

MB-02+

Original documentation

Contents

I. Introduction

- Welcome!
- Main features
- Software description and its possibilities
- Hardware description and its possibilities
- Optional available accessories
- How to set up the system

II. MB-02+: The system

- Understanding BS-DOS 308
 - New BASIC commands
 - Commands to set up the path
 - Disk commands
 - Data storage commands
 - New BASIC functions
- Understanding BS-ROM 118
 - What's new
 - An overview of repaired bugs
 - NMI Menu
 - NMI Menu Control Keys
 - Extended syntax of existing commands
 - New commands and their syntax
 - New system variable description

III. Error messages

- BS-DOS 308 error messages
- BS-ROM 118 error messages

IV. Programming under BS-DOS 308

V. MB-02+ BS-DOS 308 Assistant (the utilities disk)

VI. Hardware documentation

VII. Credits

I. Introduction

Welcome!

Dear Spectrum-addicted friend,

Thank you for buying a brand new product MB-02+ designed for your ZX Spectrum-compatible machine. Welcome to the world of a perfect combination of your computer and a new disk system. This invention lifts the possibilities of your Speccy as high as possible. This is the original documentation for MB-02+ developed by 8BitCompany. Please read these instructions carefully before setting up your hardware. Thank you.

Main features

Here comes the list of the the best features implemented into your system:

- disk space: high density: 1.800K; double density: 840K
 - speed rate: high density: 40-50K/sec.; double density: 25K/sec.
 - internal memory size: EPROM 2K (maximum 64K); SRAM 128K (maximum 512K-optional)-with backup
 - time measurement: Real Time Clock processor (optional)-with backup
 - connectable as many as 4 floppy disk drives
 - Kempston joystick/Amiga mouse port available
 - 3-channel 2-way parallel port (via i8255)
 - tape-compatible system (great advantage!)
 - full RAM-only mode
 - high quality DOS
 - as many as 256 directories on a single disk, each directory may contain as many as 65279 files!
 - software system-RESET (fully compatible with hardware RESET-doesn't work on ZX Spectrum +2A)
-

Software description and its possibilities

Together with MB-02+ you will get standard software products: BS-ROM, BS-DOS and BS-BIOS developed by Busysoft and some support utilities (up-to-date MB-02+ is able to twist files between these formats: BS-DOS, DP-DOS, M-DOS, MS-DOS). Utilities that support some other systems are available as well.

- **BS-ROM:** classical ZX Spectrum ROM which contains BASIC and a software interface for a communication with BS-DOS. There is also some extra software improvements that make the work with BASIC much more easier and comfortable.
- **BS-DOS:** operation system software itself for a communication with disks assigned by BS-ROM or by a program directly.
- **BS-BIOS:** system core that arranges the communication between software and hardware of the system.

After connecting the MB-02+, you will be equipped by all functions which are usually available on any other system and some new features. The system's concept is based on an idea of a tape emulation which means that you can communicate with disk in exactly the same way as you did with a tape. Good thing about this concept is that you do not have to modify the software (no matter if it is BASIC or machine code!). Obviously the communication speed is much faster using a disk. How does it work? The core of data transfer are two routines situated in ZX-ROM at addresses 1218 (for SAVE) and 1366 (for LOAD). Routines for the communication with disks (and tape as well) are attached to those addresses so that you can swap between use of any disk or a tape. The system also remembers the position where the disk is "rewound" to (the virtual-tape pointer - compare to tape). The system also recognizes headerless files. The utilities for data transmission from tape to disk are already available. In general, all of it means that MB-02+ together with BS-DOS is fully compatible with tape.

MB-02+ also allows to switch ZX Spectrum into full 64K RAM-only mode so there is no problem to modify memory area 0-16383 (for example, you can load here original ZX-ROM, make any changes (like installing new font) and then write-protect this memory area to emulate ROM).

