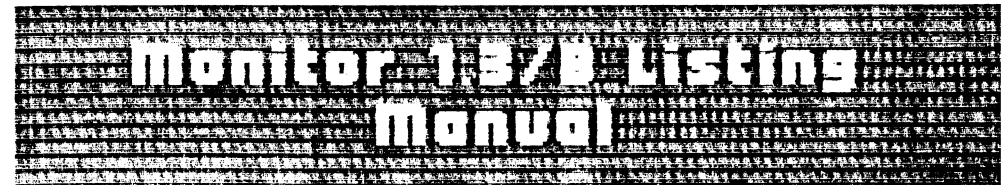


Standard Monitor 1.3

Copyright (C) 1983 by Gebr. van Montfort, Smedestraat 13, 6411 CR Heerlen, The Netherlands



Heerlen, March 1983

Standard Monitor 1.3
=====

Heerlen, March 1983

Table of contents:

Manual

Introduction.....	1
Summary.....	2
General.....	3
Commands.....	4
Special function keys.....	10
Assembly instructions.....	11

Source listing.....	12
Cross reference.....	57

Dutch manual

Inleiding.....	64
Overzicht.....	65
Algemeen.....	66
Commando's.....	67
Speciale functie toetsen.....	75
Inbouw handleiding.....	77

Chapter 1: Introduction.

This is a new monitor (operating system) for the Sorcerer computer. All known errors and peculiarities of the standard monitor version 1.0 have been removed. Maintaining the structure of version 1.0 ensures an almost complete compatibility. The monitor is delivered in 2 proms. Each set of proms has its own unique identification number. This number can be used, for example, for the recognition of your computer or to protect your own software.

Disclaimer of all warranties and liability

Gebr. van Montfort makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability or fitness for any particular purpose. Gebr. van Montfort software is sold or licensed "as is". The entire risk as to its quality and performance is with the buyer.

Should the programs prove defective following their purchase, the buyer (and not Gebr. van Montfort, its distributor, or its retailer) assumes the entire cost of all necessary servicing, repair or correction and incidental or consequential damages. In no event will Gebr. van Montfort be liable for direct, or indirect, incidental, or consequential damages resulting from any defect in software, even if Gebr. van Montfort has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liabilities for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This manual and software is copyrighted and may not in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Gebr. van Montfort.

The upgrade kit consists of the next items:

1. set of two PROMS.
2. Manual.

The following tools are needed for installation:

1. phillips head screwdriver.
2. i.c. removal tool or flat screwdriver.

* The contents of this manual are subject to change without notice.

Chapter 2: Summary.

1) Enhancements to version 1.0

PP Now works as intended and does not crash if no ROM pack is present.

DUMP <RUN/STOP> key works correctly now. <CTRL C> will stop the dumping.

ENTER Works as before, but also shows the contents of the memory. Simple <RETURN> leaves the contents of memory unchanged and moves on to the next memory location. <^>-key shows the contents of the previous memory location. Only if you type in some data the contents is changed.

MOVE MO 100 200 101 (overlapping move) works correctly now.

TEST This routine is considerably improved. Only error and succes messages are printed now.

SAVE The additional command format SA NAMEX 21E S77F saves from 21EH to 99DH.

LOAD LO NAMEX 1 8000 now always loads at 8000H.

<CTRL \> This combination of keys initiates the graphics now.

Commands are allowed in lower case letters. Parameters however must be written in upper case.

<CR> The typing of only a <CR> (carriage return) does no longer give an error.

<CLEAR> This key does not initiate the standard graphics any longer.

Video This routine is approx. 18% faster now.

Keyboard This routine is faster too, with better bounce control. The UART status of (IY+45)=(XXD6) is used, therefore the RS232 interface can be used now.

Quickcheck This routine is faster now and even checks <CTRL S>.

2) New commands:

FL FL 100 1FFF 00 fills the memory locations 100H up to 1FFFH with 00H.

SET X= SE X=100 6-12-82 put startaddress plus date in tape-header.

SET T= Using workarea-address IY+3D (XXCE)

T=0	1200 baud cassette	40H
T=1	300 baud cassette	00H
T=2	1200 baud RS232	C0H
T=3	300 baud RS232	80H
T=4	Motor 1 off	motor 2 off
T=5	Motor 1 on	motor 2 off
T=6	Motor 1 off	motor 2 on
T=7	Motor 1 on	motor 2 on
T=8	Normal numerical keyboard.	
T=9	The cursor operating keys can be used without the SHIFT-key.	

<CTRL X> Boot of disc system (only present if ordered, because of variable boot-address). For systems with exidy-drives (boot address BF00) there is a monitor 1.3/B with a reboot-utility inside.

Chapter 3: General.

In examples, replies give by the user are printed bold-faced.

Keys are denoted by <...>, f.e. :<CR>=carriage return. <CTRL ?> means pushing in the control key simultaneously with the there-after mentioned key, f.e. <CTRL X> means pushing in the control key simultaneously with the key X.

The commands are listed in the left column. Only the first two letters are necessary. The commands may be written in either upper- or lowercase! The parameters of each command are listed to the right; parameters enclosed in [square parenthesis] are optional. A letter used as a parameter must be upper case.

Between commands and parameters a delimiter must be placed (a space is normally used). Commands must be closed by a <CR> or, if they should not be executed, with the <@> key.

Addresses must be given as hexadecimal numbers. Hexadecimal values are given as XXXX and/or YYYY throughout this manual. Addresses are 2 bytes long (XXXX) and have a range from 0 to FFFF. Leading zeros may be omitted (100 is equal to 0100).

Parameters can be hexadecimal values and are then listed with XX, else they are listed by WW (or VV).

Command lines may be 60 characters long. If the video line is full no backspace is sent to the video (or the printer) in this new monitor, thus preventing the irritating flashing.

Chapter 4: Commands.

DUMP DU XXXX [XXXX]
addr.1 addr.2

Gives the contents of the memory at the VDU (visual display unit).
If only addr. 1 is given, the contents of this address is displayed. When address 2 is included, the contence from address 1 to and including address 2 will be displayed in coloms of 16 addresses.
The dump-command can temporarily be interrupted by pushing the <RUN/STOP> key. Pushing the key for the second time will start the display again. <CTRL C> breaks the command.

ENTER EN XXXX [VV [WW.....]] [/]
addr. parameter(s) terminator

Gives the possibility to change the value of one or more memory locations, one at a time or all at once in any combination and sequence that might be convenient. The "interactive" and "batch" modes are terminated by the terminator character (/) and <CR>.

There are 3 possibilities if only the address is given (interactive mode):

- 1) Only <CR> leaves the old value unchanged and shows the contents of the next address.
- 2) Typing YY <CR> replaces the old value by YY and displays the next memory location.
- 3) The <^> key shows the contents of the previous memory location.

In "batch" mode the parameters following the address, are automatically stored in sequential addresses starting at address XXXX.

Example "interactive" mode:

```

en 100 <CR>
0100 FF: <CR>      FFH is the contents of address 100H, no
                    chang.
0101 00:AA <CR>    Address 101H is changed into AAH.
0102 FF:^ <CR>     ^^ means two places back!
0100 FF:20 <CR>    Address 100H is changed into 20H.
0101 AA:^ <CR>     Address 101H is correct, back one place,
0100 20:/ <CR>    to check address 100H. / means STOP.

```

Example "batch" mode:

```

en 100 41 42 43 44 45 46 / <CR>
This puts the values 41H, 42H,...at the addresses 100H to
106H.

```

FILL FL XXXX XXXX VV [WW.....]
addr.1 addr.2 parameter(s)

This command fills the memorylocations from address 1 to address 2 with the value(s) XX. The number of parametervalues may not exceed the number of memorylocations. This command deviates in syntax from the others because this command can only be called by FL and not by its full name (fill).

Examples:

FL 0 100 FF<CR>

Fills the places from 0H to 100H with value FFH.

FL 1000 2000 41 42 43 44 45<CR>

Fills the places from 1000H to 2000H with the values 41H 42H 43H 44H 45H repeatedly.

Not allowed:

FL 0 2 00 AA FF BB<CR>

The number of values is greater than the number of memorylocations to be filled.

SAVE SA NAMEX XXXX [S]XXXX [X]
name addr.1 addr.2 or blocklength recorder #

Writes the memorycontents of adress 1 up to address 2 into the recorder specified (if no recordernumber is given then recorder 1 is used). If the "S" is added before address 2, this value is used as the number of addresses to be saved, starting from address 1. This command format sets the user free of address calculations before copying programs. The first character of the name must be a capital, the name may contain up to five characters.

Examples:

SA Prog 1000 1100 2 <CR>

Saves the program with the name "Prog" from address 1000H to 1100H at recorder #2.

SA Progn 1F50 S01EF <CR>

Saves the program with the name "Progn" from address 1F50H to 213EH at recorder #1. This relinquishes the calculation during the copying of programs.

The following (optional) recordernumbers are possible:

- 1= cassette recorder #1
- 2= cassette recorder #2
- 3= RS232 gate 300 baud
- 4= RS232 gate 1200 baud

LOAD LO[G] [NAMEX] [X [XXXX]]
name recorder addr.

Loads the program NAMEX of recorder # X to memorylocation XXXX.

If a G is inserted behind LO, the program is loaded and immediately started at the GO-address in the file-header.

(Watch it! The G is a parameter and must be a capital letter!). LOAD loads the first program met on recorder # 1 into the memory at the place which is mentioned in the file-header at the place ADRES. If you add an address, loading takes place from this address.

The following recordernumbers are possible:

- 1 = cassette recorder # 1 (f.e. LO 1)
- 2 = cassette recorder # 2 (f.e. LO 2)
- 3 = RS232 gate 300 baud (f.e. LO 3)
- 4 = RS232 gate 1200 baud (f.e. LO 4)

FILE FI [X]
recorder

Lists all programs which are on the tape in recorder # X with the following fileheader information:

- 1: program name
- 2: file type
- 3: block length
- 4: start address
- 5: go address
- 6: date if present

This command looks for 10 zeros and checks if thereafter a 01 follows (characteristic feature of the file-headers). By this method the number of CRC-ERRORS is drastically limited in comparison to version 1.0

Here 4 recordernumbers are possible again (see load).

GO GO XXXX
addr.

Starts the program at address XXXX. After ending the program a complete warmstart is carried out. Program changes of the IY register are irrelevant, because IY is recalculated as soon as the program is ended.

MOVE MO XXXX XXXX [S]XXXX
 addr.1 addr.2 blocklength or addr.3

Copies the memory contents of the memory block starting at address 1 and ending at address 3 to the memory block starting at address 2. If the "S" is added before address 3, this number of addresses is moved, starting at address 1 to the memory block starting at address 2. The syntax check on this command is improved, so that wrong commands have no longer disastrous consequences. Overlapped moves are carried out correctly now and do not longer destroy the program to be moved.

Example:

Working with a disassembler or the file-command uses only a part of the screen. Saving a part of the screen contents can be done by this command:

MO F080 F0A0 S800<CR>

The the left part of the screen is moved to the right side.

TEST TE XXXX XXXX [C]
 addr.1 addr.2 continue

Tests every bit in the range of the RAM memory from address 1 to address 2 (exclusive). During testing an asterix (*) flashes on the screen. In this version, during testing, on the printer only messages will be written, no asterixes.

3 RAM-addresses in memory can never be tested! The addresses F000H and F001H. Here the Monitor stores its top of RAM value and address F7FFH where the asterix flashes.

The test command carries out a complete read/write test, so that hardware errors, like broken addresslines are always detected.

Example:

To test the RAM addresses where the stack and workarea of the monitor are located (XF00H to XFFFH, whereby X is dependant of the size of the memory (X=3 is 16K; X=7 is 32K; X=B is 48K)), typing:

EN 0 21 FF FF C3 6 E0/<CR>

GO 0<CR>

TE XF00 XFFF C<CR>

Moves the stack to FFFFH and the workarea XF00H to XFFFH is tested.

PROMPT PR=W
 Changes the prompt sign (>) into the character here given by W.

CREATE CR Creates a so-called batch-tape on recorder # 1. Every permitted Monitor command you type in, is put on tape instead of being carried out. To terminate this command enter an empty line (<CR> only).

Example:

```
>CR <CR>
*EN 0 CD 18 E0 28 FB C9/<CR>
*DU 0 FF<CR>
*GO 0<CR>
*DU 100 1FF<CR>
*GO 0<CR>
*DU 200 2FF<CR>
```

These batchcommands take care of a memorydump per page and repeatedly waits for the pushing of a key.

```
*OV<CR>
```

```
*<CR>
```

An empty line (only <CR>) ends the create-mode.

LIST LI Gives a list of all occuring commands on the batch-tape. Only recorder # 1 can be used.

BATCH BA Execute the commands saved during the create mode (execute the batch-file from recorder #1). This tape with batch-commands can only work from recorder 1.

OVER OV This command closes the batch-tape and gives the control back to the operator. Normally used as last command on the tape. (see last command in the example of the create command).

SET SE W=W
 This command has several formats.

SE F=XX Sets the file-type in the file-header at XX. File type values above 7F hex. disable starting of the command LOG. All ascii values between 0H and FFH can be used without disturbing the screen during reading of the program. If no file-header is given a space is being printed, so that a list of file-headers is beautifully listed one under the other!

SE X=XXXX [XX-XX-XX]

The GO-address in the file-header is set at XXXX. As optional parameter the date can be included, so that this is a part of the file-header. This gives you the possibility to check which version of a program you are dealing with. The date may be given as DD-MM-YY or MM-DD-YY.

Example:

SE X=1000 12-2-82<CR>

Causes the file-header to be displayed as:
NAMEX BLCK ADDR.1 1000 12-02-1982

SE T=X Sets some values of the Monitor for the determination of the baudrate, the RS-232 status and the status of the numerical keyboard.

The following parameters are possible:

- X: meaning:
- 0 1200 baud cassette
- 1 300 baud cassette
- 2 1200 baud RS232
- 3 300 baud RS232
- 4 motor #1 off, motor #2 off
- 5 motor #1 on, motor #2 off
- 6 motor #1 off, motor #2 on
- 7 motor #1 on, motor #2 on
- 8 normal keyboard
- 9 the arrows and the home-key at the numerical keyboard work without shift.

SE S=WW Changes the printing speed at the screen. Initially this value is 00, this is the highest printing speed. The higher the hexadecimal value WW, the lower becomes the printing speed. This command is handy f.e. for studying "peculiar" basic listings!

SE O=W Sends, depending on the value of W, to different output devices.

- V sorcerer video (initial).
- L centronics printer driver (7 bits).
- P paralel printer driver (8 bits).
- S serie output driver.
- XXXX output driver at address XXXX.

SE I=W Receives, depending on the value of W, from different input devices.

- K sorcerer keyboard (initial).
- P parallel gate.
- S serie input driver.
- XXXX input driver at address XXXX.

PP PP [W]

Executes a program in the Rom-pack. If a parameter is given, then cold start (RESET); if no parameter, then warm start. If no Pack is present, a warm start to the Monitor is carried out.

Chapter 5: Special function keys.

<CLEAR>: Clears the screen (in Basic CHR\$(12)); but in contrast to the vesion 1.0 monitor the current graphics are preserved.

<CTRL \>: Resets to standard graphics (in Basic CHR\$(28)).

<RUN/STOP>: Also works during the dump command execution.

<CTRL S>: Stops Standard Basic programs or listings until a key is being pushed.

<CTRL RUN/STOP>: Works in Standard Basic like <CTRL O>.

<ESC><RESET>: Does a warm start to the rompack if present else to the monitor. WATCH IT! First push <ESC>, hold down and press both <RESET>-keys. Release both <RESET>-keys at the same time, but hold the <ESC>-key until the Rompack has gained control, detectable by the message being printed. With some packs this can take a while.

<CTRL C><RESET>: Does a cold start to the Monitor as if no rompack is present. This command is very convenient if you get f.e. problems with a crashing Basic-editor. Pushing <CTRL C><RESET> leaves the Basic program untouched. The remarks concerning the pushing of keys as by <ESC><RESET> must be applied here too.

<CTRL W>: This command can automatically boot a (floppy) disk. This shortens for typing "GO XXXX". The key <W> can be specified when ordering this PROM-set, together with the actualy bootaddress XXXX.

Chapter 6: Assembly.

This is quite simple:

- Switch off the current and remove all connectors and the pack.
- Remove the topside of the sorcerer (5 screws)
- Watch for the keyboard cable! It must also be disconnected.
- Remove the connector to the transformer from the print.
- Loosen the large print from the bottom and take it out carefully after disconnecting the ground wire. (6 screws).
- Remove the Rom-pack holder (4 screws).
- Remove the two IC's with the text EXMOL, watch for position and number !!!
- Exchange these IC's with the two new monitor PROM's, watch again for position and number!
- Assemble everything neatly in reversed order. The arrow of the keyboard connector must point to the front.

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0001 ; *****
0002 ; *
0003 ; * monitor version 1.3/B *
0004 ; *
0005 ; * written by A.C.L. van Montfort *
0006 ; * Heerlen, 16-JUN-82 *
0007 ; *
0008 ; *****
0009 ;
0010 ;
0011 ; COPYRIGHT (C) 1983 by Gebr. van Montfort
0012 ;
0013 ;
0014 ;uart equates
0015 ;
>00FD 0016 UARTS EQU 0FDH ;uart status port-address
>00FC 0017 UARTD EQU 0FCH ;uart data port-address
0018 ;
0019 ;
0020 ;ASCII equates
0021 ;
>000A 0022 LF EQU 0AH ;linefeed
>000C 0023 FF EQU 0CH ;formfeed
>000D 0024 CR EQU 0DH ;carriage return
>001B 0025 ESC EQU 1BH ;escape
>0020 0026 SPACE EQU 20H
0027 ;
>0001 0028 CTRLA EQU 'A'-64
>0003 0029 CTRLC EQU 'C'-64
>007F 0030 RUBOUT EQU 7FH
0031 ;
0032 ;
>F000 0033 RAMTOP EQU 0F000H ;top of ram storage address
>F080 0034 VID EQU 0F080H ;start address video memory
>003C 0035 LINELN EQU 3CH ;length of input buffer
>006E 0036 STORE EQU 6EH ;offset: (top of ram)-(top of stack)
0037 ;
0038 ;
0039 ;work area pointers
0040 ;
>003D 0041 TAPES EQU 3DH ;baudrate, RS232, cassette motors
0042 ; ;and keyboard flags
>003E 0043 SPEEDS EQU 3EH ;display speed
>003F 0044 OUTADD EQU 3FH ;output vector
>0041 0045 INADD EQU 41H ;input vector
>0043 0046 BATCHF EQU 43H ;batch mode flag
>0044 0047 PROMPT EQU 44H ;prompt character
>0045 0048 CMTRFG EQU 45H ;also used for baudrate, RS232, ..
>0046 0049 CRCBYT EQU 46H ;needed for CRC calculation
>0047 0050 CHEAD EQU 47H ;header from console
>0057 0051 THEAD EQU 57H ;header from tape
>0067 0052 VDHL D EQU 67H ;character under cursor
>0068 0053 LINE EQU 68H ;video line #
>006A 0054 CHR EQU 6AH ;video character #
>006C 0055 LSTKEY EQU 6CH ;needed for repeat-key

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort

0056 ;cassette header equates
>0006 0057 HTYPE EQU 6 ;file type
>0007 0058 HSIZE EQU 7 ;block size
>0009 0059 HADDR EQU 9 ;start address
>000B 0060 HXEQ EQU 11 ;go address
>0010 0061 HEADLN EQU 16 ;length header
0062 ;-----
0063 ORG 0E000H
0064 ;jump table into monitor
0065 ;
'E000 C362E0' 0066 JP INITC ;cold start
'E003 C3E8E0' 0067 JP INITW ;warm start
'E006 C377E0' 0068 JP INITU ;user start
'E009 C330E0' 0069 JP CHRIN ;console input (using vector)
'E00C C345E0' 0070 JP CHROUT ;console output (using vector)
'E00F C3DAE2' 0071 INTAPE JP TAPEIN ;serial/tape input
'E012 C3EEE2' 0072 OUTAPE JP TAPOUT ;serial/tape output
'E015 C3D1EA' 0073 JP QUIK ;quik keyboard check
'E018 C31CEB' 0074 KEYBRD JP KEYBD ;keyboard input
'E01B C3F0E9' 0075 JP CHROT1 ;video output
'E01E C376E7' 0076 PARLIN JP PARIN ;parallel input
'E021 C37FE7' 0077 PARLOT JP PAROUT ;parallel output
'E024 C38AE2' 0078 JP MOTRON ;select serial device
'E027 C3AFE2' 0079 JP MTROFF ;deselect serial device
'E02A C35AE6' 0080 JP SAVBAS ;entry for basic CLOAD
'E02D C399E7' 0081 JP LODBAS ;entry for basic CSAVE
0082 ;
'E030 FDE5 0083 CHRIN PUSH IY ;we destroy
'E032 E5 0084 PUSH HL
'E033 CDA2E1' 0085 CALL GETIY ;need for input vector
'E036 2141E0' 0086 LD HL,CHRINR
'E039 E5 0087 PUSH HL ;save return address
'E03A FD6E41 0088 LD L,(IY+INADD) ;get input vector
'E03D FD6642 0089 LD H,(IY+INADD+1)
'E040 E9 0090 JP (HL) ;call input device
'E041 E1 0091 CHRINR POP HL ;restore reg. HL & IY
'E042 FDE1 0092 POP IY
'E044 C9 0093 RET
0094 ;
'E045 FDE5 0095 CHROUT PUSH IY ;we destroy
'E047 E5 0096 PUSH HL
'E048 F5 0097 PUSH AF
'E049 CDA2E1' 0098 CALL GETIY
'E04C FD663E 0099 LD H,(IY+SPEEDS) ;get display speed (SS)
'E04F 2E01 0100 LD L,1 ;HL:=SS01H
'E051 2B 0101 OUTDLY DEC HL
'E052 7C 0102 LD A,H
'E053 B5 0103 OR L ;check if reg. HL=0000
'E054 20FB 0104 JR NZ,OUTDLY-$ ;if not, loop
'E056 F1 0105 POP AF ;restore char.
'E057 2141E0' 0106 LD HL,CHRINR
'E05A E5 0107 PUSH HL ;save return address
'E05B FD6E3F 0108 LD L,(IY+OUTADD) ;get output vector
'E05E FD6640 0109 LD H,(IY+OUTADD+1)
'E061 E9 0110 JP (HL) ;call output device

