NAS-SYS 3
advanced 2K MONITOR
For the NASCOM 1 and 2

interface components

CONTENTS

NAS-SYS 3 - IMPROVED VERSION OF NAS-SYS 1

1. All the keys on the keyboard automatically repeat when held down. (Not the @ key.) The initial repeat delay and the repeat speed are adjustable.

2. All routines are interruptable, so that interrupts can be used while executing NAS-SYS routines.

3. The CRT routine allows data to be output anywhere in memory, so headings can be output to the top line of the display.

4. The Read command has an optional parameter which allows cassette tapes to be read into any memory locations.

5. The Tabulate command is enhanced in three ways. Firstly, ASCII values of the bytes are output as well as the usual hexadecimal tabulation. Secondly, a fourth parameter can be used to specify the output of additional values on each line, to allow for printers of different widths. Thirdly, a fifth parameter allows suppression of either the hexadecimal output or the ASCII output.

6. All NAS-SYS routines can be single stepped, which makes it easier to test a program which uses NAS-SYS routines. By using the repeat keyboard feature, high speed single stepping is possible.

7. The register display is enhanced so that it shows the two byte value pointed to by each of the main registers.

8. Output from the Modify command is displaced two characters to the right, to improve readability.

9. The External (X) command has additional options, and no longer fails to output nulls. This enables the NULL command in BASIC to work correctly.

10. There are three new commands. P displays the stored user program registers, D executes a program at #D000, and Y executes a program at #B000.

11. The cursor blink rate is adjustable.

12. There are three new routines. Repeat keyboard scan, Output two spaces, and a new routine which can execute any other routine.

13. Both on breakpoint and on NMI, control passes through the $NMI jump before displaying the registers, allowing a program to take alternative action.

14. The B 0 command turns off the breakpoint completely, so that with appropriate hardware NAS-SYS can be executed in RAM.

15. Support in NAS-SYS itself for the use of paper tape has been removed, so there is no longer a Load command and the Tabulate command does not output a checksum.

NOTE: Because of improvements to the Read command, ARGN must be set to 0 before calling the Read routine from a program. Therefore when using the Nascom Tape Basic, you MUST enter the command POKE 3083,0 after each cold or warm start.

NASCOM OPERATING SYSTEM - NAS-SYS 3

NAS-SYS 3 is a 2K operating system for the Nascom 1 and Nascom 2 microcomputers. It makes it easy to enter, test and run machine code programs. It also provides a comprehensive set of routines which can be called by user programs. These routines are also used by the high level languages on the Nascom such as the 8K BASIC.

SUMMARY OF FEATURES

The contents of memory can be tabulated in hexadecimal and ASCII.

Memory locations can be modified with values entered in hexadecimal or ASCII.

Programs can be executed normally, or single stepped, or executed with a breakpoint set. During single stepping or at a breakpoint a full display of the machine registers is provided.

Commands are entered and edited using a blinking, non-destructive cursor. Comprehensive screen editing includes insertion and deletion of characters. This makes it easy to correct and reenter incorrect commands.

The keyboard routine allows every possible code to be entered when using the expanded keyboard, and provides automatic repeat when a key is held down. It also supports all features when using the original Nascom 1 keyboard.

Cassette tapes can be used to store programs, using a fast and fully checked method of recording the data. Tapes can be written which automatically load and execute machine code programs without any commands being entered at the keyboard.

All parallel I/O ports can be controlled and checked by direct commands.

Full support of serial terminals and printers is included, so that the computer can be controlled by a Teletype or equivalent device.

The computer can be used as a terminal, by connecting it with an acoustic coupler to a timesharing service, or even to another Nascom computer.

All internal codes are in ASCII, making it easy to attach other peripherals such as parallel printers.

(This document was produced on a Nascom computer, using the NASPEN word processing package, and printed on a Teletype 43 terminal.)

LIST OF COMMANDS

A xxxx yyyy	Arithmetic - in hexadecimal
B xxxx	Breakpoint set or cleared
C xxxx yyyy zzzz	Copy - move data
D	Jump to address #D000 - run ZEAP
E xxxx	Execute a program
G xxxx yyyy zzzz	Generate a self loading cassette tape
H	Half duplex terminal
I xxxx yyyy zzzz	Intelligent copy - move data safely
J	Jump to address #FFFA - BASIC cold start
K xx	Keyboard option
M xxxx	Modify or examine contents of memory
N	Normal I/O to be resumed
O xx yy	Output data to port
P	Display program registers
Q xx	Query data from port
R xxxx	Read a cassette tape
S xxxx	Single step
T xxxx yyyy zzzz vv hhaa	Tabulate contents of memory
U	User specified I/O routines activated
V	Verify cassette tape is readable
W xxxx yyyy	Write a cassette tape
X xx	External serial device activated
Y	Jump to address #B000
Z	Jump to address #FFFD - BASIC warm start

(F and L commands do not exist)

Note. All hexadecimal numbers in this manual are preceded by "#".

HOW TO ENTER A COMMAND

Type the command letter in the first position on the line, followed by any values required. Each value must be separated from the others by one or more spaces. Then press the ENTER key. The line entered is ALWAYS the line where the cursor is blinking when the ENTER key is pressed. If the first position on the line is blank, no command is processed.

ERRORS WHEN ENTERING A COMMAND

If an invalid command is entered, the message "Error" is displayed on the next line. Very careful checking is performed by NAS-SYS, so that disastrous mistakes are less likely. The following types of error are detected:-

Invalid command character. (If the command character is blank, the line is simply ignored.)

More than ten values entered after the command character. (Only five values are ever needed, but a program can use up to ten values entered after the E command.)

Any value which is not a valid hexadecimal number. (Digits 0 to 9, and letters A to F.)

A value greater than #FFFF.

HOW TO CORRECT ERRORS

Simply press BACKSPACE to correct errors which you notice while typing the command.

However the comprehensive screen editing facilities allow you to go back and reenter an incorrect line provided it is still on the screen. Simply move the cursor back up the screen, and edit the line using the cursor movement keys, including deleting and inserting characters if required. Then press ENTER to enter the line.

For details of the screen editing facilities, read the next section.

SCREEN EDITING

The screen editing facilities are available with both the expanded keyboard, and the original Nascom 1 keyboard.

NAME	HEX CODE	KEYS PRESSED	FUNCTION
NULL	#00	CTRL/SHIFT/@	Ignored on display.
BS	#08	BACKSPACE	Move back and make position blank.
LF	#0A	LF or CTRL/J	Ignored on display.
FF	#0C	CS or SHIFT/BS	Clear screen and start at top left.
CR	#0D	ENTER or NEWLINE	Carriage return, line feed. Scroll up if at bottom.
CUL	#11	Left arrow or CTRL/Q	Move cursor left.
CUR	#12	Right arrow or CTRL/R	Move cursor right.
CUU	#13	Up arrow or CTRL/S	Move cursor up.
CUD	#14	Down arrow or CTRL/T	Move cursor down.
CSL	#15	SHIFT/Left arrow or CTRL/U	Delete character at cursor, move rest of line to the left.
CSR	#16	SHIFT/Right arrow or CTRL/V	Move rest of line to the right.
CH	#17	CH or CTRL/W	Move cursor to start of line.
CCR	#18	CTRL/X	If cursor at start of line, ignore. Otherwise do CR.
ESC	#1B	ESC or SHIFT/ENTER	Delete current line, and place cursor at start of line.

The cursor blinks and is non-destructive, which means that it can be moved over existing text without changing it. The cursor blink speed can be altered by changing the value KBLINK (#0C32-#0C33) in the NAS-SYS workspace.

THE KEYBOARD

The expanded keyboard has the ability to generate any of the 256 possible 8 bit codes, as follows:-

Hex #20 to #5F are the normal ASCII codes and are available as marked keys.

The letters are upper case, unless SHIFT is held down, when they become lower case. (Also see K command.) The @ key requires SHIFT to be held down, to produce an @.

Both the CTRL and the @ key operate as CTRL keys. This enables original Nascom 1 keyboards to make use of the screen editing facilities. When one of these keys is held down, and another key pressed, bit 6 is altered. This gives codes #00 to #1F and #60 to #7F.

The GRAPHICS key sets bit 7 on while it is held down, and this can be used to give graphic characters directly, with the Nascom 2 graphic option. When used in conjunction with the other keys, codes #80 to #FF may be obtained. (Also see K command.)

All keys (except the @ key) automatically repeat if held down. The delay before repeating starts and the repeat speed can be altered by changing the values KLONG (#0C2E-#0C2F) and KSHORT (#0C30-#0C31) in the NAS-SYS workspace.

THE RESET BUTTON

The Reset button is quite different to any other key. It sends a signal to the computer telling it to reinitialize itself. Pressing Reset does not result in the loss of whatever programs are stored in memory, but the operating system resets itself and takes control. The message "-- NAS-SYS 3 --" is displayed at the top left of the screen, and the computer waits for you to enter a command.

SCROLLING DISPLAY

When Reset is pressed or the screen is cleared, the cursor is positioned at the top left, and as commands are entered, the display scrolls down the screen. When the bottom line is reached, the whole display automatically starts to scroll upwards, showing the last 15 lines.

The top line of the display is never scrolled, making it a convenient place to display headings.

Note that user programs must not change the margins in the Video RAM.

DETAILED DESCRIPTION OF EACH COMMAND

A xxxx yyyy ARITHMETIC COMMAND

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.

JJ is the displacement required in a Jump Relative instruction which starts at xxxx, to cause a jump to yyyy. If such a jump is not possible, then ?? is displayed instead.

B xxxx BREAKPOINT COMMAND

The Breakpoint command is used to insert a trap code into the program at address xxxx specified. Clearly a breakpoint cannot be inserted into a program which is in ROM (Read Only Memory). The command B 0 turns off the breakpoint.

Initially the breakpoint is turned off, since Reset deactivates it. The breakpoint command may be entered at any time. Once it has been entered, NAS-SYS remembers the breakpoint address and also keeps a record of the value at that address. Then, when an Execute command is entered, code #E7 is automatically inserted at the breakpoint. If the code is encountered during execution, then the program registers are saved, and are also displayed. The original value is replaced at the breakpoint address. Any command can then be entered, so that for example the program can be modified. If an Execute or Step command is then entered without specifying an address to start execution, the program will automatically restart where the breakpoint was. If an Execute command is entered, then the original instruction at the breakpoint is executed, and the breakpoint will only stop execution again the next time that it is encountered.

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.

BREAKPOINT

M start
xxxx værdi. <RET>
C start△ start+1△ længde-1. } fylder gu Ram lager værdi.

C xxxx yyyy zzzz COPY COMMAND

Copy a block of data, length zzzz from xxxx to yyyy. One byte is copied at a time, starting with the first byte, so if there is an overlap between the source area and the target area, data may be destroyed. (Also see Intelligent copy command.)

The command can also be used to fill an area of storage with a single value. To do this put the value into memory at xxxx using the Modify command, make yyyy one greater than xxxx, and set zzzz to the number of bytes into which the value is to be copied.

D D COMMAND

The D command starts execution of a program at address #D000. This is normally the cold start for the ROM version of ZEAP.

E xxxx EXECUTE COMMAND

The Execute command executes a program, starting at address xxxx.

If address xxxx is not entered, then the stored program counter is used. This means that to continue execution after encountering a breakpoint, it is only necessary to enter E.

G xxxx yyyy zzzz GENERATE COMMAND

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

Data from address xxxx up to but not including address yyyy is written to the tape, and zzzz is the address at which execution is to start. Start the tape mechanism before entering the command. The LED is only on during output of the program.

When reading in the tape, there is no need to enter any commands at all. Simply start the tape, and stop it when the program has started execution.

The data on the tape is as follows:-
Return → (CR) E0(CR) R(CR)
then the data in the same format as the Write command, then
Ezzzz (CR)
/Read.
Execute 0000

H — HALF DUPLEX TERMINAL

The H command executes a very simple program in NAS-SYS which waits for input characters, and outputs them. The program can only be ended by pressing Reset.

To make the Nascom act as a half duplex, ASCII, computer terminal, enter the command X 30 and then enter the H command.

I xxxx yyyy zzzz — INTELLIGENT COPY COMMAND

The Intelligent copy command is identical to the Copy command, in that it copies a block of data, length zzzz, from xxxx to yyyy.

However it is intelligent enough to start the copying process at whichever end is required to ensure that data in an overlapping region is never destroyed.

J — JUMP COMMAND

The Jump command starts execution of a program at address #FFFA. This is normally the cold start for the 8K ROM BASIC.

K xx — KEYBOARD COMMAND

The Keyboard command sets the method of operation of the keyboard.

K 0   is the normal option. This is the condition following Reset.

K 1   reverses the effect of the SHIFT key on the letters.

K 4   reverses the effect of the GRAPHICS key.

K 5   has the effect of both K 1 and K 4.

M xxxx — MODIFY COMMAND

The Modify command enables locations in memory to be examined and modified, starting at address xxxx.

The address of the location to be modified is displayed, followed by the current value. This value may be changed, and when the line is entered the new value is set at that location. Several values may be entered on the line, in which case subsequent locations also have their values changed.

Enter "." at the end of the line to end the Modify command.

Enter ":" at the end of the line to go back to the previous address instead of forward to the next.

Enter "/yyyy" at the end of the line to continue the Modify command at address yyyy.

Normally, hexadecimal values must be entered. However it is also possible to enter ASCII characters directly into memory by entering a comma followed by the character. For example ",A,B,C" would enter ABC into memory.

Any invalid values entered are detected, and an error message is displayed. In this case the command continues automatically at the address at the start of the last line.

N — NORMAL COMMAND

The Normal command resets the pointers to the input and output tables to their normal values. This has the effect of turning off an X or U command. All input will then be from the Nascom keyboard or the serial input, and output will be only to the screen.

O xx yy — OUTPUT COMMAND

The Output command sends data to a port. The value yy is sent to the port xx. For example "O 7 F" would send #0F to port #07. To learn how to use the parallel ports, read the PIO technical manual.

P DISPLAY PROGRAM REGISTERS

The P command displays the stored user program registers. See "Display of Program Registers" for a description of this display.

Q xx QUERY COMMAND

The Query command obtains data from a port and displays it in hexadecimal. The port queried is the port xx. To learn how to use the parallel ports, read the PIO technical manual.

R xxxx READ COMMAND

4 * escape ⇒ out of read.

The Read command reads a cassette tape which was written by the Write command. See the Write command for the format of the data. Normally the value xxxx is not entered, and the data is read back into the memory locations from which it was written. If xxxx is entered, then it is added to the address at which the data is stored. For example if the data on the tape was written from address #1000, and the command entered is "R 0800", then the data is stored at #1800.

As each block of data is read, the header information is displayed:-

```
SSSS BBLL .
```

SSSS is the start address of the data, BB is the block number, and LL is the length (00=256). After the last block, block 00, has been read correctly, the Read command automatically ends. During execution of the command, the LED is switched on.

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 is detected. If the checksum for the header data is incorrect, then a question mark is displayed, and the program waits for the start of the next block. The data following is not loaded. If the checksum for the actual data loaded is incorrect, then instead of a full stop, a question mark 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 locations. This technique eliminates the need for a buffer.

Do not press any keys on the keyboard while reading data 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. If any errors are encountered, simply rewind the tape for about two blocks and carry on.

To stop the Read command in the middle, press ESC (SHIFT/ENTER) four times. This only works between blocks, so if necessary first press any keys until end of block is reached.

S xxxx SINGLE STEP COMMAND

The Single Step command executes a single instruction of a program, at address xxxx.

If address xxxx is not specified, then the stored program counter is used. This means that to execute a single step after encountering a breakpoint, it is only necessary to enter S.

If the previous command was a Step command, then even S need not be entered.

After the instruction has been executed, the program registers are saved, and are also displayed.

It is possible to single step through NAS-SYS routines called by a program, although since NAS-SYS should not contain any errors it is more efficient to set a breakpoint at the next instruction after the call and enter E to execute the routine.

High speed single stepping is possible, using the repeat keyboard feature. Enter the S command and then hold down the ENTER key. The repeat speed may be altered by changing the value KSHORT (#0C30-#0C31).

T xxxx yyyy zzzz vv hhaa TABULATE COMMAND

The Tabulate command displays a block of memory, starting at address xxxx and continuing up to but not including address yyyy. Each line shows the start address followed by the values of several bytes starting at that address, in hexadecimal, followed by the same values in ASCII. Bytes with values in the ranges #00-#1F, #7F-#9F and #FF are output as ".".

zzzz lines are displayed at a time. Press any key to display the next group of lines, or press ESC (SHIFT/ENTER) to end the command. By setting zzzz to 0, all of the data will be output without a pause.

If vv is 0, then 8 bytes are output on each line. Otherwise 8+vv bytes are output. For example set vv to 8 for 16 values on each line. Fewer than 8 values can be output. For example set vv to #FC for 4 values on each line.

If aa is not 0, then the ASCII part of the display is suppressed. If hh is not 0, then the hexadecimal part of the display is suppressed.

Note that since the Tabulate command is the only command to have more that three parameters, it is not normally necessary to enter the fourth and fifth values, since they will remain unchanged, except on Reset when all values are set to 0.

Note that the repeat keyboard feature may be used to scroll up the tabulated data quickly. Set zzzz to 1 and hold down the ENTER key to scroll up.

U USER I/O COMMAND

The User I/O command activates user specified input and output routines, in addition to the keyboard input and screen output, which continue to operate normally. These routines can for example control a parallel printer. The Normal command can be used to turn these routines off again.

The user output routine must be pointed to by an address stored at #0C78-#0C79, and the user input routine must have its address stored at #0C7B-#0C7C. On Reset these addresses point to a Return instruction within NAS-SYS, so if these addresses are not altered, entering the U command will have no effect.

The U command is automatically deactivated during execution of Read, Write and Generate commands, and reactivated afterwards.

V VERIFY COMMAND

The Verify command is identical to the Read command, except that the data read from the cassette tape is not loaded into memory. The purpose of the command is to check that a tape can be read without error.

