

T4

## 2K NASCOM MONITOR

The monitor is described by the attached documentation plus the following notes which should be used to modify the software manual. Also see the various technical notes attached.

### NOTES:

- 1) CRT routine supports ICH as Cursor Home.
- 2) Tabulate and Dump output is moved on 5 spaces to the right.
- 3) Load command scrolls up all invalid lines, also;
  - the next command can always be entered normally
  - just stop tape and type fullstop to exit
  - most spurious characters completely ignored.Dump command puts out a Newline so that initial spurious characters are ignored.
- 4) Modify command also supports a colon, meaning go back to previous address, or /aaaa meaning go to the address next.
- 5) When reset is pressed, the message 'NASBUG 4' should appear.

The following actions have been taken by the monitor;

The breakpoint byte (if any) was restored to its correct value.

All monitor tables etc are reset.

The screen is cleared and the message 'NASBUG 4' is output.

The breakpoint command is turned off.

The keyboard options are set to 0 (normal).

Input is requested.

('Lock out' is now completely impossible).

### BREAKPOINT COMMAND.

B aaaa

The purpose of the breakpoint command is to insert a trap code into the program at the address aaaa specified. A breakpoint cannot be inserted into a program which is in Read Only Memory, such as, the monitor itself. Therefore, entering BO turns off the breakpoint.

Continued. ....

Initially, the breakpoint is turned off, since pressing RS deactivates it. The breakpoint command may be entered at any time. When it has been entered, the monitor keeps a record of the value at the breakpoint address. When an Execute command is entered, code E7H is inserted at this address. If the code is encountered during execution, then the program registers are saved, and displayed. The original value is replaced at the breakpoint address. Any command can then be entered. If an E or S command is entered without an execution address specified, then execution starts at the position where the breakpoint was. If an E command is entered, then the original instruction at the breakpoint is executed, and the breakpoint will only stop execution the next time that it is reached.

Note that the original value is put back into the program no matter which way the program is ended. See "How to end a program".

Note that the breakpoint must only be set at the first byte of an instruction in the program.

#### WRITE COMMAND

W aaaa bbbb

The write command outputs data four times faster than Dump.

Data from address aaaa, up to but not including address bbbb is sent to the serial output.

Data is output in blocks, each containing up to 256 bytes of data. (Only the last block may have less).

The format of each block is as follows :

FF FF FF FF	4 start of block characters, FEH
SS SS	Start address, low order first
LL	Length of data (0 = 256)
BB	Block number, this is one less for each block. The last block is block 0.
CC	Checksum for the start address, data length and block number
DD DD ....	Data
EE	Checksum for data
00 00 00 00 00 00 00 00 00 00 00 00	Ten nills (0)

As each block is written, the start address, block number and length are displayed as follows :

SSSS BBLL

Continued. ....

After the command is entered, the Drive LED switches on, there is a 2 second delay, and then the data is output. At the end, the LED goes out and the next command may be entered.

The ten nills output at the end of each block ensure that even if up to ten characters are not read in this block, the next block can still be read correctly. Also, one null is output before the start of the first block, so that an initial spurious FF would be ignored.

#### GENERATE COMMAND

G aaaa bbbb eeee

The Generate command writes a tape, which when read back in loads a program and automatically executes it.

Data from address aaaa, up to but not including address bbbb is written to the tape, in the same format as for the Write Command.

eeee is the address at which execution is to start.

The data on the tape is as follows :

New Line

EO New Line

R New Line

Data, in the format used by the Write and Read Commands.

E eeee New Line

Note :

Start the tape mechanism before entering the G Command.

The LED is only on while the data specified is being output.

When reading the tape in, no commands should be entered.

Simply start the tape, and stop it when the program has started execution.

#### READ COMMAND

R

The Read command reads data which was output by the Write command. (This is four times faster than Load).

See the Write command for the format of the data.

As each block is read, the header data is displayed :

SSSS BBLL

Start address, block number  
length (0 = 256)

After block 0 has been read, the Read command ends.

During the execution of the command the Drive LED is switched on.

Continues

The start of each block is recognised by reading the four start of block characters. All data is ignored until the start of a block. If the checksum for the header data does not agree with the data, then the message ERROR is displayed, and the program waits for the start of the next block. The data following is not loaded.

If the checksum for the data does not agree with the computed total, then the message ERROR is displayed, and the program waits for the start of the next block. In this case, invalid data will have been loaded, but only at the correct addresses.

If any errors are encountered, rewind the tape for about 2 blocks and carry on.

Do not press keys on the keyboard since this will cause errors (which will be detected).

A visual check of the display is required to ensure that all blocks have been loaded correctly.

To stop the Read command in the middle, press Clear Screen four times. This only works between blocks (cursor is at bottom left), so if necessary press any keys until the end of block is reached.

#### I COPY COMMAND

I aaaa bbbb dddd

The Intelligent Copy command is identical to the Copy command, except that data in overlapping regions is never destroyed. See the description of the Copy command.

Copying is from address aaaa to address bbbb for dddd bytes.

#### ARITHMETIC COMMAND

A aaaa bbbb

The Arithmetic command performs simple hexadecimal arithmetic.

Three results are displayed as follows :

SSSS DDDD JJ

SSSS is the sum of the two values.

DDDD is the difference of the two values,

bbbb - aaaa

JJ is the displacement required in a Jump Relative instruction which starts at aaaa, to cause a jump to bbbb.

If such a jump is not possible, ? is displayed instead.

Continued. ....

OUTPUT COMMANDO pp dd

The Output command sends a value to a port.

The value dd is sent to port pp.

pp and dd are displayed.

eg:

>O 7 F

07 0F

>

OFH has been sent to port 7.

QUERY COMMANDQ pp

The Query command obtains a value from a port.

The port number pp and the value dd obtained are displayed.

eg:

>Q 5

05 12

>

Value 12H has been obtained from port 5.

KEYBOARD COMMANDK a

K0 makes the keyboard act normally. (See keyboard layout diagram). This is the condition following Reset.

K1 reverses the effect of the shift key on the letters.

K2 causes the space bar to have a special function, if it is held down while another key is pressed, the hexadecimal value of that key is displayed. This is useful when entering text using the Modify command.

K4 reverses bit 7 of the character generated. Control keys will no longer be recognised. This feature is not useful at present.

The above values may be added, for example entering K3 has the effect of K1 and K2.

Continued. ....

THE KEYBOARD

The keyboard works as shown in the diagram. The normal value is shown on the bottom left of each key. The value above this is obtained by holding down the shift key. The value to the right is obtained by holding down the Control key, and the value above this by holding down the Control key, then the shift key, and then pressing the key itself.

Pressing shift makes the letters lower case.

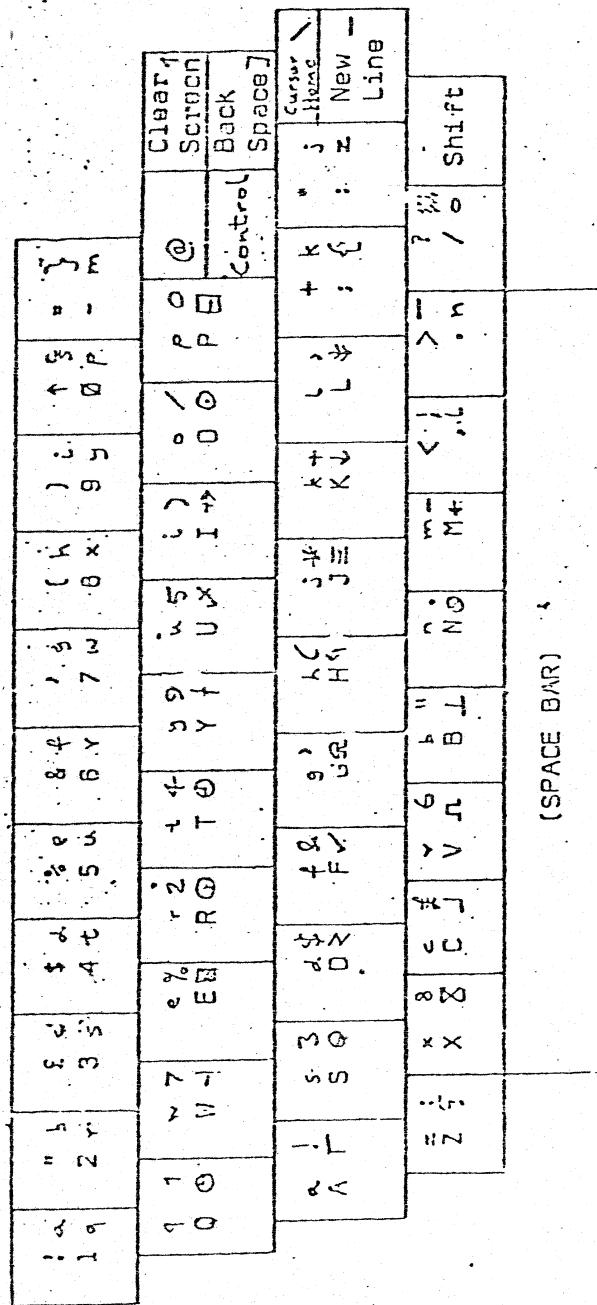
To reverse this, so that the letters are lower case, and become upper case when Shift is pressed, like a typewriter, enter the command K1 (see K command).

Note that the monitor itself will not accept a backspace over a prompt (>). Also, it will not accept the entry of a prompt (>). However, this key is of course available when running a program, for example B-Basic.

Continued. ....

(7)

RS [Reset Key]



NASCOM 1 KEYBOARD LAYOUT

Z COMMANDZ aaaa

The Z command changes the address of the command table. Normally this is 0756H, so entering Z756 has no effect.

A dummy command table is provided at 0511H, so enter Z511 to disable all commands.

(Press Reset to recover from this state).

NORMAL COMMANDN

The Normal command resets the keyboard and CRT jumps to normal. This might be used, for example, to turn off a printer.

The N command is also used to turn off the X command. (See description of the X command).

? COMMAND?

The ? command displays a list of all the commands in the command table. This provides a useful reminder of the facilities available.

ERRORS

If a non-existent command is entered, the message 'ERROR' is output.

If a command has more than three values supplied after it, then the error message is output.

Type a space as the first character on a line for the line to be ignored completely.

RESTART VECTORS

All 8 restart vectors are available for use.

A restart vector is a one byte code which calls an address in the monitor. The monitor will carry out the appropriate function.

Continued. ....

<u>Hex Code in Program</u>	<u>Restart Vector</u>	<u>Function</u>
C7	RST 0	Restart the system.
CF	RST 8	End program and return to monitor. (The registers are not saved. The user stack printer is reset).
D7	RST 10H	Simulated relative call. This is like a relative jump instruction, only it is instead a relative call. The byte following the D7H specifies the offset to the routine to be called.
DF	RST 18H	User subroutine call. This is like a normal call instruction, but the code DFH is followed by a one byte value which specifies the address of the routine to be called. If the value is 0, the address is 0E00H, if it is 1 then the address is 0E03H, up to the highest value, FFH, when the address is 10FDH.
E7	RST 20H	Breakpoint return to monitor. This code is used by the Breakpoint feature of the monitor, but it may also be used in a program.  The program registers are saved and displayed, and commands may then be entered.
EF	RST 28H	Display a string of characters. The string to be output must follow the code EFH. The string is delimited by the value 0. The routine used is that pointed to by the address at \$CRT. This is normally the CRT routine.
F7	RST 30H	Call the routine pointed to by the address at \$CRT. This is normally the CRT routine.
FF	RST 38H	Routine to provide a delay proportioned to the value in A. A is set to 0. The maximum delay possible is about 7.5 ms.

Continued. ....

HOW TO END A PROGRAM

One of the following methods should be used (and no other!) :

- 1) Press RS to restart the system, at any time.

A Halt instruction (76H) may be placed in the program to make the Halt LED light, to indicate that the program has ended.

- 2) Execute code C7H (RST 0) in the program. This is equivalent to pressing RS, and it restarts the system.
- 3) Execute code CFH (RST 8) in the program. This simply returns control to the monitor. The advantage of this over methods 1 or 2 is that the screen is not cleared. This is the recommended method for normal end of program. Note that the program registers are not saved and that the user stack pointer is set back to 1000H.
- 4) Execute code E7H (RST 20H) in the program. This stores the user registers and displays them, before returning control to the monitor. This is the recommended method for abnormal end of program, and during program testing.

Execution of the program may be continued by entering an E or S command. This command must specify the address at which execution is to continue.

