# ZEUS MONITOR May 2023

## 8-BIT HOBBY COMPUTER BOOTSTRAP CODE

**Bootup sequence**

At the bootup the Monitor..

1. ..copies memory block $0000-$2FFF from bank F (EEPROM) to address $8000 (highmem),
   copies memory block $7F20-$7FFF from bank F (EEPROM) to address $FC00 (highmem) and updates reboot jump pointer at $8002; jumps SETUP; sets SP and INT table; writes variables and pointers table (highmem);
2. ..initiates IOs; tests IO ports and prints POST letters “S”(SIO module), “P”(PIO module), “G”(GPO module);
3. ..initiates memory banks 0-E (if present), i.e. copies memory block $8000-$8070 from $8000 (highmem) to address $0000 of each bank in order to keep RST and NMI primers functional when this bank is ON;
4. ..counts number of available memory banks in the system and enumerates them (writes ID byte to $0007) and prints the calculated number of banks;
5. ..jumps to $8000, configures the system and switches to bank E; prints system register state;
6. enters HALT state and awaits for interrupt.

**The List of implemented commands**

### >clr

Fills most of TFT screen (everything except command line) with background color.

### >dump (source)

Dumps on TFT screen 0x70 bytes starting from specified source memory address.

### >send (source) (bytes)

Sends to UART specified number of bytes starting from specified source memory address.

### >load (dest) (byte1) (byte2)…

Writes specified byte set to specified destination memory address.

### >move (dest) (source) (bytes)

Copies data block of specified number of bytes from source memory address to destination memory address.

### >jump (dest)

Sets Program Counter register (jumps) to specified memory address.

### >exec (byte1) (byte2)…

Executes specified byte set. By default saves the byte set ended with 0xC9 to $0100.

### >sysl (nibble)

Sets system register low nibble value. Doesn't affect system register high nibble value. The command is intended for memory bank switching.

### >sysh (nibble)

Sets system register high nibble value. Doesn't affect system register low nibble value. The command is intended for sys reg bits control with no impact on memory bank switching.

### >out (port) (value)

Outputs specified byte value to specified IO port.

### >inp (port)

Reads register byte from specified port.

### >crc7 (byte1) (byte2) …

Calculates crc7 sum augmented with bit0 = 1 (check SD phys. layer specs) for specified byte set.
By default saves the byte set (the message) ended with calculated crc7 byte to $0100.

**Table of IO ports**

| port | description |
|---|---|
| Fxh | SYSR register (write only) |
| 00h | UART channel data (SIO) |
| 01h | PS/2 channel data (SIO) |
| 02h | UART channel instructions (SIO) |
| 03h | PS/2 channel instructions (SIO) |
| 10h | TFT + SD controls data (PIO) |
| 11h | TFT data bus data (PIO) |
| 12h | TFT + SD controls instructions (PIO) |
| 13h | TFT data bus instructions (PIO) |
| 20h/**24h**/28h/**2Ch** | channel A data (GPIO) |
| 21h/**25h**/29h/**2Dh** | channel B data (GPIO) |
| 22h/**26h**/2Ah/**2Eh** | channel A instructions (GPIO) |
| 23h/**27h**/2Bh/**2Fh** | channel B instructions (GPIO) |

