

**NewBear Computing Store**  
A Division of NEWBURY LABORATORIES LTD.



SALES & SERVICE: 7 BONE LANE NEWBURY BERKSHIRE RG14 5SH NEWBURY (0635) 49223

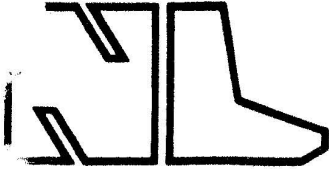
MINIMON: A soft monitor for the 7768  
MON 1 board

MINIMON (mini-monitor) is a 13 command monitor that enables the 7768 to be attached to an asynchronous terminal via the 7768 MON 1 board. MINIMON can be loaded from cassette or paper tape into the memory on the MON 1 board (FC00-FFFF) using the bootstrap loader function.

The monitor supports the following commands:

1. X,Z Relative offset calculation
2. M Examine and modify memory
3. A Alter memory
4. G Go to user program and run it
5. S Set break point
6. C Continue running a program after a break point
7. R Print 6800 registers
8. H Gives headings to print out by command R
9. B Block move
10. D Dumps a block of data to the terminal
11. P Punch unformatted binary cassette or paper tape
12. L Load from tape produced by the punch command

Issue 1 : £4.50 including cassette VAT 8%



**NewBear Computing Store**  
A Division of NEWBURY LABORATORIES LTD.



SALES & SERVICE: 7 BONE LANE NEWBURY BERKSHIRE RG14 5SH NEWBURY (0635) 49223

MINIMON: A soft monitor for the 7768  
MON 1 board.

INTRODUCTION

MINIMON is a simple 1K monitor that allows the interface of the 7768 with an asynchronous data terminal and provides a method of entering, debugging and saving programs.

It occupies the 1K space on the MON 1 board (FC00-FFFF) and uses the locations F0DD-F0FE (on the CPU board) for temporary storage. The stack used by MINIMON (may also be used by user programs) extends from F0D1 Downwards.

The ASCII send/receive terminal (assumed to operate in a full duplex mode) must be connected to ACIA (a) , or ACIA (b) if locations FC18, FD34 and FF26 are changed from 00 to 02, and locations FC11, FD05, FD2D, FF2C and FF96 are changed from 01 to 03.

USE

MINIMON can be loaded using the MON 1 Bootstrap Loader function. It is then initiated by hitting the reset button. MINIMON will respond with an asterisk and will then wait for one of the following commands:

1. X,Z Relative offset calculation
2. M Examine and modify memory
3. A Alter memory
4. G Go-run user program
5. S Set break point
6. C Continue running a program after a break point
7. R Register print-6800 registers printed
8. H Header-gives titles to print out by function R
9. B Block move

- 10. D Dumps a block of data to the terminal
- 11. P Punch-unformatted binary dump
- 12. L Load from tape produced by the punch comand

## COMMANDS

### Z (Relative offset calculation)

This command enables the calculation of branch offsets to be done automatically. MINIMON is supplied with the address of the first byte of the brach instruction and the address of destination and the offset is then calculated according to the formula:

$$\text{OFFSET} = \text{DESTINATION} - (\text{PC} + 2)$$

Format (user input underlined)

\*Z P aa D bb R=cc

where aa is the least significant two HEX digits of the branch instruction address, bb is the LSB of the destination address. cc is the offset.

Attempts to branch more than 127 bytes forward or 128 bytes back are indicated by RANGE! being printed.

### X (Relative offset calculation)

This command is similar to Z except that the addresses are the full 16 bit, 4 HEX digits to be typed. eg:

\*XP 00F0 D 0110 R=1E

### A(Alter)

If A is typed while the monitor is waiting for a comand ('\*'printed) MINIMON will reply with a space which is a prompt for an address. The user should type the address (in HEX) of the first location to be altered. The monitor will then print this address on the next line, The user can then type in the new contents of that location. The comand can then be terminated by pressing full stop, or the next memory location

can be altered by typing two HEX digits.

FORMAT

\*A 0000

0000 aabbccdd... Where aa, bb, cc, dd etc. are the data entered into locations 0000, 0001, 0002, 0003 etc. respectively.

When the end of a line is reached MINIMON will print the current address at the beginning of the next line and the command is continued.

