NASBUG 4 MONITOR

SOFTWARE MANUAL

Document No: PF/018

Issue No: 2

ISSUE	DATE	REVISED SHEETS	REMARKS
1	1.2.79	(i) - (iii), 1.1, 2.1, 3.1 - 3.8, 4.1 - 4.2, 5.1, 6.1, 7.1 - 7.3, 8.1 - 8.2, 9.1 - 9.2, 10.1 - 10.2	First Formal Issue
		REVISED SECTIONS	
2	25.5.79	(i) - (ii), 1-10	Retyped on A3 for reduction

ORIGINATOR: A.R.Rundle	APPROVAL: P.A. Greenhalgh
TITLE: NASBUG 4 Monitor - Software Manual	

CONTENTS

1. INTRODUCTION

Nasbug T4 is an extended, 2K monitor for the Nascom 1. In addition to supporting the existing Nasbug T2 commands, it contains a number of improvements and extensions, including

- ability to allow the Nascom 1 to act as an ASCII terminal, e.g. to operate over a modem.
- support for ASCII terminals interfaced via UART.
- extended register display.
- fast cassette dump and load format.
- routines to input and output data via the PIO
- intelligent copy.

This document describes the additional facilities and should be read in conjunction with Nascom 1 Programming Manual.

2. REFERENCES

1. Nascom 1 PROGRAMMING MANUAL
   Nascom Microcomputers Document No. PF/016

3. DESCRIPTION

Unless otherwise stated, commands operate as for the 1K Nasbug, version T2. The modifications and extensions are described below.

3.1 Reset

When Reset is pressed, the message "NASBUG 4" will appear. The following actions will have bee taken by the monitor.

- breakpoint (if any) restored to correct value.
- the screen is cleared prior to issuing the start up message.
- the breakpoint command is turned off (it is not necessary to initialise by setting to zero).
- keyboard options are set to 0 (normal).
- input is requested.

3.2 CRT Routine

This now interprets 1CH as "cursor home".

3.3 Tabulate and Dump

The output is now positioned two spaces in from the left hand edge of the screen.

3.4 Load

The load command scrolls up all invalid lines, also:

- the next command can always be entered normally.
- stop tape and type full stop to exit, if required.
- most spurious characters (e.g. generated by starting and stopping cassette) are ignored.

3.5 Modify

When examining or modifying store locations, a colon (:) will step back to the previous address and / aaaa is interpreted as a command to modify address aaaa next.

3.6 Breakpoint and Single Step

The Breakpoint and Single Step commands work as for the 1K monitor but produce an extended register display in the following format:

SP PC AF HL DE BC I IX IY Flags

The Flags are a decoded representation of register F. The following characters may be displaye indicating which flag bits have been set:

S Z H P N C

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

3.7 WRITE

W aaaa bbbb

The Write command outputs data four times faster than Dump. Data from address aaaa, up to but not including address bbbb are sent to the serial output.

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

The format of each is as follows:

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

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

SSSS BBLL

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

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

3.8 GENERATE

>G aaaa bbbb eeee

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

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

eeee is the address at which execution is to start.

The data on the tape is as follows:

New Line
EO New Line
R New Line
Data, in the format used by the Write and Read commands
E eeee New Line

Note:

Start the tape mechanism before entering the G command. The LED is only on while the data specified is being output. When reading the tape in, no commands should be entered. Simply start the tape, and stop it when the program has started execution.

3.9 READ

>R

The Read command reads data which was output by the Write command. (This is four times faster than Load.) See the Write command for the format of the data. As each block is read, the Leader data is displayed:

SSSS BBLL — Start address, block number. length (0 = 256)

After block 0 has been read, the Read command ends. During the execution of the command, the Drive LED is switched on.

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

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

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

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

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

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

3.10 I COPY

>I aaaa bbbb dddd

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

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

3.11 ARITHMETIC

>A aaaa bbbb

The Arithmetic command performs simple hexadecimal arithmetic.