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort

0111 ; initialize routines
0112 ;
0113 ;INITC = coldstart - finds top of ram and
0114 ; sets stack and storage there
0115 ;INITW = warmstart - uses stack from INITC
0116 ;INITU = userstart - use reg. HL as top of ram like INITC
0117 ;
0118 ; coldstart entry point (E000)
0119 ;
'E062 3EFF 0120 INITC LD A,0FFH ;init. UART: 8 bits/character,
'E064 D3FD 0121 OUT (UARTS),A ;two stopbits, no parity
'E066 1602 0122 LD D,2 ;entry flag := 2
'E068 210000 0123 LD HL,0 ;start address of RAM
'E06B 7E 0124 INITC2 LD A,(HL) ;get it
'E06C 46 0125 LD B,(HL) ;twice
'E06D 2F 0126 CPL ;complement
'E06E 77 0127 LD (HL),A ;put back
'E06F BE 0128 CP (HL) ; & check if RAM
'E070 70 0129 LD (HL),B ;put real back
'E071 23 0130 INC HL ;point next
'E072 28F7 0131 JR Z,INITC2-$ ;loop if good
'E074 2B 0132 DEC HL ;adjust
'E075 2B 0133 DEC HL ; reg. HL
'E076 01 0134 DB 1 ;skip two bytes (LD BC,nnn)
0135 ;
0136 ;
0137 ; userstart entry point (E006)
0138 ;
'E077 1601 0139 INITU LD D,1 ;gives control to monitor
'E079 2200F0 0140 LD (RAMTOP),HL ;store top of ram
'E07C 0192FF 0141 LD BC,0FF92H ;calculate top of stack
'E07F 09 0142 ADD HL,BC
'E080 F9 0143 LD SP,HL ;get a stack
'E081 E5 0144 PUSH HL
'E082 FDE1 0145 POP IY ;get valid reg. IY
'E084 CDD1EA' 0146 CALL QUIK ;check for <ESC><RESET> or others
'E087 2806 0147 JR Z,INITUA-$ ;if only <RESET>
'E089 15 0148 DEC D
'E08A D603 0149 SUB CTRLC ;test if CTRL-C
'E08C 201A 0150 JR NZ,INITUB-$ ;if <ESC><RESET>
'E08E 15 0151 DEC D ;if <CTRL><C><RESET>
'E08F 066F 0152 INITUA LD B,6FH ;get length work area
'E091 77 0153 INITU2 LD (HL),A ;store 00H
'E092 23 0154 INC HL
'E093 10FC 0155 DJNZ INITU2-$ ;loop
'E095 FD36443E 0156 LD (IY+PROMPT),'>;init prompt
'E099 FD363D40 0157 LD (IY+TAPES),40H;1200 BAUD, cassette, normal keypad
'E09D D5 0158 PUSH DE ;save entry flag
'E09E CDFDE5' 0159 CALL SETOT0 ;init output to video
'E0A1 CD20E6' 0160 CALL SETIN0 ;init input from keyboard
'E0A4 CDB1E9' 0161 CALL VIDINT ;clear screen, init standard grap.
'E0A7 D1 0162 POP DE ;restore entry flag
'E0A8 15 0163 INITUB DEC D ;if reg. D= 1, set Z-flag
'E0A9 F4AEE0' 0164 CALL P,INITUC ;if reg. D= 0 or 1, test for pack
'E0AC 181A 0165 JR INITU3-$ ;if D=0 or no prom pack, warmstart

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0166 ;check if pack there, if not, return to main
0167 ;if Z-flag set; warm, else coldstart pack
0168 ;
'E0AE 08 0169 INITUC EX AF,AF' ;save zero flag
'E0AF 3AFDDF 0170 LD A,(0DFFDH) ;see if pack there
'E0B2 FEC3 0171 CP 0C3H
'E0B4 C0 0172 RET NZ ;no, go back
'E0B5 3AFADF 0173 LD A,(0DFFAH)
'E0B8 FEC3 0174 CP 0C3H
'E0BA C0 0175 RET NZ ;no pack, go back
'E0BB 08 0176 EX AF,AF' ;restore Z-flag
'E0BC C2FDDF 0177 JP NZ,0DFFDH ;if Z-flag reset, coldstart.
'E0BF C3FADF 0178 JP 0DFFAH ;else warm start pack
0179 ;
0180 ;
0181 ; correction for LOAD command
0182 ;
'E0C2 D5 0183 LOADD PUSH DE ;save relocate load address if any
'E0C3 CD34E7' 0184 CALL SKIPFL ;skip file
'E0C6 D1 0185 POP DE ;restore
'E0C7 C9 0186 RET
0187 ;
0188 ;
0189 ; prints monitor power-up & reset heading
0190 ;
'E0C8 2162E3' 0191 INITU3 LD HL,HEDING ;print message
'E0CB CDBAE1' 0192 CALL MSGOUT
'E0CE ED5B00F0' 0193 LD DE,(RAMTOP)
'E0D2 CDE8E1' 0194 CALL ADDOUT ;print top of ram
'E0D5 21BCE3' 0195 LD HL,HEAD2
'E0D8 CDBAE1' 0196 CALL MSGOUT
'E0DB FDE5 0197 PUSH IY
'E0DD D1 0198 INITU4 POP DE
'E0DE 1B 0199 DEC DE
'E0DF CDE8E1' 0200 CALL ADDOUT ;print top of stack
'E0E2 21D5E3' 0201 LD HL,HEAD3
'E0E5 CDBAE1' 0202 INITU5 CALL MSGOUT
0203 ;
0204 ;
0205 ; warmstart entry point (E003)
0206 ;
'E0E8 CDA2E1' 0207 INITW CALL GETIY ;get a valid reg. IY of ramtop
0208 ;
0209 ;
0210 ;
0211 ;
0212 ;
0213 ;
0214 ;
0215 ;
0216 ;
0217 ;
0218 ;
0219 ;
0220 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0221 ; beginning of main program
0222 ;
'E0EB FDF9 0223 LD SP,IY ;init. stack
'E0ED CD05E2' 0224 CALL CRLF ;fresh line
'E0F0 FD7E44 0225 START1 LD A,(IY+PROMPT) ;load prompt ('>')
'E0F3 CD45E0' 0226 CALL CHROUT ;print it
'E0F6 CD3AE1' 0227 CALL LINEIN ;get a line from operator
'E0F9 FDE5 0228 PUSH IY ;reg. IY points to begin of line
'E0FB E1 0229 POP HL ;copy in reg. HL
'E0FC CD25E2' 0230 CALL SCAN ;skip delims.
'E0FF CAF0E0' 0231 JP Z,START1 ;no error if empty line
'E102 DD210EE3' 0232 LD IX,TABLE ;new start address command table
'E106 E5 0233 MAIN1 PUSH HL ;save line-input pointer
'E107 DDE5 0234 PUSH IX ;save table pointer
'E109 0602 0235 LD B,2 ;check 2 chars.
'E10B 7E 0236 MAIN2 LD A,(HL) ;get character out line-input buffer
'E10C CD18EC' 0237 CALL MAIND ;make it upper case, and compare
'E10F 2012 0238 JR NZ,MAIN4-$ ;no, try next in table
'E111 23 0239 INC HL ;else compare with second char.
'E112 DD23 0240 INC IX ;check next char.
'E114 10F5 0241 DJNZ MAIN2-$ ;loop for two
'E116 D1 0242 POP DE ;clear stack
'E117 D1 0243 POP DE ;line-input pointer
'E118 01E8E0' 0244 LD BC,INITW ;after each command a complete
'E11B C5 0245 PUSH BC ;start for reg. IY
'E11C DD6E00 0246 MAIN3 LD L,(IX) ;load
'E11F DD6601 0247 LD H,(IX+1) ; jump
'E122 E9 0248 JP (HL) ; address and call it.
'E123 DDE1 0249 MAIN4 POP IX ;reset table pointer
'E125 E1 0250 POP HL ;reset line-input pointer
'E126 DD23 0251 INC IX ;point to next
'E128 DD23 0252 INC IX ; error in table
'E12A DD23 0253 INC IX
'E12C DD23 0254 INC IX
'E12E DD7E00 0255 LD A,(IX) ;is this the
'E131 E7 0256 OR A ; end of the table?
'E132 20D2 0257 JR NZ,MAIN1-$ ;if not, continue
0258 ;
'E134 21E6E3' 0259 ERRCMD LD HL,IVCMSG ;'INVALID COMMAND'
'E137 C3C9E1' 0260 JP WHAT ;print the error
0261 ;
0262 ;
0263 ; line input routine
0264 ;
0265 ;<CR> = end line
0266 ;<RUB> = backspace
0267 ;<@> = start over
0268 ;all other ASCII characters are simply input
0269 ;the CTRL- and GRAPHIC characters will only be echoed
0270 ;
'E13A FDE5 0271 LINEIN PUSH IY
'E13C E1 0272 POP HL ;set to begin of input buffer
'E13D 3E3C 0273 LD A,LINELN ;max. buffer length
'E13F 85 0274 LINEI1 ADD A,L ;entry point used by ENTER-command
'E140 4F 0275 LD C,A

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E141 45 0276 LD B,L ;begin of input buffer
'E142 FD7E43 0277 LD A,(IY+BATCHF) ;if batch mode
'E145 B7 0278 OR A
'E146 2039 0279 JR NZ,LINE3-$ ;yes, do batch mode
'E148 CD30E0' 0280 LINE1 CALL CHRIN ;get a character
'E14B 28FB 0281 JR Z,LINE1-$ ;go back, if non
'E14D CB7F 0282 BIT 7,A
'E14F 2022 0283 JR NZ,LINE2A-$ ;if graphic
'E151 77 0284 LD (HL),A
'E152 FE0D 0285 CP CR
'E154 CA04E2' 0286 JP Z,LINECR ;if carriage return
'E157 FE18 0287 LINEBT CP 'X'-40H ;test if boot character
'E159 CAFEED' 0288 JP Z,BOOTER ;yes, go boot floppy disk system
'E15C FE20 0289 LINEBR CP SPACE
'E15E 3813 0290 JR C,LINE2A-$ ;if CTRL char.
'E160 FE7F 0291 CP RUBOUT
'E162 2814 0292 JR Z,BKSPC-$
'E164 FE40 0293 CP '@'
'E166 2005 0294 JR NZ,LINE2-$ ;if valid char. inc. buffer pointer
'E168 CD05E2' 0295 CALL CRLF ;if @, new line and
'E16B 18CD 0296 JR LINEIN-$ ; start over
0297 ;
'E16D 79 0298 LINE2 LD A,C ;test if buffer full
'E16E BD 0299 CP L
'E16F 28D7 0300 JR Z,LINE1-$ ;yes, do not accept any valid char.
'E171 7E 0301 LD A,(HL) ;get char. back
'E172 23 0302 INC HL ;inc. buffer pointer
'E173 CD45E0' 0303 LINE2A CALL CHROUT ;print it
'E176 18D0 0304 JR LINE1-$ ;wait for next char.
0305 ;
'E178 78 0306 BKSPC LD A,B ;test if buffer empty
'E179 BD 0307 CP L
'E17A 28CC 0308 JR Z,LINE1-$ ;yes, ignore
'E17C 3E08 0309 LD A,8 ;load with backspace
'E17E 2B 0310 DEC HL ;dec. buffer pointer
'E17F 18F2 0311 JR LINE2A-$ ;print backspace
0312 ;
'E181 0601 0313 LINE3 LD B,1 ;do batch mode, use recorder 1
'E183 E5 0314 PUSH HL
'E184 CD8AE2' 0315 CALL MOTRON ;select recorder 1
'E187 CD59E7' 0316 CALL TAPWT ;wait for 100 NULLs and one 01H
'E18A CDDAE2' 0317 LINE4 CALL TAPEIN ;get a byte from tape
'E18D CAD4E1' 0318 JP Z,FINISH ;if he wants us
'E190 77 0319 LD (HL),A ;store in line-input buffer
'E191 23 0320 INC HL ;attention, here is no check if the
0321 ;line input buffer overruns.
'E192 FE0D 0322 CP CR ;test carriage return
'E194 20F4 0323 JR NZ,LINE4-$ ;if not, loop
'E196 CD4EE7' 0324 CALL CKCRC ;check CRC
'E199 CDAFE2' 0325 CALL MIROFF ;stop recorder
'E19C 3600 0326 LD (HL),0 ;store endmarker for MSGOUT routine
'E19E E1 0327 POP HL ;restore start address input buffer
'E19F C3BAE1' 0328 JP MSGOUT ;print what we have got and return
0329 ;
0330 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0331 ; get reg. IY from ramtop in 0F000H
0332 ;
'E1A2 C5 0333 GETIY PUSH BC
'E1A3 F5 0334 PUSH AF
'E1A4 DBFE 0335 SEEIFR IN A,(0FEH) ;get screen status
'E1A6 CB6F 0336 BIT 5,A ;test if sync.
'E1A8 28FA 0337 JR Z,SEEIFR-$ ;if not in sync.
'E1AA F1 0338 POP AF
'E1AB 0608 0339 LD B,8 ;load wait constant
'E1AD 10FE 0340 DJNZ 0 ;wait until next sync.
'E1AF FD2A00F0 0341 LD IY,(RAMTOP) ;get ramtop
'E1B3 0192FF 0342 LD BC,-STORE ;offset
'E1B6 FD09 0343 ADD IY,BC ;'subtract'
'E1B8 C1 0344 POP BC
'E1B9 C9 0345 RET
0346 ;
0347 ;
0348 ; message output routine
0349 ;
0350 ;send string pointed by reg. HL to terminal.
0351 ;a linefeed after each CR is inserted.
0352 ;a 00 means end of text and return to main.
0353 ;
'E1BA 7E 0354 MSGOUT LD A,(HL) ;get char.
'E1BB B7 0355 OR A ;test if 00H
'E1BC C8 0356 RET Z ;if so, return
'E1BD 23 0357 INC HL
'E1BE CD45E0' 0358 MSGOT2 CALL CHROUT ;print
'E1C1 FE0D 0359 CP CR ;need linefeed?
'E1C3 20F5 0360 JR NZ,MSGOUT-$ ;no
'E1C5 3E0A 0361 LD A,LF
'E1C7 18F5 0362 JR MSGOT2-$ ;do print linefeed
0363 ;
0364 ;
0365 ; 'WHAT' error routine
0366 ;
'E1C9 E5 0367 WHAT PUSH HL ;save error message address
'E1CA 21DDE3' 0368 LD HL,ERRMSG ;'ERROR - '
'E1CD CDBAE1' 0369 CALL MSGOUT
'E1D0 E1 0370 POP HL ;get back
'E1D1 CDBAE1' 0371 CALL MSGOUT ;print diagnostic message
'E1D4 FD364300 0372 FINISH LD (IY+BATCHF),0 ;clear batch mode
'E1D8 CDB4E2' 0373 CALL MTROF1 ;turn off tape
'E1DB C3E8E0' 0374 JP INITW ;do warmstart after each error
0375 ;
'E1DE 21F6E3' 0376 ERRPAR LD HL,IVPMSG ;'INVALID PARAMETER'
'E1E1 18E6 0377 JR WHAT-$
0378 ;
'E1E3 2108E4' 0379 ERRCRC LD HL,CRCMSG ;'TAPE CRC ERROR'
'E1E6 18E1 0380 JR WHAT-$
0381 ;
0382 ;
0383 ;
0384 ;
0385 ;

```

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort	ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
			0386 ; hexadecimaal output routines					0441 ; scan routines	
			0387 ;					0442 ;	
			0388 ;ADDOUT = print hex address in reg. DE					0443 ;SCANHL = scan text started from line-input buffer,	
			0389 ;HCHOUT = print hex byte in reg. A					0444 ; then SCANLT.	
			0390 ;					0445 ;SCANLT = skip characters, then SCAN	
'E1E8	7A	0391	ADDOUT LD A,D					0446 ;SCAN = scan text pointed by reg. HL.	
'E1E9	CDEDE1'	0392	CALL HCHOUT	;print MSB	'E225	7E	0449	SCAN LD A,(HL)	;get character
'E1EC	7B	0393	LD A,E	;print LSB	'E226	FE0D	0450	CP CR	;carriage return?
'E1ED	F5	0394	HCHOUT PUSH AF		'E228	C8	0451	RET Z	;yes, thru
'E1EE	E6F0	0395	AND 0F0H	;only left part	'E229	FE2E	0452	CP '.'	;delimiter?
'E1F0	0F	0396	RRCA		'E22B	D0	0453	RET NC	;yes, return
'E1F1	0F	0397	RRCA		'E22C	23	0454	SCAN1 INC HL	;next character
'E1F2	0F	0398	RRCA		'E22D	18F6	0455	JR SCAN-\$;loop
'E1F3	0F	0399	RRCA	;move to right			0456 ;		
'E1F4	CDFAE1'	0400	CALL HCHOT2	;print left part	'E22F	FDE5	0457	SCANHL PUSH IY	
'E1F7	F1	0401	POP AF	;get back byte	'E231	E1	0458	POP HL	;set reg. HL to begin input buffer
'E1F8	E60F	0402	AND 0FH	;print right part	'E232	7E	0459	SCANLT LD A,(HL)	;get character
'E1FA	FE0A	0403	HCHOT2 CP 10	;need letter	'E233	FE0D	0460	CP CR	;carriage return?
'E1FC	3802	0404	JR C,HCHOT3-\$;no	'E235	C8	0461	RET Z	;yes, thru
'E1FE	C607	0405	ADD A,7	;adjust for letter	'E236	FE30	0462	CP '0'	;delimiter?
'E200	C630	0406	HCHOT3 ADD A,'0'	;make ASCII	'E238	38EB	0463	JR C,SCAN-\$;yes, skip delim. and return
'E202	1808	0407	JR CRLF2-\$;here I need one byte	'E23A	23	0464	INC HL	;next character
			0408 ;		'E23B	18F5	0465	JR SCANLT-\$;loop
'E204	23	0409	LINECR INC HL	;needed for compatibility			0466 ;		
			0410	;called by LINEIN routine			0467 ;		
			0411 ;				0468 ;	conversion routine	
			0412 ;				0469 ;		
			0413 ; carriage return & linefeed				0470 ;	converted ASCII string pointed by reg. HL	
			0414 ;				0471 ;	to hex address in reg. DE	
			0415 ; send <CR><LF> to terminal				0472 ;	no valid digits generate 'INVALID PARAMETER' error	
			0416 ;				0473 ;		
'E205	3E0D	0417	CRLF LD A,CR		'E23D	110000	0474	CONV LD DE,0	;start with 0000H
'E207	CD45E0'	0418	CALL CHROUT		'E240	7E	0475	CONV1 LD A,(HL)	;get character from string
'E20A	3E0A	0419	LD A,LF		'E241	FE30	0476	CP '0'	;test if delimiter
'E20C	C345E0'	0420	CRLF2 JP CHROUT	;return there	'E243	D8	0477	RET C	;yes, exit
			0421 ;		'E244	23	0478	INC HL	;point to next char.
			0422 ;		'E245	FE47	0479	CP 'F'+1	;too big?
			0423 ; address and colon output		'E247	D2DEE1'	0480	JP NC,ERRPAR	;yes
			0424 ;		'E24A	FE3A	0481	CP '9'+1	;digit?
			0425 ; prints hex address (in reg. DE), colon and a space		'E24C	3807	0482	JR C,NUMBER-\$;yes
			0426 ;		'E24E	FE41	0483	CP 'A'	;letter
'E20F	.CDE8E1'	0427	ADDCOL CALL ADDOUT	;print address	'E250	DADEE1'	0484	JP C,ERRPAR	;no
'E212	3E3A	0428	LD A,':'		'E253	C609	0485	ADD A,9	;correct for A,B,C,D,E & F
'E214	CD45E0'	0429	CALL CHROUT	;print ':'	'E255	07	0486	NUMBER RLCA	
'E217	3E20	0430	SPCOUT LD A,SPACE	;entry point print one space	'E256	07	0487	RLCA	
'E219	C345E0'	0431	JP CHROUT	;print space and return	'E257	07	0488	RLCA	
			0432 ;		'E258	07	0489	RLCA	;shift to left
			0433 ;		'E259	0604	0490	LD B,4	;four bits
			0434 ; print space and hex byte (in reg. A).		'E25B	07	0491	CONV2 RLCA	;shift into carry
			0435 ;		'E25C	CB13	0492	RL E	;thru reg. E
'E21C	F5	0436	HEXSPC PUSH AF	;save hex byte	'E25E	CB12	0493	RL D	;and reg. D
'E21D	3E20	0437	LD A,SPACE		'E260	10F9	0494	DJNZ CONV2-\$;loop
'E21F	CD45E0'	0438	CALL CHROUT	;print space	'E262	18DC	0495	JR CONV1-\$;next character
'E222	F1	0439	POP AF						
'E223	18C8	0440	JR HCHOUT-\$;print hex byte and return					

```

ADDR OBJECT      ST # SOURCE STATEMENT   Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0496 ; name find routine
0497 ;copy program name, behind command-string in input buffer
0498 ;into workarea.
0499 ;
0500 ;exit: Z-flag set if no name
0501 ;      C-flag set if bad name
0502 ;
'E264 CD2FE2' 0503 NAMFND CALL SCANHL      ;skip command
'E267 C8      0504      RET Z          ;if no name
'E268 FE41    0505      CP 'A'
'E26A D8      0506      RET C          ;if bad name
'E26B FE5B    0507      CP 'Z'+1
'E26D 3F      0508      CCF           ;complement C-flag
'E26E D8      0509      RET C          ;if bad name
'E26F E5      0510      PUSH HL       ;save string pointer
'E270 FDE5    0511      PUSH IY      ;copy reg. IY
'E272 D1      0512      POP DE        ; to reg. DE
'E273 214700 0513      LD HL,CHEAD      ;offset
'E276 19      0514      ADD HL,DE
'E277 D1      0515      POP DE        ;get string pointer
'E278 0605    0516      LD B,5        ;5 characters
'E27A FE30    0517 NAMEN1 CP '0'      ;delimiter?
'E27C 13      0518      INC DE        ;next
'E27D 3003    0519      JR NC,NAMEN2-$ ;no, skip
'E27F 1B      0520      DEC DE        ;move string pointer back
'E280 3E20    0521      LD A,SPACE   ;release with space
'E282 77      0522 NAMEN2 LD (HL),A   ;store in MWA.
'E283 23      0523      INC HL
'E284 1A      0524      LD A,(DE)   ;get next char. of string
'E285 10F3    0525      DJNZ NAMEN1-$ ;loop for 5 char.
'E287 B7      0526      OR A        ;reset Z- & C-flag
'E288 EB      0527      EX DE,HL
'E289 C9      0528      RET
0529 ;
0530 ;
0531 ; cassette motor control routines
0532 ;
0533 ;select device indicated by reg. B
0534 ;
0535 ;if reg. B=01H: select recorder 1
0536 ;      02H: select recorder 2
0537 ;      03H: select RS232 300 baud
0538 ;      04H: select RS232 1200 baud
0539 ;      other: select recorder 2
0540 ;
'E28A FDE5    0541 MOTRON PUSH IY      ;in reg. B motor #
'E28C CDA2E1' 0542      CALL GETIY
'E28F FD7E3D 0543      LD A,(IY+TAPES) ;get motor flags
'E292 05      0544      DEC B
'E293 C4FBEB' 0545      CALL NZ,MOT   ;if not motor 1
'E296 CC73E9' 0546      CALL Z,MOT1   ;if motor 1
'E299 D3FE    0547      OUT (0FEH),A
'E29B FD7745 0548      LD (IY+CMTRFG),A ;put away
'E29E FDE1    0549      POP IY
0550 ;

```