W xxxx yyyy — WRITE COMMAND

The Write command writes data to a cassette tape. Data from address xxxx up to but not including address yyyy is written to the tape.

Data is output in blocks, each of 256 bytes, except the last block, which may have less. The format of each block is as follows:-

00	Null (0).
FF FF FF FF	Four start of block characters (#FF).
SS SS	Start address, low order first.
LL	Length of data (00=256).
BB	Block number. This is one less for each block. The last block is block 00.
CC	Checksum for the header data.
DD DD	Data.
EE	Checksum for the data.
00 00 00 00 00 00 00 00 00 00	Ten nulls (0).

As each block is written, the header data, consisting of the start address, block number and length is displayed as follows:-

```
SSSS BBLL
```

When the command is entered, the LED is switched on, there is a brief delay, 256 nulls are output, and then each block is output. At the end, the LED is switched off and the command is ended.

The extra nulls at the end of each block ensure that even if several characters are lost in a block, the next block can still be read correctly. The extra null before the start of each block ensures that an initial spurious start of block character is ignored. The 256 nulls at the start ensure that error correction is always possible merely by rewinding the tape and playing it again.

X xx EXTERNAL COMMAND

The External command activates input and output routines contained within NAS-SYS, which provide comprehensive capabilities for communication with external devices such as ASCII terminals (e.g. Teletype), and mainframe computers, through the serial input/output ports. The Normal command can be used to turn these routines off again.

The value entered after the X specifies the X option, each bit of which has a specific meaning described below. However the most commonly used options are as follows:-

22 Support a terminal in full duplex mode.
Line feed is automatically supplied after carriage return.
All output is even parity.

32 Same as 22, but line feed is not supplied.

20 Support a terminal in half duplex mode.
When a character is received from the terminal, unless it is ESC (#1B), or Null (0), it is assumed that the program will try to output the character. Therefore an indicator is set so that the next character output by the program is not sent to the serial output.
Line feed is always supplied after carriage return on input and output.

30 Same as 20, but line feed is not supplied.
This option makes the Nascom into a half duplex terminal.

23, 33, 21, 31 Same as 22, 32, 20, 30, respectively, but the output parity is odd instead of even.
(Input parity is always ignored.)

To input a BACKSPACE on a terminal which does not have this key, use CTRL/H.

The X command is automatically deactivated during execution of Read, Write and Generate commands, and reactivated afterwards.

The X command also activates the U command output routine, if one has been supplied. (See the U command.) This makes it very easy to have the Nascom keyboard and display, and a Teletype keyboard and printer, and a parallel printer, all working at the same time

X OPTION BIT	FUNCTION
0	Parity of output characters. 0 = even, 1 = odd.
1	Suppress echo of input characters by user program, by setting bit 7 to 1 for each input character other than ESC (#1B) or NUL (0). 0 = suppress, 1 = do not suppress.

X OPTION BIT	FUNCTION
4	Supply LF (#0A) after CR (#0D) on output. Also output LF when CR is input unless bit 1 = 1, and regardless of the state of bit 5. 0 = output LF after CR, 1 = do not output LF after CR.
5	Make the input routine echo all input characters. 0 = echo, 1 = do not echo.
7	Indicator used to suppress the next output character. Set to 0 by output routine. See description of bit 1. 0 = do not suppress, 1 = suppress next output character.

Y

Y COMMAND

The Y command starts execution of a program at address #B000.

Z

Z COMMAND

The Z command starts execution of a program at address #FFFD. This is normally the warm start for the 8K ROM BASIC.

HOW TO ENTER AND TEST A PROGRAM

1. Write out the program on paper, in Z80 Assembler language. You will need to refer to the Z80 programming manual, and the sections of this manual describing the NAS-SYS facilities you can use for input and output, and how to end the program. There are also many useful routines in NAS-SYS which you may wish to make use of.

2. Then you can assemble the program by hand. This means that you must convert each instruction into machine code. The Z80 programming manual gives all the codes. The NAS-SYS Arithmetic command makes it easy to calculate the values to put into relative jump instructions.

3. If you have ZEAP, the Nascom Editor and Assembler package, then you can type in your program in Assembler, and ZEAP will generate all the machine code for you.

4. If you don't have an assembler, then use the Modify command to enter the program into memory.

5. Since there might be an error in the program, and this error might change any part of memory and quite possibly wipe out the program or corrupt it, it is wise to save the program on tape before starting to test it. Use the Write command to write the program to cassette tape. Unless you are confident that your tape recorder and tape are working reliably, use the Verify command to check that the tape can be read back.

6. To test the program, use the Execute command to run it. You can set a breakpoint first by using the Breakpoint command. You can also use the Single Step command to execute small parts of the program which are causing trouble. You can use the Modify or Tabulate commands to examine the program and areas of memory used by the program, after a breakpoint or during single stepping.

7. If you crash the program, you can always use the Reset button to regain control of the computer, and then use the Read command to reload the program for another attempt.

8. Although Assembler and machine code programming may seem slow and difficult at first, it is also fascinating, and it provides the only way to get the most out of the machine, in terms of both speed of execution of programs, and also in the depth of understanding which you will gain about the Z80 microprocessor and computing in general. Good luck!

DISPLAY OF PROGRAM REGISTERS

When the Single Step command is used, or when a breakpoint set by the Breakpoint command is encountered, or if code #E7 (RST #20) is executed in a program, the program registers are displayed as follows:-

```
-SP- nnvv   -PC- nnvv   -AF- nnvv   -HL- nnvv
-DE- nnvv   -BC- nnvv   I  -IX- -IY- Flags
```

The value nnvv after each of the six main registers is the two byte value at the memory locations pointed to by that register. vv is the actual byte pointed to, and nn is the next byte after that.

The flags are a decoded representation of the F register. The following characters may be displayed, indicating which flag bits have been set:-

S Z H P N C

The register display is often an essential aid when debugging a program. The Program Counter (PC) shows the address of the next instruction to be executed, and the Stack Pointer (SP) shows the position on the stack. The other registers show the effect of the program instructions on them. When the registers have been displayed, it is often necessary to investigate the actions of the program in more detail, by using the Modify command to determine the contents of various memory locations. For example you can find out what is on the stack.

SIMPLE INSTRUCTIONS FOR INPUT AND OUTPUT

NAS-SYS uses a most powerful method of controlling input and output. This is described in the section "Input and Output".

However, to simply output the contents of the A register to the screen, enter the code #F7 (RST #30) in your program. You simply type in F7 as part of your program.

To simply wait for an input character from the keyboard or from the serial input, enter the code #CF (RST 8) in your program. You simply type in CF as part of your p,ogram. The input character will be returned in the A register.

In both cases, no other registers will be affected.

Even if you use only this simple method provided by NAS-SYS for input and output, you can use the X or U commands to control terminals and printers. See the descriptions of these commands.

HOW TO END A PROGRAM

One of the following methods should be used to end each of your programs - and no other!

1. Press the Reset button at any time, to restart the system. A #76 (HALT) instruction may be placed in the program to make the Halt LED light, to indicate that the program has finished, and then you can press the Reset button to continue. This is a very primitive solution.

2. Execute code #C7 (RST 0) in the program. This is equivalent to pressing the Reset button. Like the first method this is simple but not very clever. Both these methods have the disadvantage that the screen is cleared so that you can't see what was output before the program ended.

3. The normal and recommended method of ending a program is to execute codes #DF #5B in the program. This provides a controlled return to NAS-SYS, so that you can enter commands. (The program registers are not saved, and the user stack pointer is set back to #1000.)

4. The method of ending a program at a place which it shouldn't ever get to is to execute code #E7 (RST #20) in the program. This stores the program registers and displays them, before returning control to NAS-SYS. This can be useful to indicate an abnormal end of program, as well as providing useful information for debugging. Execution of the program may then be continued by entering an Execute or Single Step command, but this command must specify the address at which execution is to start.

5. Generate a Non-Maskable Interrupt (NMI). If the computer has a hardware feature to allow the user to generate a single NMI by pushing a button, then this will have the same effect as method 4 above, except that execution can be continued by simply entering E.

Note that the action of NAS-SYS on encountering a breakpoint or an NMI is to jump to $NMI (#0C7D), which contains a jump to a NAS-SYS routine which stores and displays the program registers, and returns control to NAS-SYS so that commands can be entered. However it is possible to make a program change this instruction so that it causes a jump to a diagnostic routine of your own.

NAS-SYS RESTARTS AND ROUTINES

NAS-SYS provides many useful routines which can be called by user programs. These fall into three categories:-

1. Restart instructions. These functions are called by using the one byte Z80 RST instructions.

2. Routines which are called by special two byte instructions which in fact consist of a #DF (RST #18) instruction followed by a number which indicates the routine to be called. NAS-SYS routines should always be called by this method and never by a CALL instruction, because if NAS-SYS was changed, the routine numbers would be the same, but the absolute addresses would be different.

3. There is one routine which is called by a normal CALL instruction. This routine allows another program to perform NAS-SYS initialisation without losing control.

The routines in each of these three categories will now be described in detail, except for the low level input and output routines, which are described in the section "Input and Output".

Note that all NAS-SYS routines are interruptable, which means that no time is data held on the stack at an address less than the stack pointer. Therefore programs which use interrupts can use NAS-SYS routines without any difficulty. Interrupt routines must of course preserve the state of all registers, and the stack space must be sufficient.

NAS-SYS RESTART INSTRUCTIONS

CODE	ASSEMBLER	NAME	FUNCTION
#C7	RST 0	START	Reset computer. Initialise NAS-SYS.
#CF	RST 8	RIN	Obtain an input character in the A register.
#D7	RST #10	RCAL	Relative Call. Follow this code with the displacement to the routine to be called. This is similar to the Z80 Jump Relative instruction, and it allows relocatable code to be written.
#DF	RST #18	SCAL	Subroutine Call. Follow this code with the number of the routine to be called. This is the method used to call the NAS-SYS routines. See the next section.
#E7	RST #20	BRKPT	Store and display the program registers, then return control to NAS-SYS. This is used by the Breakpoint command.
#EF	RST #28	PRS	Output the string of characters following this code, until a 0 is encountered. Then continue execution with the next instruction. This provides a very simple way of displaying a message. The A register is set to 0.
#F7	RST #30	ROUT	Output the character in the A register.
#FF	RST #38	RDEL	Wait for a period of time dependent on the value in the A register. A is set to 0.

uden cu

NAS-SYS ROUTINES

These routines are called simply by putting the codes listed beside them into your program. For example routine ERRM outputs an error message. So to output an error message, enter the codes #DF #6B in your program. (You simply type in DF 6B as part of your program.)

It is also possible to call the routines which are the NAS-SYS commands. To do this use the code #DF followed by the ASCII code for the letter of the command. Registers HL, DE and BC should be set to the values stored in ARG1, ARG2 and ARG3 which would normally be typed in after the command. For example codes #DF #57 would call the NAS-SYS Write command routine. You would need to set HL and ARG1 to the start of the data to be written, and DE and ARG2 to the next address after the end of the data, before calling the routine. You may need to examine the listing of NAS-SYS to learn exactly how the commands work, and this is essential in the case of the Read, Verify and Tabulate commands.

You might want to define your own set of commands and routines, or add functions that are not in NAS-SYS. You can do this because the address of the start of the table of routine addresses is stored at $STAB (#0C71-#0C72). You can change this value to point to your own table of routine addresses. Examine the listing of NAS-SYS to learn how it works.

Before the list of the routines you can call, here is the description of a special routine called STMON which may be called at address #000D, by the codes #CD #0D #00. This routine reinitialises NAS-SYS and clears the screen. The point is that if the computer is set up so that on Reset it starts execution at an address other than 0, then the program which starts executing can set the Stack Pointer and then call STMON. It can then continue just as if it had been executed from NAS-SYS, and can use the NAS-SYS routines.

The listing of the NAS-SYS routines follows. You should find them easy to incorporate into your programs, and they should allow you to develop working programs more quickly.

CODES =====	NAME ====	FUNCTION ========
#DF #64	NUM	Examine an input line and convert a hexadecimal value from ASCII to binary. Set DE to point to the start of the line. Leading blanks are ignored. The value is ended by a blank or null (0). DE is returned pointing to the next position. If the value is invalid (not 0-9, A-F, or >#FFFF), then the Carry flag is set, and DE points to the invalid character. The resulting value is placed in NUMV (#0C21-#0C22) and the number of characters in the ASCII value is placed in NUMN (#0C20). The HL and A registers are modified.
#DF #66	TBCD3	Output the value in the HL register in ASCII, followed by a space. Also add H and L into the C register. The A register is modified.
#DF #67	TBCD2	Output the value in the A register in ASCII. Also add A into the C register. The A register is modified.
#DF #68	B2HEX	Output the value in the A register in ASCII. The A register is modified.
#DF #69	SPACE	Output a space. The A register is set to a space.
#DF #6A	CRLF	Output a Carriage return/line feed. A is set to a CR.
#DF #6B	ERRM	Output the message "Error" followed by a CR. A is set to a CR.
#DF #6C	TX1	Output HL in ASCII, then a space, then DE, then another space. Also, H, L, D and E are added into the C register. Register A is modified.
#DF #6D	SOUT	Send a string of characters directly to the serial output port. HL must be set to point to the start of the string, and B must be set to the length. The C register is set to 0 and then all the characters are added into it. Registers HL, B and A are also modified.
#DF #6F	SRLX	Send the character in the A register directly to the serial output port.

CODES =====	NAME ====	FUNCTION ========
#DF #5B	MRET	This is not a normal routine, but is instead used to end a program and return control to NAS-SYS. See "How to end a program".
#DF #5C	SCALJ	This is not a normal routine, but is instead used to call any other routine, when the routine number is not known when the program is written. Store the routine number at address ARGC (#0C0A) and then execute codes #DF #5C. The routine will be called for you.
#DF #5D	TDEL	Wait for about 1 second (at 4 MHz). Registers A and B are set to 0.
#DF #5E	FFLP	Flip one or more bits of output port 0, then immediately flip them back again. Register A must hav a 1 in each bit position to be flipped. A is modified
#DF #5F	MFLP	Alter the state of (turn on or off) the tape drive LED. Register A is modified.
#DF #60	ARGS	Load the contents of ARG1 into HL, ARG2 into DE and ARG3 into BC. ARG1, 2 and 3 are the first three values entered after a NAS-SYS command. You could use this routine to pick up extra values typed in as part of the E command used to run the program.
#DF #62	IN	Scan for an input character. Instead of waiting for an input, like code #CF (RST 8), it just checks to see if there has been an input. If there has been, then the Carry flag is set and the character is in A. The A register is modified.
#DF #63	INLIN	Obtain an input line. This is the routine used by NAS-SYS to get commands. It provides a blinking cursor and waits for ENTER or NEWLINE to be pressed. The DE register is set to the address of the start of the line where the cursor was when when the line was entered. The A register is modified.

A register indeholder ikke nogen karakter

CODES =====	NAME ====	FUNCTION ========
#DF #79	RLIN	Examine an input line and convert up to ten hexadecimal values separated by spaces from ASCII to binary. Set DE to point to the start of the line. The line must end with a null (0) character. RLIN uses the NUM routine to convert the values. If an invalid value is found or if there are more than ten values, the Carry flag is set. ARGN (#0C0B) is set to the number of values found. ARG1 (#0C0C-#0C0D) to ARG10 (#0C1E-#0C1F) are set to the values. NAS-SYS uses this routine to get the values from commands entered. The HL, DE, BC and A registers are modified.
#DF #7A	B1HEX	Output the low order (right) half of the A register in ASCII. The A register is modified.
#DF #7B	BLINK	Obtain an input character HEX FORM in the A register. While waiting for the input, blink the cursor on the screen. The display is not modified. Registers HL and DE are modified.
#DF #7C	CPOS	The HL register must be set to point to a position on the screen. Then the routine is called. It sets HL to point to the address of the first character on that line on the screen. Register A is modified.
#DF #7E	SP2	Output two spaces. The A register is set to a space.
#DF #7F	SCALI	This is not a normal routine, but is instead used to call any other routine, when the routine number is not known when the program is written. Store the routine number in register E and then execute codes #DF #7F. The routine will be called for you.

INPUT AND OUTPUT

This section describes in detail the powerful method which NAS-SYS uses to control input and output, so that you can make best use of it. However, it is not necessary to read or understand this section to make extensive use of NAS-SYS and its input and output facilities.

The descriptions of the N, U and X commands tell you how to control terminals and printers using NAS-SYS. The Restart instructions RIN, ROUT and PRS enable you to input characters and output single characters and messages. The NAS-SYS routines IN, BLINK and INLIN provide various input options, and TBCD3, TBCD2, B2HEX, B1HEX, SPACE, SP2, CRLF, ERRM and TX1 provide several simple ways to output data. (NUM and RLIN go a stage further and help you to get data from input lines.)

So at this point you should know how to specify input and output in your program, and how to control where this data is to go to by the N, U and X commands.

However, you can control where the data is to come from for input and go to for output, much more flexibly, if you understand the following description of how NAS-SYS works.

Both input and output work the same way, as follows:-

Every time an input or output is requested, a special routine (called ATE) in NAS-SYS calls each of a number of input/output handling routines in turn. Like all other NAS-SYS routines, these have routine numbers. There is a table of routine numbers for the input routines, and another for the output routines. Each of these tables is ended by a null (0). The address of the input table is stored at $IN (#0C75-#0C76), and the address of the output table is stored at $OUT (#0C73-#0C74).

As usual in NAS-SYS, the routine numbers are converted to actual addresses by referring to the table of subroutine addresses. The address of the start of this table is stored at $STAB (#0C71-#0C72).

The ATE routine which calls each of the input/output routines in turn, automatically preserves the HL, DE and BC registers. Output routines must preserve the AF register. Input routines must return with the Carry flag set and the input character in the A register if there is an input, otherwise the Carry flag must be reset.