The system also allows to perform disk operations and simultaneously run any other operation (using IM 2-e. g. music); none of the processes will be interrupted. MB-02+ is also the only system for ZX Spectrum which allows the use of HD (high density) disk format; that is caused by a programmable Z80-DMA chip.

Hardware description and its possibilities

What do I find inside the MB-02+?

- **WD2797A:** disk controller (FDC).
 - **Z80-DMA:** Z80-chipset family member which arranges super fast data transfer between ports and a memory in all directions (Direct Memory Access); if you learn how to control this programmable chip, you will see unseen on your own Speccy! Some DMA-demos are supplied together with MB-02+.
 - **EPROM 2K:** contains a software which is executed after Spectrum's turn-on or RESET. It allows to boot the DOS into the SRAM memory (see below) but also turn off the MB-02+ and use your Spectrum in a normal way. This chip is easily replaceable by an optional EPROM 64K which will include new graphic user interface ZX-OS/3 Vision.
 - **SRAM 128K:** static RAM (with backup!) which is split into the pages organized in the memory area #0000-#3FFF (0-16383, i. e. instead of the original ROM); due to this fact it is possible to switch your Spectrum into 64K RAM-only mode. Also DOS is loaded into this memory area (BS-ROM, BS-DOS, BS-BIOS). A software which organizes SRAM into an optional RAM-Disk is supplied together with MB-02+ so that you can use SRAM as yet another disk drive (with backup!). It's only up to your choice what SRAM size you choose (128K/256K/384K/512K). Obviously, the more the better.
 - **i8255:** a chip that arranges 3-channel 2-way parallel data transfer. Using this chip you can connect Kempston joystick, Amiga-mouse, printers, A/D or D/A convertors etc. to your MB-02+.
 - **FDD port:** floppy disk drives connection (2 drives); it's possible to connect as many as 4 external floppy disk drives (see Hardware documentation).
 - **NMI button:** to execute the NMI menu.
 - **RESET button:** to reset the entire system (both Speccy and MB-02+) NEVER press this button when there is DMA transfer (disk LOAD/SAVE etc.) in progress. Such behaviour will lock the system and you will have to turn the power down.
-

Optional available accessories

The following list describes all accessories that will improve your system:

- **real time clock processor (RTC):** real time/date measuring chip. There are plenty of things to do with RTC (timing, measuring of a certain time period etc.). New BS-DOS 400 also supports RTC information in all files (so that you know exactly when they were created). Of course, this chip is "awake" even when the MB-02+ is turned off so it will not lose the track of the time.
 - **floppy disk drive FDD 3.5" HD:** the best disk drive format to use with MB-02+; this is the most common format used all around the world. We supply only the first-rate FDDs (Sony, Teac, Mitsumi) together with FDD cable to connect to MB-02+.
 - **standard power supply:** supplies power to MB-02+ and as many as 2 FDDs.
 - **PC power supply:** supplies power to ZX Spectrum, monitor, MB-02+ and as many as 4 FDDs.
 - **Amiga-mouse:** the most common mouse to use with ZX Spectrum.
-

How to set up the system

- 1) Turn off your computer and all accessories.
- 2) Connect your MB-02+ to ZX Spectrum.
- 3) Connect floppy disk drive(s) to MB-02+. When connecting drive(s) to MB-02+, please note to turn both ends of the cable in the direction so that the red wire is near the power plug on both the MB-02+ and floppy drive(s)-pin 1 on the corresponding

connector.

- 4) Connect power supply connectors (with no power on!) to ZX Spectrum, MB-02+ and floppy drive(s). If you use PC power supply, do not forget to insert correct power connectors to ZX Spectrum and MB-02+. Do not swap them! That will result in damaging of your hardware.
- 5) Make all other usual connections, if any (TV, tape recorder, mouse etc.).
- 6) Check that all connections are correct and made firmly.
- 7) Turn the power on.
- 8) Check if the original disk is write protected. On the start-up animation, boot the BS-DOS by inserting the original disk into disk drive number 1.
- 9) If the system does not boot up, turn the power off and check all connections; due to floppy disk drive construction, it may be necessary to swap the cable in the drive upside down. Continue at point 6.