```

ADDR OBJECT      ST # SOURCE STATEMENT   Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E2A0 0604    0551 DELAY LD B,4        ;loop count
'E2A2 E5      0552 DELAY1 PUSH HL       ;we destroy
'E2A3 210000 0553 DELAY2 LD HL,0      ;clear it
'E2A6 2B      0554 DELAY3 DEC HL
'E2A7 7C      0555      LD A,H
'E2A8 B5      0556      OR L        ;test if reg. HL=0000
'E2A9 20FB    0557      JR NZ,DELAY3-$ ;loop
'E2AB 10F6    0558      DJNZ DELAY2-$ ;some more
'E2AD E1      0559      POP HL
'E2AE C9      0560      RET
0561 ;
0562 ;
'E2AF 0601    0563 MTROFF LD B,1
'E2B1 CDA2E2' 0564      CALL DELAY1 ;wait
0565 ;
'E2B4 FDE5    0566 MTROF1 PUSH IY
'E2B6 CDA2E1' 0567      CALL GETIY
'E2B9 FD7E3D 0568      LD A,(IY+TAPES) ;restore old motor flags
'E2BC FD7745 0569      LD (IY+CMTRFG),A
'E2BF C3D6EA' 0570      JP QUICK   ;send to port 0FEH, restore reg. IY
0571 ; and return
0572 ;
0573 ;
0574 ;null routine, sends 100 nulls and one 01H to uart
0575 ;
'E2C2 0664    0576 NULL LD B,100
'E2C4 AF      0577 NULL1 XOR A
'E2C5 CDEEE2' 0578      CALL TAPOUT
'E2C8 10FA    0579      DJNZ NULL1-$ ;loop 100 times
'E2CA 3C      0580      INC A      ;reg. A:=01H
'E2CB CDEEE2' 0581      CALL TAPOUT
'E2CE FD7046 0582      LD (IY+CRCBY),B ;clear CRC byte
'E2D1 C9      0583      RET
0584 ;
0585 ;
0586 ;SPACE routine, send as many spaces as the number in reg. B
0587 ;
'E2D2 3E20    0588 SPACES LD A,SPACE
'E2D4 CD45E0' 0589      CALL CHROUT
'E2D7 10F9    0590      DJNZ SPACES-$ ;loop until reg. B=00
'E2D9 C9      0591      RET
0592 ;
0593 ;
0594 ; serial input routine. (E00F)
0595 ;reads a character into reg. A from the serial-input or tape
0596 ;
'E2DA FDE5    0597 TAPEIN PUSH IY
'E2DC CDA2E1' 0598      CALL GETIY ;get reg. IY
'E2DF CDD1EA' 0599      TAPIN1 CALL QUIK ;user?
'E2E2 2025    0600      JR NZ,TAPLV-$ ;he wants us
'E2E4 DBFD    0601      IN A,(UARTS) ;get uart status
'E2E6 CB4F    0602      BIT 1,A    ;test data available
'E2E8 28F5    0603      JR Z,TAPIN1-$ ;if not, try again
'E2EA DBFC    0604      IN A,(UARTD) ;get data-byte
'E2EC 180F    0605      JR CRCOMP-$ ;go adjust CRC byte

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0606 ;serial output routine. (E012)
0607 ;writes a character from reg. A to serial-output or tape
0608 ;
'E2EE FDE5 0609 TAPOUT PUSH IY
'E2F0 CDA2E1' 0610 CALL GETIY ;get reg. IY
'E2F3 F5 0611 PUSH AF ;save byte to write
'E2F4 DBFD 0612 TAPOT1 IN A,(UARTS) ;get uart status
'E2F6 CB47 0613 BIT 0,A ;transmit buffer empty?
'E2F8 28FA 0614 JR Z,TAPOT1-$ ;no, wait
'E2FA F1 0615 POP AF
'E2FB D3FC 0616 OUT (UARTD),A ;write data to uart
0617 ;
'E2FD F5 0618 CRCOMP PUSH AF
'E2FE FD9646 0619 SUB (IY+CRCBYT) ;calculate CRC byte
'E301 2F 0620 CPL
'E302 FD7746 0621 LD (IY+CRCBYT),A
'E305 F1 0622 POP AF
'E306 FDE1 0623 TAPLV2 POP IY
'E308 C9 0624 RET
0625 ;
'E309 AF 0626 TAPLV XOR A ;set zero flag
'E30A 18FA 0627 JR TAPLV2-$
0628 ;
'E30C FFFF 0629 DB 0FFH,0FFH ;free
0630 ;
0631 ;
0632 ; COMMAND TABLE
0633 ;
0634 ; formatted as follows:
0635 ; 2 bytes ASCII command
0636 ; 2 bytes jump address
0637 ; end byte is 00H
0638 ;
'E30E 464C 0639 TABLE DB 'FL' ;new start address of command table
'E310 65E5' 0640 DEFW FILL ;entry point FILL command
'E312 4455 0641 DB 'DU' ;dump from memory
'E314 D3E4' 0642 DEFW DUMP
'E316 454E 0643 DB 'EN' ;enter to memory
'E318 38E5' 0644 DEFW ENTER
'E31A 5341 0645 DB 'SA' ;save memory to cassette
'E31C 38E6' 0646 DEFW SAVE
'E31E 4C4F 0647 DB 'LO' ;load into memory from cassette
'E320 8AE7' 0648 DEFW LOAD
'E322 4649 0649 DB 'FI' ;list cassette files
'E324 B9E6' 0650 DEFW FILES
'E326 474F 0651 DB 'GO' ;go to subroutine
'E328 97E5' 0652 DEFW GO
'E32A 4352 0653 DB 'CR' ;create batch file
'E32C 5CE8' 0654 DEFW CREAT
'E32E 5345 0655 DB 'SE' ;set parameters
'E330 A2E5' 0656 DEFW SET
'E332 4D4F 0657 DB 'MO' ;copy block memory
'E334 42E9' 0658 DEFW MOVE ;new entry point MOVE command
'E336 5445 0659 DB 'TE' ;memory test
'E338 A1E8' 0660 DEFW TEST

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E33A 4241 0661 DB 'BA' ;execute batch tape
'E33C 58E8' 0662 DEFW BATCH
'E33E 4C49 0663 DB 'LI' ;list batch tape
'E340 84E8' 0664 DEFW LIST
'E342 5052 0665 DB 'PR' ;change prompt char.
'E344 45E8' 0666 DEFW PRMPTC
'E346 4F56 0667 DB 'OV' ;end batch tape
'E348 D4E1' 0668 DEFW FINISH
'E34A 5050 0669 DB 'PP' ;prom-pack
'E34C 8AE9' 0670 DEFW PROMPK
'E34E 00 0671 DB 0 ;end command table
0672 ;
0673 ;
0674 ; SET COMMAND TABLE
0675 ;
0676 ; formatted as follows:
0677 ; 1 byte ASCII command
0678 ; 2 bytes jump address
0679 ; end byte is 00H
0680 ;
'E34F 54 0681 SETTBL DB 'T' ;set baudrate, recorders, keyboard
'E350 ABEA' 0682 DEFW TAPE0 ;new entry point 'SE T=' command
'E352 53 0683 DB 'S' ;set display speed
'E353 EAE5' 0684 DEFW SPEED
'E355 58 0685 DB 'X' ;set goaddress and date
'E356 DEEB' 0686 DEFW DATSET ;new entry point 'SE X=' command
'E358 46 0687 DB 'F' ;set file type
'E359 EEE5' 0688 DEFW SETFIL
'E35B 4F 0689 DB 'O' ;set output device
'E35C F9E5' 0690 DEFW SETOUT
'E35E 49 0691 DB 'I' ;set input device
'E35F 1CE6' 0692 DEFW SETIN
'E361 00 0693 DB 0 ;end table
0694 ;
0695 ;
0696 ; MESSAGE TABLE
0697 ;
0698 ;new power-up message
0699 ;
'E362 0D537461 0700 HEDING DB CR,'Standard Monitor ',0FFH,0FFH
6E646172
64204D6F
6E69746F
7220FFFF
'E376 4F0D076 0701 DB 0FFH,CR,CR,'version 1.3/B'
65727369
6F6E2031
2E332F42
'E386 0D284329 0702 DB CR,'(C) 1982 by Gebr. van Montfort. ',CR,CR
20313938
32206279
20476562
722E2076
616E204D
6F6E7466

```



```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
      6F72742E
      200D0D
'E3A9 54484520 0703 DB 'THE TOP OF RAM IS ',0
      544F5020
      4F462052
      414D2049
      532000
'E3BC 20484558 0704 HEAD2 DB ' HEX.',CR
      2E0D
'E3C2 53544143 0705 DB 'STACK BEGINS FROM ',0
      4E204245
      47494E53
      2046524F
      4D2000
'E3D5 20484558 0706 HEAD3 DB ' HEX.',CR,CR,0
      2E0D0D0D
'E3DD 4552524F 0707 ERRMSG DB 'ERROR - ',0
      52202D20
      00
'E3E6 494E5641 0708 IVCMSG DB 'INVALID COMMAND',0
      4C494420
      434F4D4D
      414E4400
'E3F6 494E5641 0709 IVPMSG DB 'INVALID PARAMETER',0
      4C494420
      50415241
      4D455445
      5200
'E408 54415045 0710 CRCMSG DB 'TAPE '
      20
'E40D 43524320 0711 CRC DB 'CRC ERROR',0 ;need for bootstrap
      4552524F
      5200
'E417 0D414444 0712 DHEAD DB CR,'ADDR 0 1 2 3 4 5 6 7 '
      52202020
      30202031
      20203220
      20332020
      20342020
      35202036
      20203720
'E437 20203820 0713 DB ' 8 9 A B C D E F',CR,CR,0
      20392020
      41202042
      20202043
      20204420
      20452020
      460D0D0D
'E453 0D0D4E41 0714 FILHD DB CR,CR,'NAME FILE BLCK ADDR GOADDRS',CR,CR,0
      4D452020
      2046494C
      4520424C
      434B2041
      44445220
      474F4144
    
```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
      4452530D
      0D00
'E475 0D414444 0715 TESTHD DB CR,'ADDR BIT 0 1 2 3 4 '
      52202020
      42495420
      30202020
      31202020
      32202020
      33202020
      34202020
'E495 35202020 0716 DB '5 6 7',CR,CR,0
      36202020
      370D0D00
'E4A1 42414420 0717 BADMSG DB 'BAD ',0
      00
'E4A6 4F4B2020 0718 OKMSG DB 'OK ',0
      00
'E4AB 20205041 0719 PSCMSG DB ' PASS COMPLETED.',CR,CR,0
      53532043
      4F4D504C
      45544544
      2E0D0D0D
'E4BF 0D4C4F41 0720 LDGMSG DB CR,'LOADING -',0
      44494E47
      202D00
'E4CA 464F554E 0721 FNDMSG DB 'FOUND - ',0
      44202D20
      00
      0722 ;
      0723 ;
      0724 ; DUMP command
      0725 ;
'E4D3 CD0CEC' 0726 DUMP CALL GETPAR ;get first parameter; START address
'E4D6 2852 0727 JR Z,SDUMP-$ ;if only one parameter
'E4D8 7B 0728 LD A,E
'E4D9 E6F0 0729 AND 0F0H ;only even addresses
'E4DB 5F 0730 LD E,A
'E4DC D5 0731 PUSH DE ;save START address
'E4DD CD3DE2' 0732 CALL CONV ;get second parameter, END address
'E4E0 E1 0733 POP HL ;restore START
'E4E1 EB 0734 EX DE,HL
'E4E2 04 0735 INC B ;reg. B:= 1
      0736 ;
'E4E3 CD05E2' 0737 DUMP0 CALL CRLF ;new line
'E4E6 100A 0738 DJNZ DUMP2-$ ;reg. B:=B-1, skip if reg. B not 0
'E4E8 E5 0739 DUMP1 PUSH HL ;save END address
'E4E9 2117E4' 0740 LD HL,DHEAD ;print dump heading
'E4EC CDBAE1' 0741 CALL MSGOUT
'E4EF 0610 0742 LD B,16 ;reg. B:=16
'E4F1 E1 0743 POP HL ;restore end address
'E4F2 CD23E5' 0744 DUMP2 CALL QUIKT ;see if touched my keyboard
'E4F5 CD0FE2' 0745 CALL ADDCOL ;print address
'E4F8 1A 0746 DUMP3 LD A,(DE)
'E4F9 CD1CE2' 0747 CALL HEXSPC ;print contents
'E4FC CD3CE9' 0748 CALL ENDCK ;through ?
    
```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E4FF CA05E2' 0749 JP Z,CRLF ;yes, go back after CR/LF
'E502 13 0750 INC DE ;next address
'E503 7B 0751 LD A,E
'E504 E60F 0752 AND 0FH ;test if end-of-line
'E506 28DB 0753 JR Z,DUMP0-$ ;yes, go new line
'E508 E603 0754 AND 3 ;test if space need
'E50A CC17E2' 0755 CALL Z,SPCOUT ;yes, print a space
'E50D 18E9 0756 JR DUMP3-$ ;continue
0757 ;
0758 ;
0759 ;subroutine of ENTER & FILL command
0760 ;
'E50F FE2F 0761 FILL3 CP '/' ;test if terminator
'E511 C8 0762 RET Z ;yes, go back
'E512 D5 0763 PUSH DE
'E513 CD3DE2' 0764 CALL CONV ;get hex value
'E516 7B 0765 LD A,E
'E517 D1 0766 POP DE
'E518 12 0767 LD (DE),A ;store in memory
'E519 13 0768 INC DE ;next address
'E51A C9 0769 RET
0770 ;
'E51B FF 0771 DB 0FFH ;not used yet
0772 ;
0773 ;
0774 ;subroutine of the TEST command
0775 ;
'E51C 21FFF7 0776 FLASH LD HL,0F7FFH ;point to right bottem of screen
'E51F 3E0A 0777 LD A,10
'E521 AE 0778 XOR (HL) ;exchange astrix to space or reverse
'E522 77 0779 LD (HL),A
'E523 CD77E9' 0780 QUIKT CALL QUIKWT ;check if touched my keyboard
'E526 C8 0781 RET Z ;if not, continue
'E527 C3D4E1' 0782 JP FINISH ;else, abort and do warmstart
0783 ;
0784 ;
0785 ;subroutine of the DUMP & ENTER command
0786 ;start address not changed for compatibility
0787 ;
'E52A CD0FE2' 0788 SDUMP CALL ADDCOL ;print address
'E52D 1A 0789 LD A,(DE)
'E52E C3EDE1' 0790 JP HCHOUT ;print contence and return
0791 ;
0792 ;
0793 ;this routine is used if LINEIN routine is called from outside
0794 ;of the monitor or no controller is present
0795 ;
'E531 3A58E1' 0796 BOOTR LD A,(LINEBT+1) ;get back typed control char.
'E534 C35CE1' 0797 JP LINEBR ;continue LINEIN subroutine
0798 ;
'E537 FF 0799 DB 0FFH
0800 ;
0801 ;
0802 ;
0803 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0804 ; ENTER command
0805 ;
'E538 CD0CEC' 0806 ENTER CALL GETPAR ;get address
'E53B 1819 0807 JR ENTER4-$ ;evaluate parameters if any
0808 ;
'E53D 13 0809 ENTER1 INC DE ;skip a address if only CR
'E53E CD2AE5' 0810 ENTER2 CALL SDUMP ;print contents of memory
'E541 CD17E2' 0811 CALL SPCOUT ;print space
'E544 FDE5 0812 PUSH IY
'E546 E1 0813 POP HL ;get start of input buffer
'E547 E5 0814 PUSH HL
'E548 3E36 0815 LD A,36H ;max. length input buffer
'E54A CD3FE1' 0816 CALL LINE1
'E54D E1 0817 POP HL
'E54E CD25E2' 0818 CALL SCAN
'E551 28EA 0819 JR Z,ENTER1-$ ;if empty line
'E553 01 0820 DB 1 ;skip two instructions
'E554 1B 0821 ENTER3 DEC DE ;one mem. address back
'E555 23 0822 INC HL ;next char. out of buffer
'E556 CD25E2' 0823 ENTER4 CALL SCAN ;skip to next parameter
'E559 28E3 0824 JR Z,ENTER2-$ ;if no parameters in input buffer
'E55B FE5E 0825 CP '^'
'E55D 28F5 0826 JR Z,ENTER3-$
'E55F CD0FE5' 0827 CALL FILL3 ;get parameter and store in memory
'E562 C8 0828 RET Z ;if terminator ( )
'E563 18F1 0829 JR ENTER4-$ ;continue
0830 ;
0831 ;
0832 ; FILL command (FL)
0833 ;
'E565 CD0CEC' 0834 FILL CALL GETPAR ;first parameter, START
'E568 D5 0835 PUSH DE
'E569 D5 0836 PUSH DE ;needed two times
'E56A CD0FEC' 0837 CALL GETPA1 ;second parameter, END
'E56D CADEE1' 0838 JP Z,ERRPAR ;error, if nothing to fill with
'E570 EB 0839 EX DE,HL
'E571 E3 0840 EX (SP),HL ;(SP)=END
'E572 EB 0841 EX DE,HL ;DE=START
'E573 CD25E2' 0842 FILL1 CALL SCAN ;scans for a parameter
'E576 C40FE5' 0843 CALL NZ,FILL3 ;if parameters left
'E579 20F8 0844 JR NZ,FILL1-$ ;if no terminator
0845 ;
'E57B E1 0846 FILL2 POP HL ;get END
'E57C B7 0847 OR A ;reset C-flag
'E57D ED52 0848 SBC HL,DE ;END-(START + #parameters)
'E57F 44 0849 LD B,H
'E580 4D 0850 LD C,L
'E581 E1 0851 POP HL ;get START
'E582 03 0852 FILL5 INC BC
'E583 EDB0 0853 LDIR ;copies repeatly
'E585 C9 0854 RET
0855 ;
0856 ;
0857 ;
0858 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0859 ;subroutine needed by SAVE command (S-option)
0860 ;
'E586 FE53 0861 SCONV CP 'S' ;if no S is used
'E588 C23DE2' 0862 JP NZ,CONV ;go, get end address
'E58B D5 0863 PUSH DE ;save BEGIN
'E58C CD2CE2' 0864 CALL SCAN1 ;skip 'S'
'E58F CD0FEC' 0865 CALL GETPA1 ;get LENGTH
'E592 E3 0866 EX (SP),HL ;(SP)=HL, HL:=BEGIN
'E593 19 0867 ADD HL,DE ;HL:=BEGIN+LENGTH
'E594 D1 0868 POP DE ;restore reg. HL
'E595 EB 0869 EX DE,HL ;DE=BEGIN + LENGTH
'E596 C9 0870 RET
0871 ;
0872 ;
0873 ; GO command
0874 ;
'E597 CD2FE2' 0875 GO CALL SCANHL ;skip command
'E59A CADEE1' 0876 JP Z,ERRPAR ;if no parameter
'E59D CD3DE2' 0877 CALL CONV ;get address
'E5A0 EB 0878 EX DE,HL ;put in reg. HL
'E5A1 E9 0879 JP (HL) ;jump
0880 ;
0881 ;
0882 ; SET command
0883 ;
'E5A2 CD2FE2' 0884 SET CALL SCANHL ;skip 'SE'
'E5A5 CADEE1' 0885 JP Z,ERRPAR ;if no parameter
'E5A8 DD214FE3' 0886 LD IX,SETTBL ;point to table
'E5AC DDBE00 0887 SET1 CP (IX) ;is it?
'E5AF 2811 0888 JR Z,SET2-$ ;yes
'E5B1 DD23 0889 INC IX ;skip ASCII
'E5B3 DD23 0890 INC IX ;skip
'E5B5 DD23 0891 INC IX ; jump address
'E5B7 F5 0892 PUSH AF ;save char.
'E5B8 DD7E00 0893 LD A,(IX)
'E5BB B7 0894 OR A ;end?
'E5BC CADEE1' 0895 JP Z,ERRPAR ;yes, invalid parameter
'E5BF F1 0896 POP AF ;get back
'E5C0 18EA 0897 JR SET1-$
'E5C2 23 0898 SET2 INC HL ;skip char.
'E5C3 CD25E2' 0899 CALL SCAN ;skip delimiters
'E5C6 FE3D 0900 CP '='
'E5C8 C2DEE1' 0901 JP NZ,ERRPAR ;no '=', then error
'E5CB 23 0902 INC HL ;skip '='
'E5CC CD25E2' 0903 CALL SCAN ;skip delimiters
'E5CF CADEE1' 0904 JP Z,ERRPAR ;if no second parameter
'E5D2 FE47 0905 CP 'G' ;test if hex. digit
'E5D4 DC3DE2' 0906 CALL C,CONV ;get hex. parameter
'E5D7 44 0907 LD B,H ;save pointer into line input buffer
'E5D8 4D 0908 LD C,L ;changed for SE X= command
'E5D9 DD23 0909 SET3 INC IX
'E5DB C31CE1' 0910 JP MAIN3 ;jump from table
0911 ;
0912 ;
0913 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0914 ;dummy routine, not deleted for compatibility
0915 ;
'E5DE 7B 0916 TAPE LD A,E
'E5DF B7 0917 OR A
'E5E0 3E00 0918 LD A,0
'E5E2 2002 0919 JR NZ,TAPE1-$
'E5E4 3E40 0920 LD A,40H
'E5E6 FD773D 0921 TAPE1 LD (IY+TAPES),A
'E5E9 C9 0922 RET
0923 ;
0924 ;
0925 ;set display speed, 'SE S='
0926 ;
'E5EA FD733E 0927 SPEED LD (IY+SPEEDS),E
'E5ED C9 0928 RET
0929 ;
0930 ;
0931 ;set file type, 'SE F='
0932 ;
'E5EE FD734D 0933 SETFIL LD (IY+CHEAD+HTYPE),E
'E5F1 C9 0934 RET
0935 ;
0936 ;
0937 ;part of set go-address, 'SE X='
0938 ;
'E5F2 FD7352 0939 XEQUSET LD (IY+CHEAD+HXEQ),E ;enter GO address
'E5F5 FD7253 0940 LD (IY+CHEAD+HXEQ+1),D
'E5F8 C9 0941 RET
0942 ;
0943 ;
0944 ;set output device, 'SE O='
0945 ;
'E5F9 FE56 0946 SETOUT CP 'V'
'E5FB 2003 0947 JR NZ,SETOT1-$
'E5FD 11F0E9' 0948 SETOT0 LD DE,CHROT1 ;called from INITU2
;SE O=V, point direct into monitor
0949
'E600 FE50 0950 SETOT1 CP 'P'
'E602 2003 0951 JR NZ,SETOT2-$
'E604 1121E0' 0952 LD DE,PARLOT
'E607 FE53 0953 SETOT2 CP 'S'
'E609 2003 0954 JR NZ,SETOT3-$
'E60B 1112E0' 0955 LD DE,OUTAPE
'E60E FE4C 0956 SETOT3 CP 'L'
'E610 2003 0957 JR NZ,SETOT4-$
'E612 1193E9' 0958 LD DE,CENDRV
'E615 FD733F' 0959 SETOT4 LD (IY+OUTADD),E ;set output vector
'E618 FD7240 0960 LD (IY+OUTADD+1),D
'E61B C9 0961 RET
0962 ;
0963 ;
0964 ;
0965 ;
0966 ;
0967 ;
0968 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
0969 ;set input device, 'SE I='
0970 ;
'E61C FE4B 0971 SETIN CP 'K'
'E61E 2003 0972 JR NZ,SETIN1-$
'E620 1118E0' 0973 SETIN0 LD DE,KEYBRD
'E623 FE50 0974 SETIN1 CP 'P'
'E625 2003 0975 JR NZ,SETIN2-$
'E627 111EE0' 0976 LD DE,PARLIN
'E62A FE53 0977 SETIN2 CP 'S'
'E62C 2003 0978 JR NZ,SETIN3-$
'E62E 110FE0' 0979 LD DE,INTAPE
'E631 FD7341 0980 SETIN3 LD (IY+INADD),E ;set input vector
'E634 FD7242 0981 LD (IY+INADD+1),D
'E637 C9 0982 RET
0983 ;
0984 ;
0985 ; SAVE command
0986 ;
'E638 CD64E2' 0987 SAVE CALL NAMFND ;get file name
'E63B CADEE1' 0988 SAVE0 JP Z,ERRPAR ;if no name
'E63E DADEE1' 0989 JP C,ERRPAR ;if bad name
'E641 CD32E2' 0990 CALL SCANLT ;skip rest of name, if any
'E644 CADEE1' 0991 JP Z,ERRPAR ;if no begin address
'E647 CD3DE2' 0992 CALL CONV ;get begin address
'E64A D5 0993 PUSH DE ;save
'E64B FD7350 0994 LD (IY+CHEAD+HADDR),E;store begin address
'E64E FD7251 0995 LD (IY+CHEAD+HADDR+1),D
'E651 CD25E2' 0996 CALL SCAN ;find second parameter
'E654 28E5 0997 JR Z,SAVE0-$ ;if no second parameter
'E656 1B 0998 DEC DE ;adjust for calculation of length
'E657 CD86E5' 0999 CALL SCONV ;get END address
'E65A EB 1000 SAVBAS EX DE,HL ;save line-input pointer
1001 ;(entry point E02A)
'E65B C1 1002 POP BC ;restore begin address
'E65C C5 1003 PUSH BC
'E65D 37 1004 SCF
'E65E 3F 1005 CCF ;clear carry flag
'E65F ED42 1006 SBC HL,BC ;calculate block size
'E661 23 1007 INC HL ;adjust, this is why basic programs
1008 ;always grow
'E662 E5 1009 PUSH HL ;save block size
'E663 FD754E 1010 LD (IY+CHEAD+HSIZE),L ;store block size in header
'E666 FD744F 1011 LD (IY+CHEAD+HSIZE+1),H
'E669 FD364C55 1012 LD (IY+CHEAD+5),55H ;store exidy-file type
'E66D EB 1013 EX DE,HL
'E66E 0601 1014 LD B,1 ;default cassette device
'E670 CD25E2' 1015 CALL SCAN
'E673 2804 1016 JR Z,SAVE1-$ ;if no parameters left
'E675 CD3DE2' 1017 CALL CONV ;get cassette device
'E678 43 1018 LD B,E ;into reg. B
'E679 CD8AE2' 1019 SAVE1 CALL MOTRON ;select device
'E67C CDC2E2' 1020 CALL NULL ;send 100 nulls, init CRC-byte
'E67F FDE5 1021 PUSH IY
'E681 DDE1 1022 POP IX ;copy reg. IY to reg. IX
'E683 0610 1023 LD B,HEADLN ;length header

