MT-DEBUG

Dynamic Debugging Tool
for 64Kbyte MSX-Computers

Nederlandse handleiding
Guide d'utilisation Francais
English Manual
Libretto di Istruzioni Italiano
Manual Espagnol
Deutsche Gebrauchsweisung

Registration Card
Send the registration card as soon as possible to our Customer Service Department. You will be kept informed of new programs and eventual updates.

Send the card to:

Micro Technology b.v.,
Customer Service Dept.
P.O.BOX 95
3350 NA PAPENDRECHT
NETHERLANDS

MT-DEBUG a dynamic debugging tool.
MT-DEBUG is a very usefull tool in the debugging and testing of your MSX-programs. Especially designed to work with MSX-computers, this package allows you to look into any part of the machine, setting the slots according your needs and searching through memory for a byte or byte-combination in HEX- as well as ASCII-notation. Also MT-DEBUG allows you to set breakpoints almost anywhere you want, except in ROM.
MT-DEBUG is a dynamic debugger. Fully written in Assembler, it is extremely fast and even allows you to see dynamic actions like counters and for example the stack inside your computer.

This manual is not ment to be a training-guide for novice assembly language programmers. For that purpose we reccomend you to buy existing publications on programming the Z80 like the book "PROGRAMMING THE Z80", by Rodnay Zaks, published by Sybex, ISBN 0-89588-013-X. Also a very interesting and usefull book on MSX-programming and debugging is: "THE COMPLETE MSX PROGRAMMERS GUIDE", ISBN 0-86161-173-X. Published by Melbourne House (Publishers) Ltd., Richmond, United Kingdom or Melbourne House Software Inc., Nashville, USA or Melbourne House (Australia) Pty Ltd. South Melbourne. This book can also be ordered through your bookshop.
It is a 570 pages large ring-bound manual, covering all programming and debugging techniques and explaing the special MSX capabilities. It tells you where to find the correct entry points and what they will do for you. This book is a learning guide for the novice programmer and sharpens the skills of the advanced programmer.

Therefore, this manual only contains the neccessary information about how the commands of MT-DEBUG will work for you.

Command Summary of MT-DEBUG

Pressing the ESC key always returns you to the display mode. The commands of MT-DEBUG are called by pressing keys on your keyboard. We tried as much as possible to use these keys which gives you a familiar feeling with its command. Like "D" is Display and "S" is setting the slot configuration, etc.

The following commands are available in MT-Debug:

«D» Display new address

Input address for cursor in hex. Press «ENTER» to move cursor to its new position, or «ESC» to abort

«S» Set slots

The cursor will appear at the top left hand corner. You may use the arrow keys to change the present slot configuration. The left and right arrows enable you to select the bank that you want to change. The up and down arrows change the slot number, you may also key in the digits directly. Pressing «ENTER» or «ESC» returns to display mode.

«R» Modify registers
The cursor will appear at the register display on the bottom lines of your display. You may use the arrow keys to position the cursor over the digit you want to change. (Take good care when changing stack pointer!!) Pressing «ENTER» or «ESC» returns to display mode.

«M» Modify memory
In this mode the characters you type (when legal) will change the memory contents pointed to by the cursor. In HEX mode you may only use 0 - 9 and A - F. In ASCII mode all characters accept control characters may be used. You can use the arrow keys to change the cursor position, the «SELECT» key for changing to HEX or ASCII mode, the «TAB» key for changing ASCII display mode, the «ENTER» or «ESC» key for returning to display mode.

«B» Block operation (move or clear)
The Block operation command contains sub-command to tell MT-DEBUG what to do with the block you defined. This command only works if a start and end address has been given to the debugger. To define
the start and the end addres you have to use the following commands:

«CTRL»«S» Set start of block
The cursor position will be entered as start of block (see also **«B»** Block commands). This start of block must be in the same slot and bank as the end of the block.