- 5) NMI. If the computer was modified to have an NMI button, then pressing this would have the same effect as method 4, except that execution could be continued by simply entering E.

SUPPORT OF PROGRAMS WITH ASCII CONTROL VALUES

Certain programs may be found which assume ASCII codes for keys such as New Line.

These programs may now be run without modification. Before starting execution of the program, execute the following codes :

CD B4 07                    CALL TASC(07B4H)

At the end of the program execute :

CD CO 07                    CALL FASC(07COH)

See the standard ASCII conversion table for details of the conversions carried out.

ASCII CONVERSION TABLE IN MONITORNASCOM → ASCIIASCII → NASCOM

New Line (1FH) →	CR (ODH)	CR (ODH) →	New Line (1FH)
Clear Screen (1EF) →	Escape (1BH)	Escape (1BH) →	Clear Screen (1EH)
Backspace (1DH) →	BS (08H)	BS (08H) →	BS (1DH)
Cursor Home (1CH) →	Line feed(OAH)	LF (OAH) →	Nill (OOH, ignored)
		Rubout(7FH) →	Nill (OOH, ignored)

This table is used by routines TASC and FASC (see Support of Programs with ASCII Control Values). It is also used by the X command.

Continued. ....

THE EXTERNAL (X) COMMANDX aa

The External (X) command gives the NASCOM 1 extensive capabilities for communication with external devices such as ASCII terminals (eg Teletype) and mainframe computers, through the serial input/output.

The X command is turned off by the Normal (N) command.

The value entered after the X must be one of the following :

- 0 Support a terminal in full duplex mode. Every character typed is sent back to the terminal. Line feed is automatically supplied after carriage return. All output is even parity.
- 10 Same as 0, but line feed is not supplied. Suitable for use when feeding in paper tape, which already has line feed.
- 20 Same as 0, but half duplex. Characters entered are not sent back. On output, line feed is provided after carriage return.
- 30 Same as 20, but line feed is not supplied. This makes the NASCOM 1 into a half duplex terminal.
  - 1) Same as 0, 10, 20, 30 in each case, but the output parity is odd instead of even.
  - 21)
  - 31) (Input parity is ignored).

The attached chart shows the characters sent to the serial output, depending on the origin of the characters and the option specified. Values ringed show where problems would occur. Underlined options are the correct ones to choose.

To store data on paper tape, use the Tabulate command. Use the Load command to feed the data back in. When loading back, the data is printed on the terminal, if option 10 or 11 is used.

When a character is received from the terminal, unless it is Escape, Nill, Rubout or Line feed, it is assumed that the program will try to output the character. Therefore, an indicator is set so that the next character output is not sent to the serial output.

When running a B-Basic program through a terminal, press Escape to make the program pause. Then to stop the program press Escape again, or to continue execution press Rubout (or Del).

To backspace on a teletype without this key, use Control H.

Note that the following commands cannot, or should not be used while an X command is in force :

Continued. ....

Dump (D) The routine is not designed for external devices.

Use Tabulate instead

Read (R), Write (W), Generate (G). These are designed for high speed data transfer to cassette. The R command will not recognise input data. W and G will reset the CRT pointer back to normal. (They always do this anyway).

K2 Keyboard option. This is designed for the screen and will reset the CRT pointer back to normal. (It always does this anyway).

Continued. ....

CHARACTERS SENT TO SERIAL OUTPUT

SOURCE OF INPUT	INPUT	FULL DUPLEX CONTROLLER	FULL DEPLEX CONTROLLER FOR PAPER TAPE INPUT	HALF DUPLEX CONTROLLER	HALF DUPLEX TERMINAL
		<u>X0</u>	<u>X10</u>	<u>X20</u>	<u>X30</u>
<u>COMMUNICATION WITH TERMINAL</u>					
NASCOM keyboard	X, CR	<u>X, CR, LF</u>	X, CR <u>O</u>	<u>X, CR, LF</u>	X, CR <u>C</u>
TTY keyboard	X, CR	<u>X, CR, LF</u>	X, CR <u>O</u>	-	-
TTY paper tape	X, CR, LF	X, CR, LF, <u>LF</u>	<u>X, CR, LF</u>	-	-
<u>COMMUNICATION WITH COMPUTER</u>					
NASCOM keyboard	X, CR	X, CR, <u>LF</u>	X, CR	X, CR, <u>LF</u>	<u>X, CR</u>
Mainframe Computer	X, CR, LF	<u>X, CR, LF, LF</u>	<u>X, CR, LF</u>	-	-
<u>X OPTIONS</u>					
<u>BIT</u>					
0 Output	0=Even parity 1=Odd parity				
1 Ignored					
2 Ignored					
3 Ignored					
4 Input/ Output	0=Supply LF after CR 1=Do not				
5 Input	0=Output all serial input 1=Do not output serial input				
6 Ignored					
7 Used internally:					
	1=Suppress next serial output				

PROGRAM : FULL DUPLEX TERMINAL  
ROUTINE XFD

This program makes the NASCOM into a terminal which communicates with a computer in full duplex (ie the computer echoes back all input).

First enter command X30 (or X31), then run this program.

Press Reset to end the program.

TYPEWRITER/TERMINAL PROGRAM

A very simple program (6 bytes long) has been included in the monitor. This program simply waits for an input, displays it, and then waits again.

To execute the program enter :

E 794

To stop the program press Reset.

The program can be used in conjunction with the Keyboard (K) and External (X) commands.

For example enter :

>K1

>E 794

Anything typed will appear on the screen.

The letter keys will be lower case unless Shift is held down.

Another example is to enter :

>X 30

>E 794

The computer will now act as a half-duplex, ASCII computer terminal. Connect the serial data socket to an acoustic coupler (set to full duplex!), and fit a Break key to connect the coupler input to Earth instead of RS232 out, use the 110 bps Teletype clock in the NASCOM, or a 300 bps external clock and telephone a computer timesharing service such as IBM CALL or NCSS. This works, and the NASCOM makes an excellent interactive terminal. You could also write your own intelligent terminal programs of any complexity. It should be quite easy to transfer data between NASCOMS by telephone and to control one NASCOM from another. Full duplex timesharing systems could also be used.

EXTENDED REGISTER DISPLAY

When the Stop command is used, a breakpoint is encountered, or the code E7 (RST 20H) is executed in a program the program registers are displayed.

SP PC AF HL DE BC I IX IY Flags

The Flags are a decoded representation of register F.

The following characters may be displayed, indicating which flag bits have been set :

S Z H P N G

The registers are always displayed at the bottom of the screen. If the cursor is not already at the bottom left, then the screen is scrolled up before the registers are displayed.

Continued. ....



This listing of NASBUG 4 was produced by running ZEAP V1.1  
on a NASCON 1 with 32K additional memory.  
It was printed on a TELETYPE model 43.

```
0005 ; NASBUG 4
0010 ;
0015 ; CHARACTERS
0008 0029 ABS EQU #03
000A 0025 LF EQU #0A
000D 0030 ACR EQU #0D
001B 0035 ESC EQU #1B
001C 0040 CH EQU #1C
001D 0045 BS EQU #1D
001E 0050 CS EQU #1E
001F 0055 CR EQU #1F
003E 0060 GT EQU #3E
005F 0065 CU EQU #5F
007F 0070 DEL EQU #7F
0075 ;
0080 ;
0085 ; MONITOR
0009 0090 ROM EQU #0
0095 ;
0100 ; VIDEO RAM
0000 0105 VRAM EQU #0800
000A 0110 VL1 EQU VRAM+10
004A 0115 VL2 EQU VL1+64
004A 0120 VL14 EQU VRAM+#034A
008A 0125 VL15 EQU VL14+64
0000 0130 VEND EQU VRAM+#400
0135 ;
0140 ; WORKSPACE RAM
0C00 0145 RAM EQU #0C00
0E00 0150 URSTB EQU #0E00
1000 0155 RAMTOP EQU #1000
0160 ;
0165 ; WORKSPACE
0170 ; INITIALISED TO 0
0C00 0175 RAMZ ORG RAM
0180 ; COPY OF PORT 0
0001 0185 PORT0 DEFS 1
0190 ; KEYBOARD STATUS MAP
0000 0195 KMAP DEFS 9
0200 ; COMMAND CHARACTER
```

0001 0205 ARGC DEFS 1  
0001 0210 ; NO OF ARGS  
0001 0215 ARGN DEFS 1  
0000 0220 ; UP TO 3 ARGS ENTERED  
0002 0225 ARG1 DEFS 2  
0002 0230 ARG2 DEFS 2  
0002 0235 ARG3 DEFS 2  
0001 0240 ; NO OF CHARS IN HEX VALUE  
0001 0245 NUMN DEFS 1  
0000 0250 ; HEX VALUE ENTERED  
0002 0255 NUMV DEFS 2  
0000 0260 ; BREAKPOINT ADDRESS  
0002 0265 BRKADR DEFS 2  
0000 0270 ; BREAKPOINT VALUE  
0001 0275 BRKVAL DEFS 1  
0000 0280 RAME DEFS 0  
0285 ;  
0290 ; WORKSPACE  
0295 ; NOT INITIALISED  
0300 ; CURSOR POSITION  
0002 0305 CURSOR DEFS 2  
0310 ; CONFLG NOT 0 IF E COMMAND  
0001 0315 CONFLG DEFS 1  
0320 ; MONITOR STACK  
0018 0325 MONSTK DEFS 24  
0000 0330 STACK DEFS 0  
0335 ; REGISTER SAVE AREA  
0002 0340 RBC DEFS 2  
0002 0345 RDE DEFS 2  
0002 0350 RHL DEFS 2  
0002 0355 RAF DEFS 2  
0002 0360 RPC DEFS 2  
0365 ;  
0370 ; WORKSPACE  
0375 ; INITIALISED BY TABLE  
0000 0380 INITR DEFS 0  
0385 ; USER SP  
0002 0390 RSP DEFS 2  
0395 ; END OF REG SAVE AREA  
0C3F 0400 RSAE EQU RSP+2  
0405 ; LENGTH OF KTAB  
C902 0410 \$KTABL DEFS 2  
0415 ; K OPTION  
0001 0420 \$KOPT DEFS 1  
0425 ; X OPTION  
0001 0430 \$XOPT DEFS 1  
0435 ; ADDRESS OF KTAB  
0002 0440 \$KTAB DEFS 2  
0445 ; ADDRESS OF CTAB  
0002 0450 \$CTAB DEFS 2  
0455 ; JUMP TO NMI ROUTINE

0993 0460 \$NMI DEFS 3  
0993 0465 ; JUMP TO CRT ROUTINE  
0993 0470 \$CRT DEFS 3  
0993 0475 ; JUMP TO KBD ROUTINE  
0993 0480 \$KBD DEFS 3  
0485 ;  
0490 ; END OF WORKSPACE  
0050 0495 INIT EQU \$KBD+3  
0500 ;  
0505 ;  
0510 ; START OF MONITOR  
0999 0515 START ORG ROM  
0009 31330C 0520 LD SP,STACK  
0003 C35705 0525 JP STARTB  
0006 00 0530 DEFB 0,0; SPARE  
0535 ;  
0540 ;  
0545 ; NORMAL RETURN TO MONITOR  
0008 31330C 0550 MRET LD SP,STACK  
000B C36003 0555 JP STRTC  
000E 00 0560 DEFB 0,0; SPARE  
0565 ;  
0570 ;  
0575 ; RELATIVE CALL  
0010 E5 0580 RCAL PUSH HL  
0011 E1 0585 POP HL  
0590 ; INCREMENT RETURN ADDRESS  
0012 E1 0595 POP HL  
0013 23 0600 INC HL  
0014 E5 0605 PUSH HL  
0015 C38505 0610 JP RCALB  
0615 ;  
0620 ;  
0625 ; USER CALL  
0018 E5 0630 UCAL PUSH HL  
0019 E1 0635 POP HL  
0640 ; INCREMENT RETURN ADDRESS  
001A E1 0645 POP HL  
001B 23 0650 INC HL  
001C E5 0655 PUSH HL  
001D C3C205 0660 JP UCALB  
0665 ;  
0670 ;  
0675 ; BREAKPOINT  
0680 ; DECREMENT PC ON STACK  
0020 E3 0685 BRKPT EX (SP),HL  
0021 2B 0690 DEC HL  
0022 E3 0695 EX (SP),HL  
0023 C39503 0700 JP TRAP  
0026 00 0705 DEFB 0,0; SPARE  
0710 ;