II. MB-02+: The system

Disk system MB-02+ consists of 2 parts: BS-DOS itself and corrected and improved ROM (called BS-ROM). The text below describes BS-DOS 308 and BS-ROM 118.

Understanding BS-DOS 308

Each floppy disk includes 256 directories (each directory can have a name up to 26 characters). Each directory may contain as many as 65279 files, which means that on 1 floppy disk could be stored as many as $256 \cdot 65279 = 16711424$ files.

DOS offers standard and the most common disk services together with some others. As already mentioned above, DOS can also emulate tape using a floppy disk. A difference between disk and tape is that the tape is ALWAYS rewound to a certain position (compared to disk). So if you type in LOAD "" (no file name specified) the first file available on tape will be loaded. Therefore a special load-pointer was created in order emulate tape. Load-pointer points to a position where a disk is "rewound" to. The above mentioned means that disk emulates 256-track tape (track=directory). Whenever you type in LOAD "", DOS will use a load-pointer to determine which file is supposed to be loaded.

All tape functions work as usual: LOAD, SAVE, VERIFY, MERGE-they all call ROM LOAD/SAVE routines at #4C2 (=1218) and #556 (=1366). That explains why all programs using ROM for LOAD/SAVE operations work correctly with disk as well as with tape. No modifications necessary! It is also possible to interrupt any disk operation by pressing Q+W keys, simultaneously.

New BASIC commands

In a text below:

- if any parameter is put into <these branches>, it means that it is not mandatory.
- file specification as "data" is as follows:

item <, item> <, item> <, item>...

where "item" stands for:

number1 <TO <number2>>

or

TO <number2>

...where number1 signifies start of a range (if number1 is not given all files from the start of the directory are considered) and number2 signifies the end of a range (if number2 is not given all files to the end of the directory are considered).

Example:

Let us suppose that directory contains 32 files:

TO 3,10,20,13 TO 15,28 TO

stands for these files:

1,2,3,10,20,13,14,15,28,29,30,31,32

- specification <local> sets a temporary path within a command; this specification starts by # followed by a numeric expression signifying a temporary path; such syntax could be closed by , if there are data to follow.

Example:

Let us suppose that current disk is number 4:

CAT #2,66

will view a catalogue of the directory number 66 at disk number 2.

- £ stands for pound symbol (symbol shift+X)

Further, wherever a number is expected it is possible to type in an expression that results in a number. Also wherever a string or a name is expected any string expression could be typed in instead.

Commands to set up the path

Use the following commands to set up the path to required files.

USR 15522 - setting a tape as a current device (LOAD/SAVE/VERIFY/MERGE commands will work with tape).

USR 15524 - setting a disk as a current device (LOAD/SAVE/VERIFY/MERGE commands will work with disk).

@ number

@ string

Setting a current disk drive. String signifies a disk name ("find a disk" function according to name is not implemented though). Number stands for the disk number. If number=0 LOAD/SAVE routines will work with tape (exactly the same as USR 15522). If number>0, also USR 15524 will be performed-LOAD/SAVE routines will communicate with disk.

\$ number

\$ string

To set the current directory on the current disk drive. Number stands for directory number, string signifies directory name, but only 10 characters is considered.

£ number

£ string

To set the current file; load-pointer will be set to the required position.

Any combination of these 3 commands sets the path as the part of the commands listed below (or could be used all alone as well).

Example:

@3\$5@8\$22\$ "games_a"£ "arkanoid"

means that on disk in drive 3 will be set current directory 5, on disk in drive 8 will be set current directory 22, then the first "games_a" directory found after directory 22 will be set as current, and the load-pointer will be set to "arkanoid" file. Such sequence could be ended by ; if it is required by the command's syntax.

Disk commands

The list and description of commands for disk operations and file handling follows.

.FORMAT <path>

To erase all files and directories on the disk. It is possible to unformat disk using .UNERASE command (in case SAVE is not performed after .FORMAT).