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E685 DD7E47 1024 SAVE2 LD A,(IX+CHEAD) ;get byte
'E688 CDEEE2' 1025 CALL TAPOUT ;send to UART
'E68B DD23 1026 INC IX ;next
'E68D 10F6 1027 DJNZ SAVE2-$ ;loop
'E68F CD9BE8' 1028 CALL WRCRC ;send CRC-byte
'E692 CDC2E2' 1029 CALL NULL ;write nulls after header, clear CRC
'E695 D1 1030 POP DE ;restore block size
'E696 E1 1031 POP HL ;restore begin address
'E697 CDA9E6' 1032 SAVE3 CALL BLKADJ ;get in reg. B number bytes to save
'E69A CAAFE2' 1033 JP Z,MTROFF ;if thru, turn off motor and exit
'E69D 7E 1034 SAVE4 LD A,(HL) ;get byte
'E69E CDEEE2' 1035 CALL TAPOUT ;send to serial/tape
'E6A1 23 1036 INC HL
'E6A2 10F9 1037 DJNZ SAVE4-$ ;loop for block
'E6A4 CD9BE8' 1038 CALL WRCRC ;send CRC-byte
'E6A7 18EE 1039 JR SAVE3-$ ;keep going
1040 ;
1041 ;
1042 ; calculate block-length routine
1043 ;input reg. DE: total block length
1044 ;output reg. B: short block length
1045 ;
'E6A9 AF 1046 BLKADJ XOR A
'E6AA FD7746 1047 LD (IY+CRCBY),A ;clear CRC-byte
'E6AD 47 1048 LD B,A ;also reg. B
'E6AE B2 1049 OR D ;1-255 bytes left
'E6AF 2005 1050 JR NZ,BLKAD2-$ ;no, length short block is 256 bytes
'E6B1 B3 1051 OR E
'E6B2 C8 1052 RET Z ;no bytes left
'E6B3 43 1053 LD B,E ;short block length is byte left
'E6B4 5A 1054 LD E,D ;reg. E:=00
'E6B5 C9 1055 RET
'E6B6 15 1056 BLKAD2 DEC D ;one less
'E6B7 B7 1057 OR A ;reset Z-flag
'E6B8 C9 1058 RET
1059 ;
1060 ;
1061 ; FILE command
1062 ;
'E6B9 CD2FE2' 1063 FILES CALL SCANHL ;skip 'FI'
'E6BC 0601 1064 LD B,1 ;default device
'E6BE 2804 1065 JR Z,FILES1-$ ;if no parameter
'E6C0 CD3DE2' 1066 CALL CONV ;get device
'E6C3 43 1067 LD B,E
'E6C4 2153E4' 1068 FILES1 LD HL,FILHD ;file heading
'E6C7 CDBAE1' 1069 CALL MSGOUT ;print 'NAME FILE BLCK ADDR GOADDR'
'E6CA CD8AE2' 1070 CALL MOTRON ;select device
'E6CD CD1BE7' 1071 FILES2 CALL GETHED ;get file header from UART
'E6D0 CDDEE6' 1072 CALL HEDPRT ;print header
'E6D3 FD7E5C 1073 LD A,(IY+THEAD+5);get exidy-file type
'E6D6 B7 1074 OR A
'E6D7 28F4 1075 JR Z,FILES2-$ ;if 00, then no nulls after header
'E6D9 CD34E7' 1076 CALL SKIPFL ;skip file, do CRC-check
'E6DC 18EF 1077 JR FILES2-$ ;loop, until operator interrupt, or
1078 ;CRC error occurs

```

```

Monitor 1.3/B (C) 1983 by Gebr. van Montfort
1079 ;print fileheader routine
1080 ;
'E6DE FDE5 1081 HEDPRT PUSH IY
'E6E0 DDE1 1082 POP IX ;copy reg. IY to reg. IX
'E6E2 0605 1083 LD B,5 ;length name is 5 characters
'E6E4 DD7E57 1084 FILES3 LD A,(IX+THEAD) ;get char. of filename
'E6E7 CDD6EB' 1085 CALL NOCTRL ;if CTRL-char. replace then with
1086 ;space in a filename
'E6EA DD23 1087 INC IX ;next
'E6EC 10F6 1088 DJNZ FILES3-$ ;loop for 5
'E6EE 0603 1089 LD B,3
'E6F0 CDD2E2' 1090 CALL SPACES ;three spaces
'E6F3 DD23 1091 INC IX ;skip exidy-file type byte
'E6F5 DD7E57 1092 LD A,(IX+THEAD) ;get file-type
'E6F8 CDD6EB' 1093 CALL NOCTRL ;print it, but no CTRL char.
'E6FB 0603 1094 LD B,3
'E6FD CDD2E2' 1095 CALL SPACES ;three spaces
'E700 DD23 1096 INC IX ;skip file-type
'E702 0603 1097 LD B,3 ;three address
'E704 DD5E57 1098 FILES4 LD E,(IX+THEAD) ;LSB
'E707 DD5658 1099 LD D,(IX+THEAD+1);MSB
'E70A DD23 1100 INC IX
'E70C DD23 1101 INC IX
'E70E CDE8E1' 1102 CALL ADDOUT ;print address
'E711 3E20 1103 LD A,SPACE
'E713 CD45E0' 1104 CALL CHROUT ;print space
'E716 10EC 1105 DJNZ FILES4-$ ;loop for 3
'E718 C31CE9' 1106 JP FILES5 ;go print date
1107 ;
1108 ; get header
1109 ;loads header from UART into MWA
1110 ;
'E71B CD59E7' 1111 GETHED CALL TAPWT ;wait for 10 nulls and a 01H
'E71E FDE5 1112 PUSH IY
'E720 DDE1 1113 POP IX ;copy reg. IY to reg. IX
'E722 0610 1114 LD B,HEADLN ;length header
'E724 CDDAE2' 1115 GETHD1 CALL TAPEIN ;get byte from serial/tape
'E727 CAD4E1' 1116 JP Z,FINISH ;if operator wants control
'E72A DD7757 1117 LD (IX+THEAD),A ;store in monitor work area
'E72D DD23 1118 INC IX
'E72F 10F3 1119 DJNZ GETHD1-$ ;loop
'E731 C34EE7' 1120 JP CKCRC ;check CRC and exit
1121 ;
1122 ; skip cassette file without loading in memory
1123 ;
'E734 CD59E7' 1124 SKIPFL CALL TAPWT ;wait for 10 nulls and a 01H
'E737 FD5E5E 1125 LD E,(IY+THEAD+HSIZE) ;get block size
'E73A FD565F 1126 LD D,(IY+THEAD+HSIZE+1)
'E73D CDA9E6' 1127 SKIPF1 CALL BLKADJ ;adjust in blocks
'E740 C8 1128 RET Z ;if thru
'E741 CDDAE2' 1129 SKIPF2 CALL TAPEIN ;get a byte from serial/tape
'E744 CAD4E1' 1130 SKIPF3 JP Z,FINISH ;if some one wants us
'E747 10F8 1131 DJNZ SKIPF2-$ ;loop for short block
'E749 CD4EE7' 1132 CALL CKCRC ;check CRC byte
'E74C 18EF 1133 JR SKIPF1-$

```

```

Monitor 1.3/B (C) 1983 by Gebr. van Montfort
1134 ; check CRC-byte
1135 ;
'E74E FD4646 1136 CKCRC LD B,(IY+CRCBYT)
'E751 CDDAE2' 1137 CALL TAPEIN
'E754 B8 1138 CP B
'E755 C2E3E1' 1139 JP NZ,ERRCRC
'E758 C9 1140 RET
1141 ;
1142 ;
1143 ;tape wait routine
1144 ;waits for ten NULLs followed by other NULLs till a 01H
1145 ;
'E759 C5 1146 TAPWT PUSH BC
'E75A 060A 1147 TAPWT1 LD B,10 ;wait for 10 NULL's
'E75C CDDAE2' 1148 TAPWT2 CALL TAPEIN
'E75F 28E3 1149 JR Z,SKIPF3-$ ;if operator
'E761 B7 1150 OR A
'E762 20F6 1151 JR NZ,TAPWT1-$ ;if no NULL
'E764 10F6 1152 DJNZ TAPWT2-$ ;loop for 10
'E766 CDDAE2' 1153 TAPWT3 CALL TAPEIN
'E769 28D9 1154 JR Z,SKIPF3-$ ;if operator
'E76B FE01 1155 CP 1
'E76D 38F7 1156 JR C,TAPWT3-$ ;if a NULL
'E76F 20E9 1157 JR NZ,TAPWT1-$ ;if no NULL and no '01H'
'E771 FD7046 1158 LD (IY+46H),B ;clear CRC byte
'E774 C1 1159 POP BC
'E775 C9 1160 RET
1161 ;
1162 ;
1163 ;parallel input routine. (E01E)
1164 ;
'E776 DBFE 1165 PARIN IN A,(0FEH) ;get status
'E778 CB7F 1166 BIT 7,A ;data available?
'E77A 28FA 1167 JR Z,PARIN-$ ;no, wait
'E77C DBFF 1168 IN A,(0FFH) ;get data byte
'E77E C9 1169 RET
1170 ;
1171 ;
1172 ;parallel output routine. (E021)
1173 ;
'E77F F5 1174 PAROUT PUSH AF ;save char.
'E780 DBFE 1175 PAROT1 IN A,(0FEH) ;get status
'E782 CB77 1176 BIT 6,A ;accepted?
'E784 28FA 1177 JR Z,PAROT1-$ ;no, wait
'E786 F1 1178 POP AF ;restore character
'E787 D3FF 1179 OUT (0FFH),A ;send char.
'E789 C9 1180 RET
1181 ;
1182 ;
1183 ;
1184 ;
1185 ;
1186 ;
1187 ;
1188 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
1189 ; LOAD command
1190 ;
'E78A CD2FE2' 1191 LOAD CALL SCANHL ;skip command
'E78D 2B 1192 LOAD1 DEC HL ;check for 'LOG'
'E78E 7E 1193 LD A,(HL)
'E78F FE30 1194 CP '0'
'E791 38FA 1195 JR C,LOAD1-$ ;skip delimiters
'E793 FE47 1196 CP 'G' ;is it a 'G'
'E795 F5 1197 PUSH AF ;save test flag
'E796 CD64E2' 1198 CALL NAMFND ;get name
'E799 F5 1199 LODBAS PUSH AF ;save filename flags
1200 ;(entry point E02D)
'E79A 0601 1201 LD B,1 ;defould serial device
'E79C F5 1202 PUSH AF
'E79D 2819 1203 JR Z,LOAD3-$ ;no parameters left, go load
'E79F F1 1204 POP AF ;not need
'E7A0 3807 1205 JR C,LOAD2-$ ;if bad name
'E7A2 CD32E2' 1206 CALL SCANLT ;skip rest of name
'E7A5 F5 1207 PUSH AF
'E7A6 2810 1208 JR Z,LOAD3-$ ;no parameters left, go load
'E7A8 F1 1209 POP AF
'E7A9 CD3DE2' 1210 LOAD2 CALL CONV ;get serial device
'E7AC 43 1211 LD B,E
'E7AD CD25E2' 1212 CALL SCAN ;check if other load address
'E7B0 F5 1213 PUSH AF ;Z-flag set, if not
'E7B1 2805 1214 JR Z,LOAD3-$
'E7B3 C5 1215 PUSH BC ;save serial device
'E7B4 CD3DE2' 1216 CALL CONV ;get load address
'E7B7 C1 1217 POP BC ;restore serial device
'E7B8 CD05E2' 1218 LOAD3 CALL CRLF ;fresh line
'E7BB CD8AE2' 1219 CALL MOTRON ;select serial device
'E7BE D5 1220 LOAD3A PUSH DE ;save load address
'E7BF CD1BE7' 1221 CALL GETHED ;get header
'E7C2 FD7E5C 1222 LD A,(IY+THEAD+5);get exidy-file type
'E7C5 B7 1223 OR A ;test if 00H
'E7C6 280B 1224 JR Z,LOAD3B-$ ;no printing if 00H
'E7C8 E5 1225 PUSH HL ;no funtion
'E7C9 21CAE4' 1226 LD HL,FNDMSG
'E7CC CDBAE1' 1227 CALL MSGOUT ;print 'FOUND - '
'E7CF CDDEE6' 1228 CALL HEDPRT ;print file header
'E7D2 E1 1229 POP HL
'E7D3 D1 1230 LOAD3B POP DE ;redo load address
'E7D4 F1 1231 POP AF ;load at start address?
'E7D5 F5 1232 PUSH AF
'E7D6 2006 1233 JR NZ,LOAD5-$ ;no, load at load-address
'E7D8 FD5E60 1234 LD E,(IY+THEAD+HADDR) ;load at start address
'E7DB FD5661 1235 LD D,(IY+THEAD+HADDR+1)
'E7DE E1 1236 LOAD5 POP HL ;flags in reg. HL
'E7DF F1 1237 POP AF ;get filename flags
'E7E0 F5 1238 PUSH AF ;put back
'E7E1 E5 1239 PUSH HL
'E7E2 2814 1240 JR Z,LOAD7-$ ;if no filename
'E7E4 3812 1241 JR C,LOAD7-$ ;if bad filename
'E7E6 FDE5 1242 PUSH IY
'E7E8 DDE1 1243 POP IX ;copy reg. IY to reg. IX
    
```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E7EA 0605 1244 LD B,5 ;name length
'E7EC DD7E47 1245 LOAD6 LD A,(IX+CHEAD)
'E7EF DDBE57 1246 CP (IX+THEAD) ;same?
'E7F2 DD23 1247 INC IX
'E7F4 2049 1248 JR NZ,LOADSK-$ ;skip file, if wrong name
'E7F6 10F4 1249 DJNZ LOAD6-$ ;loop for 5 char.
'E7F8 FD7E5C 1250 LOAD7 LD A,(IY+THEAD+5);get exidy-file type
'E7FB B7 1251 OR A ;test if 00H
'E7FC 2809 1252 JR Z,LOAD7A-$ ;no nulls after header if
1253 ;exidy-file = 00H
'E7FE 21BFE4' 1254 LD HL,LDGMSG
'E801 CDBAE1' 1255 CALL MSGOUT ;print 'LOADING - '
'E804 CD59E7' 1256 CALL TAPWT
'E807 EB 1257 LOAD7A EX DE,HL ;load address into reg. HL
'E808 FD5E5E 1258 LD E,(IY+THEAD+HSIZE) ;get block size
'E80B FD565F 1259 LD D,(IY+THEAD+HSIZE+1)
'E80E CDA9E6' 1260 LOAD8 CALL BLKADJ ;calc. short block length
'E811 280F 1261 JR Z,LOAD10-$ ;done, if thru
'E813 CDDAE2' 1262 LOAD9 CALL TAPEIN ;get a byte from UART
'E816 CAD4E1' 1263 JP Z,FINISH ;if some one wants us
'E819 77 1264 LD (HL),A ;store in memory
'E81A 23 1265 INC HL ;next address
'E81B 10F6 1266 DJNZ LOAD9-$ ;loop for short block
'E81D CD4EE7' 1267 CALL CKCRC ;check CRC byte
'E820 18EC 1268 JR LOAD8-$ ;loop for all blocks
'E822 CDAFE2' 1269 LOAD10 CALL MTROFF ;turn off motor
'E825 2153E4' 1270 LD HL,FILHD
'E828 CDBAE1' 1271 CALL MSGOUT ;print 'NAME FILE BLCK ADDR GOADDR'
'E82B CDDEE6' 1272 CALL HEDPRT ;print header
'E82E F1 1273 POP AF ;destroy load address flag
'E82F F1 1274 POP AF ;destroy filename flags
'E830 F1 1275 POP AF ;get 'G' flag
'E831 C0 1276 RET NZ ;if not 'LOG'
'E832 FD7E5D 1277 LD A,(IY+THEAD+HTYPE);get file type
'E835 E680 1278 AND 80H ;data file??
'E837 C0 1279 RET NZ
'E838 FD6E62 1280 LD L,(IY+THEAD+HXEQ);get go address
'E83B FD6663 1281 LD H,(IY+THEAD+HXEQ+1)
'E83E E9 1282 JP (HL) ;call go-address
'E83F CDC2E0' 1283 LOADSK CALL LOADD ;call correction routine
'E842 C3BEE7' 1284 JP LOAD3A ;go, find next file
1285 ;
1286 ; PROMPT command
1287 ;
'E845 FDE5 1288 PRMPTC PUSH IY
'E847 E1 1289 POP HL ;copy reg. IY to reg. HL
'E848 7E 1290 PRMP1 LD A,(HL)
'E849 FE0D 1291 CP CR ;end of line
'E84B CADEE1' 1292 JP Z,ERRPAR
'E84E FE3D 1293 CP '='
'E850 23 1294 INC HL
'E851 20F5 1295 JR NZ,PRMP1-$ ;loop until '=' or carriage return
'E853 7E 1296 LD A,(HL)
'E854 FD7744 1297 LD (IY+PROMPT),A
'E857 C9 1298 RET
    
```

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
		1299 ;	BATCH command	
		1300 ;		
'E858	FD7043	1301	BATCH LD (IY+BATCHF),B ;set flag	
'E85B	C9	1302	RET	
		1303 ;		
		1304 ;		
		1305 ;	CREAT command	
		1306 ;		
'E85C	3E2A	1307	CREAT LD A, '*' ;create batch-file	
'E85E	CD45E0'	1308	CALL CHROUT ;print 'creat' prompt	
'E861	CD3AE1'	1309	CALL LINEIN ;get a line from operator	
'E864	FDE5	1310	PUSH IY	
'E866	E1	1311	POP HL ;get start of line	
'E867	7E	1312	LD A,(HL)	
'E868	FE0D	1313	CP CR ;empty line?	
'E86A	C8	1314	RET Z ;ready, if so	
'E86B	0601	1315	LD B,1 ;select recorder #1	
'E86D	CD8AE2'	1316	CALL MOTRON ;turn on	
'E870	CDC2E2'	1317	CALL NULL ;send 100 nulls, A 01H and clear CRC	
'E873	7E	1318	CREAT1 LD A,(HL) ;get char. out of input buffer	
'E874	23	1319	INC HL ;next	
'E875	CDEEE2'	1320	CALL TAPOUT ;send to tape	
'E878	FE0D	1321	CP CR ;end of line?	
'E87A	20F7	1322	JR NZ,CREAT1-\$;loop until EOL	
'E87C	CD9BE8'	1323	CALL WRCRC ;send CRC byte	
'E87F	CDAFE2'	1324	CALL MTROFF	
'E882	18D8	1325	JR CREAT-\$;loop until empty line	
		1326 ;		
		1327 ;		
		1328 ;	LIST command	
		1329 ;		
'>E884		1330	LIST EQU \$;list batch-file	
'E884	0601	1331	LD B,1 ;select recorder #1	
'E886	CD8AE2'	1332	CALL MOTRON ;turn motor on	
'E889	CD05E2'	1333	LIST1 CALL CRLF ;print carriage return & linefeed	
'E88C	CD59E7'	1334	CALL TAPWT ;wait for 10 nulls and 01H,clear CRC	
'E88F	CDDAE2'	1335	LIST3 CALL TAPEIN ;get a byte from tape	
'E892	FE0D	1336	CP CR ;end of line?	
'E894	28F3	1337	JR Z,LIST1-\$;yes, go get new line	
'E896	CD45E0'	1338	CALL CHROUT ;print char.	
'E899	18F4	1339	JR LIST3-\$;loop until <RESET>	
		1340 ;		
		1341 ;		
		1342 ;	write CRC to serial/tape	
		1343 ;		
'E89B	FD7E46	1344	WRCRC LD A,(IY+CRCBYT) ;get CRC byte	
'E89E	C3EEE2'	1345	JP TAPOUT ;send and exit	
		1346 ;		
		1347 ;		
		1348 ;		
		1349 ;		
		1350 ;		
		1351 ;		
		1352 ;		
		1353 ;		