«CTRL»«E» Set end of block
The cursor position will be entered as end of block. This end of block must be in the same slot and bank as the start of the block.

Subcommands when editing in block-mode:

«M» moves the block you have defined to the block starting at the cursor position in the slot configuration that is displayed in the top left hand corner of the screen.

«C» clears the block you defined, by filling it with zeroes.
The debugger will show the action you want to take and ask you to confirm this. If you press «Y» or «ENTER» the action will take place, any other key aborts.

«F» Find
Depending on the mode you may specify up to 7 HEX bytes or 10 ASCII characters. After pressing «ENTER» the debugger will search through memory starting at the cursor address. If the sequence of characters is found the cursor will appear at the start of it, otherwise the cursor returns at byte 0000. If you don't enter a string, but press «ENTER» directly, the debugger will search for the last entered string. This enables you to search for more occurences of the same string. When the string has been found, just press «F»«ENTER» to find the next occurence. You may use «ESC» to abort when inputting the string, but not when the program is searching.

«G» Go, jump to a memory location
The program asks you to enter the address you want the program to start. (Just press «ENTER» when you want to use the value of the Program Counter. If you press the spacebar, the address of the cursor will be used.) Then the computer will ask you for a breakpoint. If you don't want to use a breakpoint, just press «ENTER». If you do want a breakpoint, you may use the arrow keys to point to the address and press the spacebar, or you may just input the address. Until you press «ENTER» on the breakpoint question you still may press «ESC» to abort. N.B. The breakpoint will only be active when it has been set in a RAM area!

«X» Toggle register / flags display
There are two kinds of register display. The first show the contents of all registers (except I and R). The second shows the flags and the block parameters you may have entered. Pressing «X» toggles between those two displays.

«P» Set up display for address in RAM at cursor position
The first two bytes in RAM at the cursor position will be used as the address where the cursor must be shown next. This might come in handy while debugging a program, for instance: C3 83 26 .. .. means a jump to 2683 (hex). Use the arrow keys to put the cursor over the 83, press «P» and the cursor will be at 2683.

«Q» Quit, return to basic
Clears the screen and returns to Basic.

«SELECT» Switches between HEX and ASCII mode
Operation mode switch, only important when modifying or searching operation is used. While modifying the mode may be changed, but when inputting a string to find this is not possible.

«TAB» Switches between standard ASCII an Graphics display
In standard ASCII display mode the control (and delete) characters are being shown as dots and the most significant bit is reset. In Graphics display mode all characters will be shown as they appear in the MSX character set.

Cursor movements:
«UP» Line up
«DOWN» Line down
«LEFT» Previous byte
«RIGHT» Next byte
« + » or « = » Page down
«-» Page up

Debugging machine-language routines in BASIC with MT-DEBUG
If you use machine-language routines in your BASIC programs, MT-DEBUG has a very usefull BASIC command, which you can write into your program. Just before or after the call to your

machine-language routine, also write the command: "CALL DEBUG". The program will jump to MT-DEBUG allowing you to inspect the results of your machine-language program.

Checking machine-language routines while executing them

While debugging a program, it can be very usefull to - after an error occured for example - write also a call into your program to MT-DEBUG. This will be a so-called Inter-Slot-Call:

```
RST     30H     ;Call function
DEFB    Slot    ;slot where MT-DEBUG is present
DEFW    4010H   ;Init Entry in MT-DEBUG
```

or

```
RST     30H
DEFB    Slot
DEFW    4013H   ;Warm Entry in MT-DEBUG
```

At the first CALL, the register contents will be initialised at zero. The effect is the same, when calling MT-DEBUG from BASIC ("CALL DEBUG").

At the second CALL, the register contents will be saved and displayed at your screen.

The Slot is decided by the place where MT-DEBUG is physical present in your MSX-Computer.
For example:

Sony HITBIT 75 on top = Slot 1
Sony HITBIT 75 at the back = Slot 3
Spectravideo 728 on top = slot 2