.ERASE <path> data

To erase files. It is possible to unerase files (if SAVE does not write them over in the meantime). Erasing the file

number 0 erases entire current directory.

`. UNERASE <path> data`

To unerase erased files/directories. Use number 0 to unerase current directory itself, files must be uneraser manually. As already mentioned, it is also possible to unerase files erased by `.FORMAT`: use `.UNERASE 0` (to unerase current directory itself) and then all necessary files inside of it. It may seem that files become uneraser right after directory unerase but they are not (file sectors are not recovered), therefore it is necessary to unerase files individually.

`. RENAME <path> number, string`

To rename file "number" to new name "string". If number=0, then directory name will be changed. Note that directory name can be 26 characters long.

`. RENAME <path> number1, number2`

To change a start address (number2) of a specified file (number1). If number1=0 then number2 specifies under what directory this directory belongs to (within the tree structure).

`. MOVE <path> number1, number2`

To move file "number1" into directory "number2". If number1=0 then the entire directory will be moved.

`. MOVE <path> number; data`

To move files specified by "data" into the directory "number".

Note: by moving a file within the current directory will put the file at the end of that directory.

`. PRESS <path>`

To press, "shake down" the current directory.

`. KILL <path>`

To erase internal CACHE memory. The use of this command is not necessary because DOS manages all operations with CACHE automatically.

`. SEARCH <!> <path> <local> <string>`

To start a search for a required string on a disk. Names of all directories (only 10 characters) and of all existing files are scanned. "!" instructs DOS to scan also through erased files and empty directory items. Search starts at the current directory, continues through directories 255 and 0 till it reaches current directory.

`CAT <.><!> <path> <local> <number>`

`COPY <.><!> <path> <local> <number>`

Displays a catalogue of a current directory or of a directory as <number>. <!> displays also erased files and empty directory items. <.> displays a catalogue starting at the load-pointer.

Data storage commands

The list of data storage commands follows:

```
SAVE <!><*> <path> <number> file name
SAVE <!><*> <path> <number> file name SCREEN$
SAVE <!><*> <path> <number> file name DATA var ( )
SAVE <!><*> <path> <number> file name CODE address, length
LOAD <!><*> <path> <number> <file name>
LOAD <!><*> <path> <number> <file name> SCREEN$
LOAD <!><*> <path> <number> <file name> DATA var ( )
LOAD <!><*> <path> <number> <file name> CODE <address <, length>>
MERGE <!><*> <path> <number> <file name>
VERIFY <!><*> <path> <number> <file name>
VERIFY <!><*> <path> <number> <file name> SCREEN$
VERIFY <!><*> <path> <number> <file name> DATA var ( )
VERIFY <!><*> <path> <number> <file name> CODE <address <, length>>
```

These commands work as usual tape commands. Number signifies load-pointer position (exactly the same as the path "L" parameter). All these commands perform operations in the current directory or in the directory specified by the path (which becomes current afterwards). SAVE command always saves the file at the end of current directory. Characters ! and * are ignored; they were implemented in order to accept commands that were originally created in BASIC 128 (LOAD !) or under D40/D80 disk system (LOAD *). VERIFY command works as expected; if any differences are found during verification, a message specifying a difference area will appear.

NEW <path> <local> parameter

This command loads and executes a file specified by a parameter. If parameter is numeric, executable file (see below) in the current directory will be loaded and executed. If parameter is string, a file name that corresponds to the string will be searched in the current directory. If no match is found, the system will look for specified file in a directory 0. If no file is found again, error message is displayed.

System considers both "Program" and "Bytes" as executable. If BASIC ("Program") file is considered, it will be loaded and executed (depends on LINE condition)-exactly the same as LOAD. If CODE ("Bytes") file is considered, the following operation will occur:

Suppose AAA is a start address of a file:

```
IF AAA>=#5D00 (=23808) THEN CLEAR AAA-1
LOAD parameter CODE AAA
CONTINUE AAA (see BS-ROM 118)
```

If no parameter is given, usual NEW command will be performed.