The way that the NAS-SYS N, U and X commands work is to change the addresses of the input and output tables, at $IN and $OUT. NAS-SYS contains alternative input/output tables and other input and output routines for this purpose.

Now that you know how the input and output works, you will see that it is possible to set up your own tables of routine numbers, and then change the addresses at $IN (#0C75-#0C76) and $OUT (#0C73-#0C74) to point to your tables. You may also wish to add your own routines, and this is done most easily by using the jump instructions provided for the U command. NAS-SYS provides some routines to make it easier for you to change the addresses of the tables. These are as follows:-

CODES	NAME	FUNCTION
=====	====	========
#DF #71	NOM	Set HL to the address of the new output table, then call this routine. It changes the address for you, and returns with the previous address in HL.
#DF #72	NIM	Set HL to the address of the new input table, then call this routine. It changes the address for you, and returns with the previous address in HL.
#DF #77	NNOM	Set the output table back to normal. It returns with the previous address in HL.
#DF #78	NNIM	Set the input table back to normal. It returns with the previous address in HL.

You now need to know the routine numbers to put in your tables. You must remember to put a 0 at the end of each table. The routine numbers are as follows:-

INPUT ROUTINES

Low level I/O rutiner

CODE	NAME	FUNCTION
#61	KBD	Scan Nascom keyboard.
#7D	RKBD	Scan Nascom keyboard and provide repeat key feature.
#70	SRLIN	Scan serial input port.
#74	XKBD	Scan external ASCII keyboard. (See X command.)
#76	UIN	User specified input routine. (See U command.)

OUTPUT ROUTINES

CODE	NAME	FUNCTION
#65	CRT	Display on Nascom screen. See "Screen Editing" for details of facilities.
#6F	SRLX	Output to serial output port. Tape, printer
#6E	XOUT	Output to external ASCII device. (See X command.) X option.
#75	UOUT	Output to user specified output routine. (See U command.)

NAS-SYS WORKSPACE

NAS-SYS requires an area of memory to use as a workspace. This occupies locations #0C00 to #0C7F, so the first free area for user programs is at #0C80. The workspace is initialised on Reset. User programs may choose to make use of or alter certain values in the workspace, so the following table is included to supplement the listing of NAS-SYS. The table shows the address and length of each value, in hexadecimal, and its name and function.

ADDR	LEN	NAME	FUNCTION
#0C00	1	PORT0	Copy of current state of output port 0.
#0C01	9	KMAP	Map of state of keyboard.
#0C0A	1	ARGC	Routine number processed by routine SCALJ.
#0C0B	1	ARGN	Number of values in input line.
#0C0C	2	ARG1	First value entered.
#0C0E	2	ARG2	Second value entered.
#0C10	2	ARG3	Third value entered.
#0C12	2	ARG4	Fourth value entered.
#0C14	2	ARG5	Fifth value entered.
#0C16	8	ARG69	Sixth to ninth values entered.
#0C1E	2	ARG10	Tenth value entered.
#0C20	1	NUMN	Number of characters in value examined by routine NUM.
#0C21	2	NUMV	Value returned by routine NUM.
#0C23	2	BRKADR	Breakpoint address.
#0C25	1	BRKVAL	Stored value from the breakpoint address.
#0C26	1	CONFLG	Normally 0, but set to -1 for the E command.
#0C27	1	$KOPT	Keyboard option. See K command.
#0C28	1	$XOPT	X option. See X command.
#0C29	2	CURSOR	Position of the cursor.
#0C2B	1	ARGX	Last command letter entered.
#0C2C	2	KCNT	Keyboard repeat counter.
#0C2E	2	KLONG	Keyboard initial repeat delay.
#0C30	2	KSHORT	Keyboard repeat speed.
#0C32	2	KBLINK	Cursor blink speed.
#0C34	#2D	MONSTK	NAS-SYS stack.
#0C61	2	RBC	Register save area. Register BC.
#0C63	2	RDE	Register save area. Register DE.
#0C65	2	RHL	Register save area. Register HL.
#0C67	2	RAF	Register save area. Register AF.
#0C69	2	RPC	Register save area. Program counter.
#0C6B	2	RSP	Register save area. Stack pointer.
#0C6D	2	$KTABL	Length of keyboard table.
#0C6F	2	$KTAB	Address of last byte in keyboard table.
#0C71	2	$STAB	Start of table of routine addresses, for routine 00. Since the first routine is in fact number #41, the actual table starts #82 beyond this address.
#0C73	2	$OUT	Start of table of output routines. (0779)
#0C75	2	$IN	Start of table of input routines. (077C)
#0C77	3	$UOUT	Jump to user specified output routine.
#0C7A	3	$UIN	Jump to user specified input routine.
#0C7D	3	$NMI	Jump to breakpoint/NMI routine. NAS-SYS sets this to display the registers.

0C01 række 0
0C02 " 1
:
0C08 " 7
0C09 " Ø

C3 2F00
C3 2F0C

77 C3
78 0C
79 2F

0C7A C3
7B 0C
7C 2F

ADDRESSING OF VIDEO RAM

The video RAM is addressed as shown in the diagram below. The top line is addressed after the other 15, and it is not scrolled by NAS-SYS. There is a 10 byte margin on the left of each line, then a 48 byte line, then a 6 byte margin on the right. When NAS-SYS clears the screen, all visible locations are made blank, and all the margins are set to nulls (0), except for the first 10 bytes, which are to the left of line 1, and the last 6 bytes, which are to the right of line 16, the top line.

LINE NO.	LEFT MARGIN	LEFT SIDE		RIGHT SIDE	RIGHT END
16	0BC0	0BCA	0BCB	0BF9	0BFF
1	0800	080A	40H : ny linie	0839	083F
2	0840	084A		0879	087F
3	0880	088A		08B9	08BF
4	08C0	08CA		08F9	08FF
5	0900	090A		0939	093F
6	0940	094A		0979	097F
7	0980	098A		09B9	09BF
8	09C0	09CA		09F9	09FF
9	0A00	0A0A		0A39	0A3F
10	0A40	0A4A		0A79	0A7F
11	0A80	0A8A		0AB9	0ABF
12	0AC0	0ACA		0AF9	0AFF
13	0B00	0B0A		0B39	0B3F
14	0B40	0B4A		0B79	0B7F
15	0B80	0B8A	0B8B	0BB9	0BBF

LEFT SIDE RIGHT SIDE

ZEAP Z80 Assembler - Source Listing

```
                0005 ; NAS-SYS 3 (R1)
                0010 ; WRITTEN BY RICHARD BEAL

                0020 ; CHARACTERS
0000            0025          ORG   0
0000 0008       0030 BS       EQU   #08
0000 000A       0035 LF       EQU   #0A
0000 000C       0040 CS       EQU   #0C
0000 000D       0045 CR       EQU   #0D
0000 0011       0050 CUL      EQU   #11 )
0000 0012       0055 CUR      EQU   #12 )
0000 0013       0060 CUU      EQU   #13 }  pile
0000 0014       0065 CUD      EQU   #14 )
0000 0015       0070 CSL      EQU   #15
0000 0016       0075 CSR      EQU   #16
0000 0017       0080 CH       EQU   #17
0000 0018       0085 CCR      EQU   #18
0000 001B       0090 ESC      EQU   #1B
0000 005F       0095 CU       EQU   #5F

                0105 ; ROM ADDRESSES
0000 0000       0110 ROM      EQU   #0
0000 D000       0115 DJMP     EQU   #D000
0000 B000       0120 YJMP     EQU   #B000
0000 FFFA       0125 BPRC     EQU   #FFFA
0000 FFFD       0130 BPRW     EQU   #FFFD

                0140 ; VIDEO RAM
0000 0800       0145 VRAM     EQU   #0800
0000 080A       0150 VL1      EQU   VRAM+10
0000 084A       0155 VL2      EQU   VL1+64
0000 0B8A       0160 VL15     EQU   VRAM+#038A
0000 0C00       0165 VEND     EQU   VRAM+#400

                0175 ; WORKSPACE RAM
0000 0C00       0180 RAM      EQU   #0C00
0000 1000       0185 USRSP    EQU   #1000


                0200 ; WORKSPACE
0C00            0205 INITZ  ORG   RAM
                0210 ; COPY OF PORT 0
```

```
0C00 0001      0215 PORTO  DEFS 1
               0220 ; KEYBOARD STATUS MAP
0C01 0009      0225 KMAP   DEFS 9
               0230 ; COMMAND CHAR
0C0A 0001      0235 ARGC   DEFS 1
               0240 ; NO OF ARGS
0C0B 0001      0245 ARGN   DEFS 1
               0250 ; UP TO 10 ARGS
0C0C 0002      0255 ARG1   DEFS 2
0C0E 0002      0260 ARG2   DEFS 2
0C10 0002      0265 ARG3   DEFS 2
0C12 0002      0270 ARG4   DEFS 2
0C14 0002      0275 ARG5   DEFS 2
0C16 0008      0280 ARG69  DEFS 8
0C1E 0002      0285 ARG10  DEFS 2
               0290 ; NO OF CHARS IN HEX VALUE
0C20 0001      0295 NUMN   DEFS 1
               0300 ; HEX VALUE ENTERED
0C21 0002      0305 NUMV   DEFS 2
               0310 ; BPT ADDRESS
0C23 0002      0315 BRKADR DEFS 2
               0320 ; BPT VALUE
0C25 0001      0325 BRKVAL DEFS 1
               0330 ; CONFLG NOT 0 IF E COMMAND
0C26 0001      0335 CONFLG DEFS 1
               0340 ; K OPTION
0C27 0001      0345 $KOPT  DEFS 1
               0350 ; X OPTION
0C28 0001      0355 $XOPT  DEFS 1
               0360 ; CURSOR POSITION
0C29 0002      0365 CURSOR DEFS 2
               0370 ; LAST COMMAND
0C2B 0001      0375 ARGX   DEFS 1
               0380 ; REPEAT COUNTER
0C2C 0002      0385 KCNT   DEFS 2
               0390 ; INITIAL REPEAT DELAY
0C2E 0002      0395 KLONG  DEFS 2
               0400 ; REPEAT SPEED
0C30 0002      0405 KSHORT DEFS 2
               0410 ; BLINK SPEED
0C32 0002      0415 KBLINK DEFS 2
               0420 ; MONITOR STACK
0C34 002D      0425 MONSTK DEFS #2D
0C61 0C61      0430 STACK  EQU  $
               0435 ; REGISTER SAVE AREA
0C61 0002      0440 RBC    DEFS 2
0C63 0002      0445 RDE    DEFS 2
0C65 0002      0450 RHL    DEFS 2
0C67 0002      0455 RAF    DEFS 2
0C69 0002      0460 RPC    DEFS 2
               0465 ; USER SP
```

```
0C6B 0002       0470 RSP    DEFS 2
                0475 ; END OF REG SAVE AREA
0C6D 0C6D       0480 RSAE   EQU  $
0C6D 0C6D       0485 INITR  EQU  $
                0490 ; LENGTH OF KTAB
0C6D 0002       0495 $KTABL DEFS 2
                0500 ; ADDRESS OF END OF KTAB
0C6F 0002       0505 $KTAB  DEFS 2
                0510 ; ADDRESS OF STAB
0C71 0002       0515 $STAB  DEFS 2
                0520 ; OUTPUT TABLE
0C73 0002       0525 $OUT   DEFS 2
                0530 ; INPUT TABLE
0C75 0002       0535 $IN    DEFS 2
                0540 ; USER JUMPS
0C77 0003       0545 $UOUT  DEFS 3
0C7A 0003       0550 $UIN   DEFS 3
                0555 ; NMI JUMP
0C7D 0003       0560 $NMI   DEFS 3

                0570 ; END OF WORKSPACE
0C80 0C80       0575 INITE  EQU  $


                0590 ; START OF MONITOR
0000            0595 START  ORG  ROM
0000 310010     0600        LD   SP,USRSP
0003 D708       0605        RCAL STMON
0005 C3FE03     0610        JP   MRET


                0625 ; GET INPUT
0008 DF62       0630 RIN    SCAL ZIN
000A D8         0635        RET  C
000B 18FB       0640        JR   RIN


                0655 ; INITIALISE MONITOR
000D C3DE03     0660 STMON  JP   STRTB


                0675 ; RELATIVE CALL
                0680 ; INC RET ADDRESS
0010 E3         0685 RCAL   EX   (SP),HL
0011 23         0690        INC  HL
0012 E3         0695        EX   (SP),HL
0013 E5         0700        PUSH HL
0014 F5         0705        PUSH AF
0015 C38405     0710        JP   RCALB
```

```
                  0725 ; SUBROUTINE CALL
0018 18F6         0730 SCAL     JR   RCAL


                  0745 ; OUTPUT HL DE, ADD TO SUM
001A D700         0750 TX1      RCAL TX2
001C DF66         0755 TX2      SCAL ZTBCD3
001E EB           0760          EX   DE,HL
001F C9           0765          RET


                  0780 ; BPT
                  0785 ; DECREMENT PC ON STACK
0020 E3           0790 BRKPT    EX   (SP),HL
0021 2B           0795          DEC  HL
0022 E3           0800          EX   (SP),HL
0023 C37D0C       0805          JP   $NMI
0026 0000         0810          DEFB 0,0; FILL


                  0825 ; OUTPUT A STRING
0028 E3           0830 PRS      EX   (SP),HL
0029 7E           0835 PRS1     LD   A,(HL)
002A 23           0840          INC  HL
                  0845 ; OUTPUT UNLESS 0
002B B7           0850          OR   A
002C 2006         0855          JR   NZ,PRS2
002E E3           0860          EX   (SP),HL
002F C9           0865 DRET     RET


                  0880 ; OUTPUT A CHAR
0030 E5           0885 ROUT     PUSH HL
0031 C35507       0890          JP   AOUT


                  0905 ; MORE OF PRS
0034 F7           0910 PRS2     RST  ROUT
0035 18F2         0915          JR   PRS1
0037 00           0920          DEFB 0; FILL


                  0935 ; DELAY
0038 3D           0940 RDEL     DEC  A
0039 C8           0945          RET  Z
003A F5           0950          PUSH AF
003B F1           0955          POP  AF
003C 18FA         0960          JR   RDEL


                  0975 ; DELAY
```

```
003E AF         0980 TDEL    XOR  A
003F 47         0985         LD   B,A
0040 FF         0990 TDEL2   RST  RDEL
0041 FF         0995         RST  RDEL
0042 10FC       1000         DJNZ TDEL2
0044 C9         1005         RET


                1020 ; SET, RESET BIT IN PO
0045 E5         1025 FFLP    PUSH HL
0046 21000C     1030         LD   HL,PORTO
0049 AE         1035         XOR  (HL)
004A D300       1040         OUT  (0),A
004C 7E         1045         LD   A,(HL)
004D D300       1050 FF2     OUT  (0),A
004F E1         1055         POP  HL
0050 C9         1060         RET

                1070 ; FLIP BIT 4 IN PO
0051 3E10       1075 MFLP    LD   A,#10
0053 E5         1080         PUSH HL
0054 21000C     1085         LD   HL,PORTO
0057 AE         1090         XOR  (HL)
0058 77         1095         LD   (HL),A
0059 18F2       1100         JR   FF2


                1115 ; SERIAL OUTPUT TO P1
005B F5         1120 SRLX    PUSH AF
005C D301       1125         OUT  (1),A
                1130 ; WAIT UNTIL OUTPUT
005E DB02       1135 SRL4    IN   A,(2)
0060 CB77       1140         BIT  6,A
0062 28FA       1145  Z/N    JR   Z,SRL4
0064 F1         1150         POP  AF
0065 C9         1155         RET


                1170 ; NMI RESTART
0066 C37D0C     1175 RNMI    JP   $NMI


                1190 ; GET INPUT
0069 E5         1195 BIN     PUSH HL
006A 2A320C     1200         LD   HL,(KBLINK)
006D DF62       1205 BIN2    SCAL ZIN
006F 3805       1210         JR   C,BIN8
0071 2B         1215         DEC  HL
0072 7C         1220         LD   A,H
0073 B5         1225         OR   L
0074 20F7       1230         JR   NZ,BIN2
```

```
0076 E1          1235 BIN8    POP  HL
0077 C9          1240         RET


                 1255 ; BLINK UNTIL INPUT
0078 2A290C      1260 BLINK   LD   HL,(CURSOR)
007B 56          1265         LD   D,(HL)
007C 365F        1270         LD   (HL),CU
007E D7E9        1275         RCAL BIN
0080 72          1280         LD   (HL),D
0081 D8          1285         RET  C
0082 D7E5        1290         RCAL BIN
0084 30F2        1295         JR   NC,BLINK
0086 C9          1300         RET


                 1315 ; CHECK SERIAL IN
0087 DB02        1320 SRLIN   IN   A,(2)
0089 17          1325         RLA
008A D0          1330         RET  NC
008B DB01        1335         IN   A,(1)
008D C9          1340         RET


                 1355 ; REPEAT KEYBOARD ROUTINE
008E DF61        1360 RKBD    SCAL ZKBD
0090 3007        1365         JR   NC,RK2
                 1370 ; KEY PRESSED
0092 2A2E0C      1375         LD   HL,(KLONG)
0095 222C0C      1380         LD   (KCNT),HL
0098 C9          1385         RET
                 1390 ; KEY NOT PRESSED
0099 2A2C0C      1395 RK2     LD   HL,(KCNT)
009C 2B          1400         DEC  HL
009D 222C0C      1405         LD   (KCNT),HL
00A0 7C          1410         LD   A,H
00A1 B5          1415         OR   L
00A2 C0          1420         RET  NZ
                 1425 ; TEST AND CLEAR TABLE
00A3 21020C      1430         LD   HL,KMAP+1
00A6 010008      1435         LD   BC,#0800
                 1440 ; SET UP MASK IN D
00A9 16FF        1445 RK3     LD   D,#FF
00AB 7D          1450         LD   A,L
00AC FE06        1455         CP   6
00AE 2002        1460         JR   NZ,RK5
00B0 16BF        1465         LD   D,#BF
00B2 FE09        1470 RK5     CP   9
00B4 2002        1475         JR   NZ,RK6
00B6 16C7        1480         LD   D,#C7
                 1485 ; TEST FOR KEY DOWN
```

```
00B8 7E         1490 RK6     LD   A,(HL)
00B9 A2         1495         AND  D
00BA 2806       1500         JR   Z,RK7
00BC 0E01       1505         LD   C,1
                1510 ; CLEAR KEYS DOWN
00BE 7A         1515         LD   A,D
00BF 2F         1520         CPL
00C0 A6         1525         AND  (HL)
00C1 77         1530         LD   (HL),A
                1535 ; NEXT ROW
00C2 23         1540 RK7     INC  HL
00C3 10E4       1545         DJNZ RK3
00C5 79         1550         LD   A,C
00C6 B7         1555         OR   A
00C7 C8         1560         RET  Z
                1565 ;  REPEAT SPEED
00C8 2A300C     1570         LD   HL,(KSHORT)
00CB 222C0C     1575         LD   (KCNT),HL


                1590 ; KEYBOARD ROUTINE
                1595 ; RESET KBD COUNTER
00CE 3E02       1600 KBD     LD   A,2       } nulstil pil
00D0 CD4500     1605         CALL FFLP      }
                1610 ; STORE ROW 0 IN MAP
00D3 21010C     1615         LD   HL,KMAP     0C01  pegepind til Ra
00D6 DB00       1620         IN   A,(0)
00D8 2F         1625         CPL                  ned > '1'
00D9 77         1630         LD   (HL),A

                1640 ; SCAN 8 ROWS
00DA 0608       1645         LD   B,8        løkketæller
                1650 ; INC KBD COUNTER
00DC 3E01       1655 KSC1    LD   A,1
00DE CD4500     1660         CALL FFLP
00E1 23         1665         INC  HL                efter cpl
                1670 ; GET ROW STATUS
00E2 DB00       1675         IN   A,(0)             | tast nede '1'   op
00E4 2F         1680         CPL
00E5 E67F       1685         AND  #7F  bit7 væk     8 gange
00E7 57         1690         LD   D,A
                1695 ; IF MAP DIFFERENT
                1700 ;  FIND OUT WHY
00E8 AE         1705         XOR  (HL)              konstater forskel
00E9 2004       1710         JR   NZ,KSC2           to ens tal XOR = 0
                1715 ; SCAN NEXT ROW                    er der slet :
00EB 10EF       1720 KSC1A   DJNZ KSC1
                1725 ; NO KEY PRESSED               B-1 if B<>0 jump
00ED B7         1730 KSC8    OR   A
00EE C9         1735         RET                CY=0 . et slags signal
                                                       tilbage
                1745 ; WAIT, TO DEBOUNCE
00EF AF         1750 KSC2    XOR   A        } venter Ams
00F0 FF         1755         RST   RDEL
                1760 ; GET ROW AGAIN
00F1 DB00       1765         IN    A,(0)
00F3 2F         1770         CPL                                 paal flek
00F4 E67F       1775         AND   #7F    -                      virker i
00F6 5F         1780         LD    E,A    - Tastatur status
                1785 ; E = NEW STATE      el. LD E, A
00F7 7A         1790         LD    A,D -
                1795 ; A = OLD STATE            D NY STATUS
00F8 AE         1800         XOR   (HL)          (HL) FØR STATUS
                1805 ; A = CHANGES
                1810 ; FIND CHANGED BIT    KUN 1 forandring
00F9 0EFF       1815         LD    C,-1        (LSB) forandring.     (C)
00FB 1600       1820         LD    D,0              ændres:  bit nr
00FD 37         1825         SCF      — set Cr. flag                 alle andre
00FE CB12       1830 KSC4    RL    D           00000001              nulstille
0100 0C         1835         INC   C
0101 1F         1840         RRA
0102 30FA       1845         JR    NC,KSC4
                1850 ; C = COL. CHANGED, selv
                1855 ; D = MASK WITH 1 AT CHANGE
0104 7A         1860         LD    A,D
0105 A3         1865         AND   E     — kontroller om vi trykkede ned el
0106 5F         1870         LD    E,A     E≠0 hvis vi har sloppet   tast.
                1875 ; E = NEW STATE,
                1880 ;  MASKED BY CHANGE
                1885 ; IF MAP STATE AND NEW
                1890 ;  STATE EQUAL, IGNORE
0107 7E         1895         LD    A,(HL)
0108 A2         1900         AND   D
0109 BB         1905         CP    E
010A 28DF       1910         JR    Z,KSC1A  sker aldrig
                1915 ; UPDATE MAP
010C 7E         1920         LD    A,(HL) }
010D AA         1925         XOR   D      }  opdater map også ikke
010E 77         1930         LD    (HL),A }
                1935 ; IF NEW STATE IS 0, ~~THEN~~
                1940 ;  KEY RELEASED, SO IGNORE
010F 7B         1945         LD    A,E
0110 B7         1950         OR    A     sætter zero otto, Z ← 0
0111 28D8       1955         JR    Z,KSC1A      ~~if tast sluppet~~
                                                 Z=1 if ..
[illegible]
                1965 ; VALUE = SRRRRCCC
                1970 ;  S=1 IF SHIFT
                1975 ;  RRRR=9-ROW NUMBER
                1980 ;  CCC=COLUMN NUMBER
0113 3A010C     1985         LD    A,(KMAP)
0116 E610       1990         AND   #10 — bit nr 4 indeholder shift n..
0118 B0         1995         OR    B
                                   000SRRRR
```

