==

*** MB-02+ ***

Original documentation

8BitCompany Interactive Entertainment Studios

presents

MB-02+
Revolutionary Step Forward

==

*** Introduction ***

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+.

Please read these instructions carefully before setting your hardware up.

==

*** Main features ***

```
*disk space: HD: 1.800K / DD: 840K
*speed rate: HD: 40-50K/sec. / DD: 25K/sec.
*internal memory size: EPROM 2K (maximum 64K)
                       SRAM 128K (maximum 512K) - backup
*time measurement: Real Time Clock processor - backup
*connect as many as 4 floppy disk drives
*Kempston joystick/Amiga mouse port available
*3-channel two-way parallel port (via i8255)
*tape-compatible system (great advantage!)
*full RAM-only mode
*first-rate 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 + some support utilities (up-to-date MB-02+ is able to twist files between formats: BS-DOS, DP-DOS, M-DOS and, of course, MS-DOS).

*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 & 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 + 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 idea is that you do not have to modify the software (no matter if it is BASIC or machine code!). Obviously the communication speed rate is much higher 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 any font) and then write-protect this memory area to emulate ROM). Due to this mode we are also considering an implementation of CP/M and UNIX systems.

The system also allows to perform disk operations and simultaneously run any other operation (under 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 will you 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 user graphic system ZX-OS/3 Vision (in preparation).

*SRAM 128K: static RAM (under 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 (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 (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 two-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 (using a special 4-FDD adaptor - see below).
*NMI button: to execute the NMI menu.
*RESET button: to reset the entire system (both Speccy and MB-02+).

==

*** Accessories available (optional) ***

*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.). BS-DOS 400 also supports RTC information in all files (so that you know exactly when they were created). Of course, this chip is running even when the MB-02+ is turned off so it will not lose the track of the time (backup!).
*floppy disk drive FDD 3.5" HD: the best disk drive to use with MB-02+; this is the most common format used all around the world. We supply only the first-rate FDDs (Mitsumi, Sony, Teac) 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.
*4-FDD adaptor: if you choose to use 3 or 4 FDDs it's necessary to use this adaptor.
*Centronics adaptor: this adaptor is used to connect Centronics-compatible printer (the most common format).
*bus expansion: if you own ZX Spectrum +2A we recommend to use this expansion due to the lack of space in the area of the Spectrum's bus and a power connector.
*Amiga-mouse: the most common mouse to use with Spectrum.

==

*** How to set up the system? ***

1) Turn off your comupter 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).
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).

===

*** MB-02+ 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.

===

*** BS-DOS 308 ***

Each floppy disk includes 256 directories (each directory may have a name up to 26 characters). Each directory may contain as many as 65279 files, which means that on 1 disk could be stored as many as 256*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 will be loaded.
Therefore a special load-pointer was created in order to point a place on a disk 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 find out what file to load.
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!

Note: it is 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 (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 (suppose current disk is number 4):

```
CAT #2,66
```

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

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 ***

USR 15522 - setting a tape as a current device (LOAD, SAVE, VERIFY, MERGE commands will communicate with tape).
USR 15524 - setting a disk as a current device (LOAD, SAVE, VERIFY, MERGE commands will communicate with disk).

```
@ number
@ string
```

Setting a current disk drive. String signifies a disk name (find a disk according to name is not implemented though). Number stands for the disk number. If number=0 load/save rountines will communicate 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. 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 in the commands below (or could be used all alone as well).
Example: @3$5@8$22$"fart"?"stinks" 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 "fart" directory found after directory 22 will be set, and the load-pointer will be set to "stinks" file. Such sequence could be ended by ";" if it is required by the command's syntax.

==

*** Disk commands ***

.FORMAT <path>
To erase all files and directories on the disk. It possible to unformat disk using .UNERASE (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 the entire current directory.

.UNERASE <path> data
To unerase ereased files/directories. Use number 0 to unerase current directory itself, files must be unerased 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 unerased right after directory unerase but they are not (file sectors are not recovered), therefore it is necessary to unerase files as well.

.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 as long as 26 characters.

.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 direcory belongs (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 to 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 within 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 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.

==

*** Commands for data handling ***

```
  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 "?" 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 the directory. Characters "!" and "*" are ignored; they were impemented 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, 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 DOS version number.