New BASIC functions

A purpose of these functions is to give an information that is usually hardly available.

FN ?

Gives a DOS version number.

FN @

Gives current drive number.

FN \$

Gives current directory number.

FN L

Gives a load-pointer position.

FN #

Gives the number of files in the current directory.

FN %

Gives the number of empty sectors on the current disk.

FN *

Gives a 32 characters long string specifying current load-pointer item.

Example:

FN *(7 TO 16) signifies current file name

FN *(6 TO 22) signifies current file header (as on tape)

If load-pointer is not positioned within the files of the current directory, empty string will be given instead.

If load-pointer >= #FF00 (=65280), 32 characters long string will specify a disk name.

FN ©

A special function that results in a string that includes memory range 0 to 65535.

Understanding BS-ROM 118

There were few bugs, issues and problems with the original Spectrum ZX-ROM 48. All of them were fixed. There were also some new features added to the original ZX-ROM 48. BS-ROM 118 is nothing but well-known ZX-ROM 48 on which the following modifications have been performed:

- new commands
- extended syntax of existing commands
- search for tape lead-in tone is 4 times shorter
- program pointer has been changed from > to *
- faster error beep signal
- implemented co-operation with hexadecimal numbers 0-65535
- BASIC lines in range 0-16383 (editable)
- faster autorepeat
- when trying to enter error command, cursor will be moved to the error position right away
- can list programs without control codes (ignores colours, TAB and AT control codes-good for hidden lines viewing)
- NMI menu
- some errors corrected
- two new characters in a character set (CHR\$ 30 and CHR\$ 31)
- program pointer * is displayed only after autolist
- line editing is possible in 4 directions (left, right, up, down)
- super-fast reset/NEW (memory cleared using PUSH instruction)
- it is possible to interrupt autolist using a "Q" key

The list of repaired bugs

These are the fixes in the original ZX-ROM 48:

- NMI routine replaced by NMI menu
- PRINT AT 0,0;CHR\$ 8;-this command moved PRINT cursor to error position (AT -1,31); corrected
- if an error message is to be printed and interrupt is disabled (DI), the system used to halt itself (corrected)
- ZX-ROM 48 used to overwrite itself (addresses 0-4); BS-ROM 118 corrected this issue so that it can even run with memory write enable mode
- suppose that last program line is N, if you entered in an empty N+1 line and pressed EDIT right away, you would be editing N line together with the program pointer; corrected

After reset or NEW, system will display the information about memory space available together with OK message. Please note that neither reset nor NEW test memory-it is expected that memory works correctly.

NMI menu

NMI menu is executed by pressing NMI button or jumping to address #66 (=102). Fast flashing border indicates that system is in NMI menu. If the stack pointer (SP) was in a memory area #0001-#4000 (1-16384), it will be changed to #5800 (=22528).

NMI menu control keys

B	warm BASIC start; all system variables are reset except of PROG, VARS and ELINE
N	exactly the same as CLEAR 24575:NEW
R	48K memory reset
E	128K memory reset
I	resets AY/FDC/DMA chips
S	saves the screen at the current device
T	USR 15522 (current device: tape)
D	USR 15524 (current device: disk)
Q	returns back into the interrupted program; note that return is correct only in case SP register was not changed to #5800 (=22528), all registers are reset to correct values (except of bits 0-6 of R-register)
M	executes MRS debugger (must be loaded in the memory!); registers are set as in Q-command
0...9	switches ZX 128K RAM pages; if SP was situated in area #C001-#0000 (=49153-0), it will be changed to #5800 (=22528); exactly the same as OUT (32765), number-using keys 8 or 9 it is possible to display the contents of VIDEORAM 2
U	user defined function that executes instruction situated at #3901 (=14593)
Z	user defined function that executes instruction situated at #3904 (=14596)

Extended syntax of existing commands

In a text below, parameter in <these branches> is not mandatory.