B: 8- rækken } x,y koordinater for nøgle matrice
C: bit nr på nøgle }

```
0119 87        2000          ADD  A,A
011A 87        2005          ADD  A,A
011B 87        2010          ADD  A,A  ←  SRRRR000
011C B1        2015          OR   C       SRRRRCCC

               2025 ; SEARCH TABLE
011D D750      2030          RCAL KSE   side 40
011F 2806      2035          JR   Z,KSC5
               2040 ; CHECK FOR UNSHIFTED
0121 E67F      2045          AND  #7F
0123 D74A      2050          RCAL KSE      scanner for småbog
0125 20C6      2055          JR   NZ,KSC8
               2060 ; SET A TO ASCII VALUE
0127 79        2065 KSC5     LD   A,C   ikke for små-bogstaver

               2075 ; SUPPORT LOWER CASE
0128 21010C    2080          LD   HL,KMAP
012B FE41      2085          CP   "A   ascii a    'x'<a
012D 3816      2090          JR   C,K20
012F FE5B      2095          CP   "Z+1             '>'
0131 3012      2100          JR   NC,K20  — nu ved man at det er
               2105 ; IT IS A LETTER
0133 CB66      2110          BIT  4,(HL)
               2115 ; 1 = SHIFT DOWN
0135 2801      2120          JR   Z,K7     hop if stor bogstav
0137 3F        2125          CCF
               2130 ; TEST OPTION
0138 3A270C    2135 K7       LD   A,($KOPT)
013B CB47      2140          BIT  0,A
013D 79        2145          LD   A,C
013E 2801      2150          JR   Z,K8
0140 3F        2155          CCF
0141 3802      2160 K8       JR   C,K20
0143 C620      2165          ADD  A,#20—  forskellen 20H, mellem
                                                 li bogstav.
               2175 ; CONTROL KEYS
               2180 ; IF NOT @, MAY MODIFY
0145 FE40      2185 K20      CP   "@
0147 2006      2190          JR   NZ,K30
               2195 ; IF SHIFT DOWN, NORMAL,   fastev nr.
               2200 ;  OTHERWISE IGNORE
0149 CB66      2205          BIT  4,(HL)
014B 28A0      2210          JR   Z,KSC8  side 52
014D 1806      2215          JR   K35
               2220 ; IF @ DOWN, MODIFY
014F CB6E      2225 K30      BIT  5,(HL)
0151 2802      2230          JR   Z,K35
0153 EE40      2235          XOR  #40
               2240 ; CONTROL
0155 CB5E      2245 K35      BIT  3,(HL)
0157 2802      2250          JR   Z,K40
```

```
0159 EE40       2255        XOR   #40
                2260 ; GRAPHIC                     række i matrice
015B 21060C     2265 K40    LD    HL,KMAP+5
015E CB76       2270        BIT   6,(HL)
0160 2802       2275        JR    Z,K55       ex. 'A' : 01000001
0162 EE80       2280        XOR   #80 ------------->  1000000
                                                      11000001 = CTH.

                2290 ; K4 OPTION
                2295 ;  CHANGE BIT 7
0164 21270C     2300 K55    LD    HL,$KOPT
0167 CB56       2305        BIT   2,(HL)
0169 2802       2310        JR    Z,K60
016B EE80       2315        XOR   #80

                2325 ; END
016D 37         2330 K60    SCF
016E C9         2335        RET

                2345 ; SEARCH KEYBOARD TABLE
016F 2A6F0C     2350 KSE    LD    HL,($KTAB)
0172 ED4B6D0C   2355        LD    BC,($KTABL)
0176 EDB9       2360        CPDR     A - HL  if eq så stop   z=1
0178 C9         2365        RET              ifnot eq så HL= HL-1
                                                BC vil indeholde ascii værdi

                2380 ; WORKSPACE INITIALISATION
                2385 ;  TABLE
0179 0179       2390 INITT  EQU   $
                2395 ; LENGTH OF KTAB
0179 6000       2400 IKTABL DEFW KTABE-KTAB
                2405 ; END OF KEYBOARD TABLE
017B 1906       2410 IKTAB  DEFW KTABE-1
                2415 ; SUBROUTINE TABLE
017D 0007       2420 ISTAB  DEFW STABA-"A-"A
                2425 ; OUTPUT TABLE
017F 7907       2430 IOUT   DEFW OUTT1
                2435 ; INPUT TABLE
0181 7C07       2440 IIN    DEFW INT1
                2445 ; USER JUMPS
0183 C32F00     2450 IUOUT  JP    DRET
0186 C32F00     2455 IUIN   JP    DRET
                2460 ; NMI JUMP
0189 C3         2465 INMI   DEFB #C3
018A 018A       2470 INITX  EQU   $

                2480 ; TABLE WITH SPEEDS
                2485 ; INITIAL REPEAT DELAY
018A 8002       2490 ILONG  DEFW #0280
                2495 ; REPEAT SPEED
018C 5000       2500 ISHORT DEFW #0050
                2505 ; BLINK SPEED
```

```
018E 0001      2510 IBLINK DEFW #0100

               2525 ; CRT ROUTINE
               2530 ; IGNORE NULL OR LF
0190 B7        2535 CRT    OR   A
0191 C8        2540        RET  Z
0192 F5        2545        PUSH AF
0193 FE0A      2550        CP   LF
0195 2824      2555        JR   Z,CRT2

               2565 ; CLEAR SCREEN
0197 FE0C      2570        CP   CS
0199 2022      2575        JR   NZ,CRT6
               2580 ; CLEAR TOP LINE
019B 210A08    2585        LD   HL,VL1
019E E5        2590        PUSH HL
019F 0630      2595        LD   B,48
01A1 3620      2600 CR1    LD   (HL),"
01A3 23        2605        INC  HL
01A4 10FB      2610        DJNZ CR1
               2615 ; SET MARGIN
01A6 0610      2620        LD   B,16
01A8 3600      2625 CR3    LD   (HL),0
01AA 23        2630        INC  HL
01AB 10FB      2635        DJNZ CR3
               2640 ; COPY DOWN SCREEN
01AD EB        2645        EX   DE,HL
01AE E1        2650        POP  HL
01AF E5        2655        PUSH HL
01B0 01B003    2660        LD   BC,VEND-VRAM-64-10-6
01B3 EDB0      2665        LDIR
               2670 ; SET TO TOP LEFT
01B5 E1        2675        POP  HL

               2685 ; SET HL TO LEFT SIDE
01B6 DF7C      2690 CRT0   SCAL ZCPOS

               2700 ; SAVE CURSOR
01B8 22290C    2705 CRT1   LD   (CURSOR),HL

               2715 ; RETURN
01BB F1        2720 CRT2   POP  AF
01BC C9        2725        RET

               2735 ; SET HL TO CURSOR
01BD 2A290C    2740 CRT6   LD   HL,(CURSOR)

               2750 ; BS, CUL
01C0 FE08      2755        CP   BS
01C2 2011      2760        JR   NZ,CRT14
```

```
01C4 F5        2765 CRT8   PUSH AF
               2770 ; IGNORE MARGINS
01C5 2B        2775 CRT10  DEC  HL
01C6 7E        2780        LD   A,(HL)
01C7 B7        2785        OR   A
01C8 28FB      2790        JR   Z,CRT10
01CA F1        2795        POP  AF
01CB FE11      2800        CP   CUL
01CD 2802      2805        JR   Z,CRT12
01CF 3620      2810        LD   (HL),"
01D1 D763      2815 CRT12  RCAL CTST
01D3 18E6      2820        JR   CRT2
01D5 FE11      2825 CRT14  CP   CUL
01D7 28EB      2830        JR   Z,CRT8

               2840 ; CURSOR HOME, ESC
01D9 FE17      2845        CP   CH
01DB 28D9      2850        JR   Z,CRT0
01DD FE1B      2855        CP   ESC
01DF 200B      2860        JR   NZ,CRT20
01E1 DF7C      2865        SCAL ZCPOS
01E3 0630      2870        LD   B,48
01E5 3620      2875 CRT18  LD   (HL),"
01E7 23        2880        INC  HL
01E8 10FB      2885        DJNZ CRT18
01EA 18CA      2890        JR   CRT0

               2900 ; NEW LINE, CCR
01EC FE0D      2905 CRT20  CP   CR
01EE 286D      2910        JR   Z,CRT38
01F0 FE18      2915        CP   CCR
01F2 200C      2920        JR   NZ,CRT25
01F4 E5        2925        PUSH HL
01F5 DF7C      2930        SCAL ZCPOS
01F7 D1        2935        POP  DE
01F8 B7        2940        OR   A
01F9 ED52      2945        SBC  HL,DE
01FB 19        2950        ADD  HL,DE
01FC 28BA      2955        JR   Z,CRT1
01FE 185D      2960        JR   CRT38

               2970 ; CUU, CUD
0200 FE13      2975 CRT25  CP   CUU
0202 2008      2980        JR   NZ,CRT28
0204 11C0FF    2985        LD   DE,-64
0207 19        2990 CRT26  ADD  HL,DE
0208 D72C      2995        RCAL CTST
020A 18AF      3000        JR   CRT2
020C FE14      3005 CRT28  CP   CUD
020E 2005      3010        JR   NZ,CRT29
0210 114000    3015        LD   DE,64
```

```
0213 18F2      3020          JR    CRT26

               3030 ; CSL, CSR
0215 FE15      3035 CRT29    CP    CSL
0217 200E      3040          JR    NZ,CRT32
0219 23        3045 CRT30    INC   HL
021A 7E        3050          LD    A,(HL)
021B 2B        3055          DEC   HL
021C B7        3060          OR    A
021D 2004      3065          JR    NZ,CRT31
021F 3620      3070          LD    (HL),"
0221 1898      3075          JR    CRT2
0223 77        3080 CRT31    LD    (HL),A
0224 23        3085          INC   HL
0225 18F2      3090          JR    CRT30
0227 FE16      3095 CRT32    CP    CSR
0229 201F      3100          JR    NZ,CRT34
022B 0620      3105          LD    B,"
022D 7E        3110 CRT33    LD    A,(HL)
022E B7        3115          OR    A
022F 288A      3120          JR    Z,CRT2
0231 70        3125          LD    (HL),B
0232 47        3130          LD    B,A
0233 23        3135          INC   HL
0234 18F7      3140          JR    CRT33

               3150 ; TEST FOR ON SCREEN
0236 110A08    3155 CTST     LD    DE,VL1
0239 B7        3160          OR    A
023A ED52      3165          SBC   HL,DE
023C 19        3170          ADD   HL,DE
023D D8        3175          RET   C
023E 11BA0B    3180          LD    DE,VL15+48
0241 B7        3185          OR    A
0242 ED52      3190          SBC   HL,DE
0244 19        3195          ADD   HL,DE
0245 D0        3200          RET   NC
0246 F1        3205          POP   AF
0247 C3B801    3210 CT8      JP    CRT1

               3220 ; CUR, OTHERS
024A FE12      3225 CRT34    CP    CUR
024C 2801      3230          JR    Z,CRT36
024E 77        3235          LD    (HL),A
               3240 ; IGNORE MARGINS
024F 23        3245 CRT36    INC   HL
0250 7E        3250          LD    A,(HL)
0251 B7        3255          OR    A
0252 28FB      3260          JR    Z,CRT36

               3270 ; TEST NEED FOR CR
```

```
0254 11CA0B   3275        LD   DE,VL15+64
0257 B7       3280        OR   A
0258 ED52     3285        SBC  HL,DE
025A 19       3290        ADD  HL,DE
025B 20EA     3295        JR   NZ,CT8

              3305 ; DO NEW LINE
025D DF7C     3310 CRT38  SCAL ZCPOS 0249H
025F 114000   3315        LD   DE,64
0262 19       3320        ADD  HL,DE
0263 D7D1     3325        RCAL CTST  0236H

              3335 ; SCROLL UP
0265 110A08   3340 CRT40  LD   DE,VL1
0268 214A08   3345        LD   HL,VL2
026B 017003   3350        LD   BC,VEND-VRAM-64-64-16
026E EDB0     3355        LDIR
              3360 ; CLEAR BOTTOM LINE
0270 0630     3365        LD   B,48
0272 2B       3370 CRT50  DEC  HL
0273 3620     3375        LD   (HL),"
0275 10FB     3380        DJNZ CRT50
0277 18CE     3385        JR   CT8

              3395 ; SET HL TO START OF LINE
0279 7D       3400 CPOS   LD   A,L
027A E6C0     3405        AND  #C0
027C C60A     3410        ADD  A,#0A
027E 6F       3415        LD   L,A
027F C9       3420        RET


              3435 ; MODIFY COMMAND
0280 DF60     3440 MODIFY SCAL ZARGS
              3445 ; OUTPUT ADDRESS
0282 220C0C   3450 MOD1   LD   (ARG1),HL
0285 DF7E     3455        SCAL ZSP2
0287 DF66     3460        SCAL ZTBCD3
0289 7E       3465        LD   A,(HL)
028A DF68     3470        SCAL ZB2HEX
              3475 ; GET INPUT LINE
028C EF       3480        RST  PRS
028D 20111111 3485        DEFB " ,CUL,CUL,CUL,0
     00
0292 D754     3490        RCAL INLS
              3495 ; GET ADDRESS
0294 DF64     3500        SCAL ZNUM
0296 384C     3505        JR   C,MOD9
0298 7E       3510        LD   A,(HL)
0299 B7       3515        OR   A
029A 2848     3520        JR   Z,MOD9
```

```
018E 0001      2510 IBLINK DEFW #0100


               2525 ; CRT ROUTINE
               2530 ; IGNORE NULL OR LF
0190 B7        2535 CRT     OR   A
0191 C8        2540         RET  Z
0192 F5        2545         PUSH AF
0193 FE0A      2550         CP   LF
0195 2824      2555         JR   Z,CRT2

               2565 ; CLEAR SCREEN
0197 FE0C      2570         CP   CS
0199 2022      2575         JR   NZ,CRT6
               2580 ; CLEAR TOP LINE
019B 210A08    2585         LD   HL,VL1
019E E5        2590         PUSH HL
019F 0630      2595         LD   B,48
01A1 3620      2600 CR1     LD   (HL),"
01A3 23        2605         INC  HL
01A4 10FB      2610         DJNZ CR1
               2615 ; SET MARGIN
01A6 0610      2620         LD   B,16
01A8 3600      2625 CR3     LD   (HL),0
01AA 23        2630         INC  HL
01AB 10FB      2635         DJNZ CR3
               2640 ; COPY DOWN SCREEN
01AD EB        2645         EX   DE,HL
01AE E1        2650         POP  HL
01AF E5        2655         PUSH HL
01B0 01B003    2660         LD   BC,VEND-VRAM-64-10-6
01B3 EDB0      2665         LDIR
               2670 ; SET TO TOP LEFT
01B5 E1        2675         POP  HL

               2685 ; SET HL TO LEFT SIDE
01B6 DF7C      2690 CRT0    SCAL ZCPOS

               2700 ; SAVE CURSOR
01B8 22290C    2705 CRT1    LD   (CURSOR),HL

               2715 ; RETURN
01BB F1        2720 CRT2    POP  AF
01BC C9        2725         RET

               2735 ; SET HL TO CURSOR
01BD 2A290C    2740 CRT6    LD   HL,(CURSOR)

               2750 ; BS, CUL
01C0 FE08      2755         CP   BS
01C2 2011      2760         JR   NZ,CRT14
```

```
01C4 F5         2765 CRT8   PUSH AF
                2770 ; IGNORE MARGINS
01C5 2B         2775 CRT10  DEC  HL
01C6 7E         2780        LD   A,(HL)
01C7 B7         2785        OR   A
01C8 28FB       2790        JR   Z,CRT10
01CA F1         2795        POP  AF
01CB FE11       2800        CP   CUL
01CD 2802       2805        JR   Z,CRT12
01CF 3620       2810        LD   (HL),"
01D1 D763       2815 CRT12  RCAL CTST
01D3 18E6       2820        JR   CRT2
01D5 FE11       2825 CRT14  CP   CUL
01D7 28EB       2830        JR   Z,CRT8

                2840 ; CURSOR HOME, ESC
01D9 FE17       2845        CP   CH
01DB 28D9       2850        JR   Z,CRT0
01DD FE1B       2855        CP   ESC
01DF 200B       2860        JR   NZ,CRT20
01E1 DF7C       2865        SCAL ZCPOS
01E3 0630       2870        LD   B,48
01E5 3620       2875 CRT18  LD   (HL),"
01E7 23         2880        INC  HL
01E8 10FB       2885        DJNZ CRT18
01EA 18CA       2890        JR   CRT0

                2900 ; NEW LINE, CCR
01EC FE0D       2905 CRT20  CP   CR
01EE 286D       2910        JR   Z,CRT38
01F0 FE18       2915        CP   CCR
01F2 200C       2920        JR   NZ,CRT25
01F4 E5         2925        PUSH HL
01F5 DF7C       2930        SCAL ZCPOS
01F7 D1         2935        POP  DE
01F8 B7         2940        OR   A
01F9 ED52       2945        SBC  HL,DE
01FB 19         2950        ADD  HL,DE
01FC 28BA       2955        JR   Z,CRT1
01FE 185D       2960        JR   CRT38

                2970 ; CUU, CUD
0200 FE13       2975 CRT25  CP   CUU
0202 2008       2980        JR   NZ,CRT28
0204 11C0FF     2985        LD   DE,-64
0207 19         2990 CRT26  ADD  HL,DE
0208 D72C       2995        RCAL CTST
020A 18AF       3000        JR   CRT2
020C FE14       3005 CRT28  CP   CUD
020E 2005       3010        JR   NZ,CRT29
0210 114000     3015        LD   DE,64
```

```
0213 18F2      3020         JR   CRT26

               3030 ; CSL, CSR
0215 FE15      3035 CRT29   CP   CSL
0217 200E      3040         JR   NZ,CRT32
0219 23        3045 CRT30   INC  HL
021A 7E        3050         LD   A,(HL)
021B 2B        3055         DEC  HL
021C B7        3060         OR   A
021D 2004      3065         JR   NZ,CRT31
021F 3620      3070         LD   (HL),"
0221 1898      3075         JR   CRT2
0223 77        3080 CRT31   LD   (HL),A
0224 23        3085         INC  HL
0225 18F2      3090         JR   CRT30
0227 FE16      3095 CRT32   CP   CSR
0229 201F      3100         JR   NZ,CRT34
022B 0620      3105         LD   B,"
022D 7E        3110 CRT33   LD   A,(HL)
022E B7        3115         OR   A
022F 288A      3120         JR   Z,CRT2
0231 70        3125         LD   (HL),B
0232 47        3130         LD   B,A
0233 23        3135         INC  HL
0234 18F7      3140         JR   CRT33

               3150 ; TEST FOR ON SCREEN
0236 110A08    3155 CTST    LD   DE,VL1
0239 B7        3160         OR   A
023A ED52      3165         SBC  HL,DE
023C 19        3170         ADD  HL,DE
023D D8        3175         RET  C
023E 11BA0B    3180         LD   DE,VL15+48
0241 B7        3185         OR   A
0242 ED52      3190         SBC  HL,DE
0244 19        3195         ADD  HL,DE
0245 D0        3200         RET  NC
0246 F1        3205         POP  AF
0247 C3B801    3210 CT8     JP   CRT1

               3220 ; CUR, OTHERS
024A FE12      3225 CRT34   CP   CUR
024C 2801      3230         JR   Z,CRT36
024E 77        3235         LD   (HL),A
               3240 ; IGNORE MARGINS
024F 23        3245 CRT36   INC  HL
0250 7E        3250         LD   A,(HL)
0251 B7        3255         OR   A
0252 28FB      3260         JR   Z,CRT36

               3270 ; TEST NEED FOR CR
```

```
0254 11CA0B    3275            LD   DE,VL15+64
0257 B7        3280            OR   A
0258 ED52      3285            SBC  HL,DE
025A 19        3290            ADD  HL,DE
025B 20EA      3295            JR   NZ,CT8

               3305 ; DO NEW LINE
025D DF7C      3310 CRT38      SCAL ZCPOS 0249H
025F 114000    3315            LD   DE,64
0262 19        3320            ADD  HL,DE
0263 D7D1      3325            RCAL CTST  0236H

               3335 ; SCROLL UP
0265 110A08    3340 CRT40      LD   DE,VL1
0268 214A08    3345            LD   HL,VL2
026B 017003    3350            LD   BC,VEND-VRAM-64-64-16
026E EDB0      3355            LDIR
               3360 ; CLEAR BOTTOM LINE
0270 0630      3365            LD   B,48
0272 2B        3370 CRT50      DEC  HL
0273 3620      3375            LD   (HL),"
0275 10FB      3380            DJNZ CRT50
0277 18CE      3385            JR   CT8

               3395 ; SET HL TO START OF LINE
0279 7D        3400 CPOS       LD   A,L
027A E6C0      3405            AND  #C0
027C C60A      3410            ADD  A,#0A
027E 6F        3415            LD   L,A
027F C9        3420            RET


               3435 ; MODIFY COMMAND
0280 DF60      3440 MODIFY SCAL ZARGS
               3445 ; OUTPUT ADDRESS
0282 220C0C    3450 MOD1       LD   (ARG1),HL
0285 DF7E      3455            SCAL ZSP2
0287 DF66      3460            SCAL ZTBCD3
0289 7E        3465            LD   A,(HL)
028A DF68      3470            SCAL ZB2HEX
               3475 ; GET INPUT LINE
028C EF        3480            RST  PRS
028D 20111111  3485            DEFB " ,CUL,CUL,CUL,0
     00
0292 D754      3490            RCAL INLS
               3495 ; GET ADDRESS