Three results are displayed, as follows:

SSSS DDDD JJ

SSSS is the sum of the two values

DDDD is the difference of the two values
bbbb - aaaa

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

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

3.12 OUTPUT

>O pp dd

The Output command sends a value to a port. The value dd is sent to port pp.

pp and dd are displayed.

e.g: >O 7 F
07 0F
>

0F H has been sent to port 7

3.13 QUERY

Q pp

The Query command obtains a value from a port. The port number pp and the value dd obtained are displayed.

e.g: >Q 5
05 12

Value 12H has been obtained from port 5.

3.14 KEYBOARD

K a.

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

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

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

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

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

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

Pressing Shift makes the letters lower case. To reverse this, so that all letters are lower case, and become upper case when Shift is pressed, like a typewriter, enter the command K1

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

3.15 Z (COMMAND TABLE)

Z aaaa

The Z command changes the address of the command table. Normally this is 0755H, so entering Z755 has no effect. A dummy command table is provided at 0511H, so enter Z511 to disable all commands. (Press Reset to recover from this state.)

3.16 NORMAL

N

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

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

3.17 ? COMMAND

?

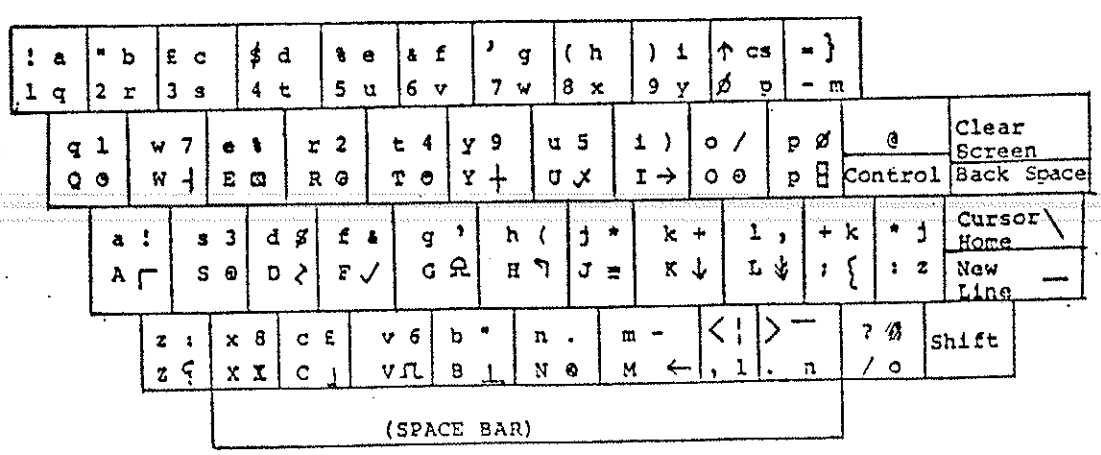
The ? command displays a list of all the commands in the command table. This provides a useful reminder of the facilities available.

3.18 ERRORS

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

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

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



KEYBOARD LAYOUT

4. RESTART VECTORS

All 8 restart vectors are available for use. A restart vector is a one byte code which calls an address in the monitor. The monitor will carry out the appropriate function.

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

5. HOW TO END A PROGRAM

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

1. Press RS to restart the system, at any time. A Halt instruction (76H) may be placed in the program to make the Halt LED light, to indicate that the program has ended.

2. Execute code C7H (RST 0) in the program. This is equivalent to pressing RS, and it restarts the system.

3. Execute code CFH (RST 8) in the program. This simply returns control to the monitor. The advantage of this over methods 1 or 2 is that the screen is not cleared. This is the recommended method for normal end of program. Note that the program registers are not saved and that the user stack pointer is set back to 1000 H.

4. Execute code E7H (RST 20 H) in the program. This stores the user registers and displays them, before returning control to the monitor. This is the recommended method for abnormal end of program, and during program testing.

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

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

6. SUPPORT OF PROGRAMS USING ASCII CONTROL CODES

Certain programs may be found which assume ASCII codes for keys such as New Line. These programs may now be run without modification. Before starting execution of the program, execute the following code:

```
CD  B4  07      Call  TASC (07B4H)
```

The monitor will then convert non-standard control codes to ASCII format.