CONTINUE <expression>

If expression is not 0, machine code at address (expression) will be executed. After return back to BASIC, IY must be set to #5C3A (=23610), but HL' does not have to be set to #2758 (=10072).

CLS <expression>

If expression is not 0, all colours will be set according to the expression (suppose expression stands for the attribute value).

Example:

```
CLS 7=BORDER 0:PAPER 0:INK 7:BRIGHT 0:FLASH 0:CLS
```

```
GO TO
GO SUB
PAUSE
BORDER
```

Numeric parameter is not mandatory; if not given, 0 will be considered.

<PRINT> item list

If the item list does not start by a number, ; , or ' a PRINT command can be skipped.

POKE address, byte

Usual POKE command.

POKE address; word

2-byte (16 bits) POKE command.

POKE address, string

Storing a given string into the memory.

VAL\$ <expression>

DEC → HEX conversion. If the expression is numeric, the result of this function is a string 4 characters long in hexadecimal form.

Example:

VAL\$ 33023="80FF"

% or & <string>

HEX → DEC conversion.

Example:

%80FF=33023

VAL <expression>

If the expression is numeric, the result of this function is 2-byte (16 bits) number on (expression) address (i.e., 2-byte PEEK).

New commands and their syntax

The list of new commands implemented into ZX-ROM 48 follows:

, <expression>

Use to edit BASIC line which is given by the expression. If the expression is not given or equals 0, the first line available will be edited.

' <expression>

Switches ZX 128K RAM pages; exactly the same as OUT (32765),expression. Expression must be in range 0-15. Please note that during page switching, RAMTOP must be below #C000 (=49152). See 0..9 keys in NMI menu above.

©

Resets all system variables except of PROG, VARS and ELINE (also see B-command in NMI menu).

128K memory reset (also see E-command in NMI menu).

?

Displays an information regarding program length, variables length and empty memory space available assigned for BASIC.

*

Reads headers from the tape and displays the following information together with each header:
file type, file name, address, length

 (underline symbol: symbol shift+0)

The same as USR 102-executes NMI menu.

#

The same as USR 54885-executes MRS (must be loaded in the memory!).

New system variable description

There is new use of 3 bits situated at address 23728 (after reset/NEW, 0 is stored into this address):

bit 0=0: BASIC lines displayed as 4-digits long

bit 0=1: BASIC lines displayed as 5-digits long

bit 1=0: on autolist, control codes work as usual

bit 1=1: on autolist, control codes work according to bit 2

bit 2=0: on autolist, control codes are ignored

bit 2=1: on autolist, control codes are replaced by ?

III. Error messages

General explanation

Error messages are displayed in the following form:

```
=====
      Error message
      drv/dir/sub: file_name
      trk, sid, sec
      Enter=ok/retry, Space=abort
=====
```

Where:

drv	error in drive number
dir	error in directory number
sub	error in file number
trk	error on track number
sid	error on side number
sec	error at sector number
file_name	specifies error area

BS-DOS 308 error messages

The following is the list of error messages. If you consider the error message displayed does not make much sense to you, BS-DOS might be damaged by some sort of software. Therefore we recommend to reboot BS-DOS again.

File not found
Specified file not found.

File too short
Attempt to load more sectors than assigned for the file.

Can't unerase
Cannot unerase due to sector damage or unerased directory never existed.

Error at N1-N2:N3
When verifying, a difference area was found; N1-N2 is a difference range, N3 signifies difference area length.

File too long
Not possible to load/create a file this long.

Disk full
File requires more space than available on the current disk.

Disk not ready
No disk in drive detected or some sort of disk error.

Unknown disk
When operating a drive, the drive number specified does not exist.

Directory too long
Attempt to access more than 65279 files in a directory.

Directory full
Attempt to add new file into the full directory.

Data integrity error N1
Error found when accessing a disk.

Can't read the disk
BS-DOS cannot read the BOOT sector on the disk.

Seek error
FDC chip cannot find track 0.