0294 DF64      3500            SCAL ZNUM
0296 384C      3505            JR   C,MOD9
0298 7E        3510            LD   A,(HL)
0299 B7        3515            OR   A
029A 2848      3520            JR   Z,MOD9
```

```
029C 23        3525        INC  HL
029D D5        3530        PUSH DE
029E 5E        3535        LD   E,(HL)
029F 23        3540        INC  HL
02A0 56        3545        LD   D,(HL)
02A1 EB        3550        EX   DE,HL
02A2 D1        3555        POP  DE
02A3 0600      3560        LD   B,0
               3565 ; GET EACH ENTRY
02A5 E5        3570 MOD2   PUSH HL
02A6 DF64      3575        SCAL ZNUM
02A8 7E        3580        LD   A,(HL)
02A9 B7        3585        OR   A
02AA 2807      3590        JR   Z,MOD3
               3595 ; PUT INTO MEMORY
02AC 23        3600        INC  HL
               3605 ; HL = NUMN+1 = NUMV
02AD 7E        3610        LD   A,(HL)
02AE E1        3615        POP  HL
02AF 77        3620 MOD2A  LD   (HL),A
02B0 04        3625        INC  B
02B1 23        3630        INC  HL
02B2 E5        3635        PUSH HL
02B3 E1        3640 MOD3   POP  HL
02B4 1A        3645        LD   A,(DE)
               3650 ; IF "." RETURN
02B5 FE2E      3655        CP   ".
02B7 C8        3660        RET  Z
               3665 ; IF "," SET CHAR
02B8 FE2C      3670        CP   ",
02BA 2005      3675        JR   NZ,MOD4
02BC 13        3680        INC  DE
02BD 1A        3685        LD   A,(DE)
02BE 13        3690        INC  DE
02BF 18EE      3695        JR   MOD2A
               3700 ; INC IF NONE
02C1 78        3705 MOD4   LD   A,B
02C2 B7        3710        OR   A
02C3 2001      3715        JR   NZ,MOD5
02C5 23        3720        INC  HL
               3725 ; IF ":" GO BACK
02C6 1A        3730 MOD5   LD   A,(DE)
02C7 FE3A      3735        CP   ":
02C9 2004      3740        JR   NZ,MOD7
02CB 2B        3745        DEC  HL
02CC 2B        3750        DEC  HL
02CD 18B3      3755        JR   MOD1
               3760 ; IF "/" SET TO VALUE
02CF FE2F      3765 MOD7   CP   "/
02D1 200A      3770        JR   NZ,MOD8
02D3 13        3775        INC  DE
```

```
02D4 DF64      3780        SCAL ZNUM
02D6 380C      3785        JR   C,MOD9
02D8 2A210C    3790        LD   HL,(NUMV)
02DB 18A5      3795        JR   MOD1
02DD B7        3800 MOD8   OR   A
02DE 28A2      3805        JR   Z,MOD1
02E0 FE20      3810        CP   "
02E2 28C1      3815        JR   Z,MOD2
02E4 DF6B      3820 MOD9   SCAL ZERRM
02E6 1898      3825        JR   MODIFY


               3840 ; ROUTINE TO GET INPUT LINE
               3845 ; STORE BPT BYTE ETC
02E8 D718      3850 INLS   RCAL BRSTO
               3855 ; RESET NMI ADDRESS
02EA 217504    3860        LD   HL,TRAP             ; come to TRAP after NMI or
02ED 227E0C    3865        LD   ($NMI+1),HL         ; BPT
               3870 ; NORMAL START OF ROUTINE
               3875 ; GET INPUT CHAR
02F0 E5        3880 INLIN  PUSH HL  <----------------  SCAL 63H
02F1 DF7B      3885 INL2   SCAL ZBLINK
02F3 F7        3890        RST  ROUT
02F4 FE0D      3895        CP   CR                  CR = 0DH
02F6 20F9      3900        JR   NZ,INL2
               3905 ; SET DE TO START OF INPUT
02F8 2A290C    3910        LD   HL,(CURSOR)
02FB 11C0FF    3915        LD   DE,-64  -   40H
02FE 19        3920        ADD  HL,DE
02FF EB        3925        EX   DE,HL
0300 E1        3930        POP  HL
0301 C9        3935        RET


               3950 ; STORE BPT BYTE AND
               3955 ;  SET CONFLG TO 0
0302 AF        3960 BRSTO  XOR  A
0303 32260C    3965        LD   (CONFLG),A
0306 2A230C    3970        LD   HL,(BRKADR)
0309 7E        3975        LD   A,(HL)
030A 32250C    3980        LD   (BRKVAL),A
030D C9        3985        RET


               4000 ; TABULATE COMMAND
               4005 ; IF HL>=DE THEN END
030E DF6A      4010 TB1    SCAL ZCRLF
0310 C1        4015        POP  BC
0311 B7        4020        OR   A
0312 ED52      4025        SBC  HL,DE
0314 19        4030        ADD  HL,DE
```

```
0315 D0       4035        RET  NC
              4040 ; CONTROL SCROLLING
0316 78       4045 TB2    LD   A,B
0317 B1       4050        OR   C
0318 2007     4055        JR   NZ,TB3
031A CF       4060        RST  RIN
031B FE1B     4065        CP   ESC
031D C8       4070        RET  Z
              4075 ; START OF ROUTINE
031E CDF604   4080 TABCDE CALL ARGS3
0321 0B       4085 TB3    DEC  BC
0322 C5       4090        PUSH BC
              4095 ; OUTPUT ADDRESS
0323 DF7E     4100        SCAL ZSP2
0325 DF66     4105        SCAL ZTBCD3
0327 E5       4110        PUSH HL
              4115 ; OUTPUT 8+ARG4 BYTES
0328 3A120C   4120        LD   A,(ARG4)
032B C608     4125        ADD  A,8
032D 47       4130        LD   B,A
032E C5       4135        PUSH BC
              4140 ; HEX OUTPUT
032F 3A150C   4145 TB4    LD   A,(ARG5+1)
0332 B7       4150        OR   A
0333 2005     4155        JR   NZ,TB4A
0335 7E       4160        LD   A,(HL)
0336 DF68     4165        SCAL ZB2HEX
0338 DF69     4170        SCAL ZSPACE
033A 23       4175 TB4A   INC  HL
033B 10F2     4180        DJNZ TB4
              4185 ; ASCII OUTPUT
033D DF7E     4190        SCAL ZSP2
033F C1       4195        POP  BC
0340 E1       4200        POP  HL
0341 3A140C   4205 TB5    LD   A,(ARG5)
0344 B7       4210        OR   A
0345 200C     4215        JR   NZ,TB8
0347 7E       4220        LD   A,(HL)
0348 3C       4225        INC  A
0349 E67F     4230        AND  #7F
034B FE21     4235        CP   #21
034D 7E       4240        LD   A,(HL)
034E 3002     4245        JR   NC,TB6
0350 3E2E     4250        LD   A,".
0352 F7       4255 TB6    RST  ROUT
0353 23       4260 TB8    INC  HL
0354 10EB     4265        DJNZ TB5
0356 18B6     4270        JR   TB1


              4285 ; OUTPUT HL THEN SPACE
```

```
0358 7C        4290 TBCD3  LD   A,H
0359 DF67      4295        SCAL ZTBCD2
035B 7D        4300        LD   A,L
035C DF67      4305        SCAL ZTBCD2


               4320 ; OUTPUT SPACE
035E 3E20      4325 SPACE  LD   A,"
0360 F7        4330        RST  ROUT
0361 C9        4335        RET


               4350 ; OUTPUT TWO SPACES
0362 D7FA      4355 SP2    RCAL SPACE
0364 18F8      4360        JR   SPACE


               4375 ; ERROR MESSAGE
0366 EF        4380 ERRM   RST  PRS
0367 4572726F  4385        DEFM /Error/
     72
036C 00        4390        DEFB 0


               4405 ; OUTPUT CR
036D 3E0D      4410 CRLF   LD   A,CR
036F F7        4415        RST  ROUT
0370 C9        4420        RET


               4435 ; ADD TO CHECKSUM, OUTPUT
0371 F5        4440 TBCD2  PUSH AF
0372 81        4445        ADD  A,C
0373 4F        4450        LD   C,A
0374 F1        4455        POP  AF


               4470 ; OUTPUT A
0375 F5        4475 B2HEX  PUSH AF
0376 1F        4480        RRA
0377 1F        4485        RRA
0378 1F        4490        RRA
0379 1F        4495        RRA
037A D701      4500        RCAL B1HEX
037C F1        4505        POP  AF

               4515 ; OUTPUT LOW HALF A
037D E60F      4520 B1HEX  AND  #0F
037F C690      4525        ADD  A,#90
0381 27        4530        DAA
0382 CE40      4535        ADC  A,#40
```

```
0384 27        4540         DAA
               4545 ; OUTPUT CHAR
0385 F7        4550         RST  ROUT
0386 C9        4555         RET


               4570 ; READ IN HEX VALUE
               4575 ; DE = INPUT LINE
               4580 ; NUMN = NO OF CHARS
               4585 ; NUMV = VALUE
0387 1A        4590 NUM     LD   A,(DE)
               4595 ; IGNORE BLANKS
0388 FE20      4600         CP   "
038A 13        4605         INC  DE
038B 28FA      4610         JR   Z,NUM
038D 1B        4615         DEC  DE
               4620 ; NUMV, NUMN = 0
038E 210000    4625         LD   HL,0
0391 22210C    4630         LD   (NUMV),HL
0394 AF        4635         XOR  A
0395 21200C    4640         LD   HL,NUMN
0398 77        4645         LD   (HL),A
               4650 ; GET CHAR
0399 1A        4655 NN1     LD   A,(DE)
               4660 ; CHECK FOR END
039A B7        4665         OR   A
039B C8        4670         RET  Z
039C FE20      4675         CP   "
039E C8        4680         RET  Z
               4685 ; CONVERT FROM ASCII
               4690 ; IF LT 0 INVALID
039F D630      4695         SUB  "0
03A1 D8        4700         RET  C
               4705 ; IF LT 10 THEN OK, SO NN2
03A2 FE0A      4710         CP   10
03A4 380B      4715         JR   C,NN2
               4720 ; CONVERT A/F FROM ASCII
03A6 D607      4725         SUB  "A-"0-10
               4730 ; IF LT 10 INVALID
03A8 FE0A      4735         CP   10
03AA D8        4740         RET  C
               4745 ; IF GE 16 INVALID
03AB FE10      4750         CP   16
03AD 3802      4755         JR   C,NN2
               4760 ; INVALID
03AF 37        4765         SCF
03B0 C9        4770         RET

               4780 ; VALID CHAR FOUND
               4785 ; POINT TO NEXT CHAR
03B1 13        4790 NN2     INC  DE
```

```
                4795 ; INC NUMN
03B2 34         4800         INC  (HL)
                4805 ; PUT VALUE IN NUMV, ROTATING
                4810 ;  PREVIOUS CONTENTS
03B3 23         4815         INC  HL
03B4 ED6F       4820         RLD
03B6 23         4825         INC  HL
03B7 ED6F       4830         RLD
03B9 2B         4835         DEC  HL
03BA 2B         4840         DEC  HL
03BB 28DC       4845         JR   Z,NN1
03BD 1B         4850         DEC  DE
03BE 37         4855         SCF
03BF C9         4860         RET


                4875 ; GET ARGUMENTS
03C0 010B0C     4880 RLIN    LD   BC,ARGN
03C3 AF         4885         XOR  A
03C4 02         4890         LD   (BC),A
                4895 ; GET VALUE
                4900 ; C SET IF INVALID
03C5 DF64       4905 RL2     SCAL ZNUM
03C7 D8         4910         RET  C
                4915 ; CHECK FOR END
03C8 7E         4920         LD   A,(HL)
03C9 B7         4925         OR   A
03CA C8         4930         RET  Z
                4935 ; COPY TO ARG1/10
03CB 23         4940         INC  HL
03CC 03         4945         INC  BC
03CD 7E         4950         LD   A,(HL)
03CE 02         4955         LD   (BC),A
03CF 23         4960         INC  HL
03D0 03         4965         INC  BC
03D1 7E         4970         LD   A,(HL)
03D2 02         4975         LD   (BC),A
                4980 ; INC ARGN
03D3 210B0C     4985         LD   HL,ARGN
03D6 34         4990         INC  (HL)
03D7 7E         4995         LD   A,(HL)
03D8 FE0B       5000         CP   11
03DA 38E9       5005         JR   C,RL2
03DC 37         5010         SCF
03DD C9         5015         RET


                5030 ; MONITOR INITIALISATION
                5035 ; RESTORE BPT BYTE
03DE D766       5040 STRTB  RCAL BRRES
                5045 ; SET WORKSPACE TO 0
```

```
03E0 11000C    5050          LD   DE,INITZ
03E3 066D      5055          LD   B,INITR-INITZ
03E5 AF        5060          XOR  A
03E6 12        5065 ST4      LD   (DE),A
03E7 13        5070          INC  DE
03E8 10FC      5075          DJNZ ST4
               5080 ; SET WORKSPACE FROM TABLE
03EA 217901    5085          LD   HL,INITT
03ED 011100    5090          LD   BC,INITE-INITR-2
03F0 EDB0      5095          LDIR
               5100 ; SET SPEEDS FROM TABLE
03F2 112E0C    5105          LD   DE,KLONG
03F5 010600    5110          LD   BC,6
03F8 EDB0      5115          LDIR
               5120 ; CLEAR SCREEN
03FA EF        5125          RST  PRS
03FB 0C00      5130          DEFB CS,0
03FD C9        5135          RET


               5150 ; USER RETURN
               5155 ; RESET STACKS
03FE 31610C    5160 MRET     LD   SP,STACK
0401 210010    5165          LD   HL,USRSP
0404 226B0C    5170          LD   (RSP),HL
0407 EF        5175          RST  PRS
0408 2D2D204E  5180          DEFM /-- NAS-SYS 3 --/
     41532D53
     59532033
     202D2D
0417 0D00      5185          DEFB CR,0
0419 D72B      5190          RCAL BRRES


               5205 , MAIN MONITOR LOOP
               5210 ; GET LINE AND OBEY
041B CDE802    5215 PARSE    CALL INLS              ; Get input
041E 012B0C    5220          LD   BC,ARGX
               5225 ; IF COMMAND IS BLANK, AND
               5230 ;   PREVIOUS COMMAND NOT S,
               5235 ;   IGNORE IT
0421 1A        5240          LD   A,(DE)            ; DE = start of input !
0422 FE20      5245          CP   " 
0424 2005      5250          JR   NZ,PA2
0426 0A        5255          LD   A,(BC)
0427 FE53      5260          CP   "S
0429 20F0      5265          JR   NZ,PARSE
               5270 ; CHECK AND STORE
042B FE41      5275 PA2      CP   "A
042D 380D      5280          JR   C,PERR
042F FE5B      5285          CP   "Z+1
```

```
0431 3009      5290            JR   NC,PERR
0433 02        5295            LD   (BC),A
0434 320A0C    5300            LD   (ARGC),A
               5305 ; POINT TO NEXT CHAR
0437 13        5310            INC  DE
               5315 ; GET ARGS
0438 DF79      5320            SCAL ZRLIN               RLIN = 3C0 H get argumen
043A 3004      5325            JR   NC,PEND                      if carry then invalid
043C DF6B      5330 PERR       SCAL ZERRM               ERRM = 366 H
043E 18DB      5335            JR   PARSE
               5340 ; CALL COMMAND ROUTINE
0440 DF60      5345 PEND       SCAL ZARGS               ARGS: 4EF   HL : ARG1
0442 DF5C      5350            SCAL ZSCALJ                          DE : " 2
0444 18D5      5355 PA7        JR   PARSE                           BC : " 3
                                                        SCALJ : 5ADH


               5370 ; RESTORE BPT BYTE
0446 2A230C    5375 BRRES      LD   HL,(BRKADR)
0449 7C        5380            LD   A,H
044A B5        5385            OR   L
044B C8        5390            RET  Z
044C 3A250C    5395            LD   A,(BRKVAL)
044F 77        5400            LD   (HL),A
0450 C9        5405            RET


               5420 ; THE EXECUTE COMMAND
               5425 ; CONFLG NOT 0 IF E COMMAND
0451 3EFF      5430 EXEC       LD   A,-1
0453 32260C    5435            LD   (CONFLG),A

               5445 ; EXECUTE AND STEP COMMANDS
               5450 ; DISCARD RETURN
0456 F1        5455 STEP       POP  AF
               5460 ; IF NO ADDRESS ENTERED,
               5465 ;  USE STORED USER PC
0457 3A0B0C    5470            LD   A,(ARGN)
045A B7        5475            OR   A
045B 2803      5480            JR   Z,EXEC2
               5485 ; USER PC = NEW ADDRESS
045D 22690C    5490            LD   (RPC),HL
               5495 ; RESTORE REGS BC DE HL AF
0460 C1        5500 EXEC2      POP  BC
0461 D1        5505            POP  DE
0462 E1        5510            POP  HL
0463 F1        5515            POP  AF
               5520 ; RESTORE USER SP
0464 ED7B6B0C  5525            LD   SP,(RSP)
               5530 ; PUT USER PC ON TOP OF STACK
0468 E5        5535            PUSH HL
0469 2A690C    5540            LD   HL,(RPC)
```

```
046C E3        5545        EX   (SP),HL
               5550 ; SET BIT 3 OF PO, TO
               5555 ;  ACTIVATE NMI
046D F5        5560        PUSH AF
046E 3E08      5565        LD   A,8
0470 D300      5570        OUT  (0),A
0472 F1        5575        POP  AF
               5580 ; EXECUTE ONE STEP OF PROGRAM
0473 ED45      5585        RETN


               5600 ; COME HERE AFTER NMI OR BPT
0475 F5        5605 TRAP   PUSH AF
0476 E5        5610        PUSH HL
               5615 ; RESET NMI BIT IN PO
0477 3A000C    5620        LD   A,(PORTO)
047A D300      5625        OUT  (0),A
               5630 ; IF CONFLG NOT 0 THEN E
               5635 ;  SO EXECUTE NORMALLY
047C 3A260C    5640        LD   A,(CONFLG)
047F B7        5645        OR   A
0480 280D      5650        JR   Z,ER1
               5655 ; STORE BPT BYTE,
               5660 ;  SET CONFLG TO 0 FOR
               5665 ;  NMI OR BPT,
               5670 ;  AND INSERT RESTART
0482 CD0203    5675        CALL BRSTO
0485 7C        5680        LD   A,H
0486 B5        5685        OR   L
0487 2802      5690        JR   Z,TRAP8
0489 36E7      5695        LD   (HL),#E7
               5700 ; EXECUTE PROGRAM NORMALLY
048B E1        5705 TRAP8  POP  HL
048C F1        5710        POP  AF
048D ED45      5715        RETN

               5725 ; RESTORE BPT BYTE,
               5730 ; STORE USER REGISTERS
048F D7B5      5735 ER1    RCAL BRRES
0491 D5        5740        PUSH DE
0492 C5        5745        PUSH BC
               5750 ; STACK HAS: PC AF HL DE BC
               5755 ; SET HL TO USER SP
0493 210000    5760        LD   HL,0
0496 39        5765        ADD  HL,SP
               5770 ; SET MONITOR SP
0497 31610C    5775        LD   SP,STACK
               5780 ; COPY USER REGS FROM USER
               5785 ;  STACK TO REG SAVE AREA
049A 11610C    5790        LD   DE,STACK
049D 010A00    5795        LD   BC,10
```

```
04A0 EDB0      5800        LDIR
               5805 ; STORE USER SP
04A2 226B0C    5810        LD   (RSP),HL
04A5 D702      5815        RCAL PREGS
04A7 189B      5820        JR   PA7

               5830 ; OUTPUT REGISTERS
04A9 EF        5835 PREGS  RST  PRS
04AA 1800      5840        DEFB CCR,0
               5845 ;  SP PC AF HL DE BC
04AC 216D0C    5850        LD   HL,RSAE
04AF 0606      5855        LD   B,6
04B1 2B        5860 ER2    DEC  HL
04B2 56        5865        LD   D,(HL)
04B3 2B        5870        DEC  HL
04B4 5E        5875        LD   E,(HL)
04B5 E5        5880        PUSH HL
04B6 EB        5885        EX   DE,HL
04B7 5E        5890        LD   E,(HL)
04B8 23        5895        INC  HL
04B9 56        5900        LD   D,(HL)
04BA 2B        5905        DEC  HL
04BB DF6C      5910        SCAL ZTX1
04BD E1        5915        POP  HL
04BE DF7E      5920        SCAL ZSP2
04C0 10EF      5925        DJNZ ER2
               5930 ; I REG
04C2 ED57      5935        LD   A,I
04C4 DF68      5940        SCAL ZB2HEX
04C6 DF69      5945        SCAL ZSPACE
               5950 ; IX IY REGS
04C8 DDE5      5955        PUSH IX
04CA E1        5960        POP  HL
04CB DF66      5965        SCAL ZTBCD3
04CD FDE5      5970        PUSH IY
04CF E1        5975        POP  HL
04D0 DF66      5980        SCAL ZTBCD3
               5985 ; F REG
04D2 3A670C    5990        LD   A,(RAF)
04D5 11E604    5995        LD   DE,ESTR-1
04D8 0608      6000        LD   B,8
04DA 13        6005 ER4    INC  DE
