

MSX BIOS

Программы BIOS хранятся в [ПЗУ](#).

Выражаем огромную благодарность А.Б. Родионову на предоставленную информацию!

BIOS

Так-же MainBIOS, MainROM, т.е. основной.

MSX BIOS Entry List

0010	CHRGTR	Gets next char from BASIC text	00E1	TAPION	Sets motor on & reads header from tape
0014	WRSLT	Sel. appropriate slot & writes to memory	00E4	TAPIN	Inputs from tape
0018	OUTDO	Outputs to current device	00E7	TAPIOF	Stops reading from tape
001C	CALSLT	Performs inter-slot call	00EA	TAPOON	Sets motor on & writes header to cas.
0020	DCOMPR	Compares HL with DE	00ED	TAPOUT	Outputs to tape
0024	ENASLT	Sel. appropriate slot & enables it	00F0	TAPOOF	Stops writing to tape
0028	GETYPR	Returns type of FAC	00F3	STMOTR	Starts cassette motor
002D-002F		Reserved area. Filled with zero	Routines used to handle queues		
0030	CALLF	Performs far-call (inter-slot) call	00F6	LFTQ	
0038	KEYINT	Performs hardware interrupt procedure	00F9	PUTQ	Places byte in queue
003B	INITIO	Performs device initialisation	Routines used by GENGPR & ADVGRP		
003E	INIFNK	Initializes function key strings	00FC	RIGHTC	Moves one pixel right
0041	DISSCR	Disables screen display	00FF	LEFTC	Moves one pixel left
0044	ENASCR	Enables screen display	0102	UPC	Moves one pixel up
0047	WRTVDP	Writes to VDP register	0105	TUPC	Moves one pixel up
004A	RDVRM	Reads VRAM addressed by HL	0108	DOWNC	Moves one pixel down
004D	WRTVRM	Writes to VRAM addressed by HL	010B	TDOWNC	Moves one pixel down
0050	SETRD	Sets up VDP for read	010E	SCALXY	Scales X-Y coordinates
0053	SETWRT	Sets up VDP for write	0111	MAPXYC	Maps coordinate to physical address
0056	FILVRM	Fills VRAM with specified data	0114	FETCHC	Fetches cur_phys_addr & mask pattern
0059	LDIRMV	Moves a VRAM block to memory	0117	STOREC	Stores physical address & mask patrn
005C	LDIRVM	Moves a memory block to VRAM	011A	SETATR	Sets attribute byte
005F	CHGMOD	Sets VDP mode according to SCRMOD	011D	READC	Reads attribute of current pixel
0062	CHGCLR	Change color of screen	0120	SETC	Sets cur_pixel to specified attribute
0066	NMI	Performs non-maskable interrupt proc.	0123	NSETCX	Sets pixel horizontally
0069	CLRSRPR	Initializes all sprites	0126	GTASPC	Returns the aspect ratio
006C	INITXT	Init.screen for TEXT1 & sets VDP	0129	PNTINI	Initializes PAINT function
006F	INIT32	Init.screen for GRAPHIC1 & sets VDP	012C	SCANR	Scans pixels to the right
0072	INIGRP	Init.screen for GRAPHIC2 & sets VDP	012F	SCANL	Scans pixels to the left
Additional entries					

0075	INIMLT	Init.screen for MULTICOLOR & sets VDP	0132	CHGCAP	Changes status of CAP lamp
0078	SETTXT	Sets VDP for TEXT1	0135	CHGSND	Changes status of 1 bit sound port
007B	SETT32	Sets VDP for GRAPHIC1	0138	RSLREG	Reads current output to prim_slot_reg
007E	SETGRP	Sets VDP for GRAPHIC2	013B	WSLREG	Writes to primary slot register
0081	SETMLT	Sets VDP for MULTICOLOR	013E	RDVDP	Reads VDP status register
0084	CALPAT	Returns address of sprite pattern table	0141	SNSMAT	Ret.status of specfd_row_of_kb_matrix
0087	CALATR	Returns address of sprite attrib. table	0144	RHYDIO	Perfrms oprtn for mass storag.devices
008A	GSPSIZ	Returns current sprite size	0147	FORMAT	Initializes mass storage devices
008D	GRPPTR	Prints a char on graphic screen	014A	ISFLIO	Checks if device I/O is being done
Routines used to access the PSG			014D	OUTDLP	Outputs to line printer
0090	GICINI	Init. PSG & static data for PLAY stmt	0150	GETVCP	Ret.addr_2nd_byte in spec_voice_buff
0093	WRTPSG	Writes data to PSG register	0153	GETVC2	Ret.addr_any_byte in spec_voice_buff
0096	RDPSG	Reads data from PSG register	0156	KILBUF	Clears keyboard buffer
0099	STRTMS	Checks/starts background tasks for PLAY	0159	CALBAS	Performs inter-slot call into BASIC
Routines used to access the KB & CRT			MSX-2 additional entries		
009C	CHSNS	Checks status of keyboard buffer	015C	SUBROM	Performs inter-slot call into SUBROM
009F	CHGET	Waits for char_be_input & ret.its codes	015F	EXTROM	Performs inter-slot call into SUBROM
00A2	CHPUT	Outputs char to console	0162	CHKSIZ	Does slot scan for SUBROM
00A5	LPTOUT	Outputs char to line printer	0165	CHKNEW	Check screen mode
00A8	LPTSTT	Checks line printer status	0168	EOL	Erase to_end_of_line
00AB	CNVCHR	Checks grph_headr_byte & converts codes	016B	BIGFIL	Same as FILVRM except few things
00AE	PINLIN	Get line_from_kb_till_CR & put to buff.	016E	1NSETRD	Set-up VDP to read
00B1	INLIN	Same as PINLIN, except if AUTFLG is set	0174	1NRDVRM	Reads VRAM_addr_by_HL:valid all bits
00B4	QINLIN	Outputs '?' & SP6CE then falls to INLIN	0177	NWRVRM	Writes A to VRAM_addr_by_HL:v.al.bts
00B7	BREAKX	Checks status of Control-STOP key			
00BA	ISCNTC	Checks status of SHIFT-STOP key			
00BD	CKCNTC	Same as ISCNTC, used by BASIC			
00C0	BEEP	Sounds buzzer			
00C3	CLS	Clears screen			
00C6	POSIT	Locates cursor at specified position			
00C9	FNKSB	Checks fun_key_displ_active? if so,does			
00CC	ERAFNK	Erases function key display			
00CF	DSPFNK	Displays function key display			
00D2	TOTEXT	Forcibly places screen in text mode			