### M (Modify)

The modify command can be used for both examining and altering memory. The function is entered by typing an M, MINIMON will respond with a space and then wait for the HEX address to be typed in. The monitor will print the contents of this location and then wait for further inputs which may be any of the following:

- 1 A space which will cause the monitor to print the contents of the next location.
- 2 A slash '/' which will cause the monitor to print the contents of the previous location and its address on the next line.
- 3 A carriage return (CR) which will cause MINIMON to print the contents of the next location and its address on the next line.
- 4 Two HEX digits to be stored in the location whose contents was last printed.

FORMAT

\*M 0000 aa bb/  
0000 aa (CR)  
0001 bbcc dd

Where aa, bb are the contents of 0000, 0001 respectively. cc was entered into location 0001 in place of bb.

When the end of a line is reached MINIMON will continue the modify function at the beginning of the next line.



The modify function, as with all commands, can be left at any time by typing a full stop.

### G (Go)

The G function enables the user program to be entered at the specified address and run from that point. After the G command has been entered the user should type the HEX address of the start of the program. This can then be checked and if it is incorrect type a full stop to exit the function (with no harm done). If it is correct any character (apart from a full stop) will cause the system to run the user program until it encounters a SWI (3F) or the system reset button is pressed.

#### FORMAT

\*G aaaaf           Where aaaa is the address of the start of the user program. Any character was then typed to run the program.

### S (Set break point)

Using this command a program can be stopped at any predetermined point. The user selects this point and types in its address. MINIMON will then save the instruction at that point and replace it with a SWI (3F). Two break points may be set at any one time.

#### FORMAT

\*S1 aaaa           Where aaaa is the address of break point 1  
OR

\*S2 bbbb           Where bbbb is the address of break point 2

### C (Continue)

After a program has been interrupted (using the S command) registers, memory, etc. may be examined and altered. Then to set the program running again from the point it left off type either C1 or C2 according to which break point was encountered.

FORMAT

\*C1

OR

\*C2

### R (Register print)

If a break point (SWI) is encountered while a user program is running, the 6800 registers will be pushed on to the stack, and control will be passed to MINIMON. Using the R command the registers can then be printed from the stack.

FORMAT

\*R CC B A X PC SP

Where CC=Condition Codes, B and A are the accumulators, X=Index reg., PC=Program Counter, SP=Stack Pointer.

These registers can then be altered on the stack using the Modify command, the HEX address is SP+1.

FORMAT

\*R CF 0B 34 0004 0017 F0D0

\*M F0D1 CF 0B00 34 00 04.      Where acc. B was changed from  
0B to 00.

The values of these registers (Altered or not) will then be reloaded into the 6800 MPU when the next G or C command is given.

### H (Header)

This command gives titles to the print out by the register print instruction.

FORMAT

\*H CC B A X PC SP

\*R CF 0B 34 0004 0017 F0D0

### B (Block Move)

Using the Block move function a block of data or program of any size can be moved to any location. For example.

The block of data with starting address  $\text{\$}100$  and finishing address  $\text{\$}110$  could be moved one location up, so that its new starting address would be  $\text{\$}101$ . This could be done as follows:

\*B S  $\text{\$}100$  F  $\text{\$}110$  TO  $\text{\$}101$

where S=Start address of block, F=Finishing address of block, and the last entry is the new start address of the block. (All entries are in HEX)

NOTE: Block move can also be used to move a block (say  $\text{\$}100$  to  $\text{\$}110$ ) one or more locations down (say to the new starting address  $\text{\$}0FF$ ) as follows:

\*B S  $\text{\$}100$  F  $\text{\$}110$  TO  $\text{\$}0FF$

In each case, although the original block will be overwritten, the relocated block will be faithfully reproduced.

#### D (Dump)

MINIMON will dump the contents of memory between two specified addresses to the terminal. There are 8 (typically but can be adjusted-see listing) Bytes per line with the address of the first Byte on each line printed first. The dump can be stopped at any time by typing any character at the terminal keyboard.

#### FORMAT

\*D S aaaa F bbbb

aaaa cc cc cc cc ...

Where aaaa is the address of the first Byte, bbbb the add. of the last Byte of the area of data to be printed. cc was the data found.

#### PUNCH

Program or data can be dumped (unformatted binary) onto casset or paper tape using this command. Note MINIMON will punch two ASCII S's at the beginning of each data dump as a start of file indicator.

#### FORMAT

\*P S aaaa F bbbb

Where aaaa, bbbb are the boundary addresses.





# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM <i>MINIMON</i>		VERSION	AUTHOR <i>ACH</i>	DATE <i>DEC. '78</i>	PAGE <i>1</i> OF <i>14</i>
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
		<i>CTRLA</i>	<i>EQU \$F401</i>	<i>ACIA(a) CTRL/STATUS</i>	
		<i>CTRLB</i>	<i>EQU \$F403</i>	<i>ACIA(b) " "</i>	
		<i>DATAA</i>	<i>EQU \$F400</i>	<i>ACIA(a) DATA REG.</i>	
		<i>DATAB</i>	<i>EQU \$F402</i>	<i>ACIA(b) " "</i>	
		<i>STACK</i>	<i>EQU \$F000</i>	<i>TOP OF MINIMON'S STACK</i>	
		<i>JNMI</i>	<i>EQU \$F000</i>	<i>SPACE FOR JUMP TO NMI SUB.</i>	
		<i>JIRQ</i>	<i>EQU \$F0E0</i>	<i>" " " " IRQ "</i>	
		<i>SAVE</i>	<i>EQU \$F0E3</i>	<i>TEMP. STORAGE FOR NEWLINE</i>	
		<i>P</i>	<i>EQU \$F0E5</i>	<i>" " " Z,X</i>	
		<i>X</i>	<i>EQU \$F0E6</i>	<i>" " " SUB</i>	
		<i>Y</i>	<i>EQU \$F0E8</i>	<i>" " " SUB</i>	
		<i>NEW</i>	<i>EQU \$F0EA</i>	<i>" " " BLOCK MOVE</i>	
		<i>M</i>	<i>EQU \$F0EC</i>	<i>" " " ZOUT</i>	
		<i>TEMPX</i>	<i>EQU \$F0E0</i>	<i>" " " PRX</i>	
		<i>START</i>	<i>EQU \$F0EF</i>	<i>" " " GETADDRESS</i>	
		<i>STOP</i>	<i>EQU \$F0F1</i>	<i>" " " GETADDRESS</i>	
		<i>Z</i>	<i>EQU \$F0F3</i>	<i>" " " . RDX</i>	
		<i>R</i>	<i>EQU \$F0F5</i>	<i>" " " PR</i>	
		<i>Q</i>	<i>EQU \$F0F6</i>	<i>" " " ZIN, DUMP</i>	
		<i>ABYTE</i>	<i>EQU \$F0F7</i>	<i>" " " SET BR. PT.</i>	
		<i>BYTE</i>	<i>EQU \$F0FB</i>	<i>" " " SET BR. PT.</i>	
		<i>PSTACK</i>	<i>EQU \$F0F0</i>	<i>" " " SWI</i>	
<i>F.C.00</i>	<i>B.0.F.0.2B</i>	<i>RD</i>	<i>JSR RDB</i>	<i>INPUT ONE CHARACTER</i>	
<i>03</i>	<i>2.7.F.8</i>		<i>BEQ RD</i>	<i>IGNOR PAPER TAPE FOLLOWER</i>	
<i>05</i>	<i>8.0.0.8</i>		<i>BSR PR</i>	<i>ECHO CHARACTER</i>	
<i>07</i>	<i>8.1.2.E</i>		<i>CMP A #'.</i>	<i>WAS CHARACTER A FULLSTOP</i>	
<i>09</i>	<i>2.6.0.3</i>		<i>BNE END</i>		
<i>0B</i>	<i>7.E.F.F.8.F</i>		<i>JMP START</i>	<i>YES - GOTO START OF MINIMON</i>	
<i>0E</i>	<i>3.9.</i>	<i>END</i>	<i>RTS</i>	<i>NO - RETURN</i>	
<i>0F</i>	<i>F.6.F.4.0.1</i>	<i>PR</i>	<i>LDA B CTRLA</i>	<i>GET ACIA(a) CTRL BYTE</i>	
<i>12</i>	<i>C.5.0.2</i>		<i>BIT B #02</i>	<i>IS ACIA(a) BUSSY?</i>	
<i>14</i>	<i>2.7.F.9</i>		<i>BEQ PR</i>	<i>YES - TRY AGAIN</i>	
<i>16</i>	<i>B.7.F.4.0.0</i>		<i>STA A DATAA</i>	<i>NO - SEND DATA</i>	
<i>19</i>	<i>7.C.F.0.F.5</i>		<i>INC R</i>	<i>CHARACTERS PRINTED + ONE</i>	
<i>1C</i>	<i>3.9.</i>		<i>RTS</i>	<i>RETURN TO CALLER</i>	

