

SIZMON 3 documentation

© Széll Imre (Siz)

History

v3.0 – 2011.12.31 – first public version

Features:

- ROM version: uses 16k ROM area and several bytes RAM for work variables and routines
- ROM configuration is autodetected (can be used in function low, c1lo and c2lo slots)
- Starts with pressing Escape during reset-routine
- Dynamic parameter input: every numeric parameter can be entered in hexadecimal (prefix: none or \$), decimal (prefix +), binary (prefix: %), ASCII (PETSCII, prefix: ") or character screen code (prefix ') form. Only the least significant digits are used (8 or 16 bit, depending on the expected parameter type)
- Listing can be continued with scrolling from screen top up(except disassembly)/bottom down. Shift (or shift lock) pauses listings (it's useful with directories and file contents)
- All outputs can be redirected to printer by prefixing the command with an asterisk (*) character (for example: `y p u` selects a centronics compatible printer on the user port and `*d cef4 cff7` prints the disassembly)

Commands:

<i>Command</i>	<i>Command name</i>	<i>Description</i>
£	List commands	Prints a list of valid commands
@[<command>]	Disk status and command	@ alone prints drive status @<command> executes disk command and drive prints status
A [<address> [<statement>]]	Assembly	Enter assembly statement
B [C S <address>]	Breakpoint	B alone displays breakpoint Breakpoint clear Breakpoint set at address
C <src-address-from> <src-address-to> <dst-address-from>	Compare memory	Compare memory area src to dst
D M [<address-from> [<address-to>]]	Disassembly	D <address-from> <address-to> disassembles the memory DM displays current settings DM\$ turns on \$ sign DM# turns off \$ sign (default) DMI turns on illegal codes DMN turns off illegal codes (default)
E [<address>]	Edit screen	Edit screen

<i>Command</i>	<i>Command name</i>	<i>Description</i>
		In the menu: E-enter edit mode R-review (restore) contents S-store changes Q-quit from editor In edit mode press return to exit.
F <address-from> <address-to> <value>	Fill memory	Fills memory from <address-from> to <address-to> with byte <value>
G <address>	Go	Start execution
H <address-from> <address-to> <value>	Hunt for data	Search memory for specific data. data can be a string or any other byte sequence
I <address-from> <address-to>	Binary dump	Dump memory in binary format
J <address-from> <address-to>	Binary dump in multicolor mode	
K <address-from> <address-to>	ASCII (PETSCII) memory dump	
L [<drive>] "<filename>" [<startaddress>]	Load	
M <address-from> <address-to>	Memory dump (hexadecimal)	
N <address-from> <address-to> <search address-from> <search address-to> <address-offset>	New address (Program relocation)	
O [<address>]	Screen finder	
P [<address>]	Picture finder	
Q [<address>]	Character set finder	
R	Registers	
S [<drive>] "<filename>" <startaddress> <endaddress>	Save	
T	Transfer memory (copy data)	
U	Instructions	
V [<drive>] "<filename>" [<startaddress>]	Verify	
W [<address>]	Walk thru program (debug)	A simple debugger with real instruction execution (except jumps) in its real environment (memory config, IRq config,

<i>Command</i>	<i>Command name</i>	<i>Description</i>
		<p>stack, etc.) Displays all CPU registers and the next instruction to be executed.</p> <p>Available keys:</p> <p><space> - execute instruction</p> <p><j> - execute jsr (real call!)</p> <p><cr> - skip instruction</p> <p><down> - skip instruction</p> <p><stop> - quit debug mode</p> <p><esc> - quit debug mode</p>
X	Exit (reset)	
Y	Setup	<p><u>Print settings</u></p> <p>Y</p> <p><u>Setting default memory</u></p> <p>YA Set RAM</p> <p>YO[<bank>] Set ROM <bank></p> <p>YI[<drv>] Set drive <drv></p> <p><u>Setting printer</u></p> <p>YP[<prt>] Set to serial <prt></p> <p>YPU Set to Centronics printer on user port</p> <p><u>Set current drive</u></p> <p>YD[<drive>]</p> <p><u>Drive and interface scan</u></p> <p>YDS</p> <p><u>Expansion area scan</u></p> <p>YE – memory scan from \$fd00-\$feff</p> <p>YES – scan for known expansions</p>
Z <address-from> <address-to>	Memory dump with screen codes	
;	Modify registers	
>	Modify memory (hexadecimal)	
<	Modify memory (ASCII/Screen)	
\$[<filenamemask>]	Directory	
[Modify memory (binary)	

<i>Command</i>	<i>Command name</i>	<i>Description</i>
]	Modify memory (multi-color binary)	
.	Enter assembly statement	
/ [<drive>] "<filename>" [<startaddress>]	Fast loader	Currently available only for serial and parallel 1541
= [<drive>] "<filename>"	Display file contents	
- <destination drive> "<filename>"	Copy file to another drive (unaccelerated)	