Slot ID



Slot ID (or slot address) is MSX-BIOS way to represent the slot and sub-slot location in a byte. The byte contains following info:

bit 0-1	Primary slot number
---------	---------------------

bit 2-3	Sub slot number (optional)
bit 4-6	Unused
bit 7 = 1	if Slot is expanded

an example to interpret this from pascal is as follows:

```

procedure slotID(nValue:byte);
(** nValue is for example mem[$F344] **)
var
    num,num2: byte;
    exp : boolean;
begin;
    num:=nValue;
    num:= (num shl 6);
    num:= (num shr 6);
    if ((nValue and 128) = 128) then exp:=True else exp:=false;
    num2:=nValue;
    num2:=(num2 shl 4);
    num2:=(num2 shr 6);
    writeln('Slot-ID',num);
    writeln('subslot:',num2);
    writeln('expanded:',exp);
end;

```

 Slot ID

SUB-ROM

Его также называют SUB-BIOS.

MSX SUBROM Entry List

Graphic handler for BASIC			Handle ROMA-KANA Conversion		
0069	PAINT	Paints the graphic screen	0135	KYKLOK	Handle KANA key and lamp
006D	PSET	Sets the point	0139	PUTCHR	Get_key_fr_KB & conv_it_KANA & put_Buff
0071	ATRSCN	Scans color attribute	Access VDP		
0075	GLINE	Draws a line	013D	SETPAG	Set VDP registers to page changes
0079	DOBOXF	Draws a filled box	Access palette		
007D	DOLINE	Draws a line	0141	INIPLT	Init. palette and VRAM for pal_sav_area
0081	BOXLIN	Draws a box	0145	RSTPLT	Restore palette from VRAM
Low level graphics			0149	GETPLT	Get color codes from palette
0085	DOGRPH	Draws a line	014D	SETPLT	Set color codes to palette
0089	GRPPRT	Prints char on graphic screen	BASIC extended statement		
008D	SCALXY	Scales X-Y coordinate	0151	PUTSPR	Put sprites
0091	MAPXYC	Maps coordinate to physical address	0155	COLOR	Change screen color, sprite color, palette
0095	READC	Reads attribute of current pixel	0159	SCREEN	Change screen mode
0099	SETATR	Sets attribute byte	015D	WIDHTS	Change text screen width
009D	SETC	Sets cur_pixel to specified attribute	0161	VDP	Set VDP register
00A1	TRIGHT	Moves one pixel right	0165	VDPF	Read current VDP reg.

00A5	RIGHTC	Moves one pixel right	0169	BASE	Set VDP bas registers
00A9	TLEFTC	Moves one pixel left	016D	BASEF	Read VDP bas registers
00AD	LEFTC	Moves one pixel left	0171	VPOKE	Write a byte to VRAM
00B1	TDOWNC	Moves one pixel down	0175	VPEEK	Read a byte from VRAM
00B5	DOWNC	Moves one pixel down	0179	SETS	Sets beep sound, scr adjust, time & data
00B9	TUPC	Moves one pixel up	Miscellaneous		
00BD	UPC	Moves one pixel up	017D	BEEP	Beeps buzzer
00C1	SCANR	Scans pixels to the right	0181	PROMPT	Displays prompt
00C5	SCANL	Scans pixels to the left	Restore screen		
00C9	NVBXLN	Draws a box	0185	SDFSCR	Restore scr_param from RAM on clock chip
00CD	NVBXFL	Draws a filled box	0189	SETSCR	Restore screen & print opening message
Access VDP			VRAM data transfer function		
00D1	CHGMOD	Sets VDP mode according to SCRMOD	018D	SCOPY	Copies VRAM, array & disk file
00D5	INITXT	Init.screen for TEXT1 & sets VDP	0191	BLTVV	Copies VRAM to VRAM
00D9	INIT32	Init.screen for GRAPHIC1 & sets VDP	0195	BLTVM	Array to VRAM
00DD	INIGRP	Init.screen for GRAPHIC2 & sets VDP	0199	BLTMV	VRAM to array
00E1	INIMLT	Init.screen for MULTICOLOR & sets VDP	019D	BLTVD	Disk file to VRAM
00E5	SETTXT	Sets VDP for TEXT1	01A1	BLTDV	VRAM to disk file
00E9	SETT32	Sets VDP for GRAPHIC1	01A5	BLTMD	Loads array data from disk file
00ED	SETGRP	Sets VDP for GRAPHIC2	01A9	BLTDM	Saves array data to disk file
00F1	SETMLT	Sets VDP for MULTICOLOR	Mouse and Track ball		
00F5	CLRSPR	Initializes all sprites	01AD	NEWPAD	Read paddle, mouse & track ball
00F9	CALPAT	Returns address of sprite pattern table	Miscellaneous		
00FD	CALATR	Returns address of sprite attrib. table	01B1	GETPUT	Get TIME & get DATE & PUT KANJI
0101	GSPSIZ	Returns current sprite size	01B5	CHGMDP	Sets VDP mode according to SCRMOD
0105	GETPAT	?	01B9	RESV1	Not used. Reserved
0109	WRTVRM	Writes to VRAM addressed by HL	01BD	KNJPRT	Put kanji char to graphic scr(5-8)
010D	RDVRM	Reads VRAM addressed by HL	Access clock chip		
0111	CHGCLR	Change color of screen	01F5	REDCLK	Read clock data
0115	CLS	Clears screen	01F9	WRTCLK	Write clock data
0119	CLRTXT	Clears the text screen			
011D	DSPFNK	Displays function key display			
0121	DELLNO	Deletes a line in text mode			
0125	INSLNO	Inserts a line in text mode			
0129	PUTVRM	Put char in text screen			
012D	WRTVDP	Writes to VDP register			
0131	VDPSTA	Read VDP status			