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
		1354 ;	TEST command	
		1355 ;		
'E8A1	3E13	1356	TEST LD A,19 ;init heading flags	
'E8A3	08	1357	EX AF,AF' ;save heading flags	
'E8A4	CD0CEC'	1358	CALL GETPAR ;skip "TE" & get START address	
'E8A7	D5	1359	PUSH DE ;save START	
'E8A8	CD0FEC'	1360	CALL GETPA1 ;get END address	
'E8AB	FE43	1361	CP 'C' ;if continue testing	
'E8AD	E1	1362	POP HL ;restore START	
'E8AE	0E01	1363	LD C,1 ;init pass counter	
'E8B0	F5	1364	TEST0 PUSH AF ;save cont. flag	
'E8B1	C5	1365	PUSH BC ;save pass counter	
'E8B2	E5	1366	PUSH HL ;save START	
'E8B3	0600	1367	LD B,0 ;get mask data	
'E8B5	70	1368	TEST1 LD (HL),B ;clear ram part under test	
'E8B6	CD3BE9'	1369	CALL ENDCK1 ;next address, test if done	
'E8B9	20FA	1370	JR NZ,TEST1-\$	
'E8BB	CD1CE5'	1371	CALL FLASH ;first flash	
'E8BE	E1	1372	POP HL ;restore START	
'E8BF	E5	1373	TEST2 PUSH HL	
'E8C0	7E	1374	TEST3 LD A,(HL)	
'E8C1	A8	1375	XOR B ;test if contents equal reg. B	
'E8C2	4F	1376	LD C,A ;save difference	
'E8C3	78	1377	LD A,B	
'E8C4	2F	1378	CPL	
'E8C5	77	1379	LD (HL),A ;test if complement can be stored	
'E8C6	AE	1380	XOR (HL)	
'E8C7	B1	1381	OR C ;add bad-bits	
'E8C8	282F	1382	JR Z,TEST4-\$;if no error	
'E8CA	D5	1383	PUSH DE ;save END address	
'E8CB	EB	1384	EX DE,HL ;get bad ram-address in reg. DE	
'E8CC	4F	1385	LD C,A ;save bad bits	
'E8CD	08	1386	EX AF,AF' ;get line counter	
'E8CE	3C	1387	INC A ;count lines	
'E8CF	2175E1'	1388	LD HL,TESTHD ;'ADDR BITS 0 1 2'	
'E8D2	FE14	1389	CP 20	
'E8D4	CCBAE1'	1390	CALL Z,MSGOUT ;if reg. A=20, print bits heading	
		1391		
'E8D7	08	1392	EX AF,AF' ;and clear reg. A	
'E8D8	C5	1393	PUSH BC ;save line counter	
'E8D9	CD0FE2'	1394	CALL ADDCOL ;save mask byte	
'E8DC	0604	1395	LD B,4 ;print bad ram-address	
'E8DE	CDD2E2'	1396	CALL SPACES ;four spaces	
'E8E1	0608	1397	LD B,8 ;a byte has 8 bits	
'E8E3	2EA1	1398	TEST32 LD L,BADMSG ;load bad message	
'E8E5	CB19	1399	RR C	
'E8E7	3802	1400	JR C,TEST33-\$;if bad, skip	
'E8E9	2EA6	1401	LD L,OKMSG ;load ok message	
'E8EB	CDBAE1'	1402	TEST33 CALL MSGOUT ;print OK/BAD	
'E8EE	10F3	1403	DJNZ TEST32-\$;loop for 8 bits	
'E8F0	C1	1404	POP BC ;restore mask byte	
'E8F1	EB	1405	EX DE,HL	
'E8F2	D1	1406	POP DE ;restore END address	
'E8F3	CD05E2'	1407	CALL CRLF ;new line	
'E8F6	CD23E5'	1408	CALL QUIKT ;test for operator	

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'E8F9 05 1409 TEST4 DEC B
'E8FA 70 1410 LD (HL),B ;store next mask byte
'E8FB 04 1411 INC B
'E8FC CD3BE9' 1412 CALL ENDCK1 ;next address, test if ready
'E8FF 20BF 1413 JR NZ,TEST3-$ ;no, continue
'E901 CD1CE5' 1414 CALL FLASH
'E904 E1 1415 POP HL ;restore START address
'E905 10B8 1416 DJNZ TEST2-$ ;one PASS is 256 tests
1417 ;
'E907 C1 1418 POP BC ;get PASS counter
'E908 D5 1419 PUSH DE ;save END
'E909 E5 1420 PUSH HL
'E90A 59 1421 LD E,C ;PASS to reg. DE
'E90B 50 1422 LD D,B
'E90C CD0FE2' 1423 CALL ADDCOL ;print the pass
'E90F 21ABE4' 1424 LD HL,PSCMSG ;'PASS COMPLETED'
'E912 CDBAE1' 1425 CALL MSGOUT
'E915 E1 1426 POP HL ;restore START address
'E916 D1 1427 POP DE ;restore END address
'E917 03 1428 INC BC ;increment PASS counter
'E918 F1 1429 POP AF ;get continue flag
'E919 C0 1430 RET NZ ;if no 'C' option
'E91A 1894 1431 JR TEST0-$ ;loop until operator abort testing
1432 ;
1433 ;
1434 ;subroutine print date if any
1435 ;
'E91C 0602 1436 FILES5 LD B,2 ;two times
'E91E DD7E57 1437 FILES6 LD A,(IX+THEAD)
'E921 B7 1438 OR A
'E922 2814 1439 JR Z,CRLF1-$ ;if NULL, no date.
'E924 CDEDE1' 1440 CALL HCHOUT
'E927 3E2D 1441 LD A,'-' ;print seperator
'E929 CD45E0' 1442 CALL CHROUT
'E92C DD23 1443 INC IX
'E92E 10EE 1444 DJNZ FILES6-$
'E930 DD5E57 1445 LD E,(IX+THEAD) ;get year
'E933 1619 1446 LD D,19H ;print 19..
'E935 CDEBE1' 1447 CALL ADDOUT ;print year
'E938 C305E2' 1448 CRLF1 JP CRLF ;print CR/LF and exit
1449 ;
1450 ;
'E93B 23 1451 ENDCK1 INC HL ;count up reg. HL and check
1452 ; if equal reg. DE
1453 ;
1454 ;
1455 ;subroutine compare register DE with HL
1456 ;
'E93C 7A 1457 ENDCK LD A,D
'E93D BC 1458 CP H
'E93E C0 1459 RET NZ
'E93F 7B 1460 LD A,E
'E940 BD 1461 CP L
'E941 C9 1462 RET
1463 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
1464 ; MOVE command
1465 ;
'E942 CD0CEC' 1466 MOVE CALL GETPAR ;get FROM address
'E945 D5 1467 PUSH DE
'E946 CD0FEC' 1468 CALL GETPA1 ;get second address
'E949 D5 1469 PUSH DE
'E94A FE53 1470 CP 'S' ;test if S-option
'E94C 280D 1471 JR Z,MOVES-$ ;yes, do it
'E94E CD3DE2' 1472 CALL CONV ;get TO address
'E951 E1 1473 POP HL ;get END
'E952 C1 1474 POP BC ;get FROM
'E953 B7 1475 OR A ;clear carry flag
'E954 ED42 1476 SBC HL,BC ;length:=FROM-END
'E956 C5 1477 PUSH BC
'E957 E3 1478 EX (SP),HL
'E958 C1 1479 POP BC
'E959 180B 1480 JR MOVE1-$
1481 ;
'E95B CD2CE2' 1482 MOVES CALL SCAN1 ;skip 'S'
'E95E CD0FEC' 1483 CALL GETPA1 ;get length
'E961 4B 1484 LD C,E
'E962 42 1485 LD B,D
'E963 0B 1486 DEC BC
'E964 D1 1487 POP DE ;get TO
'E965 E1 1488 POP HL ;get FROM
'E966 CD3CE9' 1489 MOVE1 CALL ENDCK
'E969 FA82E5' 1490 JP M,FILL5 ;if upwards moving
'E96C 09 1491 ADD HL,BC ;correct FROM address
'E96D EB 1492 EX DE,HL
'E96E 09 1493 ADD HL,BC ;correct TO address
'E96F EB 1494 EX DE,HL
'E970 03 1495 INC BC ;adjust length
'E971 ED08 1496 LDDR ;use RET instruction of MOT1 routine
1497 ;
'E973 F610 1498 MOT1 OR 10H ;set 4e bit (motor 1)
'E975 B7 1499 QUIKW2 OR A ;init Z-flag, used with subr. QUIKW1
'E976 C9 1500 RET
1501 ;
1502 ;
1503 ;this subroutine checks if operator hit ESC or CTRL C
1504 ;if ESC, waits for a second ESC,
1505 ;if CTRL C, returns Z-flag reset
1506 ;else Z-flag set
1507 ;
'E977 CDD1EA' 1508 QUIKW1 CALL QUIK
'E97A C8 1509 RET Z ;go back, nothing happened
'E97B CD30E0' 1510 CALL CHRIN ;get the key
'E97E FE03 1511 QUIKW1 CP 3
'E980 28F1 1512 JR Z,MOT1-$
'E982 C430E0' 1513 CALL NZ,CHRIN
'E985 FE1B 1514 CP 1BH
'E987 20F5 1515 JR NZ,QUIKW1-$
'E989 C9 1516 RET
1517 ;
1518 ;

```



```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
1519 ; PP command
1520 ;
'E98A CD2FE2' 1521 PROMPK CALL SCANHL ;skip command name
'E98D 32F0FB' 1522 LD (0FBF0H),A ;init rompac bus systeem
'E990 C3AE00' 1523 JP INITUC ;go, check if pack present
1524 ;
1525 ;
1526 ;centronics printer driver
1527 ;
'E993 CDF0E9' 1528 CENDRV CALL CHROT1 ;SE 0=E996, output to centronics
'E996 F5 1529 PUSH AF ; printer and not to video
'E997 FE0A 1530 CP LF
'E999 2814 1531 JR Z,CENGBK-$ ;no linefeeds to printer
'E99B F5 1532 PUSH AF
'E99C DBFF 1533 CENBSY IN A,(0FFH) ;get busy bit
'E99E CB7F 1534 BIT 7,A
'E9A0 20FA 1535 JR NZ,CENBSY-$ ;if printer busy, wait
'E9A2 F1 1536 POP AF
'E9A3 F680 1537 OR 80H ;set bit 7 (strobe)
'E9A5 D3FF 1538 OUT (0FFH),A
'E9A7 E67F 1539 AND 7FH ;reset bit 7
'E9A9 D3FF 1540 OUT (0FFH),A
'E9AB F680 1541 OR 80H ;set bit 7
'E9AD D3FF 1542 OUT (0FFH),A
'E9AF F1 1543 CENGBK POP AF
'E9B0 C9 1544 RET
1545 ;
1546 ;
1547 ;video driver routines
1548 ;
1549 ;initialize video memory
1550 ;
'E9B1 CDA3EE' 1551 VIDINT CALL GRAP ;init graphics
'E9B4 2180F0' 1552 CLEAR LD HL,0F080H ;clear screen
'E9B7 3EF8 1553 LD A,0F8H ;0F800H is no screen address
'E9B9 3620 1554 CLR1 LD (HL),SPACE ;fill with space
'E9BB 23 1555 INC HL
'E9BC BC 1556 CP H ;out of screen
'E9BD 20FA 1557 JR NZ,CLR1-$ ;no, continue
'E9BF AF 1558 HOMEUC XOR A
'E9C0 FD7768 1559 LD (IY+LINE),A ;clear line #
'E9C3 FD7769 1560 LD (IY+LINE+1),A
'E9C6 FD776A 1561 LD (IY+CHR),A ;clear character #
'E9C9 FD776B 1562 LD (IY+CHR+1),A
1563 ;
1564 ;
1565 ;write cursor
1566 ;
'E9CC CDD6E9' 1567 WCUR CALL PTRSET
'E9CF 7E 1568 LD A,(HL) ;get char. from screen
'E9D0 FD7767 1569 LD (IY+VDHLD),A ;save in MWA
'E9D3 365F 1570 LD (HL),5FH ;write cursor
'E9D5 C9 1571 RET
1572 ;
1573 ;

```

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort
1574 ;set cursor pos. into reg. HL
1575 ;
'E9D6 2180F0' 1576 PTRSET LD HL,VID ;first video pos.
'E9D9 FD5E68 1577 LD E,(IY+LINE) ;get line #
'E9DC FD5669 1578 LD D,(IY+LINE+1)
'E9DF 19 1579 ADD HL,DE
'E9E0 FD5E6A 1580 LD E,(IY+CHR) ;get character #
'E9E3 FD566B 1581 LD D,(IY+CHR+1)
'E9E6 19 1582 ADD HL,DE
'E9E7 C9 1583 RET
1584 ;
1585 ;
1586 ;the routine REC is not used any more, but is
1587 ;not deleted because other software does use it.
1588 ;
'E9E8 CDD6E9' 1589 REC CALL PTRSET
'E9EB FD7E67 1590 LD A,(IY+VDHLD)
'E9EE 77 1591 LD (HL),A
'E9EF C9 1592 RET
1593 ;
1594 ;
1595 ;entry point video driver (E01B)
1596 ;
'E9F0 FDE5 1597 CHROT1 PUSH IY
'E9F2 CDA2E1' 1598 CALL GETIY
'E9F5 F5 1599 PUSH AF
'E9F6 C5 1600 PUSH BC
'E9F7 D5 1601 PUSH DE
'E9F8 E5 1602 PUSH HL
'E9F9 CDD6E9' 1603 CALL PTRSET ;calc. cursor address
'E9FC FE20 1604 CP SPACE ;is CTRL char.
'E9FE 3833 1605 JR C,CTRL-$ ;yes, do them
'EA00 77 1606 LD (HL),A ;store in video mem.
'EA01 13 1607 INC DE ;count up char. count
'EA02 7B 1608 LD A,E
'EA03 E63F 1609 AND 3FH ;test end of line
'EA05 2807 1610 JR Z,RETUR1-$ ;yes, do not save char. count
'EA07 FD736A 1611 LD (IY+CHR),E
'EA0A FD726B 1612 LD (IY+CHR+1),D ;save char. count
'EA0D 23 1613 INC HL ;next screen loc. for cursor
'EA0E 4E 1614 RETUR1 LD C,(HL) ;write cursor
'EA0F FD7167 1615 LD (IY+VDHLD),C
'EA12 365F 1616 LD (HL),5FH
'EA14 1816 1617 JR RETURN-$ ;restore registers and return
1618 ;
'EA16 13 1619 CURRGT INC DE ;cursor right
'EA17 7B 1620 LD A,E
'EA18 E63F 1621 AND 3FH ;test if out of video
'EA1A 1802 1622 JR CURT-$
1623 ;
'EA1C B3 1624 CURLFT OR E ;cursor left
'EA1D 1B 1625 DEC DE
'EA1E 280C 1626 CURT JR Z,RETURN-$ ;if cursor out of video
'EA20 21 1627 DB 21H ;skip 2 bytes
1628 ;

```

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor	1.3/B (C) 1983 by Gebr. van Montfort
'EA21	57	1629	CARRET LD D,A		;clear char. counter
'EA22	5F	1630	LD E,A		
'EA23	FD736A	1631	LD (IY+CHR),E		;store char. counter
'EA26	FD726B	1632	LD (IY+CHR+1),D		
		1633	;		
'EA29	CDCCE9'	1634	NXLOC CALL WCUR		;write cursor
		1635	;		
'EA2C	E1	1636	RETURN POP HL		;restore all registers
'EA2D	D1	1637	POP DE		
'EA2E	C1	1638	POP BC		
'EA2F	F1	1639	POP AF		
'EA30	FDE1	1640	POP IY		
'EA32	C9	1641	RET		
		1642	;		
'EA33	FD4E67	1643	CTRL LD C,(IY+VDHLD)		;restore char. under cursor
'EA36	71	1644	LD (HL),C		
'EA37	3D	1645	DEC A		
'EA38	28E2	1646	JR Z,CURLFT-\$;if CTRL A, cursor left
'EA3A	D607	1647	SUB 7		
'EA3C	2842	1648	JR Z,BAKSPC-\$;if CTRL H, back space
'EA3E	D60B	1649	SUB 11		
'EA40	28D4	1650	JR Z,CURRGT-\$;if CTRL S, cursor right
'EA42	C606	1651	ADD A,6		
'EA44	28DB	1652	JR Z,CARRET-\$;if CTRL M or RETURN-key
'EA46	212CEA'	1653	LD HL,RETURN		
'EA49	E5	1654	PUSH HL		;save return address
'EA4A	3C	1655	INC A		
'EA4B	CAB4E9'	1656	JP Z,CLEAR		;if CTRL L or CLEAR-key
'EA4E	D605	1657	SUB 5		
'EA50	CABFE9'	1658	JP Z,HOMECU		;if CTRL Q, home cursor
'EA53	D60B	1659	SUB 11		
'EA55	CAA3EE'	1660	JP Z,GRAP		;if CTRL \, init standard graphics
'EA58	FD5E68	1661	LD E,(IY+LINE)		
'EA5B	FD5669	1662	LD D,(IY+LINE+1)		;get in reg. DE line #
'EA5E	C605	1663	ADD A,5		
'EA60	2840	1664	JR Z,CURUP-\$;if CTRL W, cursor up
'EA62	E1	1665	POP HL		;destroy return address on stack
'EA63	C60D	1666	ADD A,13		
'EA65	2804	1667	JR Z,LINFED-\$;if CTRL Z, cursor down
'EA67	D610	1668	SUB 16		
'EA69	20BE	1669	JR NZ,NXLOC-\$;if no CTRL J or LINE-FEED-key
		1670	;		
'EA6B	6B	1671	LINFED LD L,E		
'EA6C	62	1672	LD H,D		
'EA6D	29	1673	ADD HL,HL		
'EA6E	29	1674	ADD HL,HL		;multiply by 4
'EA6F	3E1D	1675	LD A,29		;in reg. H line #
'EA71	BC	1676	CP H		
'EA72	281A	1677	JR Z,LLN-\$;if scroll is need
'EA74	214000	1678	LD HL,64		;else, add only one line
'EA77	19	1679	LINF1 ADD HL,DE		
'EA78	FD7568	1680	LD (IY+LINE),L		
'EA7B	FD7469	1681	LD (IY+LINE+1),H		;store line #
'EA7E	18A9	1682	JR NXLOC-\$;go show cursor
		1683	;		

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor	1.3/B (C) 1983 by Gebr. van Montfort
'EA80	3C	1684	BAKSPC INC A		;send CTRL A
'EA81	CD45E0'	1685	CALL CHROUT		
'EA84	CD17E2'	1686	CALL SPCOUT		;print SPACE
'EA87	3E01	1687	LD A,CTRLA		
'EA89	CD45E0'	1688	CALL CHROUT		;send CTRL A
'EA8C	189E	1689	JR RETURN-\$;and go back
		1690	;		
'EA8E	1180F0	1691	LLN LD DE,VID		;top of video
'EA91	21C0E0	1692	LD HL,VID+64		
'EA94	014007	1693	LD BC,740H		
'EA97	EDE0	1694	LDIR		
'EA99	3E20	1695	LD A,SPACE		;clear lowest line
'EA9B	12	1696	LLN1 LD (DE),A		
'EA9C	1C	1697	INC E		
'EA9D	C29BEA'	1698	JP NZ,LLN1		;could be rel.
'EAA0	1887	1699	JR NXLOC-\$		
		1700	;		
'EAA2	7A	1701	CURUP LD A,D		;if on the upper line
'EAA3	B3	1702	OR E		
'EAA4	C8	1703	RET Z		
'EAA5	F1	1704	POP AF		;destroy return address
'EAA6	21C0FF	1705	LD HL,-64		;subtract one line
'EAA9	18CC	1706	JR LINF1-\$		
		1707	;		
		1708	;		
		1709	;		'SE T=' command
		1710	;		
'EAAB	7B	1711	TAPE0 LD A,E		
'EAAC	E503	1712	AND 3		
'EAAE	C85B	1713	BIT 3,E		
'EAB0	16F0	1714	LD D,0F0H		;mask bits
'EAB2	2009	1715	JR NZ,TAPE2-\$;for SE T=8 to SE T=F
'EAB4	BB	1716	CP E		
'EAB5	2812	1717	JR Z,TAPE3-\$;for SE T=0,1,2 OR 3
'EAB7	16CF	1718	LD D,0CFH		;mask bits
'EAB9	07	1719	RLCA		;for SE T=4,5,6 or 7
'EABA	07	1720	RLCA		
'EABB	07	1721	RLCA		
'EABC	07	1722	RLCA		
'EABD	5F	1723	TAPE2 LD E,A		
'EABE	FD7E3D	1724	LD A,(IY+TAPES)		;get old value
'EAC1	A2	1725	AND D		;mask
'EAC2	B3	1726	OR E		
'EAC3	FD773D	1727	LD (IY+TAPES),A		;store new value
'EAC6	C3B4E2'	1728	JP MTROF1		;send to port 0FEH
		1729	;		
'EAC9	163F	1730	TAPE3 LD D,3FH		;mask bits
'EACB	EE01	1731	XOR 1		
'EACD	0F	1732	RRCA		
'EACE	0F	1733	RRCA		
'EACF	18EC	1734	JR TAPE2-\$		
		1735	;		
		1736	;		
		1737	;		
		1738	;		