04DB 17        6010        RLA
04DC F5        6015        PUSH AF
04DD 1A        6020        LD   A,(DE)
04DE 3001      6025        JR   NC,ER6
04E0 F7        6030        RST  ROUT
04E1 F1        6035 ER6    POP  AF
04E2 10F6      6040        DJNZ ER4
04E4 DF6A      6045        SCAL ZCRLF
04E6 C9        6050        RET
```

```
               6060 ; STRING FOR FLAGS
04E7 535A0048 6065 ESTR    DEFB "S,"Z,0,"H
04EB 00504E43 6070         DEFB 0,"P,"N,"C


               6085 ; GET ARGUMENTS
04EF 2A0C0C   6090 ARGS    LD   HL,(ARG1)
04F2 ED5B0E0C 6095 ARGS2   LD   DE,(ARG2)
04F6 ED4B100C 6100 ARGS3   LD   BC,(ARG3)
04FA C9       6105         RET


               6120 ; WRITE COMMAND
04FB DF5F     6125 WRITE   SCAL ZMFLP
               6130 ; WAIT
04FD DF5D     6135         SCAL ZTDEL
               6140 ; OUTPUT TO CRT ONLY
04FF DF77     6145         SCAL ZNNOM
0501 E5       6150         PUSH HL
               6155 ; OUTPUT 256 NULLS
0502 AF       6160         XOR  A
0503 47       6165         LD   B,A
0504 DF6F     6170 W3      SCAL ZSRLX
0506 10FC     6175         DJNZ W3
               6180 ; CALCULATE LENGTH-1
0508 DF60     6185         SCAL ZARGS
050A D7E6     6190 W4      RCAL ARGS2
050C EB       6195         EX   DE,HL
050D 37       6200         SCF
050E ED52     6205         SBC  HL,DE
               6210 ; IF LEN-1 IS NEG, END
0510 DA7D06   6215         JP   C,R1Y
0513 EB       6220         EX   DE,HL
               6225 ; HL = START
               6230 ; DE = LENGTH-1
               6235 ; WAIT
0514 AF       6240         XOR  A
0515 FF       6245         RST  RDEL
               6250 ; OUTPUT 00 FF FF FF FF
0516 0605     6255         LD   B,5
0518 DF6F     6260 W5      SCAL ZSRLX
051A 3EFF     6265         LD   A,#FF
051C 10FA     6270         DJNZ W5
               6275 ; IF BLOCK 0, SET LEN TO E+1
051E AF       6280         XOR  A
051F BA       6285         CP   D
0520 2002     6290         JR   NZ,W6
0522 43       6295         LD   B,E
0523 04       6300         INC  B
               6305 ; SET E TO LENGTH
```

```
0524 58         6310 W6     LD   E,B
                6315 ; OUTPUT START ADDRESS
0525 7D         6320        LD   A,L
0526 DF6F       6325        SCAL ZSRLX
0528 7C         6330        LD   A,H
0529 DF6F       6335        SCAL ZSRLX
                6340 ; OUTPUT LENGTH OF DATA
052B 7B         6345        LD   A,E
052C DF6F       6350        SCAL ZSRLX
                6355 ; OUTPUT BLOCK NUMBER
052E 7A         6360        LD   A,D
052F DF6F       6365        SCAL ZSRLX
                6370 ; NOW DISPLAY ALL THIS
                6375 ; AND OUTPUT HEADER CHECKSUM
0531 0E00       6380        LD   C,0
0533 DF6C       6385        SCAL ZTX1
0535 79         6390        LD   A,C
0536 DF6F       6395        SCAL ZSRLX
                6400 ; OUTPUT THE BLOCK
0538 DF6D       6405        SCAL ZSOUT
                6410 ; OUTPUT CHECKSUM AND NULLS
053A 060B       6415        LD   B,11
053C 79         6420        LD   A,C
053D DF6F       6425 W9     SCAL ZSRLX
053F AF         6430        XOR  A
0540 10FB       6435        DJNZ W9
                6440 ; CRLF (READ HAS SAME TIMING)
0542 DF6A       6445        SCAL ZCRLF
0544 18C4       6450        JR   W4


                6465 ; ICOPY COMMAND
                6470 ; IF ARG1 GE ARG2, GO TO
                6475 ;  LDIR COPY
0546 B7         6480 ICOPY  OR   A
0547 ED52       6485        SBC  HL,DE
0549 19         6490        ADD  HL,DE
054A 3009       6495        JR   NC,COPY
                6500 ; SET TO END NOT START
054C 0B         6505        DEC  BC
054D EB         6510        EX   DE,HL
054E 09         6515        ADD  HL,BC
054F EB         6520        EX   DE,HL
0550 09         6525        ADD  HL,BC
0551 03         6530        INC  BC
0552 EDB8       6535        LDDR
0554 C9         6540        RET


                6555 ; COPY COMMAND
0555 EDB0       6560 COPY   LDIR
```

```
0557 C9       6565        RET


              6580 ; ARITHMETIC COMMAND
0558 EB       6585 ARITH  EX   DE,HL
0559 E5       6590        PUSH HL
              6595 ; SUM
055A 19       6600        ADD  HL,DE
055B DF66     6605        SCAL ZTBCD3
              6610 ; DIFFERENCE
055D E1       6615        POP  HL
055E B7       6620        OR   A
055F ED52     6625        SBC  HL,DE
0561 DF66     6630        SCAL ZTBCD3
              6635 ; OFFSET
0563 2B       6640        DEC  HL
0564 2B       6645        DEC  HL
0565 7C       6650        LD   A,H
0566 CB05     6655        RLC  L
0568 CE00     6660        ADC  A,0
056A 2806     6665        JR   Z,AOK
              6670 ; NO GOOD SO ??
056C EF       6675 ANG    RST  PRS
056D 3F3F0D00 6680        DEFB "?,"?,CR,0
0571 C9       6685        RET
              6690 ; OUTPUT OFFSET
0572 7D       6695 AOK    LD   A,L
0573 0F       6700        RRCA
0574 DF68     6705 A7     SCAL ZB2HEX
0576 C36D03   6710        JP   CRLF


              6725 ; OUTPUT.COMMAND
0579 44       6730 O      LD   B,H
057A 4D       6735        LD   C,L
057B ED59     6740        OUT  (C),E
057D C9       6745        RET


              6760 ; QUERY COMMAND
057E 44       6765 Q      LD   B,H
057F 4D       6770        LD   C,L
0580 ED78     6775        IN   A,(C)
0582 18F0     6780        JR   A7


              6795 ; RELATIVE CALL RESTART
0584 D5       6800 RCALB  PUSH DE
              6805 ; SET HL TO RET ADDR
0585 210600   6810        LD   HL,6
0588 39       6815        ADD  HL,SP
```

```
0589 5E       6820       LD   E,(HL)
058A 23       6825       INC  HL
058B 56       6830       LD   D,(HL)
058C EB       6835       EX   DE,HL
058D 2B       6840       DEC  HL
              6845 ; RCAL OR SCAL?
058E 2B       6850       DEC  HL
058F CB5E     6855       BIT  3,(HL)
0591 23       6860       INC  HL
0592 200B     6865       JR   NZ,SCAL2
              6870 ; GET OFFSET
0594 5E       6875       LD   E,(HL)
              6880 ; E = OFFSET, SET D
0595 7B       6885       LD   A,E
0596 17       6890       RLA
0597 9F       6895       SBC  A,A
0598 57       6900       LD   D,A
0599 23       6905       INC  HL
059A 19       6910       ADD  HL,DE
059B D1       6915 RCAL4 POP  DE
059C F1       6920       POP  AF
059D E3       6925       EX   (SP),HL
              6930 ; FAKE JUMP TO ROUTINE
059E C9       6935       RET


              6950 ; SUBROUTINE CALL RESTART
              6955 ; GET ROUTINE NO.
059F 5E       6960 SCAL2 LD   E,(HL)
05A0 1600     6965 SCAL3 LD   D,0
05A2 2A710C   6970       LD   HL,($STAB)
05A5 19       6975       ADD  HL,DE
05A6 19       6980       ADD  HL,DE
05A7 5E       6985       LD   E,(HL)
05A8 23       6990       INC  HL
05A9 56       6995       LD   D,(HL)
05AA EB       7000       EX   DE,HL
05AB 18EE     7005       JR   RCAL4        59BH


              7020 ; SUBROUTINE FOR CALL
              7025 ;  ROUTINE NO. AT ARGC ,  SP : 0C61 H
05AD E5       7030 SCALJ PUSH HL
05AE F5       7035       PUSH AF
05AF D5       7040       PUSH DE
05B0 210A0C   7045       LD   HL,ARGC
05B3 18EA     7050       JR   SCAL2        59FH


              7065 ; SUBROUTINE FOR CALL
              7070 ;  ROUTINE NO. IN E
```

```
05B5 E5       7075 SCALI   PUSH HL
05B6 F5       7080         PUSH AF
05B7 D5       7085         PUSH DE
05B8 18E6     7090         JR   SCAL3


              7105 ; KEYBOARD TABLE
05BA FFFFFFFF 7110 KTAB    DEFB #FF,#FF,#FF,#FF; #00
05BE FFFFFFFF 7115         DEFB #FF,#FF,#FF,#FF; #04
05C2 08FF8EFF 7120         DEFB #08,#FF,#8E,#FF; #08 BS,LF
05C6 8809FFFF 7125         DEFB #88,#09,#FF,#FF; #0C CS,CR
05CA FF3E2E46 7130         DEFB #FF,#3E,#2E,#46; #10  LRU
05CE 36BEAE0E 7135         DEFB #36,#BE,#AE,#0E; #14 DLR,CH
05D2 FFFFFF89 7140         DEFB #FF,#FF,#FF,#89; #18 ESC
05D6 FFFFFFFF 7145         DEFB #FF,#FF,#FF,#FF; #1C
05DA 149C9BA3 7150         DEFB #14,#9C,#9B,#A3; #20   "#
05DE 92C2BAB2 7155         DEFB #92,#C2,#BA,#B2; #24 $%&'
05E2 AAA298A0 7160         DEFB #AA,#A2,#98,#A0; #28 ()*+
05E6 290A2119 7165         DEFB #29,#0A,#21,#19; #2C ,-./
05EA 1A1C1B23 7170         DEFB #1A,#1C,#1B,#23; #30 0123
05EE 12423A32 7175         DEFB #12,#42,#3A,#32; #34 4567
05F2 2A221820 7180         DEFB #2A,#22,#18,#20; #38 89:;
05F6 A98AA199 7185         DEFB #A9,#8A,#A1,#99; #3C <=>?     3C + 2
05FA 0D2C4113 7190         DEFB #0D,#2C,#41,#13; #40 @ABC
05FE 3B334310 7195         DEFB #3B,#33,#43,#10; #44 DEFG
0602 402D3830 7200         DEFB #40,#2D,#38,#30; #48 HIJK
0606 28313925 7205         DEFB #28,#31,#39,#25; #4C LMNO
060A 1D241534 7210         DEFB #1D,#24,#15,#34; #50 PQRS
060E 4535112B 7215         DEFB #45,#35,#11,#2B; #54 TUVW
0612 443D3C1E 7220         DEFB #44,#3D,#3C,#1E; #58 XYZ[
0616 9E169A96 7225         DEFB #9E,#16,#9A,#96; #5C \]^_
061A 061A     7230 KTABE   EQU  $      5F


              7245 ; KEYBOARD COMMAND
              7250 ; STORE K OPTIONS
061A 7D       7255 KOP     LD   A,L
061B 32270C   7260         LD   ($KOPT),A
061E C9       7265         RET


              7280 ; BPT COMMAND
              7285 ; STORE BPT ADDRESS
061F 22230C   7290 BREAK   LD   (BRKADR),HL
0622 C9       7295         RET


              7310 ; GENERATE COMMAND
              7315 ; OUTPUT COMMANDS TO BOTH
0623 217407   7320 G       LD   HL,OUTT2
0626 DF71     7325         SCAL ZNOM
```

```
0628 E5       7330        PUSH HL
0629 214D06   7335        LD   HL,GDS
062C 0606     7340        LD   B,GDSE-GDS
062E 7E       7345 G2     LD   A,(HL)
062F F7       7350        RST  ROUT
              7355 ; WAIT
0630 0E14     7360        LD   C,20
0632 AF       7365        XOR  A
0633 FF       7370 G4     RST  RDEL
0634 0D       7375        DEC  C
0635 20FC     7380        JR   NZ,G4
0637 23       7385        INC  HL
0638 10F4     7390        DJNZ G2
              7395 ; OUTPUT THE DATA
063A DF57     7400        SCAL "W
              7405 ; WAIT
063C AF       7410        XOR  A
063D FF       7415        RST  RDEL
              7420 ; OUTPUT "E"
063E 3E45     7425        LD   A,"E
0640 F7       7430        RST  ROUT
              7435 ; OUTPUT EXECUTION ADDRESS
0641 2A100C   7440        LD   HL,(ARG3)
0644 DF66     7445        SCAL ZTBCD3
0646 3E0D     7450        LD   A,CR
              7455 ; FINAL CR, END
0648 F7       7460        RST  ROUT
0649 E1       7465        POP  HL
064A DF71     7470        SCAL ZNOM
064C C9       7475        RET

              7485 ; COMMANDS OUTPUT BY GENERATE
064D 0D45300D 7490 GDS    DEFB CR,"E,"0,CR,"R,CR
     520D
0653 0653     7495 GDSE   EQU  $


              7510 ; STRING TO SERIAL OUTPUT
              7515 ; HL = ADDRESS
              7520 ; B = LENGTH
              7525 ; C = CHECKSUM
0653 0E00     7530 SOUT   LD   C,0
0655 7E       7535 S01    LD   A,(HL)
0656 DF6F     7540        SCAL ZSRLX
0658 81       7545        ADD  A,C
0659 4F       7550        LD   C,A
065A 23       7555        INC  HL
065B 10F8     7560        DJNZ S01
065D C9       7565        RET
```

```
                7580 ; READ ROUTINE
065E DF5F       7585 READ   SCAL ZMFLP
                7590 ; NORMAL TABLES
0660 DF77       7595        SCAL ZNNOM
0662 E5         7600        PUSH HL
0663 DF78       7605        SCAL ZNNIM
0665 E5         7610        PUSH HL
                7615 ; LOOK FOR 4 #FF CHARS
                7620 ;  OR 4 ESC CHARS
0666 0603       7625 R1     LD   B,3
0668 4F         7630        LD   C,A
0669 CF         7635 R2     RST  RIN
066A B9         7640        CP   C
066B 20F9       7645        JR   NZ,R1
066D 10FA       7650        DJNZ R2
066F FEFF       7655        CP   #FF
0671 2810       7660        JR   Z,R3
0673 FE1B       7665        CP   ESC
0675 20EF       7670        JR   NZ,R1
                7675 ; END, RESTORE TABLES
0677 EF         7680 R1W    RST  PRS
0678 1800       7685        DEFB CCR,0
067A E1         7690 R1X    POP  HL
067B DF72       7695        SCAL ZNIM
067D E1         7700 R1Y    POP  HL
067E DF71       7705        SCAL ZNOM
0680 C35100     7710        JP   MFLP
                7715 ; GET HEADER DATA
0683 CF         7720 R3     RST  RIN
0684 6F         7725        LD   L,A
0685 CF         7730        RST  RIN
0686 67         7735        LD   H,A
0687 CF         7740        RST  RIN
0688 5F         7745        LD   E,A
0689 CF         7750        RST  RIN
068A 57         7755        LD   D,A
                7760 ; DISPLAY AND CHECK
068B 0E00       7765        LD   C,0
068D DF6C       7770        SCAL ZTX1
068F CF         7775        RST  RIN
0690 B9         7780        CP   C
0691 2029       7785        JR   NZ,R6
                7790 ; OFFSET
0693 3A0B0C     7795        LD   A,(ARGN)
0696 B7         7800        OR   A
0697 2805       7805        JR   Z,R3A
0699 ED4B0C0C   7810        LD   BC,(ARG1)
069D 09         7815        ADD  HL,BC
                7820 ; SET B TO LENGTH
069E 43         7825 R3A    LD   B,E
                7830 ; LOAD THE DATA
```

```
069F 0E00      7835          LD   C,0
06A1 3A2B0C    7840 R4       LD   A,(ARGX)
06A4 FE52      7845          CP   "R
06A6 2803      7850          JR   Z,R4A
06A8 CF        7855          RST  RIN
06A9 1802      7860          JR   R4C
06AB CF        7865 R4A      RST  RIN
06AC 77        7870          LD   (HL),A
06AD E5        7875 R4C      PUSH HL
06AE 2A290C    7880          LD   HL,(CURSOR)
06B1 77        7885          LD   (HL),A
06B2 E1        7890          POP  HL
06B3 81        7895          ADD  A,C
06B4 4F        7900          LD   C,A
06B5 23        7905          INC  HL
06B6 10E9      7910          DJNZ R4
               7915 ; CHECK AGAINST CHECKSUM
06B8 CF        7920          RST  RIN
06B9 B9        7925          CP   C
06BA 2806      7930          JR   Z,R7
               7935 ; ERROR FOUND
06BC EF        7940 R6       RST  PRS
06BD 3F2000    7945          DEFB "?," ,0
06C0 18A4      7950          JR   R1
               7955 ; CR, TEST FOR END
06C2 EF        7960 R7       RST  PRS
06C3 2E2000    7965          DEFB ".," ,0
06C6 AF        7970          XOR  A
06C7 BA        7975          CP   D
06C8 209C      7980          JR   NZ,R1
06CA 18AB      7985          JR   R1W


               8000 ; USER I/O COMMAND
06CC 217B07    8005 UP       LD   HL,INTU
06CF DF72      8010          SCAL ZNIM
06D1 217807    8015          LD   HL,OUTTU
06D4 DF71      8020          SCAL ZNOM
06D6 C9        8025          RET


               8040 ; EXTERNAL (X) COMMAND
06D7 7D        8045 XP       LD   A,L               ARG1 i HL.
06D8 32280C    8050          LD   ($XOPT),A
06DB 217F07    8055          LD   HL,INTX – 077F
06DE DF72      8060          SCAL ZNIM
06E0 217707    8065          LD   HL,OUTTX
06E3 DF71      8070          SCAL ZNOM
06E5 C9        8075          RET
```

```
                8090 ; X INPUT ROUTINE
                8095 ; CHECK FOR INPUT
06E6 DF70       8100 XKBD   SCAL ZSRLIN
06E8 D0         8105        RET  NC
                8110 ; STRIP PARITY
06E9 E67F       8115        AND  #7F
                8120 ; IF FULL DUPLEX, SEND BACK
06EB 21280C     8125        LD   HL,$XOPT
06EE CB6E       8130        BIT  5,(HL)
06F0 CC1D07     8135        CALL Z,XSOPO
                8140 ; TRANSPARENT MODE
06F3 CB4E       8145        BIT  1,(HL)
06F5 200D       8150        JR   NZ,XK4
                8155 ; SUPPLY LF
06F7 F5         8160        PUSH AF
06F8 D71B       8165        RCAL XSOPL
06FA F1         8170        POP  AF
                8175 ; IF ESCAPE OR NULL ENTERED
                8180 ;  ASSUME PROGRAM WILL NOT
                8185 ;  OUTPUT THE CHAR
06FB B7         8190        OR   A
06FC 2806       8195        JR   Z,XK4
06FE FE1B       8200        CP   ESC
0700 2802       8205        JR   Z,XK4
0702 CBFE       8210        SET  7,(HL)
0704 37         8215 XK4    SCF
0705 C9         8220        RET


                8235 ; X OUTPUT ROUTINE
0706 F5         8240 XOUT   PUSH AF
                8245 ; OUTPUT UNLESS BIT 7 SET
                8250 ;  TO SUPPRESS SERIAL OUTPUT
0707 21280C     8255        LD   HL,$XOPT
070A CB7E       8260        BIT  7,(HL)
070C CC1307     8265        CALL Z,XSOP
                8270 ; TURN OFF SUPPRESSION
070F CBBE       8275        RES  7,(HL)
0711 F1         8280        POP  AF
0712 C9         8285        RET

                8295 ; OUTPUT CHAR AND LF
0713 D708       8300 XSOP   RCAL XSOPO
                8305 ; IF IT WAS A CR AND BIT 4
                8310 ;  OF $XOPT = 0, OUTPUT LF
0715 FE0D       8315 XSOPL  CP   CR
0717 C0         8320        RET  NZ
0718 CB66       8325        BIT  4,(HL)
071A C0         8330        RET  NZ
071B 3E0A       8335        LD   A,LF
```

```
                  8345 ; OUTPUT ASCII CHAR
                  8350 ; SET PARITY ETC
071D B7           8355 XSOPO  OR   A
071E F5           8360        PUSH AF
                  8365 ; MAKE PARITY EVEN
071F EA2407       8370        JP   PE,XSOP2
0722 EE80         8375        XOR  #80
                  8380 ; IF BIT 0 SET, MAKE IT ODD
0724 CB46         8385 XSOP2  BIT  0,(HL)
0726 2802         8390        JR   Z,XSOP4
0728 EE80         8395        XOR  #80
                  8400 ; OUTPUT IT
072A CD5B00       8405 XSOP4  CALL SRLX
                  8410 ; RESTORE ORIGINAL VALUE
072D F1           8415        POP  AF
072E C9           8420        RET


                  8435 ; TERMINAL PROGRAM
072F DF63         8440 XN     SCAL ZINLIN
0731 18FC         8445        JR   XN


                  8460 ; MAKE $IN AND $OUT NORMAL
0733 DF78         8465 NORMAL SCAL ZNNIM

                  8475 ; SET NEW OUTPUT TABLE
0735 217907       8480 NNOM   LD   HL,OUTT1
0738 E5           8485 NOM    PUSH HL
0739 2A730C       8490        LD   HL,($OUT)
073C E3           8495        EX   (SP),HL
073D 22730C       8500        LD   ($OUT),HL
0740 E1           8505        POP  HL
0741 C9           8510        RET


                  8525 ; SET NEW INPUT TABLE
0742 217C07       8530 NNIM   LD   HL,INT1
0745 E5           8535 NIM    PUSH HL
0746 2A750C       8540        LD   HL,($IN)
0749 E3           8545        EX   (SP),HL
074A 22750C       8550        LD   ($IN),HL
074D E1           8555        POP  HL
074E C9           8560        RET


                  8575 ; ADDRESS TABLE EXECUTION
074F E5           8580 IN     PUSH HL
0750 21750C       8585        LD   HL,$IN
0753 1803         8590        JR   ATE
0755 21730C       8595 AOUT   LD   HL,$OUT
```