DISK-ROM



Disk Driver table

4010H (DISKIO)

Physical input/output for disk devices (See PHYDIO in main BIOS)

Input:

- F = Carry set for write, carry reset for read
- A = Drive number (0=A:)
- B = Number of sectors to read/write
- C = Media descriptor
 - F8h for 360K floppy disk, 3,5 inch, sigle-side, 9 sector
 - F9h for 720K floppy disk, 3,5 inch, double-side, 9 sector
 - FAh for 320K floppy disk, 3,5 inch, sigle-side, 8 sector
 - FBh for 640K floppy disk, 3,5 inch, double-side, 8 sector
 - FCh for 180K floppy disk, 5,25 inch, sigle-side, 9 sector
 - FDh for 360K floppy disk, 5,25 inch, double-side, 9 sector
 - FEh for 160K floppy disk, 5,25 inch, sigle-side, 8 sector
 - FFh for 320K floppy disk, 5,25 inch, double-side, 8 sector
- DE = Logical sector number (starts at 0)
- HL = Transfer address

Output:

- F = Carry set on error, carry reset on success
- A = If error: errorcode
- B = Number of sectors transferred (always)

Error codes can be:

0	Write protected
2	Not ready
4	Data (CRC) error
6	Seek error
8	Record not found
10	Write fault
12	Other error

4013H (DSKCHG)

Check if disk has been changed.

Input:

- A = Drive number (0=A:)
- B = Media descriptor
- C = Media descriptor
- HL = Base address of DPB

Output:

- F = Carry set on error, carry reset on success
- A = If error: errorcode (see DSKIO)

- B = if success: 1=Disk not changed, 0=Unknown, -1=Disk changed

Note: If the disk has been changed or may have been changed (unknown) read the boot sector or the FAT sector for a disk media descriptor and transfer a new DPB as with GETDPB

4016H (GETDPB)

Get Drive Parameter Block

Input:

- A = Drive number
- B = First byte of FAT (media descriptor)
- C = Media descriptor
- HL = Base address of DPB

Output: HL+1 ... HL+18 = DPB for specified drive

Note: DPB consists of

Name	Offset	Size	Description
MEDIA	0	1	Media type (F8..FF)
SECSIZ	1	2	Sector size (must be 2^n)
DIRMSK	3	1	$(SECSIZE/32)-1$
DIRSHFT	4	1	Number of one bits in DIRMSK
CLUSMSK	5	1	$(\text{Sectors per cluster})-1$
CLUSSHFT	6	1	$(\text{Number of one bits in CLUSMSK})+1$
FIRFAT	7	2	Logical sector number of first FAT
FATCNT	8	1	Number of FATs
MAXENT	A	1	Number of directory entries (max 254)
FIRREC	B	2	Logical sector number of first data
MAXCLUS	D	2	$(\text{Number of clusters (not including reserved, FAT and directory sectors)})+1$
FATSIZ	F	1	Number of sectors used
FIRDIR	10	2	FAT logical sector number of start of directory

4019H (CHOICE)

Get string that describes disk formatting options.

Input: None

Output: HL = Address of zero terminated character string (ASCIIZ) with the text with choices for a DSKFMT If there is no choice (only 1 format supported) return HL = 0

401CH (DSKFMT)

Format disk

Input:

- A = Choice specified by user (1-9). See CHOICE
- D = Drive number (0=A:)
- HL = Begin address of work area
- BC = Length of work area

Output:

- F = Carry flag reset on success, carry flag set on error
- A = If error: errorcode

Error codes can be:

0	Write protected
2	Not ready
4	Data (CRC) error
6	Seek error
8	Record not found
10	Write fault
12	Bad parameter
14	Out of memory
16	Other error

Notes: Also writes a MSX boot sector at sector 0, clears all FATs (media descriptor at first byte, 0FFh at second/third byte and rest zero) and clears the directory (filling it with zeros)

401FH (DSKSTP)

Stop the drive motor of the corresponding interface.

Input: None

Output: None

Modify: AF, BC, DE, HL, IX et IY.