Break
Q+W keys pressed simultaneously when operating the disk.

Ready error
Error when activating a disk drive.

Write protect
Displayed when writing to disk. Write protect switch is switched to the protect position.

CRC error
When loading a data sector, CRC addition checks do not match.

Data lost
Data lost during transfer between FDC and DMA chips. Caused probably by a software that operates DMA under interrupt mode.

Time out
Time ran out when operating a disk without a success.

Record not found
When reading from/writing to disk, sector identification was not found (caused for example by low disk quality).

Unknown format
Attempt to access a disk which is not BS-DOS format.

System error N1
Usually some sort of damage found within a BS-DOS itself. It is recommended to reboot the system immediately.

BS-ROM 118 error messages

The list of errors displayed by BS-ROM 118 follows:

Invalid I/O device
This message is displayed when "Abort" option was selected.

Invalid file name
Unexisting file/directory name given.

Invalid argument
Attempt to execute unexecutable file using NEW.

All other error messages have exactly the same meaning as in ZX-ROM 48.

V. MB-02+ BS-DOS 308 Assistant (the utilities disk)

COPY-DISK-DISK ("cdd")

This program was created for files copying. Help page is displayed when "cdd" is executed. Help page will also appear if parameters are not correct. For parameter specification, check out MOVE/ERASE commands, but use "" instead of TO.

VI. Hardware documentation

Real-time clock processor (RTC-optional)

Low address: 3, %00000011, #03

Type: out,it

HIGH ADDRESS	A3	A2	A1	A0	REGISTER	BIT3	BIT2	BIT1	BIT0	VALUE	REMARKS
0, #00	0	0	0	0	S1	s4	s3	s2	s1	0 - 9	1 - second digit register
1, #01	0	0	0	1	S10	*	s30	s20	s10	0 - 5	10 - second digit register
2, #02	0	0	1	0	MI1	mi4	mi3	mi2	mi1	0 - 9	1 - minute digit register
3, #03	0	0	1	1	MI10	*	mi30	mi20	mi10	0 - 5	10 - minute digit register
4, #04	0	1	0	0	H1	h4	h3	h2	h1	0 - 9	1 - hour digit register
5, #05	0	1	0	1	H10	*	AM/PM	h20	h10	0 - 2	AM/PM, 10 - hour register
6, #06	0	1	1	0	D1	d4	d3	d2	d1	0 - 9	1 - day digit register
7, #07	0	1	1	1	D10	*	*	d20	d10	0 - 3	10 - day digit register
8, #08	1	0	0	0	MO1	mo4	mo3	mo2	mo1	0 - 9	1 - month digit register
9, #09	1	0	0	1	MO10	*	*	*	mo10	0 - 1	10 - month digit register
10, #0A	1	0	1	0	Y1	y4	y3	y2	y1	0 - 9	1 - year digit register
11, #0B	1	0	1	1	Y10	y40	y30	y20	y10	0 - 9	10 - year digit register
12, #0C	1	1	0	0	W	*	w3	w2	w1	0 - 6	Week register
13, #0D	1	1	0	1	RegD	30sec ADJ	IRQ FLAG	BUSY	HOLD	-	Control register D
14, #0E	1	1	1	0	RegE	t1	t0	ITRPT /STND	MASK	-	Control register E
15, #0F	1	1	1	1	RegF	TEST	24/12	STOP	RESET	-	Control register F

nutno doplnit nakres pripojeni 4 FDD

Bit * does not exist.

Please mask AM/PM bit with 10's of hours operations.

Busy is read only, IRQ can be set low only.