Monitor 1.3/B (C) 1983 by Gebr. van Montfort

```

1739 ; keyboard quick check routine. (E015)
1740 ;
'EAD1 FDE5 1741 QUIK PUSH IY
'EAD3 CDA2E1' 1742 CALL GETIY
'EAD6 FD7E45 1743 QUIKD LD A,(IY+CMTRFG)
'EAD9 E6F0 1744 AND 0F0H
'EADB F5 1745 PUSH AF
'EADC D3FE 1746 OUT (0FEH),A
'EADE DBFE 1747 IN A,(0FEH)
'EAE0 1F 1748 RRA
'EAE1 3022 1749 JR NC,QUIK2-$ ;if RUN/STOP
'EAE3 E602 1750 AND 2
'EAE5 C2F7EA' 1751 JP NZ,QUIK0 ;if not CTRL
'EAE8 F1 1752 POP AF
'EAE9 F5 1753 PUSH AF
'EAEA F603 1754 OR 3
'EAEC D3FE 1755 OUT (0FEH),A
'EAAE DBFE 1756 IN A,(0FEH)
'EAF0 1F 1757 RRA
'EAF1 3017 1758 JR NC,QUIK4-$ ;if 'C'
'EAF3 E602 1759 AND 2
'EAF5 280E 1760 JR Z,QUIK2-$ ;if 'S'
'EAF7 F1 1761 QUIK0 POP AF
'EAF8 3C 1762 INC A
'EAF9 D3FE 1763 OUT (0FEH),A
'EAFB DBFE 1764 IN A,(0FEH)
'EAFD E610 1765 AND 10H
'EAFF 2805 1766 JR Z,QUIK3-$ ;if ESC
'EB01 AF 1767 XOR A
'EB02 FDE1 1768 QUIK1 POP IY
'EB04 C9 1769 RET
1770 ;
'EB05 F1 1771 QUIK2 POP AF
'EB06 3E1B 1772 QUIK3 LD A,ESC
'EB08 1803 1773 JR QUIK5-$
1774 ;
'EB0A F1 1775 QUIK4 POP AF
'EB0B 3E03 1776 LD A,CTRLC
'EB0D B7 1777 QUIK5 OR A
'EB0E 18F2 1778 JR QUIK1-$
1779 ;
1780 ;
1781 ;set standard graphics
1782 ;
'EB10 C3A3EE' 1783 WCSET JP GRAP ;go init standard graphics
'EB13 1100FC 1784 LD DE,0FC00H ;with the shorter routine
'EB16 010002 1785 LD BC,512 ;not deleted for compatibility
'EB19 EDB0 1786 LDIR
'EB1B C9 1787 RET
1788 ;
1789 ;
1790 ;
1791 ;
1792 ;
1793 ;

```

Monitor 1.3/B (C) 1983 by Gebr. van Montfort

```

1794 ; keyboard driver. (E018)
1795 ;
'EB1C FDE5 1796 KEYBD PUSH IY
'EB1E CDA2E1' 1797 CALL GETIY
'EB21 C5 1798 PUSH BC
'EB22 D5 1799 PUSH DE
'EB23 E5 1800 PUSH HL
'EB24 FD7E45 1801 LD A,(IY+CMTRFG) ;get RS232 status, motor flags, ect.
'EB27 E6F0 1802 AND 0F0H
'EB29 5F 1803 LD E,A ;row #
'EB2A 211EEC' 1804 LD HL,INSTBL ;point to start of keyboard table
'EB2D 01FE01 1805 LD BC,1FEH ;reg. B bit #, reg. C port address
'EB30 ED59 1806 OUT (C),E
'EB32 ED78 1807 IN A,(C) ;get row #0
'EB34 E61E 1808 AND 1EH ;mask SHIFT, CTRL, SHIFT-LOCK
; & GRAPHIC keys
; save in reg. D
'EB36 57 1810 LD D,A ;send row #
'EB37 ED59 1811 LOOP1 OUT (C),E ;get key's
'EB39 ED78 1812 IN A,(C) ;mask with bit #
'EB3B A0 1813 AND B ;if zero, a key is hit
'EB3C 281F 1814 JR Z,AKEY-$ ;else, inc. table pointer
'EB3E 23 1815 LOOP2 INC HL ;next bit
'EB3F CB00 1816 RLC B ;5e bit ?
'EB41 CB68 1817 BIT 5,B ;no, continue
'EB43 28F2 1818 JR Z,LOOP1-$ ;load again bit 0
'EB45 0601 1819 LD B,1 ;next row #
'EB47 1C 1820 INC E
'EB48 7B 1821 LD A,E
'EB49 E60F 1822 AND 0FH ;row # = 16 ??
'EB4B 20EA 1823 JR NZ,LOOP1-$ ;no, continue
'EB4D CB42 1824 BIT 0,D ;if no repeat key is pressed
'EB4F 287E 1825 JR Z,NOKEY-$ ;nothing at all
'EB51 0606 1826 LD B,6
'EB53 0B 1827 LOOP3 DEC BC ;delay for repeat
'EB54 78 1828 LD A,B
'EB55 B1 1829 OR C
'EB56 20FB 1830 JR NZ,LOOP3-$
'EB58 FD7E6C 1831 LD A,(IY+LSTKEY) ;get last key pressed
'EB5B 1863 1832 JR HAVIT2-$
1833 ;
'EB5D D5 1834 AKEY PUSH DE
'EB5E 1690 1835 LD D,90H ;debouns count
'EB60 ED78 1836 DBOUN IN A,(C)
'EB62 A0 1837 AND B ;key still pressed
'EB63 2069 1838 JR NZ,NOKEY1-$ ;no, keep it
'EB65 15 1839 DEC D ;count down
'EB66 20F8 1840 JR NZ,DBOUN-$
'EB68 D1 1841 POP DE
'EB69 7E 1842 LD A,(HL) ;out of keyboard table
'EB6A E61F 1843 AND 1FH ;mask
'EB6C 20D0 1844 JR NZ,LOOP2-$ ;if coded
'EB6E 7D 1845 LD A,L
'EB6F 14 1846 INC D ;set repeat flag
'EB70 FE24 1847 CP 24H
'EB72 28CA 1848 JR Z,LOOP2-$ ;if repeat

```

```

ADDR OBJECT      ST # SOURCE STATEMENT      Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'EB74 15          1849      DEC D                ;reset repeat flag
'EB75 FE5A        1850      CP 05AH             ;test edit key
'EB77 FD5E45      1851      LD E,(IY+CMTRFG)    ;default, no edit key
'EB7A CB9B        1852      RES 3,E            ;if not the '+' key
'EB7C 2006        1853      JR NZ,WAITK-$      ;edit-key active?
'EB7E CB4B        1854      BIT 1,E            ;skip, if not
'EB80 2802        1855      JR Z,WAITK-$       ;set edit key
'EB82 CBDB        1856      SET 3,E            ;if not num. keypad
'EB84 3806        1857      WAITK JR C,WAITK1-$  ;num. keypad active?
'EB86 CB43        1858      BIT 0,E            ;if not
'EB88 2802        1859      JR Z,WAITK1-$     ;set keypad flag
'EB8A CBF2        1860      SET 6,D            ;wait until release the key
'EB8C ED78        1861      WAITK1 IN A,(C)
'EB8E A0           1862      AND B
'EB8F 28FB        1863      JR Z,WAITK1-$
'EB91 015000      1864      LD BC,50H         ;offset between tables
'EB94 CB76        1865      BIT 6,(HL)
'EB96 09          1866      ADD HL,BC
'EB97 280F        1867      JR Z,NOGRP-$      ;if no graphic key
'EB99 CB4A        1868      BIT 1,D            ;GRAPHIC ??
'EB9B 200B        1869      JR NZ,NOGRP-$
'EB9D 7E          1870      LD A,(HL)
'EB9E F680        1871      OR 80H             ;set graphic bit
'EBA0 CB62        1872      BIT 4,D            ;SHIFT ??
'EBA2 2019        1873      JR NZ,HAVIT1-$
'EBA4 F640        1874      OR 40H             ;set shift-graphic
'EBA6 1815        1875      JR HAVIT1-$
                1876 ;
'EBA8 CB72        1877      NOGRP BIT 6,D      ;KEY-PAD ??
'EBA A 200E        1878      JR NZ,TABLE2-$
'EBA C CB52        1879      BIT 2,D            ;CTRL ??
'EBA E 280B        1880      JR Z,TABLE3-$
'EBB 0 CB62        1881      BIT 4,D            ;SHIFT ??
'EBB 2 2806        1882      JR Z,TABLE2-$
'EBB 4 CB5A        1883      BIT 3,D            ;SHIFT LOCK ??
'EBB 6 2801        1884      JR Z,TABLE1-$
'EBB 8 09          1885      ADD HL,BC          ;calc. the correct table
'EBB 9 09          1886      TABLE1 ADD HL,BC
'EBB A 09          1887      TABLE2 ADD HL,BC
'EBB B 09          1888      TABLE3 ADD HL,BC
'EBB C 7E          1889      HAVIT LD A,(HL)   ;get ASCII code from table
'EBB D FD7345      1890      HAVIT1 LD (IY+CMTRFG),E ;resave
'EBC 0 FDCB455E    1891      HAVIT2 BIT 3,(IY+CMTRFG) ;edit-key?
'EBC 4 2803        1892      JR Z,HAVIT3-$     ;no, skip
'EBC 6 FD7E67      1893      LD A,(IY+VDHLD)   ;else get char. under cursor
'EBC 9 0C          1894      HAVIT3 INC C       ;reset Z-flag
'EBC A FD776C      1895      LD (IY+LSTKEY),A
'EBC D 21          1896      DB 21H            ;skip two bytes
'EBC E D1          1897      NOKEY1 POP DE
'EBC F AF          1898      NOKEY XOR A         ;set Z-flag, clear reg. A
'EBD 0 E1          1899      POP HL
'EBD 1 D1          1900      POP DE
'EBD 2 C1          1901      POP BC
'EBD 3 FDE1        1902      POP IY
'EBD 5 C9          1903      RET

```

```

ADDR OBJECT      ST # SOURCE STATEMENT      Monitor 1.3/B (C) 1983 by Gebr. van Montfort
                1904 ;
                1905 ;
                1906 ;print char. in reg. A, if CTRL, then print a space
                1907 ;
'EBD 6 FE20        1908      NOCTRL CP SPACE
'EBD 8 D245E0      1909      JP NC,CHROUT      ;print if no CTRL char.
'EBD B C317E2      1910      JP SPCOUT         ;print space
                1911 ;
                1912 ;
                1913 ; 'SE X=' command
                1914 ;
'EBD E 60          1915      DATSET LD H,B      ;get back pointer to input buffer
'EBD F 69          1916      LD L,C
'EBE 0 FDE5        1917      PUSH IY
'EBE 2 DDE1        1918      POP IX            ;IY > IX
'EBE 4 D5          1919      PUSH DE          ;save go address
'EBE 5 0E03        1920      LD C,3            ;three parameters
'EBE 7 CD25E2      1921      DAT1 CALL SCAN
'EBE A 280B        1922      JR Z,DAT2-$
'EBE C CD3DE2      1923      CALL CONV
'EBE F DD7354      1924      LD (IX+54H),E
'EBF 2 DD23        1925      INC IX
'EBF 4 0D          1926      DEC C
'EBF 5 20F0        1927      JR NZ,DAT1-$
'EBF 7 D1          1928      DAT2 POP DE
'EBF 8 C3F2E5      1929      JP XEQSET        ;set GO address
                1930 ;
                1931 ;
                1932 ;subroutines called from MOTRON
                1933 ;
'EBF B 05          1934      MOT DEC B
'EBF C 05          1935      DEC B
'EBF D 2805        1936      JR Z,MOT3-$      ;motor #3
'EBF F 1008        1937      DJNZ MOT2-$     ;motor #2
'EC 0 1 F6C0        1938      OR 0C0H          ;motor #4
'EC 0 3 C9          1939      RET
                1940 ;
'EC 0 4 E53F        1941      MOT3 AND 3FH
'EC 0 6 F680        1942      OR 80H
'EC 0 8 C9          1943      RET
                1944 ;
'EC 0 9 F620        1945      MOT2 OR 20H
'EC 0 B C9          1946      RET
                1947 ;
                1948 ;
                1949 ;subroutine get parameter behind command string
                1950 ;
'EC 0 C CD2FE2      1951      GETPAR CALL SCANHL ;skip command
'EC 0 F CADEE1      1952      GETPA1 JP Z,ERRPAR  ;if no parameter
'EC 1 2 CD3DE2      1953      CALL CONV        ;get parameter
'EC 1 5 C325E2      1954      JP SCAN          ;scan for next one
                1955 ;
                1956 ;
                1957 ;
                1958 ;

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort

```

1959 ;subroutine for accept lowercase commands
1960 ;
'EC18 E65F 1961 MAIND AND 5FH ;make uppercase
'EC1A DDBE00 1962 CP (IX)
'EC1D C9 1963 RET
1964 ;
1965 ;
1966 ;keyboard codes tables
1967 ;
1968 ; bit: 0 1 2 3 4
1969 ;
'EC1E 00908882 1970 INSTBL DB 0 ,90H,88H,82H,84H;0 instruction codes
84
'EC23 00800040 1971 DB 0 ,80H, 0 ,40H, 0 ;1
00
'EC28 40404040 1972 DB 40H,40H,40H,40H,40H;2
40
'EC2D 40404040 1973 DB 40H,40H,40H,40H,40H;3
40
'EC32 40404040 1974 DB 40H,40H,40H,40H,40H;4
40
'EC37 40404040 1975 DB 40H,40H,40H,40H,40H;5
40
'EC3C 40404040 1976 DB 40H,40H,40H,40H,40H;6
40
'EC41 40404040 1977 DB 40H,40H,40H,40H,40H;7
40
'EC46 40404040 1978 DB 40H,40H,40H,40H,40H;8
40
'EC4B 40404040 1979 DB 40H,40H,40H,40H,40H;9
40
'EC50 40404040 1980 DB 40H,40H,40H,40H,40H;A
40
'EC55 40000040 1981 DB 40H, 0 , 0 ,40H,40H;B
40
'EC5A 40404040 1982 DB 40H,40H,40H,40H, 0 ;C
00
'EC5F 40404040 1983 DB 40H,40H,40H,40H,40H;D
40
'EC64 40400040 1984 DB 40H,40H, 0 ,40H,40H;E
40
'EC69 00000040 1985 DB 0 , 0 , 0 ,40H,40H;F
40
1986 ;
'EC6E 00000000 1987 DB 0 , 0 , 0 , 0 , 0 ;0 graphic codes
00
'EC73 0C00000D 1988 DB FF, 0 , 0 ,0DH, 0 ;1
00
'EC78 28271A0E 1989 DB 28H,27H,1AH,0EH, 0 ;2
00
'EC7D 291C1B0F 1990 DB 29H,1CH,1BH,0FH, 1 ;3
01
'EC82 1D111003 1991 DB 1DH,11H,10H, 3 , 2 ;4
02
'EC87 2B2A1E12 1992 DB 2BH,2AH,1EH,12H, 4 ;5

```

ADDR OBJECT ST # SOURCE STATEMENT Monitor 1.3/B (C) 1983 by Gebr. van Montfort