Notes: Not all diskinterfaces support this entry. Only valid when 401FH <> 00H

Kernel table

4022H (BASENT)

BASIC cold start. Start the Basic environment and eventually run a file of Basic program from a program in machine language.

Set the variable REBOOT (0F340h) to 0 to run the AUTOEXEC.BAS file (at the root of the disk). Otherwise, there will be a return to Basic. Under MSX-DOS, when the REBOOT variable (0F340h) contains a value other than 0 it is possible to run another Basic program whose file name and length is specified in DTA (0080h).

Input: None

Output: None (Does not return)

Examples:

```
1. ; Quit your DOS program and launch a BASIC program saved as NAME.BAS
```

```
CALSLT equ 001Ch
DTA equ 0080h
BASENT equ 04022h
REBOOT equ 0F340h
MASTER equ 0F348h

org 0100h
```

```

    ld    a,1      ; Not 0 to ignore the AUTOEXEC.BAS if present
    ld    (REBOOT),a

    ld    hl,NAME
    ld    de,DTA
    ld    bc,END-NAME
    ldir                   ; Copy the file name

    ld    ix,BASENT
    ld    iy,(MASTER-1)    ; Slot of the Master Disk-ROM
    jp    CALSLT
NAME:
    db    END-NAME        ; Length
    db    "NAME.BAS",0
END:

```

2. ; Quit your DOS program and go to BASIC

```

CALSLT equ 001Ch
DTA equ 0080h
BASENT equ 04022h
REBOOT equ 0F340h
TEMPST equ 0F67Ah
MASTER equ 0F348h

    org    0100h

    ld    a,1      ; Not 0 to ignore the AUTOEXEC.BAS if present
    ld    (REBOOT),a

    ld    a,0
    ld    (DTA),a    ; No file name

    ld    ix,(TEMPST)    ; Erases 3 bytes
    ld    (ix),0        ; from the
    ld    (ix+1),0      ; area reserved for
    ld    (ix+2),0      ; the BASIC program

    ld    ix,BASENT
    ld    iy,(MASTER-1)    ; Slot of the Master Disk-ROM
    jp    CALSLT

```

3. ; Quit your DOS program and launch the AUTOEXEC.BAS under BASIC

```

CALSLT equ 001Ch
BASENT equ 04022h
REBOOT equ 0F340h
TEMPST equ 0F67Ah
MASTER equ 0F348h

    org    0100h

    ld    a,0      ; 0 to run the AUTOEXEC.BAS file
    ld    (REBOOT),a

    ld    ix,(TEMPST)    ; Erases 3 bytes
    ld    (ix),0        ; from the
    ld    (ix+1),0      ; area reserved for
    ld    (ix+2),0      ; the BASIC program

    ld    ix,BASENT
    ld    iy,(MASTER-1)    ; Slot of the Master Disk-ROM
    jp    CALSLT

```

4. ENASLT equ 0024H
MASTER equ 0F348H
BASENT equ 04022H

```
LD A,(MASTER)
```



```
LD H, 40H
CALL ENASLT
JP BASENT
```

4025H (FORMTM)

Format a disk in BASIC by asking the user for settings.

This routine sets the Carry flag to 1 and then calls the FORMTK (04026h) routine below.

Input: None

Output: None

Note: This is interactive process with user (prompts for drive and format choice)

4026H (FORMTK)

Format a disk from a program with user-specified parameters.

Input:

- F = Reset the Carry flag
- HL = Start of buffer
- BC = Size of buffer

Output: None

4029H (MTOFF)

Stop all drives of all controllers

Input: None

Output: None

Modify: AF, BC, DE, HL, IX et IY.

Note: This routine only exists if the interface manages removable disks. Otherwise, 04029h will contain the zero byte (00h). Some HDD interfaces have wrongly implemented this as a HDD powerdown (HSH/MAK)

402DH (GETSLT)

Get Slot ID of disk ROM

Input: None

Output: A = Slot ID

4030H

Get MSX-DOS system bottom

Input: None

Output: HL = Lowest address used by the base MSX-DOS system

Рабочая область

Some workspace in MSX work area

Диапазон	Имя	Длина (дес./шест.)		Описание
F41F-F55C	KBUF	318	13E	tokenised form of input line in Direct Mode
F55E-F660	BUF	259	103	text from console collected by INLIN
F67A-F697	TEMPST	30	01E	stack for string descriptors
F6E8-F74B	PARM1	100	064	local variables of «FN» currently evaluted
F750-F7B3	PARM2	100	064	used to construct the local vars owned by current «FN»
F7BC-F7C3	SWPTMP	8	008	1st operand of SWAP
F7C5-F7EF	FBUFFR	43	02B	text produced during numeric output conversion
F7F6-F805	DAC	16	010	primary decimal accumulator
F806-F846	HOLD8	65	041	for double precision multiplication
F847-F856	ARG	16	010	secondary decimal accumulator
F857-F85E	RNDX	8	008	current double precision random number
F866-F870	FILNAM	11	00B	user-specified filename
F871-F87B	FILNM2	11	00B	filename from I/O device for comparsion with FILNAM
F87F-F91E	FNKSTR	160	0A0	10 function key strings
F959-F970	QUETAB	24	018	parameters for VOICAQ, VOICBQ, VOICCQ and RS232
F971-F974	QUEBAK	4	004	putback characters for queues (unused)
F975-F9F4	VOICAQ	128	080	voice A queue
F9F5-FA74	VOICBQ	128	080	voice B queue
FA75-FAF4	VOICCQ	128	080	voice C queue
FAF5-FB35	RS2IQ	64	040	RS232 queue
FB41-FB65	VCBA	37	025	current parameters for voice A
FB66-FB8A	VCBB	37	025	current parameters for voice B
FB8B-FBAF	VCBC	37	025	current parameters for voice C
FBB2-FBC9	LINTTB	24	018	overflow of screen line
FBCE-FBD7	FNKFLG	10	00A	normally zero, 1 for KEY(n) ON
FBF0-FC17	KEYBUF	40	028	circular queue driven by GETPNT & PUTPNT
FC18-FC3F	LINWRK	40	028	complete line of screen characters
FC40-FC47	PATWRK	8	008	8×8 pixel pattern
FC4C-FC81	TRPTBL	51	033	(17×3) current state of interrupt generated devices
FC82-FC99		24	018	rest of TRPTBL (unused)
FD09-FD88	SLTWRK	128	080	64×2 bytes workspaces for each of 64 possible slots
FD89-FD98	PROCNM	16	010	device or statement name of extension ROM
FD9A-FFC9	HKEYI	560	230	hooks
FFCA-...				