At the end of the program, execute

```
CD  C0  07      Call  FASC (07C0H)
```

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

ASCII CONVERSION TABLE IN MONITOR

NASCOM → ASCII		ASCII → NASCOM	
New Line (1FH)	→ CR (0DH)	CR (0DH)	→ New Line (1FH)
Clear Screen (1EH)	→ Escape (1BH)	Escape (1BH)	→ Clear Screen (1EH)
Backspace (1DH)	→ BS (08H)	BS (08H)	→ BS (1DH)
Cursor home (1CH)	→ Line Feed (0AH)	LF (0AH)	→ Null (00H, ignored)
		Rubout (7FH)	→ Null (00H, ignored)

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

7. SUPPORT OF ASCII PERIPHERALS

7.1 The External (X) Command

X aa

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

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

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

0 Support a terminal in full duplex mode. Every character typed is sent back to the terminal. Line feed is automatically supplied after carriage return. All output is even parity.

10 Same as 0, but Line feed is not supplied. Suitable for use when feeding in paper tape, which already has line feeds.

20 Same as 0, but half duplex. Characters entered are not sent back. On output, Line feed is provided after Carriage return.

30 Same as 20, but Line feed is not supplied. This makes the Nascom-1 into a half duplex terminal.

1, 11, 21, 31 Same as 0, 10, 20, 30, in each case, but the output parity is odd instead of eve
(Input parity is ignored).
The chart in figure 7.1 shows the characters sent to the serial output, depending on the origin
of the characters, and the option specified. Values ringed show where problems would occur.
Underlined options are the correct ones to choose.

To store data on paper tape, use the Tabulate command.

Use the Load command to feed the data back in. When loading back, the data is printed on the terminal, if option 10 or 11 is used.

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

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

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

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

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

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

K2 Keyboard option. This is designed for the screen, and will reset the CRT pointer back to normal.

CHARACTERS SENT TO SERIAL OUTPUT

Source of Input	Input	Full duplex Controller	Full duplex Controller for paper tape input	Half duplex Controller	Half duplex Terminal
		X0	X10	X20	X30
COMMUNICATION WITH TERMINAL					
NASCOM keyboard	X,CR	X,CR,LF	X,CR	X,CR,LF	X,CR
TTY keyboard	X,CR	X,CR,LF	X,CR	-	
TTY Paper Tape	X,CR,LF	X,CR,LF (LF)	X,CR,LF	-	-
COMMUNICATION WITH COMPUTER					
NASCOM keyboard	X,CR	X,CR, (LF)	X,CR	X,CR, (LF)	X,CR
Mainframe Computer	X,CR,LF	(X,CR,LF,LF)	(X,CR,LF)	-	-

X OPTIONS

BIT

0. Output: 0 = Even parity, 1 = Odd parity
1. Ignored
2. Ignored
3. Ignored
4. Input/: Output: 0 = Supply LF after CR, 1 = Do not.
5. Input: 0 = Output all serial input, 1 = Do not output serial input
6. Ignored
7. Used internally: 1 = Suppress next serial output.

8. USING THE NASCOM AS A TERMINAL

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

```
E 794
```

To stop the program, press Reset.

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

For example, enter:

```
>K1
>E 794
```

Anything typed will appear on the screen.

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

Another example is to enter:

```
>X30
>E 794
```

The computer will now act as a half-duplex, ASCII computer terminal. Connect the serial data socket to an acoustic complex (set to full duplex!), and fit a Break key to connect the complex input to Earth instead of RS 232 out, use the 110 bps Teletype clock in the Nascom, or a 300 bps external clock, and telephone a computer timesharing service such as IBM CALL. This makes the Nascom an excellent interactive terminal. You could also write your own intelligent terminal programs of any complexity. It should be quite easy to transfer data between Nascoms by telephone, and to control one Nascom from another.

In order to use the Nascom as a full duplex terminal, the program detailed in figure 8.1 must be entered and executed.