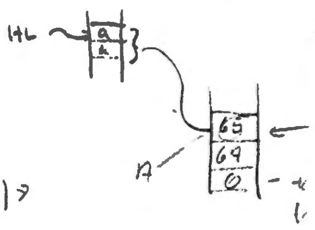
SCAL 62H.

C=1, der var et tegn

0C75H

0C73H

Ved indgang 'til AT:



```
              8600 ; GET START OF TABLE
0758 D5       8605 ATE     PUSH DE
0759 C5       8610         PUSH BC
075A 5E       8615         LD   E,(HL)
075B 23       8620         INC  HL
075C 56       8625         LD   D,(HL)
              8630 ; GET ROUTINE NUMBER
075D F5       8635 AT4     PUSH AF
075E 1A       8640         LD   A,(DE)(0779)>
075F 13       8645         INC  DE
              8650 ; CHECK FOR END
0760 B7       8655         OR   A  <- sætter Zero flag hvis F
                                                          carry
0761 280C     8660         JR   Z,AT6
0763 6F       8665         LD   L,A  -> HL -> 0C65
0764 F1       8670         POP  AF
              8675 ; CALL ROUTINE
0765 D5       8680         PUSH DE
0766 B7       8685         OR   A  <- 65
0767 5D       8690         LD   E,L
0768 CDB505   8695         CALL SCALI  Areg sendt til [illegible]
076B D1       8700         POP  DE
076C 30EF     8705         JR   NC,AT4
076E F5       8710         PUSH AF
076F F1       8715 AT6     POP  AF
0770 C1       8720         POP  BC
0771 D1       8725         POP  DE
0772 E1       8730         POP  HL
0773 C9       8735         RET

                                 DB
              8750 ; OUTPUT TABLES
0774 65       8755 OUTT2   DEFB ZCRT
0775 6F       8760         DEFB ZSRLX
0776 00       8765         DEFB 0
0777 6E       8770 OUTTX   DEFB ZXOUT
0778 75       8775 OUTTU   DEFB ZUOUT
0779 65       8780 OUTT1   DEFB ZCRT <- EFTER RESET
077A 00       8785         DEFB 0

              8800 ; INPUT TABLES
077B 76       8805 INTU    DEFB ZUIN
077C 7D       8810 INT1    DEFB ZRKBD <- EFTER RESET
077D 70       8815         DEFB ZSRLIN
077E 00       8820         DEFB 0
077F 74       8825 INTX    DEFB ZXKBD
0780 7D       8830         DEFB ZRKBD
0781 00       8835         DEFB 0

              8850 ; SUBROUTINE TABLE
```