0715 ;  
0720 ; OUTPUT A STRING  
0028 E3 0725 PRS EX (SP),HL  
0029 7E 0730 PRS1 LD A,(HL)  
002A 23 0735 INC HL  
0740 ; OUTPUT UNLESS 0  
002B B7 0745 OR A  
002C 2016 0750 JR NZ,PRS2  
002E E3 0755 EX (SP),HL  
002F C9 0760 RET  
0765 ;  
0770 ;  
0775 ; OUTPUT A CHARACTER  
0030 C34A0C 0780 RCRT JP \$CRT  
0033 90 0785 DEFB 0,0; SPARE  
0790 ;  
0795 ;  
0800 ; DELAY ROUTINE  
0035 AF 0805 KDEL XOR A  
0036 F5 0810 KDEL2 PUSH AF  
0037 F1 0815 POP AF  
0820 ; DELAY RESTART  
0038 F5 0825 RDEL PUSH AF  
0039 F1 0830 POP AF  
003A 3D 0835 DEC A  
003B 20F9 0840 JR NZ,KDEL2  
003D C9 0845 EDEL RET  
0850 ;  
0855 ;  
0860 ; GET A CHARACTER  
003E CD400C 0865 CHIN CALL \$KBD  
0041 D8 0870 RET C  
0042 18FA 0875 JR CHIN  
0880 ;  
0885 ;  
0890 ; MORE OF PRS  
0044 F7 0895 PRS2 RST RCRT  
0045 18E2 0900 JR PRS1  
0047 00 0905 DEFB 0,0,0; SPARE  
0910 ;  
0915 ;  
0920 ; SET, RESET BIT IN P0  
004A F5 0925 FLPFLP PUSH AF  
004B CD5300 0930 CALL FLIP  
004E F1 0935 POP AF  
004F 1802 0940 JR FLIP  
0945 ;  
0950 ; FLIP BIT 4 IN P0  
0051 3E10 0955 NOTFLP LD A, #10  
0960 ;  
0965 ; FLIP BIT(S) IN P0

0053 E5 0970 FLIP PUSH HL  
0054 21000C 0975 LD HL,PORT0  
0057 AE 0980 XOR (HL)  
0058 D309 0985 OUT (0),A  
005A 77 0990 LD (HL),A  
005B E1 0995 POP HL  
005C C9 1000 RET  
1005 ;  
1010 ;  
1015 ; SERIAL OUTPUT TO P1  
1020 ; DISPLAY CHARACTER  
005D F7 1025 SRLOUT RST RCRT  
1030 ; SEND TO P1  
005E D301 1035 SRLX OUT (1),A  
1040 ; WAIT UNTIL OUTPUT  
0060 DB62 1045 SRL4 IN A,(2)  
0062 87 1050 ADD A,A  
0063 FB 1055 RET H  
0064 18FA 1060 JR SRL4  
1065 ;  
1070 ;  
1075 ; NMI RESTART  
0066 C3470C 1080 RNMI JP \$NMI  
1085 ;  
1090 ;  
1095 ; KEYBOARD ROUTINE  
0069 C5 1100 KBD PUSH BC  
006A D5 1105 PUSH DE  
006B E5 1110 PUSH HL  
1115 ; RESET KBD COUNTER  
006C 3E02 1120 LD A,2  
006E CD4A09 1125 CALL FLPFLP  
1130 ; STORE ROW 0 IN MAP  
0071 21010C 1135 LD HL,KMAP  
0074 DB00 1140 IN A,(0)  
0076 2F 1145 CPL  
0077 77 1150 LD (HL),A  
1155 ;  
1160 ; SCAN 8 ROWS  
0078 0608 1165 LD B,B  
1170 ; INCREMENT KBD COUNTER  
007A 3E01 1175 KSC1 LD A,1  
007C CD4A09 1180 CALL FLPFLP  
007F 23 1185 INC HL  
1190 ; GET ROW STATUS  
0080 DB00 1195 IN A,(0)  
0082 2F 1200 CPL  
0083 57 1205 LD D,A  
1210 ; IF MAP DIFFERENT  
1215 ; GO TO FIND OUT WHY  
0084 AE 1220 XOR (HL)

0085 2007 1225 JR NZ,KSC2  
1230 ; SCAN NEXT ROW  
0087 10F1 1235 KSC1A DJNZ KSC1  
1240 ; NO KEY PRESSED  
0089 B7 1245 KSC8 OR A  
1250 ; GO TO RESTORE REGS  
008A C37001 1255 KSC9 JP CRT9  
1260 ;  
1265 ;  
1270 ; DEFINE MONITOR TYPE  
008D 00 1275 MONTYP DEFB 0 .  
1280 ;  
1285 ;  
1290 ; MORE OF KBD ROUTINE  
1295 ; WAIT, TO DEBOUNCE  
008E C03500 1300 KSC2 CALL KDEL  
1305 ; GET ROW AGAIN  
0091 DB00 1310 IN A,(0)  
0093 2F 1315 CPL  
0094 5F 1320 LD E,A  
1325 ; E = NEW STATE  
0095 7A 1330 LD A,D  
1335 ; A = OLD STATE  
0096 AE 1340 XOR (HL)  
1345 ; A = CHANGES  
1350 ; FIND CHANGED BIT  
0097 0EFF 1355 LD C,-1  
0099 1600 1360 LD D,0  
009B 37 1365 SCF  
009C CB12 1370 KSC4 RL D  
009E 0C 1375 INC C  
009F 1F 1380 RRA  
00A0 30FA 1385 JR NC,KSC4  
1390 ; C = COL CHANGED  
1395 ; D = MASK WITH 1 AT CHANGE  
00A2 7A 1400 LD A,D  
00A3 A3 1405 AND E  
00A4 5F 1410 LD E,A  
1415 ; E = NEW STATE,  
1420 ; MASKED BY CHANGE  
1425 ; IF MAP STATE AND NEW  
1430 ; STATE ARE EQUAL, IGNORE  
00A5 7E 1435 LD A,(HL)  
00A6 A2 1440 AND D  
00A7 BB 1445 CP E  
00A8 28DD 1450 JR Z,KSC1A  
1455 ; UPDATE MAP  
00AA 7E 1460 LD A,(HL)  
00AB AA 1465 XOR D  
00AC 77 1470 LD (HL),A  
1475 ; IF NEW STATE IS 0, THEN

1480 ; KEY RELEASED, SO IGNORE  
00AD 7B 1485 LD A,E  
00AE B7 1490 OR A  
00AF 2806 1495 JR Z,KSC1A  
1500 ;  
1505 ; SET UP VALUE: SRRRRCCC  
1510 ; S=1 IF SHIFT DOWN  
1515 ; RRRR=9-ROW NUMBER  
1520 ; CCC=COLUMN NUMBER  
00B1 3A919C 1525 LD A,(KMAP)  
00B4 E610 1530 AND #10  
00B6 B0 1535 OR B  
00B7 87 1540 ADD A,A  
00B8 87 1545 ADD A,A  
00B9 87 1550 ADD A,A  
00BA B1 1555 OR C  
1560 ;  
1565 ; SEARCH TABLE  
00BB ED4B3F9C 1570 LD BC,(\$KTABL)  
00BF 2A439C 1575 LD HL,(\$KTAB)  
00C2 EDB1 1580 CPIR  
00C4 280B 1585 JR Z,KSC5  
1590 ; CHECK AGAIN FOR  
1595 ; UNSHIFTED CHAR  
00C6 2A439C 1600 LD HL,(\$KTAB)  
00C9 ED4B3F9C 1605 LD BC,(\$KTABL)  
00CD E67F 1610 AND #7F  
00CF EDB1 1615 CPIR  
00D1 2086 1620 KSC5 JR NZ,KSC8  
1625 ; CALC ASCII VALUE  
00D3 ED4B439C 1630 LD BC,(\$KTAB)  
00D7 37 1635 SCF  
00D8 ED42 1640 SBC HL,BC  
00DA 7D 1645 LD A,L  
1650 ;  
1655 ; SUPPORT LOWER CASE  
00DB FE41 1660 CP "À"  
00DD 381E 1665 JR C,K20  
00DF FE5B 1670 CP "Z+1"  
00E1 301A 1675 JR NC,K20  
1680 ; IT IS A LETTER  
00E3 21019C 1685 LD HL,KMAP  
00E6 CB66 1690 BIT 4,(HL)  
1695 ; 1 = SHIFT DOWN  
00E8 21419C 1700 LD HL,\$KOPT  
00EB 2008 1705 JR NZ,K8  
1710 ; SHIFT NOT DOWN  
00ED CB46 1715 BIT 0,(HL)  
00EF 280C 1720 JR Z,K20  
00F1 C620 1725 ADD A,#20  
00F3 1808 1730 JR K20

1735 ; SHIFT WAS DOWN  
00F5 C620 1740 KB ADD A, #20  
00F7 CB46 1745 BIT 0, (HL)  
00F9 2802 1750 JR Z, K20  
00FB D620 1755 SUB #20  
1760 ;  
1765 ; CONTROL KEYS ROUTINE  
00FD C00D04 1770 K20 CALL KBDC  
1775 ;  
1780 ; K4 OPTION  
1785 ; CHANGE BIT .7  
0100 21410C 1790 LD HL, \$KOPT  
0103 CB56 1795 BIT 2, (HL)  
0105 2802 1800 JR Z, K40  
0107 EE80 1805 XOR #80  
1810 ;  
1815 ; K2 OPTION  
1820 ; HEX KEYBOARD  
0109 37 1825 K40 SCF  
010A CB4E 1830 BIT 1, (HL)  
010C CA8A00 1835 K42 JP Z, KSC9  
1840 ; IF SPACE, NORMAL  
010F FE20 1845 CP "  
0111 37 1850 SCF  
0112 28F8 1855 JR Z, K42  
1860 ; IF SPACE NOT DOWN, NORMAL  
0114 21080C 1865 LD HL, KMAP+7  
0117 CB66 1870 BIT 4, (HL)  
0119 28F1 1875 JR Z, K42  
011B E1 1880 POP HL  
011C D1 1885 POP DE  
011D C1 1890 POP BC  
1895 ; RESET CRT  
011E CDC697 1900 CALL FASC2  
1905 ; DISPLAY HEX VALUE AND SPACE  
1910 ; RETURN WITH CARRY NOT SET  
0121 CD4402 1915 CALL B2HEX  
0124 B7 1920 OR A  
0125 C33C02 1925 JP SPACE  
1930 ;  
1935 ;  
1940 ; WORKSPACE INITIALISATION  
1945 ; TABLE  
0000 1950 INIT DEFS 0  
1955 ; USER STACK POINTER  
0128 0010 1960 ISP DEFW RAMTOP  
1965 ; LENGTH OF KTAB  
012A 6000 1970 IKTAB DEFW KTAB-EKTAB  
1975 ; K OPTION  
012C 00 1980 IKOPT DEFB 0  
1985 ; X OPTION

012D 09 1999 IXOPT DEFB 0  
012E D495 1995 ; KEYBOARD TABLE  
0130 5597 2000 IKTAB DEFU KTAB  
0131 ; COMMAND TABLE  
0132 5597 2010 ICTAB DEFU CTAB  
0133 ; NMI JUMP  
0132 C39503 2020 INMI JP TRAP  
0134 ; CRT JUMP  
0135 C33B01 2030 ICRT JP CRT  
0136 ; KBD JUMP  
0138 C3F204 2040 IKBO JP TIN  
0055 2045 INITX DEFS 0  
2050 ;  
2055 ;  
2060 ; CRT ROUTINE  
2065 ; IGNORE NULLS  
013B B7 2070 CRT OR A  
013C C8 2075 RET Z  
013D C5 2080 PUSH BC  
013E D5 2085 PUSH DE  
013F E5 2090 PUSH HL  
0140 F5 2095 PUSH AF  
2100 ;  
2105 ; CLEAR SCREEN  
0141 FE1E 2110 CP CS  
0143 202F 2115 JR NZ,CRTA6  
2120 ; SET TOP LEFT  
0145 210908 2125 LD HL,VL1-1  
0148 36FF 2130 LD (HL),#FF  
2135 ; CLEAR TOP LINE  
014A 23 2140 INC HL  
014B 0630 2145 LD B,48  
014D 3620 2150 CRTA4 LD (HL),"  
014F 23 2155 INC HL  
0150 10FB 2160 DJNZ CRTA4  
2165 ; SET MARGIN  
0152 0610 2170 LD B,16  
0154 3600 2175 CRTA5 LD (HL),0  
0156 23 2180 INC HL  
0157 10FB 2185 DJNZ CRTA5  
2190 ; COPY DOWN SCREEN  
0159 EB 2195 EX DE,HL  
015A 210A08 2200 LD HL,VL1  
015D 01B903 2205 LD BC,VEND-VRAM-64-10-6  
0160 EDB0 2210 LDIR  
2215 ; SET BOTTOM RIGHT  
0162 3EFF 2220 LD A,#FF  
0164 32BA08 2225 LD (VL15+48),A  
2230 ;  
2235 ; SET HL TO BOTTOM LEFT  
0167 218A08 2240 CRT0 LD HL,VL15