6. Рабочая область

Рабочая область находится в RAM и делится на две части:

- 1) область с и с т е м н ы х переменных MSX-BASIC;
- 2) область л о в у ш е к.

Таблица системных переменных

Имя	Адрес	Длина	Содержание
(16-ри- чный)	(деся- тичная)		
RDPRIM	F380	5	Программа чтения из установочного места
WRPRIM	F385	7	Программа записи в установочное место
CLPRIM	F38C	14	Переход на запись в установочное место
USRTAB	F39A	20	Начальные адреса для DEFUSR
LINL40	F3AE	1	Длина строки в режиме SCREEN 0
LINL32	F3AF	1	Длина строки в режиме SCREEN 1
LINLEN	F3B0	1	Длина строки в текущем режиме
CRTCNT	F3B1	1	Число доступных строк на экране
CUVLST	F3B2	1	Интервал между столбцами
TXTNAM	F3B3	2	Адрес Таблицы имен (PNT) в SCREEN 0
TXTCOL	F3B5	2	Адрес Таблицы цветов (CT) в SCREEN 0
TXTCGP	F3B7	2	Адрес Таблицы символов (PGT) в SCREEN 0
TXATTR	F3B9	2	
TXTPAT	F3BB	2	
T32NAM	F3BD	2	Адрес Таблицы имен (PNT) в SCREEN 1
T32COL	F3BF	2	Адрес Таблицы цветов (CT) в SCREEN 1
T32CGP	F3C1	2	Адрес Таблицы символов (PGT) в SCREEN 1
T32ATR	F3C3	2	Адрес атрибутов спрайтов в SCREEN 1
T32PAT	F3C5	2	Адрес шаблонов спрайтов в SCREEN 1
GRPNAM	F3C7	2	Адрес Таблицы имен (PNT) в SCREEN 2
GRPCOL	F3C9	2	Адрес Таблицы цветов (CT) в SCREEN 2
GRPCGP	F3CB	2	Адрес изображений (PGT) в SCREEN 2
GRPATR	F3CD	2	Адрес атрибутов спрайтов в SCREEN 2
GRPPAT	F3CF	2	Адрес шаблонов спрайтов в SCREEN 2
MLTNAM	F3D1	2	Адрес Таблицы имен (PNT) в SCREEN 3
MLTCOL	F3D3	2	Адрес Таблицы цветов (CT) в SCREEN 3
MLTCGP	F3D5	2	Адрес изображений (PGT) в SCREEN 3
MLATTR	F3D7	2	Адрес атрибутов спрайтов в SCREEN 3
MLTPAT	F3D9	2	Адрес шаблонов спрайтов в SCREEN 3
CUKSW	F3DB	1	Отзвук клавиши (0 - нет отзвука)
CSRY	F3DC	1	Колонка текстового курсора
CSRX	F3DD	1	Строка текстового курсора
CNCDFG	F3DE	1	Отображение текста ключей (0 - KEY OFF)
RG0SAV	F3DF	1	Содержимое VDP(0)
RG1SAV	F3E0	1	Содержимое VDP(1)
RG2SAV	F3E1	1	Содержимое VDP(2)
RG3SAV	F3E2	1	Содержимое VDP(3)
RG4SAV	F3E3	1	Содержимое VDP(4)
RG5SAV	F3E4	1	Содержимое VDP(5)
RG6SAV	F3E5	1	Содержимое VDP(6)
RG7SAV	F3E6	1	Содержимое VDP(7)
STATFL	F3E7	1	Статус VDP
TRGFLG	F3E8	1	
FORCLR	F3E9	1	Текущий цвет изображения
BAKCLR	F3EA	1	Текущий цвет фона
BDRCLR	F3EB	1	Текущий цвет бордюра
MAXUPD	F3EC	3	
MINUPD	F3EF	3	
ATRBYT	F3F2	1	Характеристика байта