FN @
Gives current drive number.

FN $
Gives current directory number.

FN ?
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 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.

==

*** BS-ROM 118 ***

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 shorter 4 times
program pointer has been changed ">" 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
*ability to list programs without control codes (ignores colours, TAB & AT controls - good for viewing of hidden lines)
*NMI menu
*some errors corrected
*two new characters in a character set
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

==

*** An overview of corrected errors ***

*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), system halts itself (corrected)
*ZX-ROM 48 used to overwrite itself (addressed 0-4); BS-ROM 118 was corrected so that it can even run under 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 program pointer; corrected

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

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

Control keys within NMI menu:

```
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 to 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 6-0 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 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.
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
Supposing 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
Putting a given string into the memory.

VAL$ <expression>
DEC -> HEX conversion. If expression is numeric, the result of this function is string 4 characters long in hexadecimal form.
Example: VAL$ 33023 = "80FF"

% or &
HEX -> DEC conversion.
Example: %80FF = 33023

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

===

*** New commands and their syntax ***

, <expression>
Use to edit BASIC line given by an expression. If 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 in NMI menu above.

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

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

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

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

_ [underline]
The same as USR 102 - executes NMI menu.

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

==

*** New system variable ***

There is new use of 3 bits situated at address 23728 (after reset/NEW, 0 is put 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 "?"
```

==

*** BS-DOS error messages ***

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 closely error area
```

The following is the list of error messages. If you consider the error message displayed does not make much sense to you, BS-DOS can be damaged by some sort of software. Therefore we recommend to boot up 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 on the disk. Write protect switch is switched to the proctect position.

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

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

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

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

Unknown format
Attempt to access a disk which was not formatted by MB-02+.

System error N1

Ususally some sort of damage found within a BS-DOS itself. It is recommened to reboot the system immeditately.

==

*** BS-ROM error messages ***

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

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.

==

*** Description of supplied utilities ***

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.

==

*** What is in the progress? ***

*MB-SoundFace: a sound card which enables replaying of the MOD files (PC/Amiga most common music files) on your Speccy (requires MB-02+ with SRAM 512K).
*MB-Level2: a card which will allow to connect and control harddisks and CD-ROM drives on the MB-02+.
*MB-Modem: by connecting an external modem to MB-02+ you can communicate on your ZX Spectrum with your friend, transfer files or get attached to the Internet or BBS.
*MB-VideoFace: a hardware which transforms the external video signal into the ZX Spectrum's graphic format.
*ZX-OS/3 Vision: A software in EPROM 64K chip which will include new user graphic system ZX-OS/3 Vision.
*MB-fully interactive game: using the powers of MB-02+, you will discover the world of an interactive horror adventure. Thousands of high quality pictures and film-animations, wonderful sound reproduction and perfect playability - you have never seen all of it at one time on your Speccy. All pictures are created by a revolutionary technology using 72 colours and full screen multicolor (MultiTech72). You cannot miss that!!!

==

*** Epilogue: ...and what did they say?! ***

"MB-02+ is what every Speccy-guy can only dream of (until he actually gets it)."

Omega

"There is no doubt the MB-02+ is the best system for ZX Spectrum ever developed."

Jaxon Hollis (JSH)

"By and large - SUPER!!!!!!!!!!"

Triton

"MB-02+ is a disk system for ZX Spectrum of the 3rd millenium."

Mr. Johny Renegade

"The disk system that overcame tommorow's products."

Sharks

"MB-02+ is all I ever wanted."

Last Monster (LMN)

"After all those sexless systems there finally comes something with a pussy."

Blazko/systems (BLS)

==

*** Credits ***

MB-02+, BS-DOS, BS-ROM, BS-BIOS
© 8BitCompany Interactive Entertainment Studios Czech Rep. Europe
© 1994-1997 8BC/IES. All rights reserved.

Product specifications may vary due to progress.

Hardware: Robert Letko/ROBO, Jan Ku?era/LMN
Software: Slavomír Lábsky/BUSY
Documentation: Slavomír Lábsky/BUSY
Translation: Martin Blažek/BLS

Contact 8BitCompany : tel. +420 67 43424

Index: 8BCies/014107100297/English version

==

GOOD LUCK!!!

==