2245 ;  
2250 ; POSITION CURSOR  
016A 365F 2255 CRT1 LD (HL),CU  
016C 22180C 2260 LD (CURSOR),HL  
2265 ;  
2270 ; RETURN  
016F F1 2275 CRT2 POP AF  
0170 E1 2280 CRT9 POP HL  
0171 D1 2285 POP DE  
0172 C1 2290 POP BC  
0173 C9 2295 RET  
2300 ;  
2305 ; REMOVE CURSOR  
0174 2A180C 2310 CRTA6 LD HL,(CURSOR)  
0177 3620 2315 LD (HL),"  
2320 ;  
2325 ; BACKSPACE  
0179 FE1D 2330 CP BS  
017B 269B 2335 JR NZ,CRTAB  
2340 ; IGNORE MARGINS  
017D 2B 2345 CRTA7 DEC HL  
017E 7E 2350 LD A,(HL)  
017F B7 2355 OR A  
0180 28FB 2360 JR Z,CRTA7  
2365 ; CHECK FOR TOP LEFT  
0182 3C 2370 INC A  
0183 20E5 2375 JR NZ,CRT1  
0185 23 2380 INC HL  
0186 10E2 2385 JR CRT1  
2390 ;  
2395 ; CURSOR HOME  
0188 FE1C 2400 CRTAB CP CH  
018A 28DB 2405 JR Z,CRT0  
2410 ;  
2415 ; NEW LINE  
018C FE1F 2420 CP CR  
018E 2809 2425 JR Z,CRT3  
2430 ;  
2435 ; ANY OTHER CHARACTER  
0190 77 2440 LD (HL),A  
2445 ; IGNORE MARGINS  
0191 23 2450 CRT2A INC HL  
0192 7E 2455 LD A,(HL)  
0193 B7 2460 OR A  
0194 28FB 2465 JR Z,CRT2A  
2470 ; CHECK FOR BOTTOM RIGHT  
0196 3C 2475 INC A  
0197 2001 2480 JR NZ,CRT1  
2485 ;  
2490 ; SCROLL UP  
0199 110A08 2495 CRT3 LD DE,VLI

019C 214A08 2500 LD HL,VL2  
019F 017903 2505 LD BC,VEND-VRAM-64-64-16  
01A2 ED56 2510 LDH R  
2515 ; CLEAR BOTTOM LINE  
01A4 0639 2520 LD B,48  
01A6 2B 2525 CRT3A DEC HL  
01A7 3620 2530 LD (HL),"  
01A9 10FB 2535 DJNZ CRT3A  
01AB 18BA 2540 JR CRT0  
2545 ;  
2550 ;  
2555 ; MODIFY COMMAND  
01AD 2A0C0C 2560 MODIFY LD HL,(ARG1)  
2565 ; DISPLAY CURRENT VALUE  
01B0 CD3202 2570 MOD1 CALL TBCD3  
01B3 7E 2575 LD A,(HL)  
01B4 CD4402 2580 CALL B2HEX  
2585 ; GET INPUT LINE  
01B7 CD8105 2590 CALL BRKST  
01BA 11520B 2595 LD DE,VL14+8  
01BD 0600 2600 LD B,0  
2605 ; GET EACH ENTRY  
01BF E5 2610 MOD2 PUSH HL  
01C0 CD5A02 2615 CALL NEXNUM  
2620 ; IF END OR INVALID, MOD3  
01C3 7E 2625 LD A,(HL)  
01C4 B7 2630 OR A  
01C5 2808 2635 JR Z,MOD3  
2640 ; PUT VALUE INTO MEMORY  
01C7 23 2645 INC HL  
2650 ; HL = NUMN+1 = NUMV  
01C8 7E 2655 LD A,(HL)  
01C9 E1 2660 POP HL  
01CA 77 2665 LD (HL),A  
2670 ; INC COUNT OF BYTES CHANGED  
01CB 04 2675 INC B  
01CC 23 2680 INC HL  
01CD 18F0 2685 JR MOD2  
2690 ;  
2695 ; LOOK AT INVALID CHAR  
01CF E1 2700 MOD3 POP HL  
01D0 1A 2705 LD A,(DE)  
2710 ; IF "." RETURN  
01D1 FE2E 2715 CP ".  
01D3 C8 2720 RET Z  
2725 ; HOW MANY BYTES CHANGED?  
01D4 78 2730 LD A,B  
01D5 B7 2735 OR A  
2740 ; JUMP TO EXTENSION  
01D6 C38005 2745 JP MOD4B  
01D9 00 2750 DDEFB 0,0; SPARE

2755 ;  
2760 ;  
2765 ; ROUTINE TO GET INPUT LINE  
2770 ; OUTPUT PROMPT  
01DB EF 2775 INLINE RST PRS  
61DC 3E 2780 DEFB GT,\$  
2785 ; GET INPUT CHARACTER  
01DE CD3E00 2790 INL0 CALL CHIN  
2795 ; IF CR, RETURN  
01E1 FE1F 2800 CP CR  
01E3 285B 2805 JR Z,CRLF  
2810 ; DO NOT ALLOW GT, OR  
2815 ; BACKSPACE OVER GT  
01E5 F5 2820 PUSH AF  
01E6 FE1D 2825 CP BS  
01E8 2096 2830 JR NZ,INL4  
01EA ED5B180C 2835 LD DE,(CURSOR)  
01EE 1B 2840 DEC DE  
01EF 1A 2845 LD A,(DE)  
01F0 FE3E 2850 INL4 CP GT  
01F2 2894 2855 JR Z,INL6  
01F4 F1 2860 INL5 POP AF  
2865 ; OUTPUT CHARACTER  
01F5 F7 2870 INL5A RST RCRT  
01F6 18E6 2875 JR INL0  
2880 ; IGNORE BY MAKING NULL  
01F8 F1 2885 INL6 POP AF  
01F9 AF 2890 XOR A  
01FA 18F9 2895 JR INL5A  
2900 ;  
2905 ;  
2910 ; TABULATE COMMAND  
01FC CD9B06 2915 TABCODE CALL ARG52  
2920 ; IF HL<DE GO TO TBCD1C  
01FF B7 2925 TBCD1B OR A  
0200 EB52 2930 SBC HL,DE  
0202 19 2935 ADD HL,DE  
0203 3805 2940 JR C,TBCD1C  
2945 ; END, SO OUTPUT ".",CR  
0205 EF 2950 RST PRS  
0206 2E 2955 DEFB ",",CR,\$  
0209 C9 2960 RET  
2965 ; INIT CHECKSUM  
020A 0E00 2970 TBCD1C LD C,\$  
2975 ; OUTPUT ADDRESS  
020C EF 2980 RST PRS  
020D 20 2985 DEFB ",",\$  
0210 CD3202 2990 CALL TBCD3  
2995 ; OUTPUT 8 BYTES  
0213 0608 3000 LD B,B  
0215 7E 3005 TBCD1A LD A,(HL)

0216 CD2B02 3919 CALL TBCD2  
0219 23 3915 INC HL  
021A CD3C02 3920 CALL SPACE  
021B 10F6 3925 DJNZ TBCD1A  
3939 ; OUTPUT CHECKSUM ETC  
021F 79 3935 LD A,C  
0220 CD4402 3949 CALL B2HEX  
0223 EF 3945 RST PRS  
0224 10 3959 DEFB BS,BS,CR,  
0228 1805 3955 JR TBCD1B  
022A 00 3960 DEFB \$; SPARE  
3965 ;  
3970 ;  
3975 ; ADD A TO CHECKSUM AND PRINT  
022B F5 3989 TBCD2 PUSH AF  
022C 81 3985 ADD A,C  
022D 4F 3999 LD C,A  
022E F1 3995 POP AF  
022F C34402 3100 JP B2HEX  
3105 ;  
3110 ;  
3115 ; PRINT HL THEN SPACE  
0232 7C 3129 TBCD3 LD A,H  
0233 CD2B02 3125 CALL TBCD2  
0236 7D 3139 LD A,L  
0237 CD2B02 3135 CALL TBCD2  
023A 00 3140 DEFB \$,\$; SPARE  
3145 ;  
3150 ;  
3155 ; PRINT SPACE  
023C 3E20 3160 SPACE LD A," "  
023E 1817 3165 JR JCRT  
3170 ;  
3175 ;  
3180 ; PRINT NEW LINE  
0240 3E1F 3185 CRLF LD A,CR  
0242 1813 3190 JR JCRT  
3195 ;  
3200 ;  
3205 ; PRINT A IN HEX  
0244 F5 3210 B2HEX PUSH AF  
0245 1F 3215 RRA  
0246 1F 3220 RRA  
0247 1F 3225 RRA  
0248 1F 3230 RRA  
0249 CD4D02 3235 CALL B2HEX1  
024C F1 3240 POP AF  
3245 ;  
3250 ; PRINT RIGHT HALF OF A  
024D E60F 3255 B2HEX1 AND #0F  
024F C630 3260 ADD A,"0

0251 FE3A 3265 CP "9+1  
0253 3892 3270 JR C,JCRT  
0255 C697 3275 ADD A,"A-"0-10  
3280 ;  
3285 ; OUTPUT CHARACTER  
0257 C34A0C 3290 JCRT JP \$CRT  
3295 ;  
3300 ;  
3305 ; READ IN HEX VALUE  
3310 ; DE POINTS TO INPUT LINE  
3315 ; NUMN = CHARS IN INPUT  
3320 ; NUMV = VALUE ENTERED  
025A 1A 3325 NEXNUM LD A,(DE)  
3330 ; IGNORE BLANKS  
025B FE20 3335 CP "  
025D 13 3340 INC DE  
025E 26FA 3345 JR Z,NEXNUM  
0260 1B 3350 DEC DE  
3355 ; SET NUMN TO 0  
0261 AF 3360 XOR A  
0262 21120C 3365 LD HL,NUMN  
0265 77 3370 LD (HL),A  
3375 ; SET NUMV TO 0  
0266 23 3380 INC HL  
0267 77 3385 LD (HL),A  
0268 23 3390 INC HL  
0269 77 3395 LD (HL),A  
3400 ; SET A TO CHAR ENTERED  
026A 1A 3405 NN1 LD A,(DE)  
3410 ; POINT HL TO NUMN  
026B 2B 3415 DEC HL  
026C 2B 3420 DEC HL  
3425 ; CONVERT FROM ASCII  
3430 ; IF LT 0 RETURN  
026D D630 3435 SUB "0  
026F F8 3440 RET M  
3445 ; IF LT 10 THEN OK, SO NN2  
0270 FE9A 3450 CP 10  
0272 3898 3455 JR C,NN2  
3460 ; CONVERT A/F FROM ASCII  
0274 D697 3465 SUB "A-"0-10  
3470 ; IF LT 10 INVALID SO RETURN  
0276 FE9A 3475 CP 10  
0278 F0 3480 RET M  
3485 ; IF GE 16 INVALID SO RETURN  
0279 FE10 3490 CP 16  
027B F0 3495 RET P  
3500 ;  
3505 ; VALID CHARACTER FOUND  
3510 ; POINT TO NEXT CHARACTER  
027C 13 3515 NN2 INC DE

