
DB _GET_FNDVOLUME (0x03)

Registers:

in: **HL** – address for copying table. Write 256 bytes to this address.
out: none

Sets 8 bytes per each founded volume, maximum number of founded volumes is 26 (correspond to letters A...Z). Structure of table:

+0 = 0-drive A
1-drive B

- 2-drive C
- 3-drive D
- 4-internal SD card (ZC)
- 5-SD card of NeoGS
- 6-HD controller Nemo (EVO)
- 7-HD controller SMUC
- 8-HD controller DIVIDE
- 9-HD controller PROFI
- 0x0A-HD controller TURBO2+
- +1 = for SD card always 0
for hard drives 0-Master, 1-Slave
- +2 = FAT type of volume:
0 - FAT12 (1)
1 - FAT16 (4,6,0x0E)
2 - FAT32 (0x0B,0x0C)
- +3 = first sector of volume
- +7 = reserv

Floppy-drives not supported.

5. Reinitialize variables of selected volume.

```
RST 8
DB _COM_DEV (0x50)
DB _TO_DRV (0x04)
```

6. Set bits of device for call device driver.

```
RST 8
DB _COM_DEV (0x50)
DB _SET_DEVICE (0x05)
```

7. Control of SD card.

```
RST 8
DB _COM_DEV (0x50)
DB _CONTROL_SD (0x06)
```

8. Direct call to HDD driver.

```
RST 8
DB _COM_DEV (0x50)
DB _COMHDDN (0x07)
```

9. Reinitialize volume.

```
RST 8
DB COM_DEV (0x50)
DB _FREINIT_VOL (0x08)
```

FAT driver functions

1. Initialize variables of current volume.
Used by device manager, not recommended for external use.

RST 8
DB _COM_FAT (0x51)
DB _INIT_FATVARS (0x00)

Registers:

in: none
out: none

2. Get file descriptor.

RST 8
DB _COM_FAT (0x51)
DB _READ_DIR (0x01)

Registers:

in: **HL**-address for file descriptor (32 bytes)
BC-file number
out: none

3. Select subdirectory or parent directory.

Internal pointer must be set on name of sub-directory or on «..» for parent directory.

RST 8
DB _COM_FAT (0x51)
DB _ENTER_DIR (0x02)

Registers:

in: none
out: none

4. Get long file name.

When long name is absent got short name (8.3).

RST 8
DB _COM_FAT (0x51)
DB _GET_LONGNAME (0x03)

Registers:

in: **HL**-address for long file name (256 bytes).
none: none

5. Search file or directory by name.

Search by text string like <filename.ext>. String must ended by symbol lower than 0x21 (space symbol is ending symbol too). Name and extension not need to fill 8.3 pattern (for example string “abc.\$C” is necessary to search).

RST 8
DB _COM_FAT (0x51)
DB _FIND_NAME (0x04)

Registers:

in: **HL**-address of string with filename to search
out: **BC**-number of record
flag **C**=1 nothing found

6. Search all files and sub-directories in current directory.

You can search by extension. Directories founded always (not depend to extension). Descriptor "." is ignored.

RST 8
DB _COM_FAT (0x51)
DB _FIND_FILEITEM (0x06)

Registers:

in: none
out: **BC**-number of founded descriptors
A-page number which content table of descriptors

7. Set extensions for searching (up to 16 extensions).

RST 8
DB _COM_FAT (0x51)
DB _SET_MASK_EXT (0x07)

Registers:

in: **A**=0 discard search by extension (otherwise set extensions for searching)
HL-address of string with extensions ended by 0.
out: none

8. Open file.

RST 8
DB _COM_FAT (0x51)
DB _OPEN_FILE (0x08)

Registers:

in: none
out: none

9. Sequential read from opened file.

RST 8
DB _COM_FAT (0x51)
DB _READ_FILE (0x09)

Registers:

in: **A**-number of sectors to read
HL-address to read
out: flag **C**=1 read fail

10. Create clusters table of current directory.

Not recommend to use.

RST 8
DB _COM_FAT (0x51)
DB _INIT_TEKDIR (0x0A)

Registers:

in: none
out: none

11. Working with current position in current directory.
Has some sub-routines.

RST 8
DB _COM_FAT (0x51)
DB _POS_FILES (0x0B)

Registers:

in: **A**-number of sub-routine
BC-position
out: **BC**-current position

Sub-routine:

00-save current position of file (save internal)
01-restore current position of file (restore internal)
02-reset current position to 0 and search first "legal" record
03-move to "B" files back
04-move to "B" files forward
05-count number of "legal" record
06-set index of "legal" record from "BC"
07-return to "BC" current index of "legal" record

"Legal" record – actual file descriptor in 8.3 format. Long file name is not used.

Draw window

Address of window parameters allow in register IX.

Structure of parameters:

+00 (byte) - X coordinate (in character positions)
+01 (byte) - Y coordinate (in character positions)
+02 (byte) - V windows height (in character positions)
+03 (byte) - H windows width (in character positions)
+04 (byte) - C window color
+05 (byte) - C window menu cursor color
+06 (byte) - F flags, bits:
 7 – 1-window with bottom header (string on window bottom by inverse color), 0-without
 6 – 1-window without frame, 0-with frame
 5 - 1-window with up header (string on window up by inverse color), 0-without
 4 – 1-print text in window, 0-no text
 3 -
 2 -
 1 -
 0 -
+07 (byte) - N index of string in window. If bit7=1, then bytes +8-+9 specified address for storing menu item. If bit 7=0, then bytes +8-+9 store index of menu item in window.
+08 (word) index of menu item in window or it address (see above)
+0A (word) number of menu items
+0C (word) address of text to print in window
+0E (word) address of addresses list which correspond to menu items call
+10 (word) address of mouse activity areas list

+12 (word) address of hotkeys list

If the number of menu items greater than the height of the window is made -2 scrolling when trying to move beyond the height of the window -2. Made primarily to display a list of files from any drives, for the menu is not included. The number of menu items should not exceed the height of the window -2.

Structure of mouse activity area (6 bytes):

- +0 - reserved
- +1 - X coordinate
- +2 - Y coordinate
- +3 - V height
- +4 - H width
- +5 - K hot-key code

Print text string

When printing text on the screen, you can use some of the control codes. When printing on a standard screen ZX character cell size is 6x8 pixels.

Control codes:

- 1 – set print position in window to start position. For ZX screen mode it calculate from left-up corner of window by expression $X*8+1$, $Y*8$ (exactly per pixel on horizontal and exactly per character cell on vertical). For text mode calculates by expression $X+1$, Y (per character cell).
- 3 - when printing text on a standard screen ZX center the printing line across the width of the window. Alignment is performed for the character code 0x20 ... 0xFF. The code 0x20 is the end of the aligned line. When text mode used code 3 is ignored, centering line is not performed.
- 9,x – tab, x-number of character cells.
- 0x0D – carriage return and next line. For ZX screen mode new line started from left side of window +1 pixel, for text mode - left side of window +1 character cell.
- 0x16,y,x - force set the printing position. For ZX Screen printing position is set in pixels, for the text screen in the character cells.
- 0x17,c – set color for printing. If $c=0xFF$ color not change.