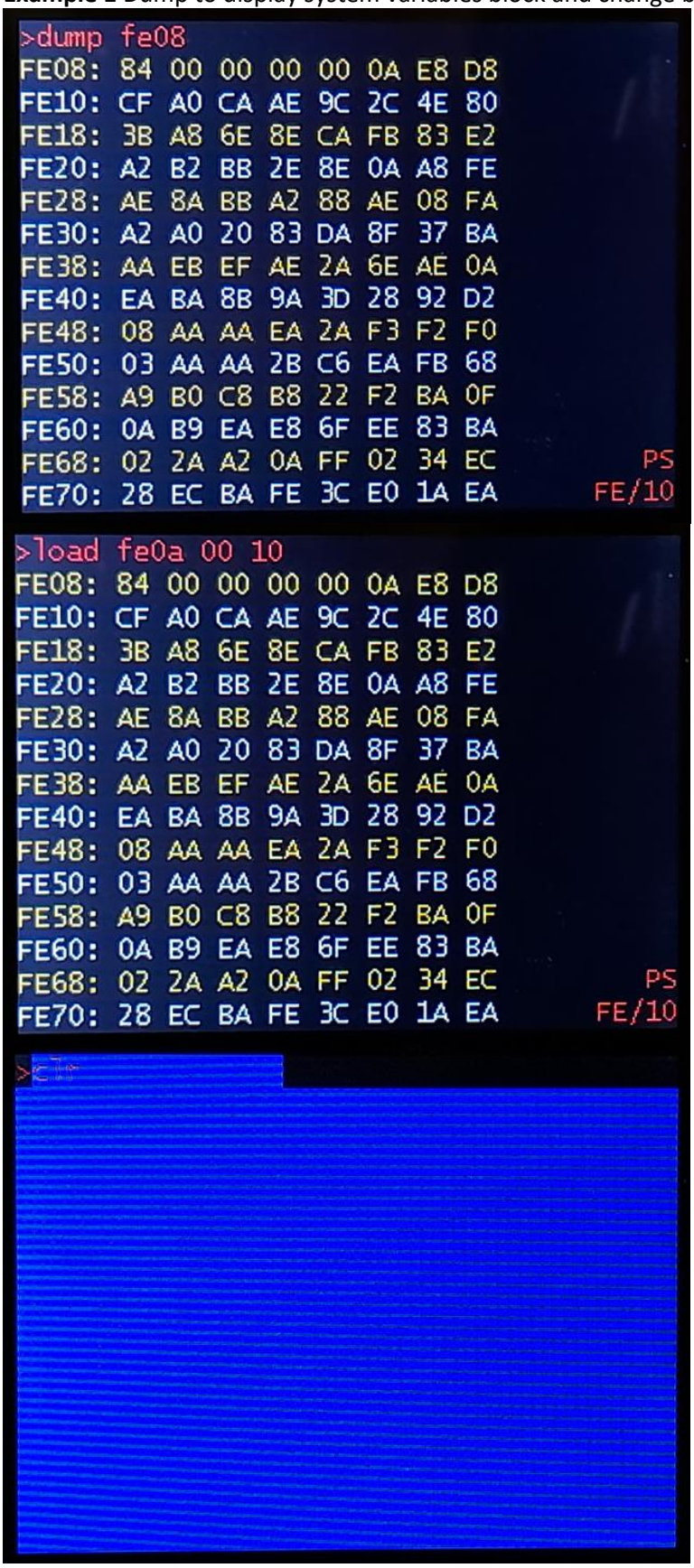
**Table of variables and pointers ($FE00h)**

| address | description | state on reset |
|---|---|---|
| FE00h | font table pointer | 8200h |
| FE02h | keyboard scan codes table pointer | 9200h |
| FE04h | keyb buffer pointer | FD00h |
| FE06h | UART buffer pointer | 0100h |
| FE08h | font color (R=0080h, Y=0084h, W=1084h) | 0080h |
| FE0Ah | background color | 0000h |
| FE0Ch | TFT char position row/col | 0000h |
| FE80h | prev keyb char | 00h |
| FE81h | system register state | FFh |