3529 ; INCREMENT NUMN  
027D 34 3525 INC (HL)  
3530 ; PUT VALUE IN NUMV, ROTATING  
3535 ; PREVIOUS CONTENTS  
027E 23 3540 INC HL  
027F ED6F 3545 RLD  
0281 23 3550 INC HL  
0282 ED6F 3555 RLD  
0284 18E4 3560 JR NN1  
3565 ;  
3570 ;  
3575 ; MAIN MONITOR LOOP  
3580 ; GET LINE AND OBEY  
0286 CD8105 3585 PARSE CALL BRKST  
0289 114B9B 3590 LD DE,VL14+1  
028C 019A9C 3595 LD BC,ARGC  
3600 ; IF COMMAND IS BLANK, AND  
3605 ; PREVIOUS COMMAND NOT \$,  
3610 ; IGNORE IT.  
028F 1A 3615 LD A,(DE)  
0290 FE20 3620 CP "  
0292 2005 3625 JR NZ,PA2  
0294 0A 3630 LD A,(BC)  
0295 FE53 3635 CP "\$  
0297 20E0 3640 JR NZ,PARSE  
3645 ; STORE COMMAND IN ARGC  
0299 02 3650 PA2 LD (BC),A  
029A 03 3655 INC BC  
3660 ; POINT TO NEXT INPUT CHAR  
029B 13 3665 INC DE  
3670 ; SET ARGN TO \$  
029C AF 3675 XOR A  
029D 02 3680 LD (BC),A  
3685 ; GET THE ARGUMENTS  
029E 03 3690 PLOOP INC BC  
029F CD5A02 3695 CALL NEXNUM  
3700 ; IF NUMN = \$ THEN PEND  
02A2 7E 3705 LD A,(HL)  
02A3 B7 3710 OR A  
02A4 281C 3715 JR Z,PEND  
3720 ; COPY NUMV TO ARG1/2/3  
02A6 23 3725 INC HL  
02A7 7E 3730 LD A,(HL)  
02A8 02 3735 LD (BC),A  
02A9 23 3740 INC HL  
02AA 03 3745 INC BC  
02AB 7E 3750 LD A,(HL)  
02AC 02 3755 LD (BC),A  
3760 ; INCREMENT ARGN  
02AD 210B0C 3765 LD HL,ARGN  
02B0 34 3770 INC (HL)

02B1 7E 3775 LD A,(HL)  
02B2 FE94 3780 ; IF LT 4 ARGS, OK SO PLOOP  
02B4 38EB 3785 DP 4  
02B6 F5 3790 JR C,PLOOP  
02B7 3899 PUSH AF  
02B8 EF 3795 ;  
02B9 45 3800 ; OUTPUT ERROR MESSAGE  
02B0 1F 3810 ERM POP AF  
02B1 18C4 3815 RST PRS  
02B2 45 3820 DEFM /error/  
02B3 1F 3825 DEFB CR,0  
02B4 18C4 3830 JR PARSE  
02B5 3835 ;  
02B6 1F 3840 ; FIND ADDRESS OF COMMAND  
02B7 1F 3845 ; ROUTINE AND CALL IT,  
02B8 1F 3850 ; RETURNING TO PARSE  
02B9 3A9A0C 3855 PEND LD A,(ARGC)  
02B0 2A450C 3860 LD HL,(\$C1AB)  
02B1 CD6604 3865 CALL TABLE  
02B2 118602 3870 LD DE,PARSE  
02B3 D5 3875 PUSH DE  
02B4 E9 3880 JP (HL)  
02B5 3885 ;  
02B6 3890 ;  
02B7 3895 ; THE EXECUTE COMMAND  
02B8 3EFF 3900 ; CONFLG NOT 0 IF E COMMAND  
02B9 321A0C 3905 EXEC LD A,-1  
02B0 3910 LD (CONFLG),A  
02B1 3915 ;  
02B2 3920 ; EXECUTE AND STEP COMMANDS  
02B3 3925 ; SET NMI JUMP ADDRESS  
02B4 210503 3930 EXEC1 LD HL,TRAP  
02B5 22430C 3935 LD (\$NMI+1),HL  
02B6 E1 3940 ; DISCARD RETURN TO PARSE  
02B7 3945 POP HL  
02B8 3950 ; IF NO ADDRESS ENTERED,  
02B9 3955 ; USE STORED USER PC  
02B0 3A9B0C 3960 LD A,(ARGN)  
02B1 B7 3965 OR A  
02B2 2804 3970 JR Z,EXEC2  
02B3 3975 ; SET USER PC TO NEW ADDRESS  
02B4 2A0C0C 3980 LD HL,(ARG1)  
02B5 223B0C 3985 LD (RPC),HL  
02B6 3990 ; RESTORE USER REGS BC DE AF  
02B7 C1 3995 EXEC2 POP BC  
02B8 D1 4000 POP DE  
02B9 F1 4005 POP AF; IN FACT RHL  
02B0 F1 4010 POP AF  
02B1 4015 ; RESTORE USER SP  
02B2 2A3D0C 4020 LD HL,(RSP)  
02B3 F9 4025 LD SP,HL

4030 ; PUT USER PC ON TOP OF STACK  
02F9 2A3B0C 4035 LD HL,(RPC)  
02F3 E5 4040 PUSH HL  
4045 ; RESTORE USER HL  
02F4 2A370C 4050 LD HL,(RHL)  
4055 ; SET BIT 3 OF P0, TO  
4060 ; ACTIVATE NMI  
02F7 F5 4065 PUSH AF  
02F8 3E08 4070 LD A,8  
02FA D300 4075 OUT (0),A  
02FC F1 4080 POP AF  
4085 ; EXECUTE ONE STEP OF PROGRAM  
02FD ED45 4090 RETN  
4095 ;  
4100 ;  
4105 ; THE STEP COMMAND  
4110 ; CONFLG 0 IF S COMMAND  
02FF AF 4115 STEP XOR A  
0300 321A0C 4120 LD (CONFLG),A  
0303 18D0 4125 JR EXEC1  
4130 ;  
4135 ;  
4140 ; COME HERE AFTER NMI  
0305 F5 4145 TRAP PUSH AF  
0306 E5 4150 PUSH HL  
4155 ; RESET NMI BIT IN P0  
0307 3A900C 4160 LD A,(PORT0)  
030A D300 4165 OUT (0),A  
4170 ; IF CONFLG NOT 0 THEN E  
4175 ; COMMAND, SO EXECUTE THE  
4180 ; PROGRAM NORMALLY  
030C 3A1A0C 4185 LD A,(CONFLG)  
030F B7 4190 OR A  
0310 2813 4195 JR Z,TR1  
4200 ; STORE BREAKPOINT BYTE  
4205 ; AND INSERT RESTART  
0312 2A150C 4210 LD HL,(BRKADDR)  
0315 7E 4215 LD A,(HL)  
0316 32170C 4220 LD (BRKVAL),A  
0319 36E7 4225 LD (HL),#E7  
4230 ; SET CONFLG TO 0, TO  
4235 ; SUPPORT NMI BUTTON  
031B AF 4240 XOR A  
031C 321A0C 4245 LD (CONFLG),A  
031F 00 4250 DEFB 0,0; SPARE  
4255 ; EXECUTE PROGRAM NORMALLY  
0321 E1 4260 POP HL  
0322 F1 4265 POP AF  
0323 ED45 4270 RETN  
4275 ;  
4280 ; STORE USER REGISTERS

0325 D5	4285 TR1	PUSH DE
0326 C5	4295	PUSH BC
	4300 ;	STACK HAS: PC AF HL DE BC
	4305 ;	SET HL TO USER SP
0327 210000	4310	LD HL,0
032A 39	4315	ADD HL,SP
	4320 ;	COPY USER REGS FROM USER
	4325 ;	STACK TO REG SAVE AREA
	4330 ;	ALSO SET MONITOR SP
032B 11330C	4335	LD DE,STACK
032E 31330C	4340	LD SP,STACK
0331 010000	4345	LD BC,0
0334 EDB9	4350	LDIR
	4355 ;	SET DE TO PC ON USER STACK
0336 5E	4360	LD E,(HL)
0337 23	4365	INC HL
0338 5A	4370	LD D,(HL)
0339 23	4375	INC HL
033A 00	4380	DEFB 0; SPARE
	4385 ;	STORE USER PC
033B ED533B9C	4390	LD (RPC),DE
	4395 ;	STORE USER SP
033F 223D0C	4400	LD (RSP),HL
	4405 ;	
	4410 ;	ENSURE CURSOR AT BOTTOM
	4415 ;	LEFT, SET HL TO END OF
	4420 ;	REG SAVE AREA
0342 CDA505	4425	CALL REGX
	4430 ;	PRINT REGISTERS
	4435 ;	SP PC AF HL DE BC
0345 0606	4440	LD B,6
0347 2B	4445 REGS1	DEC HL
0348 7E	4450	LD A,(HL)
0349 CD4402	4455	CALL B2HEX
034C 2B	4460	DEC HL
034D 7E	4465	LD A,(HL)
034E CD4402	4470	CALL B2HEX
0351 CD3C02	4475	CALL SPACE
0354 10F1	4480	DJNZ REGS1
	4485 ;	DISPLAY I IX IY F
0356 CDD906	4490	CALL ERI
0359 00	4495	DEFB 0,0,0; SPARE
	4500 ;	
	4505 ;	
	4510 ;	RESTORE BREAKPOINT BYTE
035C 2A150C	4515 STRT0	LD HL,(BRKADR)
035F 3A170C	4520	LD A,(BRKVAL)
0362 77	4525	LD (HL),A
	4530 ;	
	4535 ;	COME HERE BEFORE STARTING

4540 ; MAIN MONITOR LOOP  
4545 ; SET CONFLG TO 0, IN CASE  
4550 ; NMI WITHIN MONITOR  
0363 AF 4555 STRTX XOR A  
0364 321A9C 4560 LD (CONFLG),A  
4565 ; ENSURE CURSOR IS AT  
4570 ; BOTTOM LEFT  
0367 CDA505 4575 CALL REGX  
036A C38602 4580 JP PARSE  
4585 ;  
4590 ;  
4595 ; COME HERE FROM MRET  
4600 ; RESET USER STACK POINTER  
036D 2A2801 4605 STRTC LD HL,(ISP)  
0379 223D0C 4610 LD (RSP),HL  
0373 18E7 4615 JR STRT0  
4620 ;  
4625 ;  
4630 ; ROUTINE TO DISPLAY AND  
4635 ; OUTPUT CHAR (AF MODIFIED)  
0375 CD3B01 4640 SRLDIS CALL CRT  
0378 C35E09 4645 JP SRlx  
037B 00 4650 DEFB 0; SPARE  
4655 ;  
4660 ;  
4665 ; LOAD COMMAND  
4670 ; SWITCH ON LED  
037C CD5100 4675 LOAD CALL MOTFLP  
4680 ; CURSOR HOME  
037F EF 4685 LOD1 RST PRS  
0380 1C 4690 DEFB CH,0  
0382 00 4695 DEFB 0; SPARE  
4700 ; GET INPUT CHARACTER  
0383 CD3E09 4705 LOD1B CALL CHIN  
4710 ; IF "." THEN END SO LOD1C  
0386 FE2E 4715 CP ".  
0388 2809 4720 JR Z,LOD1C  
4725 ; IF CR THEN END OF LINE,  
4730 ; SO LOD1A TO PROCESS IT  
038A FE1F 4735 CP CR  
038C 280C 4740 JR Z,LOD1A  
4745 ; DISPLAY IT UNLESS  
4750 ; LT #1F OR GT #9E  
038E F44A0C 4755 CALL P,\$CRT  
0391 18F0 4760 JR LOD1B  
4765 ;  
4770 ; END, SO OUTPUT "." AND GO  
4775 ; TO TURN OFF LED  
0393 EF 4780 LOD1C RST PRS  
0394 1C 4785 DEFB CH,".",CR,0  
0398 1863 4790 JR EFLOP

4795 ;  
4800 ; POINT DE TO START  
037A 118A0B 4805 LODIA LD DE,VL15  
4810 ; GET ADDRESS  
039D CD5A02 4815 CALL NEXNUM  
4820 ; IF INVALID, LOD2C  
03A0 7E 4825 LD A,(HL)  
03A1 B7 4830 OR A  
03A2 2828 4835 JR Z,LOD2C  
4840 ; PUT ADDRESS INTO CHECKSUM  
03A4 2A130C 4845 LD HL,(NUMV)  
03A7 7D 4850 LD A,L  
03AB 84 4855 ADD A,H  
03A9 4F 4860 LD C,A  
4865 ; PUSH ADDRESS TO LOAD AT  
03AA E5 4870 PUSH HL  
4875 ; GET 8 BYTES INTO BUFFER  
4880 ; AND ADD INTO CHECKSUM  
03AB 21000B 4885 LD HL,VRAM  
03AE 44 4890 LD B,H; \*\*\* H=8 \*\*\*  
03AF E5 4895 PUSH HL  
03B0 E5 4900 LOD2 PUSH HL  
03B1 CD5A02 4905 CALL NEXNUM  
03B4 23 4910 INC HL  
4915 ; HL=NUMV  
03B5 7E 4920 LD A,(HL)  
03B6 E1 4925 POP HL  
03B7 77 4930 LD (HL),A  
03B8 23 4935 INC HL  
03B9 81 4940 ADD A,C  
03BA 4F 4945 LD C,A  
03BB 10F3 4950 DJNZ LOD2  
4955 ; GET CHECKSUM  
4960 ; IF WRONG, LOD2C  
03BD CD5A02 4965 CALL NEXNUM  
03C0 23 4970 INC HL  
03C1 7E 4975 LD A,(HL)  
03C2 B9 4980 CP C  
03C3 E1 4985 POP HL  
03C4 D1 4990 POP DE  
03C5 2005 4995 JR NZ,LOD2C  
5000 ; COPY FROM BUFFER  
03C7 4C 5005 LD C,H; \*\*\* H=8 \*\*\*  
03C8 EDB0 5010 LDIR  
03CA 18B3 5015 JR LOD1  
5020 ; BAD DATA, SO SCROLL UP  
03CC CD4002 5025 LOD2C CALL CRLF  
03CF 18AE 5030 JR LOD1  
5035 ;  
5040 ;  
5045 ; DUMP COMMAND