QUEUES	F3F3	2	Адрес таблицы очередей	
FRCNEW	F3F5	1		
SCNCNT	F3F6	1	Развертка синхронизации	
REPCNT	F3F7	1		
PUTPNT	F3F8	2	Заполнение буфера клавиатуры	
GETPNT	F3FA	2	Выборка из буфера клавиатуры	
CS120	F3FC	10	Параметры ввода/вывода с кассеты	
LOW	F406	2		
HIGH	F408	2		
HEADER	F40A	1		
ASPCT1	F40B	2		
ASPCT2	F40D	2		
ENDPRG	F40F	5	Указатель оператора RESUME NEXT	
ERRFLG	F414	1	Код последней ошибки	
LPTPOS	F415	1	Позиция головки принтера	
PRTFLG	F416	1	1- принтер MSX; 0- экран	
NTMSXP	F417	1	1- принтер MSX; 0- не MSX	
RAWPRT	F418	1	Если содержимое Щ0, то выводится	
			необработанный символ	
VLZADR	F419	2		
VLZDAT	F41B	1		
CURLIN	F41C	2		
KBUF	F41F	318	Скоростной буфер	
BUFMIN	F55D	1	Запятая для оператора INPUT	
BUF	F55E	258	Содержимое буфера клавиатуры	
ENDBUF	F660	1	Конец буфера клавиатуры	
TTYPOS	F661	1	Позиция на терминале	
DIMFLG	F662	1	Флаг массива	
VALTYP	F663	1	Тип переменной в слове DAC	
OPRTYP	F664	0	?	
DORES	F664	1	Тип оператора	
DONUM	F665	1	Для скоростной работы	
CONXT	F666	2	Указатель текста для выбора символа	
CONSAV	F668	1		
CONTYP	F669	1	Тип используемой константы	
CONLO	F66A	8	Значение используемой константы	
MEMSIZ	F672	2	Максимально возможный адрес для PIT	
STKTOP	F674	2	Верхний адрес стека	
TXTTAB	F676	2	Начальный адрес PIT	
TEMPPT	F678	2		
TEMPST	F67A	30		
DSCTMP	F698	3		
FRETOP	F69B	2	Верхний адрес строкового пространства	
TEMP3	F69D	2	Работа по "сборке мусора"	
TEMP8	F69F	2		
ENDFOR	F6A1	2	Указатель оператора FOR	
DATLIN	F6A3	2	Указатель оператора DATA	
SUBFLG	F6A5	1	Рабочий флаг для FOR и USR	
USFLG	F6A6	0	?	
FLGINP	F6A6	1	Рабочий флаг для INPUT и READ	
TEMP	F6A7	2	Работа операторов	
PTRFLG	F6A9	1	=0, если не программная строка	
AUTFLG	F6AA	1	Щ0, если включен режим AUTO	
AUTLIN	F6AB	2	Первая строка нумерации для AUTO	
AUTINC	F6AD	2	Шаг нумерации для AUTO	
SAVTXT	F6AF	2		
SAVSTK	F6B1	2	Сохранить стек для обработки ошибок	
ERRLIN	F6B3	2	Номер строки, в которой была допущена	
			последняя ошибка	
DOT	F6B5	2	Номер текущей строки программы	
ERRTXT	F6B7	2		
ONELIN	F6B9	2	Адрес начала подпрограммы обработки	
			ошибок в PIT	
ONEFLG	F6BB	1		

TEMP2	F6BC	2	
OLDLIN	F6BE	2	Номер последней упомянутой строки
OLDTXT	F6C0	2	Старый текстовый указатель
VARTAB	F6C2	2	Начальный адрес Таблицы переменных
ARYTAB	F6C4	2	Начальный адрес Таблицы массивов
STREND	F6C6	2	Верхний адрес используемого пространства
DATPTR	F6C8	2	Указатель считывания для DATA/READ
DEFTBL	F6CA	26	Типы переменных A-Z
PRMSTK	F6E4	2	Стек для "сборки мусора"
PRMLEN	F6E6	2	Длина используемой таблицы
PARM1	F6E8	100	Таблицы параметров функций пользователя
PRMPRV	F74C	2	Указатель блока параметров
PRMLN2	F74E	2	Длина блока параметров
PARM2	F750	100	Адреса параметров
PRMFLG	F7B4	1	Флаг источника параметров
ARYTA2	F7B5	2	Конец источника параметров
NOFUNS	F7B7	1	
TEMP9	F7B8	2	
FUNACT	F7BA	2	Число назначаемых функций
SWPTMP	F7BC	8	
TRCFLG	F7C4	1	1 - TRON; 0 - TROFF
FBUFFR	F7C5	43	
DECTMP	F7F0	2	
DECTM2	F7F2	2	
DECCNT	F7F4	1	
DAC	F7F6	16	Арифметический аккумулятор
HOLD8	F806	48	
HOLD2	F836	8	
HOLD1	F83E	8	
ARG	F847	16	Вторичный арифметический аккумулятор
RNDX	F857	8	Последнее случайное число
MAXFIL	F85F	1	Установка оператора MAXFILES
FILTAB	F860	2	Адрес блока управления файлами
NULBUF	F862	2	Адрес буфера файла #0
PTRFIL	F864	2	
RUNFLG	F866	0	?
FILNAM	F866	11	
FILNM2	F871	11	
NLONLY	F87C	1	
SAVEND	F87D	2	Конец двоичных данных для BSAVE
FNKSTR	F87F	160	Значения функциональных клавиш
CGPNT	F91F	3	Адрес ячейки в ROM, начиная с которой
			хранятся образы символов
NAMBAS	F922	2	Текущий адрес PNT
CGPBAS	F924	2	Текущий адрес PGT
PATBAS	F926	2	Текущий адрес SGT
ATRBAS	F928	2	Текущий адрес SAT
CLOC	F92A	2	
CMASK	F92C	1	
MINDEL	F92D	2	
MAXDEL	F92F	2	
ASPECT	F931	2	Рабочая область для оператора CIRCLE
CENCNT	F933	2	
CUNEF	F935	1	
CNPNTS	F936	2	
CPL0TF	F938	1	
CPCNT	F939	2	
CPCNT8	F93B	2	
CRCSUM	F93D	2	
CSTCNT	F93F	2	
CSC LX Y	F941	1	
CSAVEA	F942	2	
CSAVEM	F944	1	
CXOFF	F945	2	

