

# Aleste Debugger (DBG)

Short manual

## Features

- Size 23KB
- Easy to use (Simple GUI)
- Separated video buffer
- CPU Register's Dump
- Memory Dump
- Breakpoints
- Custom Labels
- Memory management (mapper support)
- Embedded Dissassembler
- Embedded Assembler
- Embedded Calculator
- Embedded memory operations

## Introduction

Debugger can be started by DBG command. The disk should have DBG.COM application. After start the debugger shows a framed window (look at screenshoot). Each frame display a frame specific information. The debugger located at (CORRECT ME) memory and use (CORRECT ME) page for video buffer, so the video buffer of the OS would not be changed by DBG.

Input	File:0:■	Area:0100-D5FF	PC=0100 DI I=00 SP=D5FF:C370 MAP:04/05/06/07 A=00 , szxhxpnc BC=0000: C3 "q" DE=0000: C3 "q" HL=0000: C3 "q" IX=0000: C3 "q" IY=0000: C3 "q"
>>00<<	LABEL: 0100:F3	DI	A'=00 , szxhxpnc BC'=0000: C3 "q" DE'=0000: C3 "q" HL'=0000: C3 "q"
	0101:ED 7B 06 00	LD SP,(0006)	
	0105:21 A1 01	LD HL,01A1	
	0108:11 00 D6	LD DE,D600	
	010B:01 F7 00	LD BC,00F7	
	010E:ED B0	LDIR	
	0110:ED 73 01 D6	LD (D601), SP	
	0114:21 EA FF	LD HL,FFEA	
	0117:11 2A D6	LD DE,D62A	
	011A:06 16	LD B,16	
	011C:0E 02	LD C,02	CM{...!...ж.В.М. Ms.ж!W■.жк...~ (.ж.В.~...М...г.. No enough memory \$6.)Ф.жю.ж... y! ...E...М.!жж"...> q2...!ж"...:жк...~М y!.....@М.!..."
	011E:7E	LD A,(HL)	
	0100:F3 ED 7B 06 00 21 A1 01 11 00 D6 01 F7 00 ED B0		
	0110:ED 73 01 D6 21 EA FF 11 2A D6 06 16 0E 02 7E A7		
	0120:28 1F 23 10 F9 0E 09 11 2E 01 CD 05 00 C7 0D 0A		
	0130:4E 6F 20 65 6E 6F 75 67 68 20 6D 65 6D 6F 72 79		
	0140:24 36 05 7D E6 1F F6 C0 12 23 13 0D 20 D5 21 00		
	0150:00 11 04 45 01 05 00 ED B0 21 77 D6 22 01 00 3E		
	0160:C3 32 00 00 21 00 D6 22 06 00 3A 2A D6 06 7E ED		
	0170:79 21 00 00 11 00 80 01 00 40 ED B0 21 98 02 22		

				REGS / MAP	
BP	LB	AD:HEX DIS		ALT REGS	
	HLB	AD:HEX			ADUMP

### Frames

State
Input field
List
Registers
Dump

### Subframes

BP – Breakpoints, LB and HLB – labels, AD – address, HEX – hex dump, ADUMP – ascii dump, REGS – registers, ALT REGS – alternate (') registers, MAP – mapper, DIS - disassembler

Difference between frames and subframes is: move to frame possible by special key sequence but moving between subframes in one frame by cursor movements. Also all subframes of one frame has the same scroll position.

### Frames and Subframes description

- **State**

Display current state of debugger (ESC, List, Dump, Regs, Trace, Exec, Go, Break, Quit, Input, Output). Depend on current state cursore can be relocated to specific frame.

- **Input field**

Some of states required user input then cursor will be moved to this frame and neede information will be requested. Enter will finish entering but TAB will move cursore to next column of requested information.

- **List**

- **Breakpoints**

Add or remove BP by space or any another key.

- **Label**

This is labels for memo. Just mark eny line by label. Put cursor and enter the label text then Enter or just left the line or frame.

- **Address**

Address of dissassembler or dump, change put on anddress cursor and edit it as hex number – it will change address. Or use up/down keys to scroll address.

- **Hex dump**

Edit it easy

- **Dissassembler**

Edit it easy and DBG will assemple it

- **Hex**

- **Label**

This is labels for memo. Just mark eny line by label. Put cursor and enter the label text then Enter or just left the line or frame.

- **Address**  
Address of disassembler or dump, change put on address cursor and edit it as hex number – it will change address. Or use up/down keys to scroll address.
- **Hex dump**  
Edit it easy
- **ASCII dump**  
Just edit it easy
- **Registers**  
CPU registers and mapper edit same as dup. Register MAP is a mapper has 4 numbers. Each is the number of RAM page for one 16KB page. On the screenshot 4/5/6/7 means:

0000-3FFF – 4 page  
 4000-7FFF – 5 page  
 8000-BFFF – 6 page  
 C000-FFFF – 7 page

## **States**

<b>ESC</b>	Was pressed ESC button and DBG is waiting for next character
<b>List</b>	Disassembler List
<b>Dump</b>	Memory Dump
<b>Regs</b>	Registers/Mapper Dump
<b>Trace</b>	Trace <i>The command will not be executed, but debugger will simulate the command's result (side effect)</i>
<b>Exec</b>	Execute <i>The command will be really executed.</i>
<b>Go</b>	Go
<b>Break</b>	Break at the breakpoint
<b>Quit</b>	Exit from DBG
<b>Input</b>	Input (Read) file
<b>Output</b>	Output (Write) file

## **Keyboard shortcuts**

The character + means at the same time, but character → means sequentially.

## **File operations**

ESQ → I    Read file  
 ESQ → O    Write file

## **Frames**

ESQ → L    List mode  
 ESQ → D    Dump mode  
 ESQ → R    Register's dump mode

## **Memory operations**

ESQ → S    Search  
ESQ → F    Fill memory  
ESQ → W    Compare memory  
ESQ → M    Move memory

### ***Execution***

CTR + P    Load current address to PC  
CTR + T    Trace one line  
CTR + X    Execute one line (who can explain me difference with trace)  
CTR + G    Go up to BP  
CTR + V    Show DOS video screen (press any key return back to DBG)

### ***Utilities***

ESC → V    Refresh screen  
ESC → C    Calculator  
ESC → Q    Exit

---

EOF