5050 ; TURN ON LED  
0301 CD5100 5055 DUMP CALL MOTFLP  
5060 ; WAIT FOR 2 SECS  
0304 AF 5065 XOR A  
0305 47 5070 LD B,A  
0306 FF 5075 RST RDEL  
0307 10FD 5080 DJNZ D4  
5085 ; SAVE ADDRESS OF CRT ROUTINE  
0309 2A4B0C 5090 LD HL,(\$CRT+1)  
030C E5 5095 PUSH HL  
5100 ; PUT IN SRLDIS INSTEAD  
030D 217503 5105 LD HL,SRLDIS  
03E0 224B0C 5110 LD (\$CRT+1),HL  
5115 ; OUTPUT INITIAL CR  
03E3 CD4002 5120 CALL CRLF  
5125 ; OUTPUT THE DATA  
03E6 CBFC01 5130 CALL TABCDE  
5135 ; RESTORE CRT ADDRESS  
03E9 E1 5140 POP HL  
03EA 224B0C 5145 LD (\$CRT+1),HL  
5150 ; GO TO TURN OFF LED  
03ED 100E 5155 JR EFLOP  
5160 ;  
5165 ;  
5170 ; COPY COMMAND  
03EF 2A0C0C 5175 COPY LD HL,(ARG1)  
03F2 ED5B0E0C 5180 LD DE,(ARG2)  
03F6 ED4B100C 5185 LD BC,(ARG3)  
03FA EDB0 5190 COP5 LDIR  
03FC C9 5195 RET  
5200 ;  
5205 ;  
5210 ; TURN OFF DRIVE LED  
03FD C35100 5215 EFLOP JP MOTFLP  
5220 ;  
5225 ;  
5230 ; WRITE COMMAND  
5235 ; TURN ON LED  
0400 CD5100 5240 WRITE CALL MOTFLP  
5245 ; WAIT FOR 2 SECS  
0403 AF 5250 XOR A  
0404 47 5255 LD B,A  
0405 FF 5260 RST RDEL  
0406 10FD 5265 DJNZ W2  
5270 ; CALCULATE LENGTH-1  
0408 2A0C0C 5275 LD HL,(ARG1)  
040B ED5B0E0C 5280 W4 LD DE,(ARG2)  
040F EB 5285 EX DE,HL  
0410 37 5290 SCF  
0411 ED52 5295 SBC HL,DE  
5300 ; IF LEN-1 IS NEG, TURN OFF

5395 ; LED AND RETURN  
0413 DA5100 5310 JP C,MOTFLP  
0416 EB 5315 EX DE,HL  
5320 ; HL = START  
5325 ; DE = LENGTH-1  
5330 ; WAIT FOR 7.5NS  
0417 AF 5335 XOR A  
0418 FF 5340 RST RDEL  
0419 00 5345 DEF0 9; SPARE  
5350 ; OUTPUT 00 FF FF FF FF  
041A 0605 5355 LD B,5  
041C AF 5360 XOR A  
041D CD5E00 5365 W5 CALL SRLX  
0420 3EFF 5370 LD A,#FF  
0422 10F9 5375 DJNZ W5  
5380 ; IF BLOCK 0, SET LEN TO E+1  
0424 AF 5385 XOR A  
0425 BA 5390 CP D  
0426 2902 5395 JR NZ,W6  
0428 43 5400 LD B,E  
0429 04 5405 INC B  
5410 ; SET E TO LENGTH  
042A 58 5415 W6 LD E,B  
5420 ; OUTPUT START ADDRESS  
042B 70 5425 LD A,L  
042C CD5E00 5430 CALL SRLX  
042F 7C 5435 LD A,H  
0430 CD5E00 5440 CALL SRLX  
5445 ; OUTPUT LENGTH OF DATA  
0433 7B 5450 LD A,E  
0434 CD5E00 5455 CALL SRLX  
5460 ; OUTPUT BLOCK NUMBER  
0437 7A 5465 LD A,D  
0438 CD5E00 5470 CALL SRLX  
5475 ; NOW DISPLAY ALL THIS  
5480 ; AND OUTPUT HEADER CHECKSUM  
043B 0E00 5485 LD C,0  
043D CD5804 5490 CALL TX1  
0440 79 5495 LD A,C  
0441 CD5E00 5500 CALL SRLX  
5505 ; OUTPUT THE BLOCK  
0444 CDCC06 5510 CALL SOUT  
5515 ; OUTPUT CHECKSUM AND NULLS  
0447 060B 5520 LD B,11  
0449 79 5525 LD A,C  
044A CD5E00 5530 W9 CALL SRLX  
044D AF 5535 XOR A  
044E 10FA 5540 DJNZ W9  
5545 ; CRLF (READ HAS SAME TIMING)  
0450 CD4002 5550 CALL CRLF  
5555 ; NEXT BLOCK

0453 18B6 5560 JR W4  
5565 ;  
5570 ;  
5575 ; COMMANDS OUTPUT BY GENERATE  
0455 1F 5580 GDS DEFB CR,"E,"\$,CR,"R,CR  
0600 5585 GDSE DEFS \$  
5590 ;  
5595 ;  
5600 ; ROUTINE USED BY READ/WRITE  
5605 ; RESET CRT ROUTINE ADDRESS  
045B CDC697 5610 TX1 CALL FASC2  
5615 ; DISPLAY HL AND DE, AND  
5620 ; ADD INTO CHECKSUM.  
045E CD6104 5625 CALL TX2  
0461 CD3202 5630 TX2 CALL TBCD3  
0464 EB 5635 EX DE,HL  
0465 C9 5640 RET  
5645 ;  
5650 ;  
5655 ; TABLE SEARCH ROUTINE  
5660 ; HL = START OF TABLE  
5665 ; A = VALUE TO FIND  
0466 D5 5670 TABLE PUSH DE  
0467 5F 5675 LD E,A  
0468 7E 5680 TB1 LD A,(HL)  
0469 23 5685 INC HL  
5690 ; JUMP OUT IF END  
046A B7 5695 OR A  
046B 2807 5700 JR Z,TB3  
5705 ; JUMP OUT IF MATCH  
046D BB 5710 CP E  
046E 2804 5715 JR Z,TB3  
0470 23 5720 INC HL  
0471 23 5725 INC HL  
0472 18F4 5730 JR TB1  
5735 ; PUT RESULT INTO HL  
0474 5E 5740 TB3 LD E,(HL)  
0475 23 5745 INC HL  
0476 56 5750 LD D,(HL)  
0477 EB 5755 EX DE,HL  
0478 D1 5760 POP DE  
0479 C9 5765 RET  
5770 ;  
5775 ;  
5780 ; EXTERNAL (X) COMMAND  
5785 ; CHANGE KBD AND CRT  
047A 21CF07 5790 XP LD HL,XKBD  
047D 224E0C 5795 LD (\$KBD+1),HL  
0480 21BA04 5800 LD HL,XCRT  
0483 224B0C 5805 LD (\$CRT+1),HL  
5810 ; STORE X OPTIONS

0486 3A9C9C 5815 LD A,(ARG1)  
0489 32420C 5820 LD (\$XOPT),A  
049C C9 5825 RET  
5830 ;  
5835 ;  
5840 ; EXTENSION OF XKBD  
5845 ; IF ESCAPE OR NOW NULL,  
5850 ; ASSUME PROGRAM WILLE NOT  
5855 ; OUTPUT THE CHAR  
048D 2805 5860 XKBD JR Z,XK4  
048F B7 5865 OR A  
0490 2802 5870 JR Z,XK4  
0492 CBFE 5875 SET 7,(HL)  
0494 E1 5880 XK4 POP HL  
0495 37 5885 SCF  
0496 C9 5890 RET  
0497 00 5895 DEFB 0; SPARE  
5900 ;  
5905 ;  
5910 ; ? COMMAND  
5915 ; DISPLAY LIST OF COMMANDS  
049B 2A450C 5920 QP LD HL,(\$CTAB)  
049B 7E 5925 QP2 LD A,(HL)  
049C B7 5930 OR A  
049D CA4002 5935 JP Z,CRLF  
04A0 F7 5940 RST RCRT  
04A1 CD3C02 5945 CALL SPACE  
04A4 23 5950 INC HL  
04A5 23 5955 INC HL  
04A6 23 5960 INC HL  
04A7 18F2 5965 JR QP2  
5970 ;  
5975 ;  
5980 ; TABLE FOR NASCOM TO ASCII  
5985 ; AND ASCII TO NASCOM  
04A9 1F 5990 ATAB DEFB CR,ACR,CR  
04AC 1E 5995 DEFB CS,ESC,CS  
04AF 1D 6000 DEFB BS,ABS,BG  
04B2 1C 6005 DEFB CH,LF,0  
04B5 7F 6010 DEFB DEL,DEL,0  
04B8 00 6015 DEFB 0,0  
6020 ;  
6025 ;  
6030 ; X VERSION OF CRT ROUTINE  
04BA CD3B01 6035 XCRT CALL CRT  
04BD F5 6040 PUSH AF  
6045 ; CONVERT TO ASCII  
04BE CD0205 6050 CALL AK1  
04C1 E5 6055 PUSH HL  
6060 ; OUTPUT UNLESS BIT 7 SET  
6065 ; TO SUPPRESS SERIAL OUTPUT

04C2 21420C 6070 LD HL,\$XOPT  
04C5 CB7E 6075 BIT 7,(HL)  
04C7 CCCF04 6080 CALL Z,XSOP  
6085 ; TURN OFF SUPPRESSION  
04CA CBDE 6090 RES 7,(HL)  
04CC E1 6095 POP HL  
04CD F1 6100 POP AF  
04CE C9 6105 RET  
6110 ;  
6115 ;  
6120 ; X SERIAL OUTPUT ROUTINE  
6125 ; OUTPUT THE CHARACTER  
04CF C0ED07 6130 XSOP CALL XSOP0  
6135 ; IF IT WAS A CR AND BIT 4  
6140 ; OF \$XOPT = 0, OUTPUT LF  
04D2 FE0D 6145 CP ACR  
04D4 C0 6150 RET NZ  
04D5 CB66 6155 BIT 4,(HL)  
04D7 C0 6160 RET NZ  
04D8 3E6A 6165 LD A,LF  
04DA C3ED07 6170 JP XSOP0  
6175 ;  
6180 ;  
6185 ; KBD CONTROL KEYS  
04DD 21010C 6190 KBDC LD HL,KMAP  
6195 ; IF NOT 0, MAY MODIFY  
04E0 FE49 6200 CP "0  
04E2 2007 6205 JR NZ,K30  
6210 ; IF SHIFT DOWN, NORMAL,  
6215 ; OTHERWISE IGNORE  
04E4 CB66 6220 BIT 4,(HL)  
04E6 C0 6225 RET NZ  
6230 ; DISCARD RETURN, IGNORE  
04E7 F1 6235 POP AF  
04E8 C38900 6240 JP KSCB  
6245 ; IF 0 DOWN, MODIFY  
04EB CB6E 6250 K30 BIT 5,(HL)  
04ED 2802 6255 JR Z,K35  
04EF EE40 6260 XOR #40  
04F1 C9 6265 K35 RET  
6270 ;  
6275 ;  
6280 ; CHECK FOR INPUT  
6285 ; CHECK KEYBOARD  
04F2 CD6900 6290 TIN CALL KBD  
04F5 D8 6295 RET C  
6300 ; CHECK SERIAL IN  
04F6 DB02 6305 SRLIN IN A,(2)  
04F8 17 6310 RLA  
04F9 D0 6315 RET NC  
04FA DB01 6320 IN A,(1)