CYOFF	F947	2	
LOHMSK	F949	1	Рабочая область для оператора PAINT
LOHDIR	F94A	1	
LOHADR	F94B	2	
LOHCNT	F94D	2	
SKPCNT	F94F	2	
MOVCNT	F951	2	
PDIREC	F953	1	
LFPROG	F954	1	
RTPROG	F955	1	
MCLTAB	F956	2	Рабочая область для оператора PLAY
MCLFLG	F958	1	
QUETAB	F959	24	
QUEBAK	F971	4	
VOICAQ	F975	128	
VOICBQ	F9F5	128	
VOICCQ	FA75	128	
RS2IQ	FAF5	64	
PRSCNT	FB35	1	
SAVSP	FB36	2	
VOICEN	FB38	1	
SAVVOL	FB39	2	
MCLLEN	FB3B	1	
MCLPTR	FB3C	2	
QUEUEN	FB3E	1	
MUSICF	FB3F	1	
PLYCNT	FB40	1	
VCBA	FB41	37	
VCBB	FB66	37	
VCBC	FB8B	37	
ENSTOP	FBB0	1	Если значение ненулевое, то возможна остановка программы посредством нажатия комбинации клавиш CTRL+GRAPH+SHIFT+PUC или комбинации клавиш CTRL+GRAPH+SHIFT+1+3 (нажатие CTRL+STOP игнорируется!)
BASROM	FBB1	1	
LINTTB	FBB2	24	
FSTPOS	FBCA	2	
CODSAV	FBCC	1	
FNKSWI	FBCD	1	
FNKFLG	FBCE	10	Разрешение прерываний от клавиш F1-F10
ONGSBF	FBD8	1	
CUKFL	FBD9	1	
OLDKEY	FBDA	11	Старый статус клавиш
NEWKEY	FBE5	11	Статус клавиш (матрица клавиатуры)
KEYBUF	FBF0	40	Буфер клавиатуры
BUFEND	FC18	0	?
LINWRK	FC18	40	Работа экранного обработчика
PATWRK	FC40	8	
BOTTOM	FC48	2	Начальный адрес ОЗУ
HIMEM	FC4A	2	Адрес начала рабочей области
TRPTBL	FC4C	78	Таблица ловушек
RTYCNT	FC9A	1	
INTFLG	FC9B	1	
PADY	FC9C	1	Y-координата графического планшета
PADX	FC9D	1	X-координата графического планшета
JIFFY	FC9E	2	
INTVAL	FCA0	2	Установка интервала
INTCNT	FCA2	2	Счетчик интервалов
LOWLIM	FCA4	1	Чтение с кассеты
WINWID	FCA5	1	
GRPHED	FCA6	1	
ESCCNT	FCA7	1	

INSFLG	FCA8	1	
CSRSW	FCA9	1	1 - LOCATE,,1; 0 - LOCATE,,0
CSTYLE	FCAA	1	Код символа в позиции курсора
CAPST	FCAB	1	Щ0 - клавиша CAPS считается нажатой
KANAST	FCAC	1	Щ0 - клавиша РУС считается нажатой
KANAM	FCAD	1	
FLBMEM	FCAE	1	=0, пока загружается программа
SCRMOD	FCAF	1	Текущий режим экрана
OLDSCR	FCB0	1	Последний упомянутый текстовый режим
CASPRV	FCB1	1	Символ для устройства CAS:
BRDATR	FCB2	1	Цвет границы для оператора PAINT
GXPOS	FCB3	2	X-координата графического курсора
GYPOS	FCB5	2	Y-координата графического курсора
GRPACX	FCB7	2	X-координата графического аккумулятора
GRPACY	FCB9	2	Y-координата графического аккумулятора
DRWFLG	FCBB	1	
DRWSCL	FCBC	1	Масштабный множитель для DRAW
DRWANG	FCBD	1	Значение угла в операторе DRAW
RUNBNF	FCBE	1	BLOAD/BSAVE
SAVENT	FCBF	2	Начало данных для BSAVE
BIOSL	FCC0	1	
EXPTBL	FCC1	4	Рабочая область установочного места
SLTTBL	FCC5	4	
SLTATH	FCC9	64	
SLTWRK	FD09	128	
PROCNM	FD89	16	
DEVICE	FD99	1	

Таблица ловушек

Когда это возможно, указывается ключевое слово, "переводящее" на ловушку. Если за ним следует символ "0", то это означает, что ловушка обрабатывается всякий раз, когда при инициализации опрашивается дисковод.