```

04
'EC8C 2D2C1F13 1993 DB 2DH,2CH,1FH,13H, 5 ;6
05
'EC91 21152014 1994 DB 21H,15H,20H,14H, 6 ;7
06
'EC96 2E221608 1995 DB 2EH,22H,16H, 8 , 7 ;8
07
'EC9B 302F2317 1996 DB 30H,2FH,23H,17H, 9 ;9
09
'ECA0 25241918 1997 DB 25H,24H,19H,18H,0AH;A
0A
'ECA5 2600000C 1998 DB 26H, 0 , 0 ,0CH,0BH;B
0B
'ECAA 3C383531 1999 DB 3CH,38H,35H,31H, 0 ;C
00
'ECAF 3D393633 2000 DB 3DH,39H,36H,33H,32H;D
32
'ECB4 3E3A0037 2001 DB 3EH,3AH, 0 ,37H,34H;E
34
'ECB9 0000003F 2002 DB 0 , 0 , 0 ,3FH,3BH;F
3B
2003 ;
'ECBE 0F000000 2004 DB 0FH, 0 , 0 , 0 , 0 ;0 control keys
00
'ECC3 0C00200B 2005 DB FF, 0 , ' ',0BH,ESC;1
1B
'ECC8 181A0111 2006 DB 18H,1AH,01H,11H,'1';2
31
'ECCD 03041317 2007 DB 3H, 4H,13H,17H,'2';3
32
'ECD2 06120534 2008 DB 6H,12H, 5H,'4','3';4
33
'ECD7 02160714 2009 DB 2H,16H, 7H,14H,'5';5
35
'ECDC 0D0E0819 2010 DB 0DH,0EH, 8H,19H,'6';6
36
'ECE1 0B090A15 2011 DB 0BH, 9H,0AH,15H,'7';7
37
'ECE6 2C0C0F39 2012 DB ',','0CH,0FH,'9','8';8
38
'ECEB 2F2E3B10 2013 DB '/','.',',','10H,'0';9
30
'ECF0 1C001D1B 2014 DB 1CH, 0H,1DH,1BH,':';A
3A
'ECF5 1F0D0A1E 2015 DB 1FH, CR, LF,1EH,'-';B
2D
'ECFA 2B2A2F2D 2016 DB '+','*','/','-',',';C
20
'ECFF 30310117 2017 DB '0','1', 1H,17H,'7';D
37
'ED04 2E1A1113 2018 DB '.',1AH,11H,13H,'9';E
39
'ED09 0000003D 2019 DB 0 , 0 , 0 , '=','3';F
33
2020 ;

```

Monitor 1.3/B (C) 1983 by Gebr. van Montfort

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'ED0E	1B000000 00	2021	DB ESC, 0, 0, 0, 0;0	shift table
'ED13	0C002009 1B	2022	DB FF, 0, ' ', 9H,ESC;1	
'ED18	585A4151 21	2023	DB 'X','Z','A','Q','!';2	
'ED1D	43445357 22	2024	DB 'C','D','S','W','!';3	
'ED22	46524524 23	2025	DB 'F','R','E','\$','#';4	
'ED27	42564754 25	2026	DB 'B','V','G','T','%';5	
'ED2C	4D4E4859 26	2027	DB 'M','N','H','Y','&';6	
'ED31	4B494A55 27	2028	DB 'K','I','J','U',27H;7	
'ED36	3C4C4F29 28	2029	DB '<','L','O','')','(';8	
'ED3B	3F3E2B50 30	2030	DB '?','>','+','P','@';9	
'ED40	7C607D7B 2A	2031	DB ' ',60H,7DH,7BH,'*';A	
'ED45	7F0D0A7E 3D	2032	DB 7FH, CR, LF,7EH,'=';B	
'ED4A	2B2A2F2D 20	2033	DB '+','*','/','-',' ' ;C	
'ED4F	30310117 37	2034	DB '@','1', 1H,17H,'7';D	
'ED54	2E1A1113 39	2035	DB '.','1AH,11H,13H,'9';E	
'ED59	0000003D 33	2036	DB 0, 0, 0, '=','3';F	
'ED5E	1B000000 00	2037 ; 2038	DB ESC, 0, 0, 0, 0;0	shift lock table
'ED63	0C00200B 1B	2039	DB FF, 0, ' ',0BH,ESC;1	
'ED68	585A4151 31	2040	DB 'X','Z','A','Q','!';2	
'ED6D	43445357 32	2041	DB 'C','D','S','W','!';3	
'ED72	46524534 33	2042	DB 'F','R','E','4','3';4	
'ED77	42564754 35	2043	DB 'B','V','G','T','5';5	
'ED7C	4D4E4859 36	2044	DB 'M','N','H','Y','6';6	
'ED81	4B494A55 37	2045	DB 'K','I','J','U','7';7	
'ED86	2C4C4F39 38	2046	DB ',','L','O','9','8';8	
'ED8B	2F2E3B50 30	2047	DB '/','.'',';','P','@';9	
'ED90	5C405D5B 3A	2048	DB '\\','@',']','[';:';A	

Monitor 1.3/B (C) 1983 by Gebr. van Montfort

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'ED95	5F0D0A5E 2D	2049	DB 5FH, CR, LF,5EH,'-';B	
'ED9A	2B2A2F2D 20	2050	DB '+','*','/','-',' ' ;C	
'ED9F	30313438 37	2051	DB '@','1','4','8','7';D	
'EDA4	2E323536 39	2052	DB '.','2','5','6','9';E	
'EDA9	0000003D 33	2053	DB 0, 0, 0, '=','3';F	
'EDAE	1B000000 00	2054 ; 2055	DB ESC, 0, 0, 0, 0;0	unshift lock table
'EDB3	0C00200B 1B	2056	DB FF, 0, ' ',0BH,ESC;1	
'EDB8	787A6171 31	2057	DB 'x','z','a','q','!';2	
'EDBD	63647377 32	2058	DB 'c','d','s','w','!';3	
'EDC2	66726534 33	2059	DB 'f','r','e','4','3';4	
'EDC7	62766774 35	2060	DB 'b','v','g','t','5';5	
'EDCC	6D6E6879 36	2061	DB 'm','n','h','y','6';6	
'EDD1	6B696A75 37	2062	DB 'k','i','j','u','7';7	
'EDD6	2C6C6F39 38	2063	DB ',','1','o','9','8';8	
'EDDB	2F2E3B70 30	2064	DB '/','.'',';','p','@';9	
'EDE0	5C405D5B 3A	2065	DB '\\','@',']','[';:';A	
'EDE5	5F0D0A5E 2D	2066	DB 5FH, CR, LF,5EH,'-';B	
'EDEA	2B2A2F2D 20	2067	DB '+','*','/','-',' ' ;C	
'EDEF	30313438 37	2068	DB '@','1','4','8','7';D	
'EDF4	2E323536 39	2069	DB '.','2','5','6','9';E	
'EDF9	0000003D 33	2070	DB 0, 0, 0, '=','3';F	
		2071 ;		
		2072 ;		
		2073 ;		
		2074 ;		
		2075 ;		
		2076 ;		
		2077 ;		
		2078 ;		
		2079 ;		
		2080 ;		
		2081 ;		
		2082 ;		

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'EEBE	FF80	2185 ;	2186 SHAPE	DB 1111111B,80H
'EEC0	FF40	2187		DB 1111111B,40H
'EEC2	FF20	2188		DB 1111111B,20H
'EEC4	FF10	2189		DB 1111111B,10H
'EEC6	81003C7E FFFF7E3C	2190		DB 10000001B,0,3CH,7EH,0FFH,0FFH,7EH,3CH
'EECE	FF04	2191		DB 1111111B,4
'EED0	FF02	2192		DB 1111111B,2
'EED2	FF01	2193		DB 1111111B,1
'EED4	81003C42 8181423C	2194		DB 10000001B,0,3CH,42H,81H,81H,42H,3CH
'EEDC	7F00FF	2195		DB 0111111B,0,0FFH
'EEDF	BF00FF	2196		DB 1011111B,0,0FFH
'EEE2	DF00FF	2197		DB 1101111B,0,0FFH
'EEE5	EF00FF	2198		DB 1110111B,0,0FFH
'EEE8	E0071BE 242424	2199		DB 11100000B,0,71H,0BEH,24H,24H,24H
'EEEF	81814224 18182442	2200		DB 10000001B,81H,42H,24H,18H,18H,24H,42H
'EEF7	F000106 0808	2201		DB 11110000B,0,1,6,8,8
'EEFD	F00C030 0808	2202		DB 11110000B,0,0C0H,30H,8,8
'EF03	7F80FF	2203		DB 0111111B,80H,0FFH
'EF06	7F01FF	2204		DB 0111111B,1,0FFH
'EF09	80FFFEFC F8F0E0C0 80	2205		DB 10000000B,0FFH,0FEH,0FCH,0F8H,0F0H,0E0H,0C0H,80H
'EF12	80FF7F3F 1F0F0703 01	2206		DB 10000000B,0FFH,7FH,3FH,1FH,0FH,7,3,1
'EF1B	F000F0F 0F0F	2207		DB 11110000B,0,0FH,0FH,0FH,0FH
'EF21	F00F0F0 F0F0	2208		DB 11110000B,0,0F0H,0F0H,0F0H,0F0H
'EF27	F700FF	2209		DB 1111011B,0,0FFH
'EF2A	8210387C FEFE7C38	2210		DB 1000010B,10H,38H,7CH,0FEH,0FEH,7CH,38H
'EF32	810066FF FF7E3C18	2211		DB 10000001B,0,66H,0FFH,0FFH,7EH,3CH,18H
'EF3A	E0080601 000000	2212		DB 11100000B,8,6,1,0,0,0
'EF41	E00830C0 000000	2213		DB 11100000B,8,30H,0C0H,0,0,0
'EF48	FE30FF	2214		DB 11111110B,80H,0FFH
'EF4B	FE01FF	2215		DB 11111110B,1,0FFH
'EF4E	8080C0E0 F0F8FCFE FF	2216		DB 10000000B,80H,0C0H,0E0H,0F0H,0F8H,0FCH,0FEH,0FFH
'EF57	80010307 0F1F3F7F FF	2217		DB 10000000B,1,3,7,0FH,1FH,3FH,7FH,0FFH
'EF60	F00F0000 0000	2218		DB 11110000B,0FH,0,0,0,0,0

ADDR	OBJECT	ST #	SOURCE STATEMENT	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
'EF66	F0F00000 0000	2219		DB 11110000B,0F0H,0,0,0,0
'EF6C	FF08	2220		DB 1111111B,8
'EF6E	82183C7E FF7E3C00	2221		DB 10000010B,18H,3CH,7EH,0FFH,7EH,3CH,0
'EF76	C11C6B7F 6B0808	2222		DB 11000001B,1CH,6BH,7FH,6BH,8,8
'EF7D	F0F00F0F 0F0F	2223		DB 11110000B,0F0H,0FH,0FH,0FH,0FH
'EF83	F00FF0F0 F0F0	2224		DB 11110000B,0FH,0F0H,0F0H,0F0H,0F0H
'EF89	FFF0	2225		DB 1111111B,0F0H
'EF8B	FFF0	2226		DB 1111111B,0FH
'EF8D	F0FF0000 0000	2227		DB 11110000B,0FFH,0,0,0,0
'EF93	F000FFFF FFFF	2228		DB 11110000B,0,0FFH,0FFH,0FFH,0FFH
'EF99	80010204 08102040 80	2229		DB 10000000B,1,2,4,8,10H,20H,40H,80H
'EFA2	80804020 10080402 01	2230		DB 10000000B,80H,40H,20H,10H,8,4,2,1
'EFAB	F708FF	2231		DB 1111011B,8,0FFH
'EFAE	FB00FF	2232		DB 1111011B,0,0FFH
'EFB1	FD00FF	2233		DB 1111101B,0,0FFH
'EFB4	FD00FF	2234		DB 1111110B,0,0FFH
'EFB7	AA55AAAA AAAA	2235		DB 10101010B,55H,0AAH,0AAH,0AAH,0AAH
'EFBD	F008FF00 0000	2236		DB 11110000B,8,0FFH,0,0,0
'EFC3	3F00FFFF	2237		DB 0011111B,0,0FFH,0FFH
'EFC7	F7080F	2238		DB 1111011B,8,0FH
'EFCA	AA50AA00 A0A0	2239		DB 10101010B,50H,0A0H,0A0H,0A0H,0A0H
'EFD0	FFC0	2240		DB 1111111B,0C0H
'EFD2	FF03	2241		DB 1111111B,3
'EFD4	F00055AA 55AA	2242		DB 11110000B,0,55H,0AAH,55H,0AAH
'EFDA	F708F8	2243		DB 1111011B,8,0F8H
'EFDD	FC00FFFF	2244		DB 1111100B,0,0FFH,0FFH
'EFE1	F000FF08 0808	2245		DB 11110000B,0,0FFH,8,8,8
'EFE7	F0000F08 0808	2246		DB 11110000B,0,0FH,8,8,8
'EFED	F0000F08 0808	2247		DB 11110000B,0,0F8H,8,8,8
'EFF3	F0080F00 0000	2248		DB 11110000B,8,0FH,0,0,0
'EFF9	F008F800 0000	2249		DB 11110000B,8,0F8H,0,0,0
'EFFF	00	2250 ; 2251 ; 2252 ; 2253 ;		DB 0

SYMBOL	VALUE	STMT	STATEMENT REFS	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
ADDCOL	'E20F	0427	1423 1394 0788 0745	
ADDOUT	'E1E8	0391	1447 1102 0427 0200 0194	
AKEY	'EB5D	1834	1814	
BADMSG	'E4A1	0717	1398	
BAKSPC	'EA80	1684	1648	
BATCH	'E858	1301	0662	
BATCHF	0043	0046	1301 0372 0277	
BKSPC	'E178	0306	0292	
BLKAD2	'E6B6	1056	1050	
BLKADJ	'E6A9	1046	1260 1127 1032	
BOOT1	'EE0D	2097	2117	
BOOT2	'EE45	2129	2131	
BOOT3	'EE62	2146	2135	
BOOTER	'EDFE	2085	0288	
BOOTR	'E531	0796	2111 2091	
CARRET	'EA21	1629	1652	
CENBSY	'E99C	1533	1535	
CENDRV	'E993	1528	0958	
CENGBK	'E9AF	1543	1531	
CHEAD	0047	0050	1245 1024 1012 1011 1010 0995 0994 0940 0939 0933 0513	
CHR	006A	0054	1632 1631 1612 1611 1581 1580 1562 1561	
CHRIN	'E030	0083	1513 1510 0280 0069	
CHRINR	'E041	0091	0106 0086	
CHROT1	'E9F0	1597	1528 0948 0075	
CHROUT	'E045	0095	1909 1688 1685 1442 1338 1308 1104 0589 0438 0431 0429 0420 0418 0358 0303 0226 0070	
CKCRC	'E74E	1136	1267 1132 1120 0324	
CLEAR	'E9B4	1552	1656	
CLR1	'E9B9	1554	1557	
CMTRFG	0045	0048	1891 1890 1851 1801 1743 0569 0548	
CONV	'E23D	0474	1953 1923 1472 1216 1210 1066 1017 0992 0906 0877 0862 0764 0732	
CONV1	'E240	0475	0495	
CONV2	'E25B	0491	0494	
CR	000D	0024	2066 2049 2032 2015 1336 1321 1313 1291 0720 0719 0716 0715 0714 0713 0712 0706 0704 0702 0701 0700 0460 0450 0417 0359 0322 0285	
CRC	'E40D	0711	2147	
CRCBYT	0046	0049	1344 1136 1047 0621 0619 0582	
CRCMSG	'E408	0710	0379	
CRCOMP	'E2FD	0618	0605	
CREAT	'E85C	1307	1325 0654	
CREAT1	'E873	1318	1322	
CRLF	'E205	0417	1448 1407 1333 1218 0749 0737 0295 0224	
CRLF1	'E938	1448	1439	
CRLF2	'E20C	0420	0407	
CTRL	'EA33	1643	1605	
CTRLA	0001	0028	1687	
CTRLC	0003	0029	1776 0149	
CURLFT	'EA1C	1624	1646	
CURRGT	'EA16	1619	1650	
CURT	'EA1E	1626	1622	
CURUP	'EAA2	1701	1664	
DAT1	'EBE7	1921	1927	

SYMBOL	VALUE	STMT	STATEMENT REFS	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
DAT2	'EBF7	1928	1922	
DATSET	'EBDE	1915	0686	
DBOUN	'EB60	1836	1840	
DELAY	'E2A0	0551		
DELAY1	'E2A2	0552	0564	
DELAY2	'E2A3	0553	0558	
DELAY3	'E2A6	0554	0557	
DHEAD	'E417	0712	0740	
DNR	'EE8A	2160	2155	
DUMP	'E4D3	0726	0642	
DUMPO	'E4E3	0737	0753	
DUMP1	'E4E8	0739		
DUMP2	'E4F2	0744	0738	
DUMP3	'E4F8	0746	0756	
ENDCK	'E93C	1457	1489 0748	
ENDCK1	'E93B	1451	1412 1369	
ENTER	'E538	0806	0644	
ENTER1	'E53D	0809	0819	
ENTER2	'E53E	0810	0824	
ENTER3	'E554	0821	0826	
ENTER4	'E556	0823	0829 0807	
ERRCMD	'E134	0259		
ERRCRC	'E1E3	0379	1139	
ERRMSG	'E3DD	0707	0368	
ERRPAR	'E1DE	0376	1952 1292 0991 0989 0988 0904 0901 0895 0885 0876 0838 0484 0480	
ESC	001B	0025	2056 2055 2039 2038 2022 2021 2005 1772	
FF	000C	0023	2056 2039 2022 2005 1988	
FILES	'E6B9	1063	0650	
FILES1	'E6C4	1068	1065	
FILES2	'E6CD	1071	1077 1075	
FILES3	'E6E4	1084	1088	
FILES4	'E704	1098	1105	
FILES5	'E91C	1436	1106	
FILES6	'E91E	1437	1444	
FILHD	'E153	0714	1270 1068	
FILL	'E565	0834	0640	
FILL1	'E573	0842	0844	
FILL2	'E57B	0846		
FILL3	'E50F	0761	0843 0827	
FILL5	'E582	0852	1490	
FINISH	'E1D4	0372	1263 1130 1116 0782 0668 0318	
FLASH	'E51C	0776	1414 1371	
FNDMSG	'E4CA	0721	1226	
GETHD1	'E724	1115	1119	
GETHED	'E71B	1111	1221 1071	
GETTY	'E1A2	0333	1797 1742 1598 0610 0598 0567 0542 0207 0098 0085	
GETPA1	'EC0F	1952	1483 1468 1360 0865 0837	
GETPAR	'EC0C	1951	1466 1358 0834 0806 0726	
GO	'E597	0875	0652	
GRAP	'EEA3	2166	1783 1660 1551	
GRAP1	'EEA9	2168	2183	
GRAP2	'EEAD	2172	2182	
GRAP3	'EEB5	2178	2175	

SYMBOL	VALUE	STMT	STATEMENT REFS	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
HADDR	0009	0059	1235 1234 0995 0994	
HAVIT	'EBBC	1889		
HAVIT1	'EBBD	1890	1875 1873	
HAVIT2	'EBC0	1891	1832	
HAVIT3	'EBC9	1894	1892	
HCHOT2	'E1FA	0403	0400	
HCHOT3	'E200	0406	0404	
HCHOUT	'E1ED	0394	1440 0790 0440 0392	
HEAD2	'E3BC	0704	0195	
HEAD3	'E3D5	0706	0201	
HEADLN	0010	0061	1114 1023	
HEDING	'E362	0700	0191	
HEDPRT	'E6DE	1081	1272 1228 1072	
HEXSPC	'E21C	0436	0747	
HOMECU	'E9BF	1558	1658	
HSIZE	0007	0058	1259 1258 1126 1125 1011 1010	
HTYPE	0006	0057	1277 0933	
HXEQ	000B	0060	1281 1280 0940 0939	
INADD	0041	0045	0981 0980 0089 0088	
INITC	'E062	0120	0066	
INITC2	'E06B	0124	0131	
INITU	'E077	0139	0068	
INITU2	'E091	0153	0155	
INITU3	'E0C8	0191	0165	
INITU4	'E0DD	0198		
INITU5	'E0E5	0202	2157	
INITUA	'E08F	0152	0147	
INITUB	'E0A8	0163	0150	
INITUC	'E0AE	0169	1523 0164	
INITW	'E0E8	0207	2108 0374 0244 0067	
INSTBL	'EC1E	1970	1804	
INTAPE	'E00F	0071	0979	
IVCMMSG	'E3E6	0708	0259	
IVPMSG	'E3F6	0709	0376	
KEYBD	'EB1C	1796	0074	
KEYBRD	'E018	0074	0973	
LDGMSG	'E4BF	0720	1254	
LF	000A	0022	2066 2049 2032 2015 1530 0419 0361	
LINE	0068	0053	1681 1680 1662 1661 1578 1577 1560 1559	
LINE1	'E148	0280	0308 0304 0300 0281	
LINE2	'E16D	0298	0294	
LINE2A	'E173	0303	0311 0290 0283	
LINE3	'E181	0313	0279	
LINE4	'E18A	0317	0323	
LINEBR	'E15C	0289	0797	
LINEBT	'E157	0287	0796	
LINECR	'E204	0409	0286	
LINEI1	'E13F	0274	0816	
LINEIN	'E13A	0271	1309 0296 0227	
LINELN	003C	0035	0273	
LINF1	'EA77	1679	1706	
LINFED	'EA6B	1671	1667	
LIST	'E884	1330	0664	
LIST1	'E889	1333	1337	

SYMBOL	VALUE	STMT	STATEMENT REFS	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
LIST3	'E88F	1335	1339	
LLN	'EA8E	1691	1677	
LLN1	'EA9B	1696	1698	
LOAD	'E78A	1191	0648	
LOAD1	'E78D	1192	1195	
LOAD10	'E822	1269	1261	
LOAD2	'E7A9	1210	1205	
LOAD3	'E7B8	1218	1214 1208 1203	
LOAD3A	'E7BE	1220	1284	
LOAD3B	'E7D3	1230	1224	
LOAD5	'E7DE	1236	1233	
LOAD6	'E7EC	1245	1249	
LOAD7	'E7F8	1250	1241 1240	
LOAD7A	'E807	1257	1252	
LOAD8	'E80E	1260	1268	
LOAD9	'E813	1262	1266	
LOADD	'E0C2	0183	1283	
LOADSK	'E83F	1283	1248	
LODBAS	'E799	1199	0081	
LOOP1	'EB37	1811	1823 1818	
LOOP2	'EB3E	1815	1848 1844	
LOOP3	'EB53	1827	1830	
LOSTD	'EE99	2161	2150	
LSTKEY	006C	0055	1895 1831	
MAIN1	'E106	0233	0257	
MAIN2	'E10B	0236	0241	
MAIN3	'E11C	0246	0910	
MAIN4	'E123	0249	0238	
MAIND	'EC18	1961	0237	
MOT	'EBFB	1934	0545	
MOT1	'E973	1498	1512 0546	
MOT2	'EC09	1945	1937	
MOT3	'EC04	1941	1936	
MOTRON	'E28A	0541	1332 1316 1219 1070 1019 0315 0078	
MOVE	'E942	1466	0658	
MOVE1	'E966	1489	1480	
MOVES	'E95B	1482	1471	
MSGC	'EE78	2157	2154 2151 2148	
MSGOT2	'E1BE	0358	0362	
MSGOUT	'E1BA	0354	1425 1402 1390 1271 1255 1227 1069 0741 0371 0369 0360 0328	
			0202 0196 0192	
MTROF1	'E2B4	0566	1728 0373	
MTROFF	'E2AF	0563	1324 1269 1033 0325 0079	
NAMEN1	'E27A	0517	0525	
NAMEN2	'E282	0522	0519	
NAMFND	'E264	0503	1198 0987	
NOCTRL	'EBD6	1908	1093 1085	
NOGRP	'EBA8	1877	1869 1867	
NOKEY	'EBCF	1898	1825	
NOKEY1	'EBCE	1897	1838	
NULL	'E2C2	0576	1317 1029 1020	
NULL1	'E2C4	0577	0579	
NUMBER	'E255	0486	0482	
NXLOC	'EA29	1634	1699 1682 1669	

SYMBOL	VALUE	STMT	STATEMENT REFS	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
OKMSG	'E4A6	0718	1401	
OUTADD	003F	0044	0960	0959 0109 0108
OUTAPE	'E012	0072	0955	
OUTDLY	'E051	0101	0104	
PARIN	'E776	1165	1167	0076
PARLIN	'E01E	0076	0976	
PARLOT	'E021	0077	0952	
PAROT1	'E780	1175	1177	
PAROUT	'E77F	1174	0077	
PRMP1	'E848	1290	1295	
PRMPTC	'E845	1288	0666	
PROMPK	'E98A	1521	0670	
PROMPT	0044	0047	1297	0225 0156
PSCMSG	'E4AB	0719	1424	
PTRSET	'E9D6	1576	1603	1589 1567
QUIK	'EAD1	1741	2107	1508 0599 0146 0073
QUIK0	'EAF7	1761	1751	
QUIK1	'EB02	1768	1778	
QUIK2	'EB05	1771	1760	1749
QUIK3	'EB06	1772	1766	
QUIK4	'EB0A	1775	1758	
QUIK5	'EB0D	1777	1773	
QUIKD	'EAD6	1743	0570	
QUIKT	'E523	0780	1408	0744
QUIKW1	'E97E	1511	1515	
QUIKW2	'E975	1499		
QUIKW3	'E977	1508	0780	
RAMTOP	F000	0033	0341	0193 0140
REC	'E9E8	1589		
RETUR	'EE06	2091	2087	
RETUR1	'EA0E	1614	1610	
RETURN	'EA2C	1636	1689	1653 1626 1617
RNF	'EE7B	2159	2153	
RUBOUT	007F	0030	0291	
SAVBAS	'E65A	1000	0080	
SAVE	'E638	0987	0646	
SAVE0	'E63B	0988	0997	
SAVE1	'E679	1019	1016	
SAVE2	'E685	1024	1027	
SAVE3	'E697	1032	1039	
SAVE4	'E69D	1034	1037	
SCAN	'E225	0449	1954	1921 1212 1015 0996 0903 0899 0842 0823 0818
SCAN1	'E22C	0454	1482	0864
SCANHL	'E22F	0457	1951	1521 1191 1063 0884 0875 0503
SCANLT	'E232	0459	1206	0990 0465
SCONV	'E586	0861	0999	
SDUMP	'E52A	0788	0810	0727
SEEIFR	'E1A4	0335	0337	
SET	'E5A2	0884	0656	
SET1	'E5AC	0887	0897	
SET2	'E5C2	0898	0888	
SET3	'E5D9	0909		
SETFIL	'E5EE	0933	0688	
SETIN	'E61C	0971	0692	

SYMBOL	VALUE	STMT	STATEMENT REFS	Monitor 1.3/B (C) 1983 by Gebr. van Montfort
SETIN0	'E620	0973	0160	
SETIN1	'E623	0974	0972	
SETIN2	'E62A	0977	0975	
SETIN3	'E631	0980	0978	
SETOT0	'E5FD	0948	0159	
SETOT1	'E600	0950	0947	
SETOT2	'E607	0953	0951	
SETOT3	'E60E	0956	0954	
SETOT4	'E615	0959	0957	
SETOUT	'E5F9	0946	0690	
SETTBL	'E34F	0681	0886	
SHAPE	'EEBE	2186	2167	
SKIPF1	'E73D	1127	1133	
SKIPF2	'E741	1129	1131	
SKIPF3	'E744	1130	1154	1149
SKIPFL	'E734	1124	1076	0184
SPACE	0020	0026	1908	1695 1604 1554 1103 0588 0521 0437 0430 0289
SPACES	'E2D2	0588	1396	1095 1090 0590
SPCOUT	'E217	0430	1910	1686 0811 0755
SPEED	'E5EA	0927	0684	
SPEEDS	003E	0043	0927	0099
START1	'E0F0	0225	0231	
STORE	006E	0036	0342	
TABLE	'E30E	0639	0232	
TABLE1	'EBB9	1886	1884	
TABLE2	'EBBA	1887	1882	1878
TABLE3	'EBBB	1888	1880	
TAPE	'E5DE	0916		
TAPE0	'EAAB	1711	0682	
TAPE1	'E5E6	0921	0919	
TAPE2	'EABD	1723	1734	1715
TAPE3	'EAC9	1730	1717	
TAPEIN	'E2DA	0597	1335	1262 1153 1148 1137 1129 1115 0317 0071
TAPES	003D	0041	1727	1724 0921 0568 0543 0157
TAPIN1	'E2DF	0599	0603	
TAPLV	'E309	0626	0600	
TAPLV2	'E306	0623	0627	
TAPOT1	'E2F4	0612	0614	
TAPOUT	'E2EE	0609	1345	1320 1035 1025 0581 0578 0072
TAPWT	'E759	1146	1334	1256 1124 1111 0316
TAPWT1	'E75A	1147	1157	1151
TAPWT2	'E75C	1148	1152	
TAPWT3	'E766	1153	1156	
TEST	'E8A1	1356	0660	
TEST0	'E8B0	1364	1431	
TEST1	'E8B5	1368	1370	
TEST2	'E8BF	1373	1416	
TEST3	'E8C0	1374	1413	
TEST32	'E8E3	1398	1403	
TEST33	'E8EB	1402	1400	
TEST4	'E8F9	1409	1382	
TESTHD	'E475	0715	1388	
THREAD	0057	0051	1445	1437 1281 1280 1277 1259 1258 1250 1246 1235 1234 1222
			1126	1125 1117 1099 1098 1092 1084 1073

Monitor 1.3/B (C) 1983 by Gebr. van Montfort

SYMBOL	VALUE	STMT	STATEMENT REFS
-----	-----	-----	-----
UARTD	00FC	0017	0616 0604
UARTS	00FD	0016	0612 0601 0121
VDHLD	0067	0052	1893 1643 1615 1590 1569
VID	F080	0034	1692 1691 1576
VIDINT	E9B1	1551	0161
WAIT	EE1D	2107	2114
WAITK	EB84	1857	1855 1853
WAITK1	EB8C	1861	1863 1859 1857
WCSET	EB10	1783	
WCUR	E9CC	1567	1634
WFDC	EE59	2141	2143
WHAT	E1C9	0367	0380 0377 0260
WRCRC	E89B	1344	1323 1038 1028
XEQSET	E5F2	0939	1929

ERRORS=0000

WARNINGS=0000

Inleiding.

STANDAARD MONITOR 1.3 en 1.3/B zijn verbeterde monitors voor de sorcerer computer. Voor zover dit wenselijk was zijn alle bekende fouten en eigenaardigheden uit de standaard monitor versies 1.0 en 1.1 gekorrigeerd (voor zover dit niet in konflikt komt met al bestaande software). Tevens zijn diverse nieuwe mogelijkheden toegevoegd.

Door het handhaven van de versie 1.0 structuur is de compatibiliteit nagenoeg gewaarborgd. Dit in tegenstelling met andere monitor versies.

De monitor wordt geleverd in 2 proms. Elke set proms heeft een eigen identificatie nummer (bij ons geregistreerd samen met de naam van de koper). Dit nummer kan gebruikt worden voor de beveiliging van uw eigen programmatuur en bestanden of bijvoorbeeld ter identificatie van uw computer.

Wij hopen dat u over dit nieuwe produkt tevreden zult zijn. U zult begrijpen dat aan deze nieuwe monitor een flinke hoeveelheid werk ten grondslag ligt. De ontwikkeling hiervan is door ons in onze schaarse vrije tijd gebeurd. Verder worden bij de ontwikkeling van projecten als deze uiteraard onkosten gemaakt en risico's gelopen. Wij zijn dan ook van mening dat deze onkosten gedragen moeten worden door degenen die hier het nut van hebben. Daarom verzoeken wij u, erop toe te zien dat deze monitor alleen op een legale manier verspreid wordt.

Overzicht

1) Veranderingen ten op zichte van versie 1.0

PP Werkt nu naar behoren en "crashed" niet als geen rom-pack aanwezig is.

DUMP De <RUN/STOP> toets onderbreekt/hervat de output nu korrekt. Met <CTRL C> kan het dump-commando worden afgebroken.

ENTER Werkt als vanouds maar laat tevens de inhoud van het geheugen zien. Alleen <CR> laat de geheugeninhoud ongewijzigd en gaat verder naar de volgende geheugenplaats. <^>-toets laat na <CR> de inhoud van de vorige geheugenlokatie zien. Door het intypen van een hexadecimaal getal gevolgd door een <CR> wordt de inhoud gewijzigd en een adres verder gegaan.

MOVE MO 100 200 101 (overlappend moven) werkt nu korrekt.

TEST Deze routine is aanzienlijk verbeterd. Bovendien gaan nu alleen foutmeldingen en teksten naar de printer.

SAVE Het save-commando is uitgebreid met een S-optie zodat de bloklengte kan worden opgegeven. Bij copieerwerk hoeft dan niet steeds weer het einde van het programma te worden berekend.

LOAD LO NAMEX 1 8000 laadt nu wel altijd op het opgegeven adres (in dit voorbeeld op 8000).

FILE Niet printbare karakters in de programmaam worden door spaties vervangen.

<CTRL \> De ASCII waarde (1C) initialiseert nu de graphics.

<CTRL C><RESET> Doet een koud start naar de Monitor alsof er geen rompack is.

<CR> Het typen van alleen een <CR> geeft geen error meer.

<CLEAR> De ASCII waarde (0C) initialiseert de standaard graphics niet meer, zodat uw zelfgedefinieerde graphics ook na het wissen van het beeldscherm nog intact zijn.

Commando's mogen met kleine letters getypt worden, parameters moeten met hoofdletters.

Video Deze routine is nu ca. 18% sneller.

Keyboard Ook deze routine is sneller, heeft een betere ontleding en gebruikt nu bovendien de UART status uit IY+45(=XXD6) waardoor eindelijk de RS232 interface, zonder problemen, kan worden gebruikt.

Quickcheck Deze routine is sneller en test ook op <CTRL S>.

2) Nieuwe commando's

FILL Dit nieuwe commando geeft u de mogelijkheid om een opgegeven geheugendeel te vullen met een door u bepaalde (kombinatie van) waarde(n).

SET X=.. Het is nu ook mogelijk om de datum in de fileheader mee te geven.

SET T=XX Er kunnen nu meerdere opties worden ingeschakeld via dit commando.

<CTRL @> Het typen van de ASCII NUL (00) is nu wel mogelijk.

<CTRL X> Boot het disk systeem (wordt ingebouwd als boot adres wordt opgegeven). In monitor 1.3/B zit bovendien de Complete bootroutine voor de 30,40 en 77 tracks floppy disks ingebouwd (normaal boot adres op BF00) De bootstrap in deze floppy's kan worden uitgeschakeld. Hierdoor is 56K CP/M mogelijk.

Hoofdstuk 1: Algemeen.

Bij voorbeelden worden de teksten die de monitor afdruckt normaal geprint. De antwoorden die de gebruiker ingeeft zijn **vetgedrukt**.

Toetsen worden aangeduid met <beschrijving> bv.:<CR>=carriage return. <CTRL > betekent de controltoets gelijktijdig indrukken met de daarachter vermelde toets bv.: <CTRL X> betekent de controltoets als shift toets voor X gebruiken.

De commando's staan in de linker kolom. Alleen de twee eerste letters zijn noodzakelijk. De commando's mogen in hoofd en/of in kleine letters (upper- of lowercase) worden getypt! De parameters van ieder commando staan rechts; parameters tussen [vierkante haken] mogen opgegeven worden. Letters, opgegeven als parameters moeten hoofdletters zijn.

Tussen een commando en zijn parameters en tussen de parameters onderling moet een separator staan (een spatie is o.a. een goede separator). Commando's worden afgesloten met een <CR>, of indien ze niet uitgevoerd mogen worden met de <@> toets.

Adressen moeten hexadecimaal worden opgegeven. In deze handleiding worden hexadecimale waarden aangegeven als X. Hexadecimale adressen zijn in principe 2 bytes lang (XXXX) en lopen in de range van 0 tot FFFF. Voorgevoegde nullen mogen worden weggelaten (b.v. 0100 mag ook zijn 100).

Parameters kunnen soms hexadecimale waarden zijn en worden dan aangeduid met XX. Soms kunnen dit ook alphanumerieke waarden zijn, dan wordt dit aangegeven door WW (of VV).

Commando regels mogen 60 karakters lang zijn, indien de regel vol is wordt geen backspace meer naar het output-device gezonden. Hierdoor wordt vervelend geknipper voorkomen als het output-device de video is.

Hoofdstuk 2: Commando's.

DUMP DU XXXX [XXXX]
adres-1 [adres-2]

Met het dump-commando wordt de inhoud van het geheugen op het beeldscherm weergegeven. Indien alleen adres-1 is opgegeven verschijnt alleen de inhoud van dat adres. Wanneer ook adres-2 wordt opgegeven, worden alle geheugen inhouds vanaf adres-1 tot en met adres-2 getoond.

Het dump-commando kan tijdelijk onderbroken worden door de <RUN/STOP> toets in te drukken. Zodra de toets voor de tweede keer wordt indrukkt, gaat het dump-commando weer verder. <CTRL C> breekt het commando af.

ENTER EN XXXX [XX [XX ...]] [/]
adres parameter(s)

Het enter-commando geeft de mogelijkheid om de waarde van een geheugenplaats te wijzigen, een voor een of meerdere achter elkaar.

Er zijn drie mogelijkheden:

- 1) Er wordt een nieuwe waarde en <CR> ingetypt. De oude waarde is dan vervangen door de nieuwe waarde en de volgende geheugenlokatie plus inhoud wordt weergegeven op het beeldscherm.
- 2) Er wordt alleen <CR> ingetypt. Hierdoor blijft de waarde ongewijzigd en volgende geheugenlokatie plus inhoud wordt weergegeven op het beeldscherm.
- 3) Er wordt <^> <CR> ingetypt. Hierdoor wordt de inhoud van de voorgaande geheugenlokatie getoond. Meerdere <^> op een regel laat het enter-commando evenzoveel plaatsen teruggaan.

Worden er op dezelfde regel na het adres parameters [XX] [XX] ... vermeld, dan worden deze automatisch vanaf adres XXXX ingebracht.

Het enter-commando kan worden afgebroken door een </> en <CR> te typen.

Voorbeelden:

en 100 41 42 43 44 45 46 47 48 49 / <CR>
Dit zet de waarde 41 op adres 100, 42 op 101 etc.