04FC 37 6325 SCF  
04FD C9 6339 RET  
6335 ;  
6340 ;  
6345 ; INPUT FOR ASCII PROGRAM  
04FE CDF204 6350 AKBD CALL TIN  
0501 D0 6355 RET NC  
6360 ; CONVERT TO ASCII  
0502 E5 6365 AK1 PUSH HL  
0503 21A904 6370 LD HL,ATAB  
0506 C39A07 6375 JP AK2  
0509 00 6380 DEFB 0; SPARE  
6385 ;  
6390 ;  
6395 ; Z COMMAND  
6400 ; CHANGE COMMAND TABLE  
050A 2A0C0C 6405 ZP LD HL,(ARG1)  
050D 22450C 6410 LD (\$CTAB),HL  
0510 C9 6415 RET  
6420 ;  
6425 ; DUMMY COMMAND TABLE  
0511 00 6430 ZTAB DEFB 0  
0512 3D00 6435 DEFW EDEL  
6440 ;  
6445 ;  
6450 ; ICOPY COMMAND  
6455 ; GET VALUES  
0514 CD9706 6460 ICOPY CALL ARGS  
6465 ; IF ARG1 GE ARG2, GO TO  
6470 ; LDIR COPY  
0517 B7 6475 OR A  
0518 EB52 6480 SBC HL,DE  
051A 19 6485 ADD HL,DE  
051B D2FA03 6490 JP NC,COPS  
6495 ; SET TO END NOT START  
051E 0B 6500 DEC BC  
051F EB 6505 EX DE,HL  
0520 09 6510 ADD HL,BC  
0521 EB 6515 EX DE,HL  
0522 09 6520 ADD HL,BC  
0523 03 6525 INC BC  
0524 EDB8 6530 LDDE  
0526 C9 6535 RET  
6540 ;  
6545 ;  
6550 ; ARITHMETIC COMMAND  
6555 ; GET VALUES  
0527 CD9806 6560 ARITH CALL ARGS2  
052A EB 6565 EX DE,HL  
052B E5 6570 PUSH HL  
6575 ; SUM

052C 19 6580 ADD HL,DE  
052D CD3202 6585 CALL TBCD3  
6590 ; DIFFERENCE  
0530 E1 6595 PUP HL  
0531 B7 6600 OR A  
0532 ED52 6605 SBC HL,DE  
0534 CD3202 6610 CALL TBCD3  
6615 ; OFFSET  
0537 2B 6620 DEC HL  
0538 2B 6625 DEC HL  
0539 7C 6630 LD A,H  
053A FEFF 6635 CP #FF  
053C 200A 6640 JR NZ,A2  
053E CB7D 6645 BIT 7,L  
0540 200D 6650 JR NZ,AOK  
6655 ; NO GOOD SO ??  
0542 EF 6660 ANG RST PRS  
0543 3F 6665 DEFB "?,"?",CR,  
0547 C9 6670 RET  
0548 B7 6675 A2 OR A  
0549 20F7 6680 JR NZ,ANG  
054B CB7D 6685 BIT 7,L  
054D 20F3 6690 JR NZ,ANG  
6695 ; DISPLAY OFFSET  
054F 7D 6700 ADK LD A,L  
0550 CD4402 6705 CALL B2HEX  
0553 C34002 6710 JP CRLF  
0556 00 6715 DEFB 0; SPARE  
6720 ;  
6725 ;  
6730 ; MONITOR INITIALISATION  
6735 ; RESTORE BREAKPOINT BYTE  
0557 2A150C 6740 STARTB LD HL,(BRKADR)  
055A 3A170C 6745 LD A,(BRKVAL)  
055D 77 6750 LD (HL),A  
6755 ; SET WORKSPACE TO 0  
055E 21000C 6760 LD HL,RAMZ  
0561 0618 6765 LD B,RAME-RAMZ  
0563 3600 6770 STA LD (HL),0  
0565 23 6775 INC HL  
0566 10FB 6780 DJNZ STA  
6785 ; SET WORKSPACE FROM TABLE  
0568 212801 6790 LD HL,INITT  
056B 11300C 6795 LD DE,INITR  
056E 011300 6800 LD BC,INITE-INITR  
0571 EDB0 6805 LDIR  
6810 ; INITIALISE SCREEN  
0573 EF 6815 RST PRS  
0574 1E 6820 DEFB C\$  
0575 4E 6825 DEFM /NASSBUG 4/  
057D 00 6830 DEFB 0

057E C36393 6935 JP STRX  
6840 ;  
6845 ;  
6850 ; STORE BREAKPOINT BYTE  
6855 ; BEFORE EVERY INPUT LINE  
0581 E5 6860 BRKST PUSH HL  
0582 2A150C 6865 LD HL,(BRKADR)  
0585 7E 6870 LD A,(HL)  
0586 32170C 6875 LD (BRKVAL),A  
0589 E1 6880 POP HL  
058A C3DB91 6885 JP INLINE  
6890 ;  
6895 ;  
6900 ; EXTENSION TO MODIFY  
6905 ; INCREMENT IF NO DATA INPUT  
058D 2001 6910 MOD4B JR NZ,MOD5  
058F 23 6915 INC HL  
6920 ; IF LAST CHAR WAS ":" THEN  
6925 ; GO BACK ONE BYTE  
0590 1A 6930 MOD5 LD A,(DE)  
0591 FE3A 6935 CP ":"  
0593 2002 6940 JR NZ,MOD7  
0595 2B 6945 DEC HL  
0596 2B 6950 DEC HL  
6955 ; IF IT WAS "/" SET TO VALUE  
0597 FE2F 6960 MOD7 CP "/"  
0599 2007 6965 JR NZ,MOD8  
059B 13 6970 INC DE  
059C CD5A92 6975 CALL NEXNUM  
059F 2A130C 6980 LD HL,(NUMV)  
05A2 C3B001 6985 MOD8 JP MOD1  
6990 ;  
6995 ;  
7000 ; ENSURE CURSOR IS AT BOTTOM  
7005 ; LEFT. (SET HL TO RSAE)  
05A5 2A180C 7010 REGX LD HL,(CURSOR)  
05AB 118A0B 7015 LD DE,VL15  
05AB B7 7020 OR A  
05AC ED52 7025 SBC HL,DE  
05AE 213F0C 7030 LD HL,RSAE  
05B1 C24002 7035 JP NZ,CRLF  
05B4 C9 7040 RET  
7045 ;  
7050 ;  
7055 ; RELATIVE CALL RESTART  
05B5 2B 7060 RCALB DEC HL  
05B6 3B 7065 DEC SP  
05B7 3B 7070 DEC SP  
05B8 F5 7075 PUSH AF  
05B9 D5 7080 PUSH DE  
05BA 5E 7085 LD E,(HL)

7090 ; E = OFFSET, SET D

05BB 7B	7095	LD A,E
05BC 17	7102	RLA
05BD 9F	7105	SBC A,A
05BE 57	7110	LD D,A
05BF 23	7115	INC HL
05C0 180D	7120	JR UCAL4
	7125 ;	
	7130 ;	
	7135 ;	USER CALL RESTART
05C2 2B	7140 UCAL8 DEC HL	
05C3 3B	7145 DEC SP	
05C4 3B	7150 DEC SP	
05C5 F5	7155 PUSH AF	
05C6 D5	7160 PUSH DE	
05C7 5E	7165 LD E,(HL)	
05C8 1609	7170 LD D,0	
05CA 21000E	7175 LD HL,URSTB	
05CD 19	7180 ADD HL,DE	
05CE 19	7185 ADD HL,DE	
05CF 19	7190 UCAL4 ADD HL,DE	
05D0 D1	7195 POP DE	
05D1 F1	7200 POP AF	
05D2 E3	7205 EX (SP),HL	
	7210 ; FAKE JUMP TO ROUTINE	
05D3 C9	7215 RET	
	7220 ;	
	7225 ;	
	7230 ; KEYBOARD TABLE	
05D4 FF	7235 KTAB DEFB #FF,#FF,#FF,#FF; #00	
05D8 FF	7240 DEFB #FF,#FF,#FF,#FF; #04	
05DC FF	7245 DEFB #FF,#FF,#FF,#FF; #08	
05E0 FF	7250 DEFB #FF,#FF,#FF,#FF; #0C	
05E4 FF	7255 DEFB #FF,#FF,#FF,#FF; #10	
05E8 FF	7260 DEFB #FF,#FF,#FF,#FF; #14	
05EC FF	7265 DEFB #FF,#FF,#FF,#FF; #18	
05F0 89	7270 DEFB #89,#08,#88,#09; #1C	
05F4 14	7275 DEFB #14,#9C,#9B,#A3; #20 "#	
05F8 92	7280 DEFB #92,#C2,#BA,#B2; #24 \$%&	
05FC AA	7285 DEFB #AA,#A2,#98,#A0; #28 ()*+	
0600 29	7290 DEFB #29,#0A,#21,#19; #2C ,-. /	
0604 1A	7295 DEFB #1A,#1C,#1B,#23; #30 0123	
0608 12	7300 DEFB #12,#42,#3A,#32; #34 4567	
060C 2A	7305 DEFB #2A,#22,#18,#20; #38 89: ;	
0610 A9	7310 DEFB #A9,#8A,#A1,#99; #3C < ?	
0614 0D	7315 DEFB #0D,#2C,#41,#13; #40 0ABC	
0618 3B	7320 DEFB #3B,#33,#43,#10; #44 DEFG	
061C 49	7325 DEFB #49,#20,#38,#30; #48 HIJK	
0620 28	7330 DEFB #28,#31,#39,#25; #4C LMNO	
0624 1D	7335 DEFB #1D,#24,#15,#34; #50 PQRS	
0628 45	7340 DEFB #45,#35,#11,#29; #54 TUWV	

062C 44 7345 DEFB #44,#30,#3C,#FF; #58 XYZC  
0630 FF 7350 DEFB #FF,#FF,#9A,#FF; #5C \J†  
0000 7355 KTABE DEFS 0  
7360 ;  
7365 ;  
7370 ; KEYBOARD COMMAND  
7375 ; STORE KEYBOARD OPTIONS  
0634 3A9C9C 7380 KOP LD A,(ARG1)  
0637 32410C 7385 LD (%KOPT),A  
063A C9 7390 RET  
7395 ;  
7400 ;  
7405 ; BREAKPOINT COMMAND  
7410 ; STORE BREAKPOINT ADDRESS  
063B 2A9C9C 7415 BREAK LD HL,(ARG1)  
063E 22159C 7420 LD (BRKADR),HL  
0641 C9 7425 RET  
7430 ;  
7435 ;  
7440 ; EXTENSION TO GENERATE  
7445 ; OUTPUT EXECUTION ADDRESS  
0642 CD3202 7450 GB CALL T8CD3  
7455 ; RESET CRT ADDRESS  
0645 CDC697 7460 CALL FASC2  
0648 3E1F 7465 LD A,CR  
7470 ; OUTPUT FINAL CR  
064A C35D00 7475 JP SRLOUT  
064D 00 7480 DEFB 0,0,0,0; SPARE  
7485 ;  
7490 ;  
7495 ; EXTENSION TO READ  
7500 ; LOAD THE DATA  
0651 0E00 7505 READB LD C,0  
0653 CD3E00 7510 R4 CALL CHIN  
0656 77 7515 LD (HL),A  
0657 81 7520 ADD A,C  
0658 4F 7525 LD C,A  
0659 23 7530 INC HL  
065A 19F7 7535 DJNZ R4  
7540 ; CHECK AGAINST CHECKSUM  
065C CD3E00 7545 CALL CHIN  
065F B9 7550 CP C  
0660 280A 7555 JR Z,R7  
7560 ; ERROR FOUND  
0662 EF 7565 R6 RST PRS  
0663 45 7570 DEFN /Error/  
0668 1F 7575 DEFB CR,0  
066A 1808 7580 JR R8  
7585 ; CR, TEST FOR END  
066C CD4002 7590 R7 CALL CRLF  
066F AF 7595 XOR A