Имя	Адрес	Длина	Оператор	Комментарий
H.ATTR	FE1C	5	0	ATTR\$ attribute
H.BAKU	FEAD	5		back up
H.BINL	FE76	5		BLOAD
H.BINS	FE71	5		BSAVE
H.BUFL	FF8E	5		buffer line
H.CHGE	FDC2	5		character get
H.CHPU	FDA4	5		character put
H.CHRG	FF48	5		
H.CLEA	FED0	5		CLEAR
H.CMD	FE0D	5	0	CMD
H.COMP	FF57	5		
H.COPY	FE08	5	0	COPY
H.CRDO	FEE9	5		crif do
H.CRUN	FF20	5		
H.CRUS	FF25	5		
H.CVD	FE49	5	0	CVD
H.CVI	FE3F	5	0	CVI
H.CVS	FE44	5	0	CVS
H.DEVN	FEC1	5		Определение новых имен,если
			устройство подтверждено	
H.DGET	FE80	5	0	GET,PUT
H.DIRD	FF11	5		direct statement do
H.DOGR	FEF3	5		do graph
H.DSKC	FEFE	5		disk character input
H.DSKF	FE12	5	0	DSKF
H.DSKI	FE17	5	0	DSKI\$ disk input

H.DSK0	FDEF	5	0	DSK0\$	disk output
H.DSPC	FDA9	5			Высветить курсор
H.DSPF	FDB3	5		KEY ON	
H.EOF	FEA3	5	0	EOF	
H.ERAC	FDAE	5			Стереть курсор
H.ERAF	FDB8	5		KEY OFF	
H.ERRF	FF02	5			
H.ERRO	FFB1	5		ERROR	
H.ERRP	FEFD	5			Печать сообщения об ошибке
H.EVAL	FF70	5			
H.FIEL	FE2B	5	0	FIELD	
H.FILE	FE7B	5	0	FILES	
H.FILO	FE85	5			file out 1
H.FINE	FF1B	5			
H.FING	FF7A	5			
H.FINI	FF16	5			
H.FINP	FF5C	5			
H.FORM	FFAC	5		FORMAT	
H.FPOS	FEA8	5	0	FPOS	
H.FRET	FF9D	5			free up to temporaries
H.FRME	FF66	5			
H.FRQI	FF93	5		POKE	
H.GEND	FEC6	5			Устройство не является
				дискководом	
H.GETP	FE4E	5			get file pointer
H.GONE	FF43	5			
H.INDS	FE8A	5			input disk character
H.INIP	FDC7	5			Инициализация символов
H.INLI	FDE5	5		LINE INPUT	
H.IPL	FE03	5	0	IPL	
H.ISFL	FEDF	5			is file I/O
H.ISMI	FF7F	5		MID\$	
H.ISRE	FF2A	5			
H.KEYC	FDCC	5			Кодирование клавиш (KEYCOD)
H.KEYI	FD9A	5			Обработка прерываний
H.KILL	FD9A	5	0	KILL	
H.KYEA	FDD1	5			Присваивание клавишам
				функций	
H.LIST	FF89	5		LIST,LLIST	
H.LOC	FE99	5		LOC	
H.LOF	FE9E	5	0	LOF	
H.LOPD	FED5	5			loop and set default
H.LPT0	FFB6	5		LPRINT	
H.LPTS	FFBB	5			line printer status
H.LSET	FE21	5	0	LSET	
H.MAIN	FF0C	5			at the MAIN entry
H.MERG	FE67	5	0	MERGE	
H.MKDS	FE3A	5	0	MKD\$	
H.MKIS	FE30	5	0	MKI\$	
H.MKSS	FE35	5	0	MKS\$	
H.NAME	FDF9	5	0	NAME	rename
H.NEWS	FF3E	5			
H.NMI	FDD6	5			Немаскируемые прерывания
H.NODE	FEB7	5			Определение новых имен,если
				устройство не подтверждено	
H.NOFO	FE58	5	0	OPEN	
H.NOTR	FF34	5			
H.NTFL	FE62	5	0	CLOSE	not file number 0
H.NTFN	FF2F	5			
H.NTPL	FF6B	5			
H.NULO	FE5D	5	0	KILL,LOAD	null open file
H.OKNO	FF75	5			
H.ONGO	FDEA	5		ON GOTO	
H.OUTD	FEE4	5			Вывод символов (out do)

H.PARD	FEB2	5			Анализ имени устройства
H.PHYD	FFA7	5			physical disk I/O
H.PINL	FDD8	5			Начало программной строки
H.PLAY	FFC5	5		PLAY	
H.POSD	FEBC	5			possibly disk
H.PRGE	FEF8	5		END	
H.PRTF	FF52	5			pointer get
H.PTRG	FFA2	5			Question mark and input line
H.QINL	FDE0	5			at the ready entry
H.READ	FF07	5			
H.RETU	FF4D	5		RETURN	
H.RSET	FE26	5	0	RSET	
H.RSLF	FE8F	5		INPUT\$	
H.RUNC	FECB	5		NEW,RUN	
H.SAVD	FE94	5			to save current drive
H.SAVE	FE6C	5		SAVE	
H.SCNE	FF98	5			
H.SCRE	FFC0	5		SCREEN	
H.SETF	FE53	5			set file pointer
H.SETS	FDF4	5	0	SET	
H.SNGF	FF39	5			
H.STKE	FEDA	5			stack error
H.TIMI	FD9F	5			Начало обработки прерываний
H.TOTE	FDBD	5			force screen to text mode
H.TRMN	FF61	5			
H.WIDT	FF84	5		WIDTH	

Ссылки

MSX BIOS calls

 [Main-ROM BIOS](#)

[Calling the BIOS from MSX-DOS](#)

MSX System Variables

 [System variables and work area](#)

 [System hooks](#)

<http://sysadminmosaic.ru/msx/bios>

2023-01-10 13:41