```

en 100<CR>
0100 41: <CR>      41 is de inhoud van adres 100
0101 42: A0<CR>   adres 101 wordt gewijzigd in A0.
0102 43: ^^<CR>   ^^ betekent twee plaatsen terug!
0100 41: 20<CR>   adres 100 wordt nu pas gewijzigd
0101 AA: ^<CR>    adres 101 is korrekt nu nog een
0100 20: /<CR>    plaats terug om adres 100 te
                  controleren. De / betekent STOP.

```

FILL FL XXXX XXXX XX [XX [XX...]]
adres-1 adres-2 data

Fill een nieuw commando vult de geheugenplaatsen vanaf adres-1 tot adres-2 met de waarde(n) XX. Dit commando wijkt qua syntax af van de overige doordat dit alleen met FL is aan te roepen en niet met zijn volledige naam (fill). Het aantal parameterwaarden mag het aantal te vullen geheugenplaatsen niet overschrijden.

Voorbeelden:

```

>FL 0 100 FF<CR>  Vult de plaatsen van 0 t/m 100
                  met de waarde FF.
>FL 1000 2000 41 42 43 44 45<CR>
                  Vult de plaatsen van 1000 t/m 2000 met
                  de groepwaarden {41 42 43 44 45}
                  repeterend.

```

Niet mag: FL 0 2 00 AA FF BB<CR>
Het aantal bytes is dan namelijk groter dan het aantal te vullen geheugen locaties.

LOAD LO[G] [NAAMX] [X [XXXX]]
naam recorder adres

Laadt het programma met NAAMX van recorder #X vanaf geheugenplaats XXXX. Een G achter LO (LOG) zorgt ervoor dat het programma geladen en meteen gestart wordt op het go-adres in de file-header. (werkt alleen als het file type in de file-header kleiner dan 80 hex. is).

Let op! De G is een parameter en moet dus een hoofdletter zijn!. Alleen LO laadt het eerstvolgende programma van recorder #1 in het geheugen op de plaats die vermeld staat in de "file-header" op de plaats ADRES. Indien u een adres toevoegt, dan wordt geladen vanaf dat adres.

Volgende recordernummers zijn mogelijk:

- 1 =cassette recorder #1 (default)
- 2 =cassette recorder #2
- 3 =RS232 poort 300 baud
- 4 =RS232 poort 1200 baud

De recordermotor wordt bij 3 of 4 niet aangeschakeld.

SAVE SA NAMEX XXXX [S]XXXX [X]
naam adres-1 bloklengte of adres-2 recorder #

Save de inhoud van de geheugenlokatie van adres-1 t/m adres-2 op de recorder die u opgeeft (als geen recordernummer is opgegeven dan recorder 1). Als een "S" voor adres-2 staat, dan wordt het tweede adres de lengte van de te saven geheugen inhoud met als start adres-1 naar de recorder. Dit voorkomt dat u bij het copieren van uw programma's telkens het eindadres moet uitrekenen. De eerste positie van de naam moet een hoofdletter zijn. De eerste vijf letters worden de naam.

Voorbeelden:

SA Prog 1F50 213E 2 <CR>
Saved het programma met naam "Prog" vanaf adres 1F50 tot 213E op recorder #2.

SA Progn 1F50 S1EF <CR>
Saved het programma met de naam "Progn" vanaf adres 1F50 tot 213E op recorder #1. Dit bespaart tijdrovend rekenwerk bij het copieren van uw programma's.

De betekenis van de recordernummers is dezelfde als bij het load-commando.

FILE FI [X]
recorder

List alle file-headers van de recorder #X.

Indien een datum in de fileheader staat dan wordt deze afgedrukt en indien het filetype een niet printbare karakter is (CTRL-code) dan wordt hiervoor in de plaats een spatie afgedrukt zodat de lijst met file-headers netjes onder elkaar komen. Ook wordt nu naar 10 nullen en een 01 gezocht (kenmerk van een fileheader). Door deze methode wordt het aantal CRC - ERRORS aanzienlijk beperkt.

Ook bij dit file-commando zijn weer dezelfde vier recorder # mogelijk zoals bij het load-commando.

GO GO XXXX
adres

Met het go-commando wordt een programma op adres XXXX gestart. Na beëindiging van zo'n programma wordt een volledige warmstart verricht waarbij het IY-register opnieuw wordt berekend.

MOVE MO XXXX XXXX [S]XXXX
adres-1 adres-2 bloklengte of adres-3

Copieert de geheugeninhouden van adres-1 tot adres-2 naar adres-3 of als de "S" voor adres-3 staat, de geheugeninhouden met het aantal vermeld achter de "S" van adres-1 naar adres-2.

De syntax controle op het move-commando is verbeterd. Foutieve opdrachten hebben geen nadelige consequenties meer. Ook overlappende move's worden nu korrekt uitgevoerd en vernielen niet meer het te moven programma. Bovendien is de bloklengte thans korrekt.

Voorbeeld:

Als u met een disassembler werkt of met het file-commando, dan wordt meestal maar een helft van het beeldscherm gebruikt. Als u een deel hiervan wilt bewaren dan kan dat door het volgende commando uit te voeren:

MO F080 F0A0 S77F<CR>

Hierdoor wordt de linker beeldhelft verplaatst naar rechter kant van het beeldscherm.

TEST TE XXXX XXXX [C]
adres-1 adres-2 continue

Test ieder bit in het RAM geheugen vanaf adres-1 tot adres-2. Gedurende het testen knippert een asterix (*) op het scherm. Deze wordt niet meer naar het output-device gestuurd zodat ook tijdens het testen de printer aanstaan kan. Alleen teksten worden geprint. Drie ram adressen in het geheugen kunnen niet worden getest! De adressen F000 en F001 Hex. Hier houdt de monitor zijn hoogste ramadres bij en adres F7FF waar de asterix knippert. Het test-commando voert een volledige read/write test uit zodat verbroken kontakten altijd worden gesignaleerd.

Voorbeeld:

Indien in de ram adressen waar de monitor stack en workarea staan wilt testen dan moet u het volgende hulp programma intypen:

>EN 0 21 FF FF C3 6 E0 / <CR>
>GO 0 <CR>

Hierdoor wordt de stack verplaatst naar FFFF en kan het gebied 0000 tot XXFF getest worden.

>TE 0000 XXFF C <CR>

Test 0000 tot XXFF waarbij XX afhankelijk is van de geheugen grootte (XX=3F is 16K, XX=7F is 32K, XX=BF is 48K en XX=DF is 56K), het getal XX staat op adres F001 Hex.

PROMPT PR=W

Verandert het prompt teken (>) in het teken dat u op de plaats W vermeldt.

Voorbeeld:

>PR=i <CR>
i

CREATE

CR

Maakt een zogenaamde batch-tape op recorder #1. Ieder monitorcommando dat u intypt, wordt op tape gezet in plaats van uitgevoerd. Het create-commando wordt gestopt door op een lege regel alleen de <CR> in te typen.

Voorbeeld:

>CR <CR>
*EN 0 CD 18 E0 28 FB C9/<CR> Deze batch-commando's
*DU 0 FF<CR> zorgen voor een geheugen
*GO 0<CR> dump per pagina en wacht
*DU 100 1FF<CR> telkens op het indrukken
*GO 0<CR> van een toets.
*DU 200 2FF<CR> etc.

*OV<CR>
*<CR> Een <CR> zonder tekst
> beëindigd het create-
commando.

LIST

LI

Geeft een lijst van alle voorkomende commando's op de batch-tape.

BATCH

BA

Zorgt er voor dat de commando's op de batch-tape sequentieel worden uitgevoerd. De batch-commando's worden alleen vanaf recorder #1 uitgevoerd.

OVER

OV

Is het afsluit-commando waarmee een batch-tape moet worden afgesloten. Zodra dit commando is inlezen van de batch-tape en uitgevoerd, gaat de besturing weer via het input device van de monitor.

SET SE W=V

Het set-commando kent diverse onderdelen. Ieder deel zal hier apart behandeld worden.

SE F=XX Stelt het file-type in de file-header in op XX. Bij het file-commando wordt dan het ASCII equivalent van de hexadecimale waarde XX afgedrukt. Bij waarden groter dan 7F hex. kan het ingelezen programma niet met het commando LOG gestart worden.

Alle ASCII waarden tussen 00 en FF hex. kunnen gebruikt worden. Een control karakter in de fileheader wordt bij het printen door een spatie vervangen, zodat een lijst van file-headers goed onder elkaar komt te staan! Default file-type is 00.

SE X=XXXX [YY-YY-YY] Stelt het go-adres in de file-header in op XXXX.

Als optionele parameter kan de datum worden meegegeven zodat ook deze in de file-header wordt afgedrukt. Dit geeft u de mogelijkheid om te controleren met welke programmaversie u te doen heeft.

Voorbeeld:

SE X=1000 13-10-82 <CR>

Zorgt ervoor dat de file-header als volgt komt uit te zien:

NAAMX XXXX XXXX 1000 13-11-1982
Naam bloklengthe beginadres goadres datum

SE T=X Stelt enige waarden in voor de monitor ter bepaling van de baudrate, de RS-232 status en de status van het numerieke toetsenbord.

De volgende parameters zijn mogelijk:

- X: betekenis:
- 0 1200 baud cassette (default)
- 1 300 baud cassette
- 2 1200 baud RS232
- 3 300 baud RS232
- 4 Motor #1 off, motor #2 off (default)
- 5 Motor #1 on, motor #2 off
- 6 Motor #1 off, motor #2 on
- 7 Motor #1 on, motor #2 on
- 8 Normaal keyboard (default)
- 9 De pijlen en de hometoets op het numeriek keyboard werken zonder shift.

SE S=XX Verandert de printsnelheid van het output-device (beeldscherm, printer o.i.d.). Na een koudstart is deze waarde 00, dit is de hoogste printsnelheid. Hoe hoger de hexadecimale waarde XX, hoe lager de printsnelheid wordt. Dit commando is o.a. handig bij het bestuderen van "vreemde" basic listings!

SE O=W Afhangelijk van de waarde van W de output naar:
W output-device

- V sorcerer video (default)
- L centronic's printerdriver (7 bits)
- P parallel printerdriver (8 bits)
- S serie outputdriver
- XXXX outputdriver op adres XXXX

SE I=W Afhangelijk van de waarde van W de input van:
W input-device

- K sorcerer keyboard (default)
- P parallelpoort inputdriver
- S serie inputdriver
- XXXX inputdriver op adres XXXX

Voorbeeld: Aansturen van een seriele printer.

- >SE T=3<CR> Instelling 300 baud RS232.
- >SE O=S<CR> Serie driver wordt output-device.

PP PP [W]
Het prompack-commando geeft de besturing over aan het programma in het rom-pack. Als een parameter is opgegeven dan volgt een koud start (reset). Indien geen parameter is opgegeven dan volgt een warm start. Dit commando controleert nu of er inderdaad een rom-pack aanwezig is zodat bij het ontbreken hiervan geen crash meer ontstaat. Het commando wordt genegeerd als geen pack (geen koud of warm start adres) aanwezig is.

Hoofdstuk 3: Speciale functietoetsen

- <CLEAR> Wist het beeldscherm (in basic CHR\$(12)) maar, in tegenstelling tot de monitor versie 1.0, worden de standaard graphic's niet gereset!
- <CTRL \> Reset de standaard graphics (in Basic CHR\$(28)).
- <CTRL S> Stopt basic programma's of listings totdat een willekeurige toets ingedrukt wordt.
- <CTRL @> Via het toetsenbord kan nu wel ASCII NUL worden getypt (is vaak nodig bij printers).
- <CTRL RUN/STOP> Bij standaard basic wordt na het indrukken van deze combinatie van toetsen alle output geblokkeerd terwijl het programma doorloopt. Door een tweede keer op deze toetsen te drukken wordt deze optie weer uitgeschakeld. Dit is vooral erg handig bij het testen van programma's waarvan de output naar de printer gaat.(equivalent aan <CTRL O>)
- <RUN/STOP> Werkt als tijdelijke stop bij het dump- en test-commando.
- <ESC><RESET> Doet een warm start naar het rom-pack (indien aanwezig) en anders naar de monitor!
LET OP! Eerst <ESC> Indrukken, daarna de beide <RESET> toetsen. De <ESC> toets vasthouden totdat het rom-pack de controle overgenomen heeft! Bij sommige packs kan dit wel even duren.
- <CTRL C><RESET> Doet een koud start naar de monitor. Dit commando is vooral erg handig als u problemen krijgt met b.v. een basic-editor die crashed. Indien u dan <CTRL C><RESET> drukt, wordt de invloed van de editor uitgeschakeld, maar het basic programma blijft onaangetast. De opmerkingen over het indrukken van de toetsen bij <ESC><RESET> gelden ook hier.

<CTRL X> Indien u in het bezit bent van een floppy-disk, dan is het mogelijk om een extra functie op te laten nemen in deze monitor! (in Monitor 1.3/B is deze reeds aanwezig). Deze functie start dan automatisch uw disk op. Dit bespaart iedere keer het typen van "GO XXXX". Indien u later een floppy disk aanschaft, dan kunt u deze functie zelf eenvoudig toevoegen d.m.v. een epromprogrammer.

Op adres E157 en E158 staat FE FF. Op de plaats waar deze FF staat moet u 18 hex. (d.i. <CTRL X>) programmeren. Op de adressen E159 t/m E15B staat CA FF FF. Op de plaats van deze FF FF staan dient u het "Bootadres" (GO-adres) van uw floppy disk systeem te programmeren.

Voorbeeld:

Als uw floppy een opstart adres heeft van DC00 dan zet u op adres E158 18 en op de adressen E15A en E15B respectievelijk 00 en DC.

MONITOR 1.3/B is speciaal geschikt gemaakt voor een van de 30, 40 of 77 tracks softsector floppydisk systemen (GO BF00). De originele bootrom in de disk controller kan worden uitgeschakeld omdat de bootstrap reeds in monitor 1.3/B is ingebouwd. De nieuwe bootloader begint op adres EE09 hex. In combinatie met een van onze SRAM-packs kunt u met een volledige 56Kb CP/M werken.

Hoofdstuk 4: Inbouw handleiding.

Voer de volgende handelingen aan de sorcerer uit:

DEMONTAGE:

- Zet de spanning af, maak alle randapparatuur van de computer los.
- Neem het rompack uit de sorcerer en verwijder de bovenkant (5 schroeven en de keyboard connector).
- Verwijder de stekker van de transformator naar de voedings stabilisatie print.
- Maak de grote print los van de bodem en neem deze er voorzichtig uit (eventueel de aardlijn loskoppelen).
- Verwijder de rom-pack houder (4 schroeven).

OMBOUW:

- Verwijder de twee IC's met opschrift EXM01, Let op positie, inkeping en nummer !!!
- Plaats hierna de nieuwe monitor proms. Let weer op positie, inkeping en nummer!
- Kras de jumper tussen E17 en E13 door (naast de keyboard connector).
- Verbind E13 via een draad met de dikke +5 Volt baan.

U kunt meteen nog de volgende modificatie aanbrengen:
(is niet noodzakelijk voor monitor 1.3)

Aansluiting \overline{WR} lijn naar het rom-pack.
Verbind jumper E18 met E19. Nodig voor het gebruik van een SRAM-pack. LET OP! Jumpers zijn vierkant, doormetalisering rond!

MONTAGE:

- In omgekeerde volgorde van de demontage. Let op de onderstaande punten.
- 1) Let op het pijltje van de keyboard connector, die moet naar de voorkant wijzen (plaats van de programmeur).
 - 2) Vergeet niet de stekker van de voeding, en de aardlijn aan te sluiten.