0670 BA 7699 CP D  
0671 CA5100 7695 JP Z,NOTFLP  
7610 ; GO FOR NEXT BLOCK  
0674 C30F07 7615 R8 JP R1  
7620 ;  
7625 ;  
7630 ; OUTPUT COMMAND  
0677 ED4B0C9C 7635 0 LD BC,(ARG1)  
0678 3A0E0C 7640 LD A,(ARG2)  
067E ED79 7645 OUT (C),A  
0680 1806 7650 JR ROP  
7655 ;  
7660 ; QUERY COMMAND  
0682 ED4B0C0C 7665 0 LD BC,(ARG1)  
0686 ED78 7670 IN A,(C)  
7675 ;  
7680 ; DISPLAY PORT AND VALUE  
0688 F5 7685 ROP PUSH AF  
0689 79 7690 LD A,C  
068A CD4402 7695 CALL B2HEX  
068B CD3C02 7700 CALL SPACE  
0690 F1 7705 POP AF  
0691 CD4402 7710 CALL B2HEX  
0694 C34002 7715 JP CRLF  
7720 ;  
7725 ;  
7730 ; GET ARGUMENTS  
0697 ED4B100C 7735 ARGS LD BC,(ARG3)  
0698 ED580E0C 7740 ARGS2 LD DE,(ARG2)  
069F 2A0C9C 7745 LD HL,(ARG1)  
06A2 C9 7750 RET  
7755 ;  
7760 ;  
7765 ; GENERATE COMMAND  
7770 ; OUTPUT COMMANDS  
06A3 215504 7775 0 LD HL,GDS  
06A6 0606 7780 LD B,GDSE-GDS  
06A8 CDC602 7785 CALL FASC2  
06AB 7E 7790 G2 LD A,(HL)  
06AC CD5D09 7795 CALL SRLOUT  
7800 ; WAIT 22MS  
06AF AF 7805 XOR A  
0CB0 FF 7810 RST RDEL  
06B1 FF 7815 RST RDEL  
06B2 FF 7820 RST RDEL  
06B3 23 7825 INC HL  
06B4 10F5 7830 DJNZ G2  
7835 ; OUTPUT THE DATA  
06B6 CD0004 7840 CALL WRITE  
7845 ; WAIT 7.5MS  
06B9 AF 7850 XOR A

06BA FF 7855 RST RDEL  
06BB 3E45 7860 ; OUTPUT "E"  
06BD C05D06 7865 LD A,"E"  
CALL SRLOUT  
7870 ; SET CRT TO OUTPUT AS WELL  
06C0 217503 7880 LD HL,SRLDIS  
06C3 224B0C 7885 LD (\$CRT+1),HL  
7890 ; OUTPUT EXECUTE ADDRESS  
06C6 2A100C 7895 LD HL,(ARG3)  
06C9 C34206 7900 JP GB  
7905 ;  
7910 ;  
7915 ; STRING TO SERIAL OUTPUT  
7920 ; HL = ADDRESS  
7925 ; B = LENGTH  
7930 ; C = CHECKSUM  
06CC 0E00 7935 SOUT LD C,0  
06CE 7E 7940 S01 LD A,(HL)  
06CF 81 7945 ADD A,C  
06D0 4F 7950 LD C,A  
06D1 7E 7955 LD A,(HL)  
06D2 CD5E06 7960 CALL SRlx  
06D5 23 7965 INC HL  
06D6 10F6 7970 DJNZ S01  
06D8 C9 7975 RET  
7980 ;  
7985 ;  
7990 ; EXTENDED REGISTER DISPLAY  
7995 ; I REG  
06D9 ED57 8000 ER1 LD A,I  
06DB CD4402 8005 CALL B2HEX  
06DE CD3C02 8010 CALL SPACE  
8015 ; IX IY REGS  
06E1 DDE5 8020 PUSH IX  
06E3 E1 8025 POP HL  
06E4 CD3202 8030 CALL TBCD3  
06E7 FDE5 8035 PUSH IY  
06E9 E1 8040 POP HL  
06EA CD3202 8045 CALL TBCD3  
8050 ; F REG  
06EB 3A390C 8055 LD A,(RAF)  
06F0 11FF06 8060 LD DE,ESTR-1  
06F3 0608 8065 LD B,8  
06F5 13 8070 ER4 INC DE  
06F6 17 8075 RLA  
06F7 F5 8080 PUSH AF  
06F8 1A 8085 LD A,(DE)  
06F9 DC4A0C 8090 CALL C,\$CRT  
06FC F1 8095 POP AF  
06FD 10F6 8100 DJNZ ERA  
06FF C9 8105 RET

8110 ; STRING FOR FLAGS  
0700 53 8115 ESTR DEF8 "S,"Z,0,"H  
0704 00 8120 DEF8 0,"P,"N,"C  
0708 00 8125 DEF8 0,0,0,0; SPARE  
8130 ;  
8135 ;  
8140 ; READ ROUTINE  
8145 ; TURN ON LED  
070C CD5100 8150 READ CALL MOTFLP  
8155 ; LOOK FOR 4 #FF CHARS  
070F CD3E00 8160 R1 CALL CHIN  
0712 FFFF 8165 R1A CP #FF  
0714 2000 8170 JR NZ,R1D  
0716 0603 8175 LD B,3  
0718 CD3E00 8180 R1C CALL CHIN  
071B FFFF 8185 CP #FF  
071D 2004 8190 JR NZ,R1D  
071F 10F7 8195 DJNZ R1C  
0721 1812 8200 JR R3B  
8205 ; LOOK FOR 4 CS CHARS  
0723 FE1E 8210 R1D CP CS  
0725 20E8 8215 JR NZ,R1  
0727 0603 8220 LD B,3  
0729 CD3E00 8225 R1F CALL CHIN  
072C FE1E 8230 CP CS  
072E 20E2 8235 JR NZ,R1A  
0730 10F7 8240 DJNZ R1F  
0732 C35100 8245 JP MOTFLP  
8250 ; GET HEADER DATA  
0735 CD3E00 8255 R3B CALL CHIN  
0738 6F 8260 LD L,A  
0739 CD3E00 8265 CALL CHIN  
073C 67 8270 LD H,A  
073D CD3E00 8275 CALL CHIN  
0740 5F 8280 LD E,A  
0741 CD3E00 8285 CALL CHIN  
0744 57 8290 LD D,A  
8295 ; DISPLAY AND CHECK  
0745 0E00 8298 LD C,0  
0747 CD5804 8305 CALL TX1  
074A CD3E00 8310 CALL CHIN  
074D B9 8315 CP C  
074E C26206 8320 JP NZ,R6  
8325 ; SET B TO LENGTH  
0751 43 8330 LD B,E  
0752 C35100 8335 JP READ8  
8340 ;  
8345 ;  
8350 ; COMMAND TABLE  
0755 41 8355 CTAB DEF8 "A  
0756 2705 8360 DEF8 ARITH

0758 42	8365	DEFB "B
0759 3806	8370	DEFW BREAK
075B 43	8375	DEFB "C
075C EF03	8380	DEFW COPY
075E 44	8385	DEFB "D
075F D103	8390	DEFW DUMP
0761 45	8395	DEFB "E
0762 D002	8400	DEFW EXEC
0764 47	8405	DEFB "G
0765 A306	8410	DEFW G
0767 49	8415	DEFB "I
0768 1405	8420	DEFW ICOPY
076A 4B	8425	DEFB "K
076B 3406	8430	DEFW KOP
076D 4C	8435	DEFB "L
076E 7C03	8440	DEFW LOAD
0770 4D	8445	DEFB "M
0771 AD01	8450	DEFW MODIFY
0773 4E	8455	DEFB "N
0774 C007	8460	DEFW NORMAL
0776 4F	8465	DEFB "O
0777 7206	8470	DEFW O
0779 51	8475	DEFB "Q
077A 8206	8480	DEFW Q
077C 52	8485	DEFB "R
077D 0C07	8490	DEFW READ
077F 53	8495	DEFB "S
0780 FF02	8500	DEFW STEP
0782 54	8505	DEFB "T
0783 FC01	8510	DEFW TABCODE
0785 57	8515	DEFB "U
0786 0004	8520	DEFW WRITE
0788 58	8525	DEFB "X
0789 7A04	8530	DEFW XP
078B 5A	8535	DEFB "Z
078C 0A05	8540	DEFW ZP
078E 3F	8545	DEFB "?"
078F 9804	8550	DEFW OP
0791 00	8555	DEFB Ø
0792 B702	8560	DEFW ERRM
0900	8565	CTABE DEFS Ø
	8570 ;	
	8575 ;	
	8580 ;	TERMINAL PROGRAM
0794 CD3E00	8585 XN	CALL CHIN
0797 F7	8590	RST RCRT
0798 18FA	8595	JR XN
	8600 ;	
	8605 ;	
	8610 ;	CONVERT TO/FROM ASCII
079A F5	8615 AK2	PUSH AF

079B CD6604 8620 CALL TABLE  
079E B7 8625 OR A  
079F 2804 8630 JR Z,AK4  
8635 ; MATCH FOUND  
07A1 F1 8640 POP AF  
07A2 7D 8645 LD A,L  
07A3 E1 8650 POP HL  
07A4 C9 8655 RET  
8660 ; NO MATCH  
07A5 F1 8655 AK4 POP AF  
07A6 E1 8670 POP HL  
07A7 C9 8675 RET  
8680 ;  
8685 ;  
8690 ; CONVERT TO ASCII FOR CRT  
07A8 CDAE07 8695 ACRT CALL AFA  
07AB C33B91 8700 JP CRT  
8705 ;  
8710 ;  
8715 ; CONVERT FROM ASCII  
07AE E5 8720 AFA PUSH HL  
07AF 21AA04 8725 LD HL,ATAB+1  
07B2 18E6 8730 JR AK2  
8735 ;  
8740 ;  
8745 ; CONVERT KBD AND CRT FOR  
8750 ; ASCII PROGRAM  
07B4 21FE04 8755 TASC LD HL,AKBD  
07B7 224E0C 8760 LD (\$KBD+1),HL  
07BA 21A807 8765 TASC2 LD HL,ACRT  
07BD E5 8770 TASC3 PUSH HL  
07BE 180A 8775 JR FASC3  
8780 ;  
8785 ;  
8790 ; MAKE KBD AND CRT NORMAL  
07C0 2A3901 8795 NORMAL LD HL,(IKBD+1)  
07C0 8800 FASC EQU NORMAL  
07C3 224E0C 8805 LD (\$KBD+1),HL  
8810 ;  
07C6 E5 8815 FASC2 PUSH HL  
07C7 2A3601 8820 LD HL,(ICRT+1)  
8825 ;  
07CA 224B0C 8830 FASC3 LD (\$CRT+1),HL  
07CD E1 8835 POP HL  
07CE C9 8840 RET  
8845 ;  
8850 ;  
8855 ; X VERSION OF KBD  
8860 ; RETURN IF KBD INPUT  
07CF CD6900 8865 XKBD CALL KBD  
07D2 D8 8870 RET C

8875 ; RETURN IF NO SERIAL INPUT  
07D3 CDF604 8880 XKBDIC CALL SRLIN  
07D6 D0 8885 RET NC  
8890 ; STRIP PARITY  
07D7 E67F 8895 AND #7F  
07D9 E5 8900 PUSH HL  
07DA F5 8905 PUSH AF  
8910 ; IF FULL DUPLEX, SEND BACK  
07DB 21420C 8915 LD HL,\$XOPT  
07DE CB6E 8920 BIT 5,(HL)  
07E0 CCCF04 8925 CALL Z,XSOP  
07E3 F1 8930 POP AF  
8935 ; CONVERT FROM ASCII  
07E4 CDAE07 8940 CALL AFA  
8945 ; CHECK FOR ESCAPE  
07E7 FE1E 8950 CP CS  
07E9 C38004 8955 JP XKBOB  
07EC 00 8960 DEFB 9; SPARE  
8965 ;  
8970 ;  
8975 ; OUTPUT ASCII CHARACTER  
8980 ; SET PARITY FLAG ETC  
07ED B7 8985 XSOP0 OR A  
8990 ; IGNORE NULLS  
07EE C8 8995 RET Z  
07EF F5 9000 XSOPC PUSH AF  
9005 ; MAKE PARITY EVEN  
07F0 EAF507 9010 JP PE,XSOP2  
07F3 EE80 9015 XOR #80  
9020 ; IF BIT 0 SET, MAKE IT ODD  
07F5 CB46 9025 XSOP2 BIT 0,(HL)  
07F7 2802 9030 JR Z,XSOP4  
07F9 EE80 9035 XOR #80  
9040 ; OUTPUT IT  
07FB CD5E00 9045 XSOP4 CALL SRLX  
9050 ; RESTORE ORIGINAL VALUE  
07FE F1 9055 POP AF  
07FF C9 9060 RET  
9065 ;  
9070 ; END OF LISTING