

```
#####  
# BSrom118 - Modified ZX Spectrum ROM - (c) Busy soft - Release 13.03.94 #  
#####  
Original rom: (c) Amstrad
```

#### Generally =====

It's classic ZX Spectrum 48k (+) ROM with this modifications:

- some new commands - see below
- additional syntax of some commands - see below
- cca four times more speed of pilot signal synchronize from tape
- program cursor changed from ">" to "\*" (for autolisting)
- increase frequency of error beep (RASP)
- full work with 16-bit hexadecimal numbers in range 0-65535
- more speed autorepeat after initialization (new,reset)
- enter of error line caused move of cursor into error position
- option for list by ignore control codes (for colors,at,tab)
- extendable NMI handle program - see below
- corrected some errors - see below
- two new characters - small underline and square (code 30,31)
- program cursor "\*" displayed only on autolist
- full movement on edit area (left,rigth,up,down)
- more more more more speed reset and new (memory clears by push)
- stopable autolist by the "Q" key in all moment
- some enhancements for ZX Spectrum 128k
- line number range 0-16383 (original rom was 1-9999)  
(You can correctly edit and enter line with number zero!)

#### Corrected errors =====

- Error in original NMI handle routine. But error byte is not modified [if some programs will test it]. There new NMI menu handles it now.
- PRINT AT 0,0;CHR\$ 8; moves cursor into position AT -1,31  
There position of cursor is not changed by CHR\$ 8 in this rom.
- Display of error message if interrupt is disabled. There interrupt is enabled before displayng error message in this rom.
- Original rom writes five bytes into address range 0-4.  
There range address 0-4 is not changed in this rom. This feature is useful for not disabled write access into RAM address 0-#3FFF.
- If last line in basic program is N, you enter empty line N+1 and press EDIT key, then line N is edited into edit area WITH program cursor. There program cursor is not transfer into edit area in this rom.

#### Initialization (reset,new) =====

After start is system report no classic message. There is displayed information about length of basic, variables and free memory.  
48k and 128k reset do not test memory. I think damaged memory is past...

#### Extendable NMI handler =====

RESET and NMI signals causes jump into address #00 and #66 and this event is indicated by snowing on red border. If SP was in range 0-#4000 then is set to #5800 value. Control keys:

B - warm start of basic system. All variables are inicialized

but PROG, VARS and ELINE was not changed. It causes that basic program with variables is not deleted.

N - the same as CLEAR #5FFF: NEW - memory above #6000 is not changed.

R - the same as CLEAR #FFFF: NEW - classic 48k reset.

E - extended 128k reset. It clears all 128k memory on Spectrum 128k

I - sets AY to quiet and on MB-02 resets FDC,DMA and stops disk drive

S - save "" screen\$ on tape (on MB-02 can save on disk)

T - like USR 15522 - set tape as actual device (only on MB-02)

D - like USR 15524 - set disk as actual device (only on MB-02)

Q - return into broken program. Warning: return will be correct only if SP was not set to #5800! All registers (beside low 7 bit of refresh) are restored to original values.

M - jump into MRS 08/09 debugger. MRS must be loaded into memory.

All registers on register panel will be set to values from broken program - you can continue broken program by using any trace mode of MRS debugger. Warning: This feature will works only if SP was not set to #5800 (like as Q-quit).

0..9 - switching of RAM banks on ZX Spectrum 128k. If SP is in range #C000-#FFFF then it will be set to #5800.

This is the same as OUT (#7FFD),number - you can see second videoram by pressing 8 or 9 key.

U - user function. There are JP XXXX at address #3901. You can set address of this jump to you own routine. Your routine can end with ret - it causes return into NMI handler.

Z - user function like 'U' but this key is reserved for using in MB-02 applications (for example make snapshots,...)

However, if you want use this key, place address of you routine into address #3904.

#### Additional syntax of old commands

=====

Legend: num ..... any expresion with number result

string ... any expresion with string result

Items in brackets < > are optional.

CONTINUE <num> ... if num is nonzero then run machine code from this address. Different from USR is if machine code returns (very seldom event:)) IY must be set to #5C3A but HL' can be different #2758.

CLS <num> ... if num is nonzero then all attributes and border is set to num. For example: CLS 7 = BORDER 0: PAPER 0: INK 7: BRIGHT 0: FLASH 0.

GOTO, GOSUB, PAUSE, BORDER ... numer parameter of this commands is optional, not compulsory. If parameter not present, means zero value.

<PRINT> list of items ... Yes, it means that PRINT command is optional !

If list of items is not beginning with characters , ; ' then you may not write PRINT command in your program :)

POKE add,byte ... classic original poke (use comma)

POKE add;word ... stores 16-bit number into address add (use semicolon)

POKE add,string ... stores string into address add (use comma) - first character into address add, second into add+1, third into add+2...etc.

VAL num ... 16-bit PEEK. Returns 16-bit word from address num.

This function is opposition to 'POKE add;word' command.

VAL\$ num ... If parameter is number type then result of this function is four character string contains hexadecimal digits of this number.

This is very useful for convert number into hexadecimal base.

%,& ... prefix characters for writing hexadecimal numbers.

The same as BIN for writing binary number. Number must be max 16-bit.

#### New commands and its syntax

=====

, <num> ... comma ... edit of basic line num.  
 ' <num> ... apostrophe ... switching of RAM banks on ZX Spectrum 128k.  
 The same as OUT #7FFD,num. (Be sure that RAMTOP is under #C000!)  
 □ ... 'C' in circle ... the same as 'B' in NMI handler.  
 Initialize all system variables besides PROG, VARS and ELINE.  
 ^ ... up arrow ... the same as E in NMI handler. Clears all 128k memory.  
 ? ... displays information about length of basic, variables and free  
 memory. This command is called internally after each reset or new.  
 \* ... header ... displays header information from tape in following format:  
     type of file : name of file : address : length  
 \_ ... the same as USR 00102 - jump into NMI handler ('Q' continues)  
 # ... the same as USR 54885 - jump into MRS 08/09 (MRS must be loaded)

#### One new system variable =====

NMI handler does not need vector at address 23728.  
 This address is used for storing three bits with following meaning:

bit 0=0: number of basic lines is displayed with four digits  
 bit 0=1: number of basic lines is displayed with five digits  
 bit 1=0: control codes in autolist is displayed normally  
 bit 1=1: control codes in autolist is displayed by bit 2  
 bit 2=0: do not display control codes in autolist  
 bit 2=1: display all control codes as "?" character

This variable is initialized to zero value after reset or new.

Five digits displaying of basic numbers is very useful  
 if you use line range 0 to 16383. This feature has no  
 effect to running of basic program. If you edit any basic  
 line then this line is edited with five digit numbers always.  
 If number is under 10000 then it's space blanked from left.

-----

There is a little free space on address #3C98-#3CEF in this rom.  
 The space is reserved for MB-02 extensions. If you have no MB-02  
 then you can use the space for some your routines as you want.

```
=====
Address of author:   E-mail: busy@elf.stuba.sk
Slavko Labsky      http://www.elf.stuba.sk/~busy
Gallayova 13
841 02 Bratislava   Telephone:
Slovak republic     (07) 764 705 (only in slovak!)
=====
```