scrx ch — 077C

indholdet af adr.

```
                 8855 ; STARTS WITH "A"
0782 5805        8860 STABA  DEFW ARITH         "A" : 41H
0784 1F06        8865        DEFW BREAK         "B" : 42H
0786 5505        8870        DEFW COPY          "C" : 43H
0788 00D0        8875        DEFW DJMP          "D" : 44H.
078A 5104        8880        DEFW EXEC
078C 6603        8885        DEFW ERRM
078E 2306        8890        DEFW G
0790 2F07        8895        DEFW XN
0792 4605        8900        DEFW ICOPY
0794 FAFF        8905        DEFW BPRC
0796 1A06        8910        DEFW KOP
0798 6603        8915        DEFW ERRM
079A 8002        8920        DEFW MODIFY
079C 3307        8925        DEFW NORMAL
079E 7905        8930        DEFW O
07A0 A904        8935        DEFW PREGS
07A2 7E05        8940        DEFW Q
07A4 5E06        8945        DEFW READ
07A6 5604        8950        DEFW STEP
07A8 1E03        8955        DEFW TABCDE
07AA CC06        8960        DEFW UP
07AC 5E06        8965        DEFW READ
07AE FB04        8970        DEFW WRITE
07B0 D706        8975        DEFW XP
07B2 00B0        8980        DEFW YJMP
07B4 FDFF        8985        DEFW BPRW
-------------------------------------------------------
07B6 FE03        8990        DEFW MRET;     #5B
07B8 AD05        8995        DEFW SCALJ;    #5C
07BA 3E00        9000        DEFW TDEL;     #5D
07BC 4500        9005        DEFW FFLP;     #5E
07BE 5100        9010        DEFW MFLP;     #5F
07C0 EF04        9015        DEFW ARGS;     #60
07C2 CE00        9020        DEFW KBD;      #61
07C4 4F07        9025        DEFW IN;       #62
07C6 F002        9030        DEFW INLIN;    #63
07C8 8703        9035        DEFW NUM;      #64
07CA 9001        9040        DEFW CRT;      #65
07CC 5803        9045        DEFW TBCD3;    #66
07CE 7103        9050        DEFW TBCD2;    #67
07D0 7503        9055        DEFW B2HEX;    #68
07D2 5E03        9060        DEFW SPACE;    #69
07D4 6D03        9065        DEFW CRLF;     #6A
07D6 6603        9070        DEFW ERRM;     #6B
07D8 1A00        9075        DEFW TX1;      #6C
07DA 5306        9080        DEFW SOUT;     #6D
07DC 0607        9085        DEFW XOUT;     #6E
07DE 5B00        9090        DEFW SRLX;     #6F
07E0 8700        9095        DEFW SRLIN;    #70
07E2 3807        9100        DEFW NOM;      #71
07E4 4507        9105        DEFW NIM;      #72
```

```
07E6 5807     9110          DEFW ATE;     #73
07E8 E606     9115          DEFW XKBD;    #74
07EA 770C     9120          DEFW $UOUT;   #75
07EC 7A0C     9125          DEFW $UIN;    #76
07EE 3507     9130          DEFW NNOM;    #77
07F0 4207     9135          DEFW NNIM;    #78
07F2 C003     9140          DEFW RLIN;    #79
07F4 7D03     9145          DEFW B1HEX;   #7A
07F6 7800     9150          DEFW BLINK;   #7B
07F8 7902     9155          DEFW CPOS;    #7C
07FA 8E00     9160          DEFW RKBD;    #7D
07FC 6203     9165          DEFW SP2;     #7E
07FE B505     9170          DEFW SCALI;   #7F


              9185 ; SUBROUTINE CALL TABLE
0800 005B     9190 ZMRET  EQU  #5B
0800 005C     9195 ZSCALJ EQU  #5C
0800 005D     9200 ZTDEL  EQU  #5D
0800 005E     9205 ZFFLP  EQU  #5E
0800 005F     9210 ZMFLP  EQU  #5F
0800 0060     9215 ZARGS  EQU  #60
0800 0061     9220 ZKBD   EQU  #61
0800 0062     9225 ZIN    EQU  #62
0800 0063     9230 ZINLIN EQU  #63
0800 0064     9235 ZNUM   EQU  #64
0800 0065     9240 ZCRT   EQU  #65
0800 0066     9245 ZTBCD3 EQU  #66
0800 0067     9250 ZTBCD2 EQU  #67
0800 0068     9255 ZB2HEX EQU  #68
0800 0069     9260 ZSPACE EQU  #69
0800 006A     9265 ZCRLF  EQU  #6A
0800 006B     9270 ZERRM  EQU  #6B
0800 006C     9275 ZTX1   EQU  #6C
0800 006D     9280 ZSOUT  EQU  #6D
0800 006E     9285 ZXOUT  EQU  #6E
0800 006F     9290 ZSRLX  EQU  #6F
0800 0070     9295 ZSRLIN EQU  #70
0800 0071     9300 ZNOM   EQU  #71
0800 0072     9305 ZNIM   EQU  #72
0800 0073     9310 ZATE   EQU  #73
0800 0074     9315 ZXKBD  EQU  #74
0800 0075     9320 ZUOUT  EQU  #75
0800 0076     9325 ZUIN   EQU  #76
0800 0077     9330 ZNNOM  EQU  #77
0800 0078     9335 ZNNIM  EQU  #78
0800 0079     9340 ZRLIN  EQU  #79
0800 007A     9345 ZB1HEX EQU  #7A
0800 007B     9350 ZBLINK EQU  #7B
0800 007C     9355 ZCPOS  EQU  #7C
0800 007D     9360 ZRKBD  EQU  #7D
```

```
0800 007E      9365 ZSP2   EQU  #7E
0800 007F      9370 ZSCALI EQU  #7F


               9385 ; SPARE
               9390 ; --- NONE ---


0800 0800      9405 NEND   EQU  $
               9410 ; END OF LISTING
```

ZEAP Z80 Assembler - Symbol Table

```
0C75H 0535 $IN          0C27H 0345 $KOPT
0C6FH 0505 $KTAB        0C6DH 0495 $KTABL
0C7DH 0560 $NMI         0C73H 0525 $OUT
0C71H 0515 $STAB        0C7AH 0550 $UIN
0C77H 0545 $UOUT        0C28H 0355 $XOPT
0574H 6705 A7           056CH 6675 ANG
0572H 6695 AOK          0755H 8595 AOUT
0C0CH 0255 ARG1         0C1EH 0285 ARG10
0C0EH 0260 ARG2         0C10H 0265 ARG3
0C12H 0270 ARG4         0C14H 0275 ARG5
0C16H 0280 ARG69        0C0AH 0235 ARGC
0C0BH 0245 ARGN         04EFH 6090 ARGS
04F2H 6095 ARGS2        04F6H 6100 ARGS3
0C2BH 0375 ARGX         0558H 6585 ARITH
075DH 8635 AT4          076FH 8715 AT6
0758H 8605 ATE          037DH 4520 B1HEX
0375H 4475 B2HEX        0069H 1195 BIN
006DH 1205 BIN2         0076H 1235 BIN8
0078H 1260 BLINK        FFFAH 0125 BPRC
FFFDH 0130 BPRW         061FH 7290 BREAK
0C23H 0315 BRKADR       0020H 0790 BRKPT
0C25H 0325 BRKVAL       0446H 5375 BRRES
0302H 3960 BRSTO        0008H 0030 BS
0018H 0085 CCR          0017H 0080 CH
0C26H 0335 CONFLG       0555H 6560 COPY
0279H 3400 CPOS         000DH 0045 CR
01A1H 2600 CR1          01A8H 2625 CR3
036DH 4410 CRLF         0190H 2535 CRT
01B6H 2690 CRT0         01B8H 2705 CRT1
01C5H 2775 CRT10        01D1H 2815 CRT12
01D5H 2825 CRT14        01E5H 2875 CRT18
01BBH 2720 CRT2         01ECH 2905 CRT20
0200H 2975 CRT25        0207H 2990 CRT26
020CH 3005 CRT28        0215H 3035 CRT29
0219H 3045 CRT30        0223H 3080 CRT31
0227H 3095 CRT32        022DH 3110 CRT33
024AH 3225 CRT34        024FH 3245 CRT36
025DH 3310 CRT38        0265H 3340 CRT40
```

```
0272H 3370 CRT50          01BDH 2740 CRT6
01C4H 2765 CRT8           000CH 0040 CS
0015H 0070 CSL            0016H 0075 CSR
0247H 3210 CT8            0236H 3155 CTST
005FH 0095 CU             0014H 0065 CUD
0011H 0050 CUL            0012H 0055 CUR
0C29H 0365 CURSOR         0013H 0060 CUU
D000H 0115 DJMP           002FH 0865 DRET
048FH 5735 ER1            04B1H 5860 ER2
04DAH 6005 ER4            04E1H 6035 ER6
0366H 4380 ERRM           001BH 0090 ESC
04E7H 6065 ESTR           0451H 5430 EXEC
0460H 5500 EXEC2          004DH 1050 FF2
0045H 1025 FFLP           0623H 7320 G
062EH 7345 G2             0633H 7370 G4
064DH 7490 GDS            0653H 7495 GDSE
018EH 2510 IBLINK         0546H 6480 ICOPY
0181H 2440 IIN            017BH 2410 IKTAB
0179H 2400 IKTABL         018AH 2490 ILONG
074FH 8580 IN             0C80H 0575 INITE
0C6DH 0485 INITR          0179H 2390 INITT
018AH 2470 INITX          0C00H 0205 INITZ
02F1H 3885 INL2           02F0H 3880 INLIN
02E8H 3850 INLS           0189H 2465 INMI
077CH 8810 INT1           077BH 8805 INTU
077FH 8825 INTX           017FH 2430 IOUT
018CH 2500 ISHORT         017DH 2420 ISTAB
0186H 2455 IUIN           0183H 2450 IUOUT
0145H 2185 K20            014FH 2225 K30
0155H 2245 K35            015BH 2265 K40
0164H 2300 K55            016DH 2330 K60
0138H 2135 K7             0141H 2160 K8
00CEH 1600 KBD            0C32H 0415 KBLINK
0C2CH 0385 KCNT           0C2EH 0395 KLONG
0C01H 0225 KMAP           061AH 7255 KOP
00DCH 1655 KSC1           00EBH 1720 KSC1A
00EFH 1750 KSC2           00FEH 1830 KSC4
0127H 2065 KSC5           00EDH 1730 KSC8
016FH 2350 KSE            0C30H 0405 KSHORT
05BAH 7110 KTAB           061AH 7230 KTABE
000AH 0035 LF             0051H 1075 MFLP
0282H 3450 MOD1           02A5H 3570 MOD2
02AFH 3620 MOD2A          02B3H 3640 MOD3
02C1H 3705 MOD4           02C6H 3730 MOD5
02CFH 3765 MOD7           02DDH 3800 MOD8
02E4H 3820 MOD9           0280H 3440 MODIFY
0C34H 0425 MONSTK         03FEH 5160 MRET
0800H 9405 NEND           0745H 8535 NIM
0399H 4655 NN1            03B1H 4790 NN2
0742H 8530 NNIM           0735H 8480 NNOM
0738H 8485 NOM            0733H 8465 NORMAL
```

```
0387H 4590 NUM          0C20H 0295 NUMN
0C21H 0305 NUMV         0579H 6730 O
0779H 8780 OUTT1        0774H 8755 OUTT2
0778H 8775 OUTTU        0777H 8770 OUTTX
042BH 5275 PA2          0444H 5355 PA7
041BH 5215 PARSE        0440H 5345 PEND
043CH 5330 PERR         0C00H 0215 PORTO
04A9H 5835 PREGS        0028H 0830 PRS
0029H 0835 PRS1         0034H 0910 PRS2
057EH 6765 Q            0666H 7625 R1
0677H 7680 R1W          067AH 7690 R1X
067DH 7700 R1Y          0669H 7635 R2
0683H 7720 R3           069EH 7825 R3A
06A1H 7840 R4           06ABH 7865 R4A
06ADH 7875 R4C          06BCH 7940 R6
06C2H 7960 R7           0C67H 0455 RAF
0C00H 0180 RAM          0C61H 0440 RBC
0010H 0685 RCAL         059BH 6915 RCAL4
0584H 6800 RCALB        0C63H 0445 RDE
0038H 0940 RDEL         065EH 7585 READ
0C65H 0450 RHL          0008H 0630 RIN
0099H 1395 RK2          00A9H 1445 RK3
00B2H 1470 RK5          00B8H 1490 RK6
00C2H 1540 RK7          008EH 1360 RKBD
03C5H 4905 RL2          03C0H 4880 RLIN
0066H 1175 RNMI         0000H 0110 ROM
0030H 0885 ROUT         0C69H 0460 RPC
0C6DH 0480 RSAE         0C6BH 0470 RSP
0018H 0730 SCAL         059FH 6960 SCAL2
05A0H 6965 SCAL3        05B5H 7075 SCALI
05ADH 7030 SCALJ        0655H 7535 S01
0653H 7530 SOUT         0362H 4355 SP2
035EH 4325 SPACE        005EH 1135 SRL4
0087H 1320 SRLIN        005BH 1120 SRLX
03E6H 5065 ST4          0782H 8860 STABA
0C61H 0430 STACK        0000H 0595 START
0456H 5455 STEP         000DH 0660 STMON
03DEH 5040 STRTB        031EH 4080 TABCDE
030EH 4010 TB1          0316H 4045 TB2
0321H 4085 TB3          032FH 4145 TB4
033AH 4175 TB4A         0341H 4205 TB5
0352H 4255 TB6          0353H 4260 TB8
0371H 4440 TBCD2        0358H 4290 TBCD3
003EH 0980 TDEL         0040H 0990 TDEL2
0475H 5605 TRAP         048BH 5705 TRAP8
001AH 0750 TX1          001CH 0755 TX2
06CCH 8005 UP           1000H 0185 USRSP
0C00H 0165 VEND         080AH 0150 VL1
0B8AH 0160 VL15         084AH 0155 VL2
0800H 0145 VRAM         0504H 6170 W3
050AH 6190 W4           0518H 6260 W5
```

```
0524H 6310 W6          053DH 6425 W9
04FBH 6125 WRITE       0704H 8215 XK4
06E6H 8100 XKBD        072FH 8440 XN
0706H 8240 XOUT        06D7H 8045 XP
0713H 8300 XSOP        0724H 8385 XSOP2
072AH 8405 XSOP4       0715H 8315 XSOPL
071DH 8355 XSOPO       B000H 0120 YJMP
0060H 9215 ZARGS       0073H 9310 ZATE
007AH 9345 ZB1HEX      0068H 9255 ZB2HEX
007BH 9350 ZBLINK      007CH 9355 ZCPOS
006AH 9265 ZCRLF       0065H 9240 ZCRT
006BH 9270 ZERRM       005EH 9205 ZFFLP
0062H 9225 ZIN         0063H 9230 ZINLIN
0061H 9220 ZKBD        005FH 9210 ZMFLP
005BH 9190 ZMRET       0072H 9305 ZNIM
0078H 9335 ZNNIM       0077H 9330 ZNNOM
0071H 9300 ZNOM        0064H 9235 ZNUM
007DH 9360 ZRKBD       0079H 9340 ZRLIN
007FH 9370 ZSCALI      005CH 9195 ZSCALJ
006DH 9280 ZSOUT       007EH 9365 ZSP2
0069H 9260 ZSPACE      0070H 9295 ZSRLIN
006FH 9290 ZSRLX       0067H 9250 ZTBCD2
0066H 9245 ZTBCD3      005DH 9200 ZTDEL
006CH 9275 ZTX1        0076H 9325 ZUIN
0075H 9320 ZUOUT       0074H 9315 ZXKBD
006EH 9285 ZXOUT
```