Data Bit	AM/PM	ITRPT/STND D	24/12
1	PM	ITRPT	24
0	AM	STND	12

ITRPT/STND = INTERRUPT/STANDARD

HARDDISK:

ADDRESS 7, %00000111, #07; TYPE: OUT, IN

DMA:

ADDRESS 11, %00001011, #0B; TYPE: OUT, IN

FDC:

ADDRESS 15, %00001111, #0F; TYPE: OUT, IN
ADDRESS 47, %00101111, #2F; TYPE: OUT, IN
ADDRESS 79, %01001111, #4F; TYPE: OUT, IN
ADDRESS 111, %01101111, #6F; TYPE: OUT, IN

FLOPPY:

ADDRESS 19, %00010011, #13; TYPE: OUT

BIT0 /ACTIVE A
BIT1 /MOTOR ACTIVE A
BIT2 /ACTIVE B
BIT3 /MOTOR ACTIVE B
BIT4 /ACTIVE C
BIT5 /MOTOR ACTIVE C
BIT6 /ACTIVE D
BIT7 /MOTOR ACTIVE D

ADDRESS 19, %00010011, #13; TYPE: IN

BIT0 /DRQ
BIT1 /DISK CHANGE
BIT2 /INTRQ
BIT3 HDIN
BIT4 /MOTOR ACTIVE A
BIT5 /MOTOR ACTIVE B
BIT6 /MOTOR ACTIVE C
BIT7 /MOTOR ACTIVE D

MEMORY:

ADDRESS 23, %00010111, #17; TYPE: OUT

BIT0 - BIT4 page (0-31)
BIT5 WRITE ENABLE
BIT6 SRAM ENABLE
BIT7 EPROM ENABLE

If BIT6 and BIT7 are set /RESET is generated.

BACK-UP BATTERY INFORMATION:

Lithium-based battery used for RTC & memory back-up.
Type: CR-2032

PIO:

BASE ADDRESS 31, %0xx11111, #1F; TYPE OUT, IN

or

BASE ADDRESS 27, %0xx11011, #1B; TYPE OUT, IN

xx - A6, A5	REGISTER
00	PA
01	PB
10	PC
11	CWR

CANNON 37:

PIN	FUNCTION
1	GND - ground
2	PA0
3	PA1
4	PA2
5	PA3
6	PA4
7	PA5
8	PA6
9	PA7
10	PB0
11	PB1
12	PB2
13	PB3
14	PB4
15	PB5
16	PB6
17	PB7
19	GND - ground
20	UCC - power 5V
21	PC0
22	PC1
23	PC2
24	PC3
25	PC4
26	PC5
27	PC6
28	PC7
36	*/RESET - output only!
37	UCC - power 5V

CANNON 9:

PIN	FUNCTION	KEMPSTON	MOUSE
1	PA3	UP	HOR. 1
2	PA2	DOWN	VER. 1
3	PA1	LEFT	HOR. 2
4	PA0	RIGTH	VER. 2
6	PA4	FIRE	LEFT BUTTON
7	UCC - power	UCC - power	UCC - power
8	GND - ground	GND - ground	GND - ground
9	PA5	-	RIGHT BUTTON

PRINTER:

CANNON 25	CANNON 37	FUNCTION	TYPE
1	PC1	/STROBE	output
2-9	PB0-PB7	D1-D8	output

10	PC2	/ACK	input
11	PC7	BUSY	input
12	PC6	PAPEROUT	input
13	PC5	SELECT	input
14	PC3	/AUTOFEED	output
15	PC4	/ERROR	input
25	GND	GND	ground

8BitCompany Laboratories Index: 9708122112-8BCL

VII. Credits

Copyright

MB-02+, BS-DOS, BS-ROM, BS-BIOS
© 1994-1998 8BitCompany Laboratories
© 1994-1998 8BitCompany Czech Republic Europe
All rights reserved.

Hardware

Robert Letko (Robo)
Jan Kučera (Last Monster)

Software

Slavomír Lábský (Busysoft)

Documentation

Slavomír Lábský (Busysoft)
Martin Blažek (Blazko/systems)

Contact us

8BitCompany
Luční 4570
760 05 Zlín
Czech Republic
tel.: +420 67 43424, +420 603 543256
e-mail: 8bc@mbox.mrp.cz, 8bc@publikum.cz

Product specifications may vary due to progress.

Index: 8BCL/014107100297/English version