8.1 FULL DUPLEX TERMINAL PROGRAM - XFD

P.C.	MACHINE CODE	LABEL	MNEMONIC	COMMENTS
OE00	CD6900	XFD:	CALL KBD	
OE03	3005		JR NC, XFD4	When keyboard input found, convert to ASCII and send out. (Do not display.)
OE05	CDBD04		CALL XCRTC	
OE08	18F6		JR XFD	
OE0A	CDD307	XFD4:	CALL XKBDC	Check for serial input
OE0D	30F1		JR NC, XFD	
OE0F	F7		RST 30H	Call $CRT Display input value.
OE10	18EE		JR XFD	Start again

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

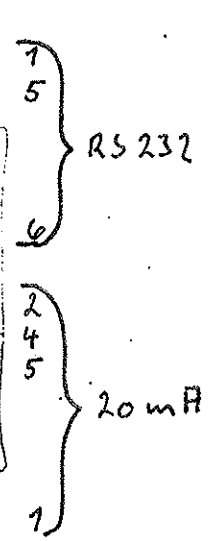
First enter command X30 (or X31), then run this program.
Press Reset to end the program.

9. CONNECTING THE NASCOM TO EXTERNAL DEVICES

9.1 NASCOM TO TERMINAL

Both 20 mA loop and RS 232 serial interfaces are brought to SK2 of the Nascom. We suggest you make a lead bringing these signals to a Cannon D type socket with the following pin allocations.

Signal Name	SK2 Name and Pin No.	Cannon Pin No.	
Chassis Ground	-	1	101
Data in V24	RS232 in (2)	2	103
Data out V24	RS232 out (14)	3●	104
Request to send (RTS)	Generated by terminal	4	105
Clear to send (CTS)	*	5	106
Data set Ready (DSR)	*	6	107
Ov (Ground)	RS232 Com (8)	7●	102
Carrier Detect	*	8	109
20mA input -ve	KBD - (4)	10	
20mA output +ve	PTR + (12)	16	
20mA output -ve	PTR - (11)	17	
Reader step -ve	-	18	
Reader step +ve	-	19	
Data Terminal Ready (DTR)	Generated by terminal*	20	108.1
20mA Input +ve	KBD + (5)	24	



* link Data set Ready, Carrier Detect and Clear to send to the Data Terminal Ready pin.

Note that the pin numbers refer to the wiring side of the female connector. (Counting left to right for each row, with the row of 13 pins on top).

When attaching a Teletype to the Nascom-1, the clock must be set to the correct speed, by adjusting the screw in VR1.

Turn the screw clockwise to make the clock slower.

If the clock is too fast, output to the terminal appears garbled.

If the clock is too slow, the output may be all right, and the terminal keyboard may work, but paper tape input will contain errors such as missing characters.

9.2 NASCOM TO MODEM

A separate lead is required, with a male connector.
Connect RS232 in and out the opposite way round because now the Nascom is acting as a terminal
Connect RS232 common. Connect Request to Send and Data Terminal Ready to +12 (approx), althou..
many complexes disregard these lines.

Data set ready, Carrier detect and Clear to send may be ignored, or, using appropriate additional components, detect the state of each with an LED. Only type in data when all three LEDs are on.

Connect a push button to the output RS232 signal, so that the input to the complex is earthed instead of being connected to the Nascom output. This then acts as a Break key.

10 USEFUL LOCATIONS

In addition to the restart jumps detailed in section 4, many subroutines are available to the user. All the major Nasbug T2 subroutine locations and reflections are preserved and can be called in the same way, and use of the additional facilities of T4 are described below.

Many of the calls use ARG1, ARG2 and/or ARG3 for transferring 16 bit parameters (locations 0C0C, 0C0E and 0C10 respectively).

10.1 Write

Location: 0400H
Parameters: ARG1 - aaaa (start address)
ARG2 - bbbb (end address)

Invokes a "Write" command under program control

10.2 Generate

Location: 06A3H
Parameters: As for write

Invokes a "Generate" command under program control.

10.3 Read

Location: 070C

Invokes a "Read" under program control.

10.4 Serial Output

SRLOUT, at 005DH, now outputs a character to the CRT as well as to the UART. Output to the UART alone is now via SRLX (005EH)

10.5 Serial Input

SRLIN (04F6H) scans the UART for input and returns with any character in A, with the carry flag set. If no character is present, the carry flag is not set.

TIN (04F2H) scans both the keyboard and UART for input, setting A and the carry flag as above

10.6 Keyboard Options

The keyboard options can be changed under program control by a call to 0634H. The param.. r a should be set in ARG1.

10.7 External

The "X" command can be invoked by a call to 047A, with aa set in ARG1.