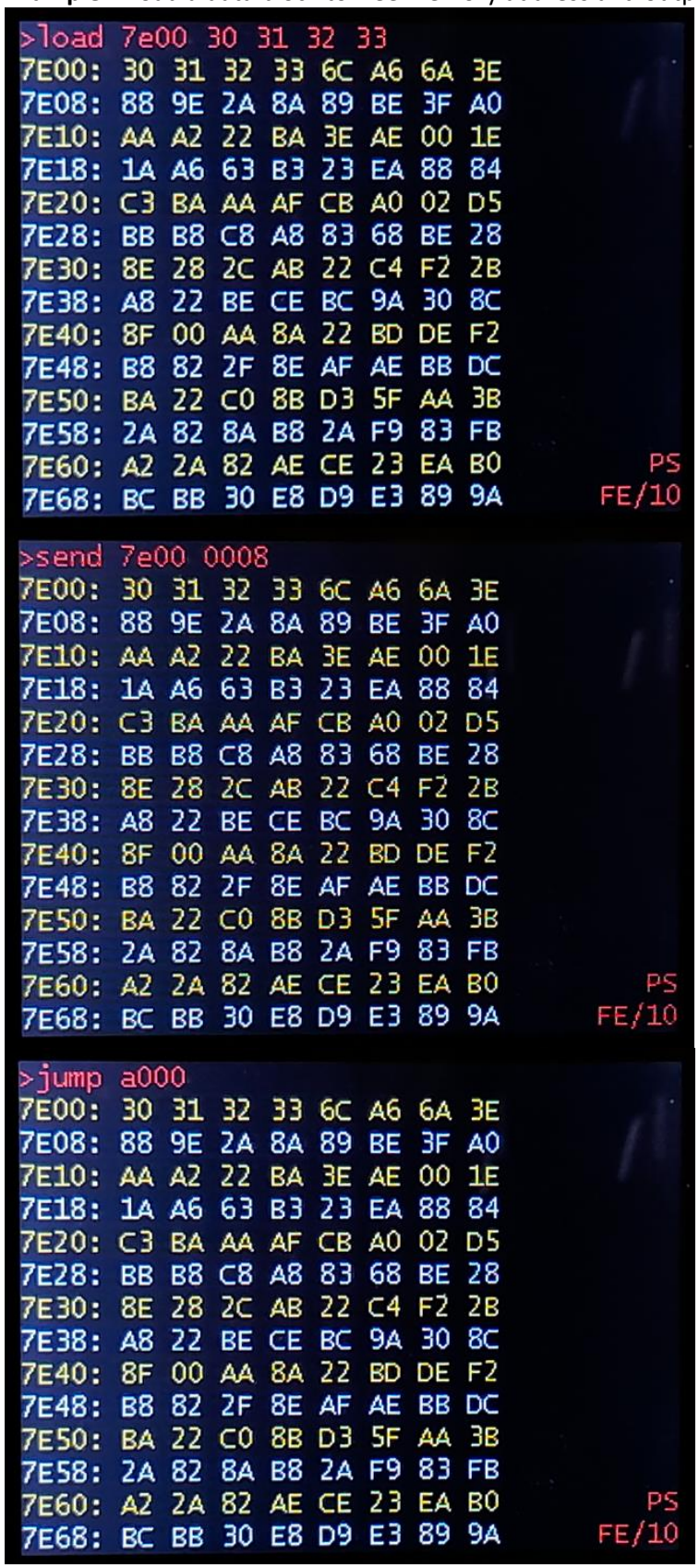
**Table of bootload memory blocks (highmem bank)**

| address | memory block description | jump table range |
|---|---|---|
| 8000h | page 0 | |
| 8100h | interrupt vectors table | |
| 8200h | font table (8x16) | |
| 9200h | keyboard scan codes table | |
| 9300h | bios subroutines | FC00h – FC2Ch |
| 9500h | command line parser | FC30h – FC4Ch |
| 9700h | command routines | FC50h – FC9Ch |
| 9B00h | SD command routines | FCA0h – FCDCh |
| A000h | setup | |

**Example 1** Dump to display system variables block and change background color.



**Example 2** Load a data block to free memory address and output it to UART, then jump to bootup address.



**Example 3** Calculate crc7 sum of a message and dump it to display.