→

→



# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

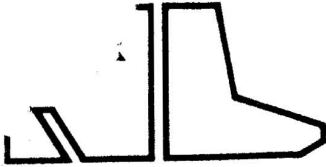
BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM		VERSION		AUTHOR	DATE	PAGE
MINIMON				ACH	DEC '78	20 of 44
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS		
F.C.1.D	8.1.2.F	VHEX	CMP A #2F	A < 30;		
1.F	2.F0E		BLE NO	NOT HEX		
21	8.1.3.9		CMP A #39	A > 39;		
23	2.20.2		BHI N	NOT A NUMERAL		
25	0.C	YES	CLC	ITS OK		
26	3.9		RTS	SO RETURN		
27	8.1.4.0	N	CMP A #40	A < 41		
29	2.F0.4		BLE NO	NOT HEX		
2B	8.1.4.6		CMP A #46	A ≤ 46		
2D	2.F.F.6		BLE YES	THEN ITS HEX		
2F	0.0	NO	SEC	IF NOT SET CARRY		
30	3.9		RTS	AND RETURN		
31	8.5.3.0	BINARY	BIT A #30	IS ACC A A LETTER?		
33	2.7.0.C		BEQ LETT			
35	4.8	R10A	ASL A	ITS A NUMBER SO		
36	4.8		ASL A	GET RID OF MOST		
37	4.8		ASL A	SIGNIFICANT 4 BITS.		
38	4.8		ASL A			
39	C.5.3.0		BIT B #30	IS ACC B A LETTER?		
3B	2.7.0.8		BEQ LETTER			
3D	C.4.0.F	R10B	AND B #0F	DESTROY MS 4 BITS.		
3F	1.B		ABA	FORM BINARY CHAR.		
40	3.9		RTS	RETURN		
41	8.B.0.9	LETT	ADD A #09	MAKE ACC A BINARY		
43	2.0.F.0		BRA R10A	AS BEFORE		
45	C.B.0.9	LETTER	ADD B #09	MAKE ACC B BINARY		
47	2.0.F.4		BRA R10B	AS BEFORE		
49	8.0.B.5	ZIN	BSR R0	GET A CHARACTER		
4B	8.0.D.0		BSR VHEX	IS IT 0 → 9 or A → F?		
4D	2.5.1.0		BCS ?	NO - ITS RUBBISH SO ?		
4F	B.7.F.0.F.6		STA A Q	YES - ITS OK SO SAVE IT		
52	8.0.A.C		BSR R0	GET SECOND CHARACTER		
54	8.0.C.7		BSR VHEX	IS IT HEX		
56	2.5.0.7		BCS ?	NO - PRINT A'?		
58	1.6		TAB	YES - PUT IT INTO ACC B		





# NewBear Computing Store

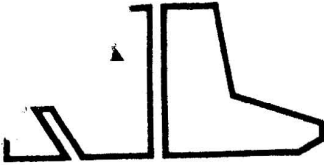
A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM MINIMON		VERSION	AUTHOR ACH	DATE DEC '78	PAGE 3 of 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
FC59	86F0F6		LDA A Q	RETRIEVE FIRST CHARACTER	
5C	8003		BSR BINARY	CONVERT ACCA,B TO BINARY	
5E	39		RTS	RETURN	
5F	863F	?	LDA A #'?	LOAD ACCA WITH ASCII '?'	
61	80AC		BSR PR	PRINT IT	
63	20E4		BRA ZIN	GO BACK TO START	
65	80FD40	RDX	JSR PR	PRINT A SPACE ( )	
68	800F		BSR ZIN	INPUT TWO HEX CHARACTERS	
6A	87F0F3		STA A Z	SAVE BYTE IN MS. Z	
6D	800A		BSR ZIN	GET LS. HEX CHARACTERS	
6F	87F0F4		STA A Z+1	STORE IN LS. Z	
72	FEF0F3		LDX Z	X ← Z	
75	39		RTS	RETURN	
76	30	STRING	TSX	POINT X AT LOCATION ON	
77	EE00		LDX X	STACK WHERE RETURN	
79	09		DEX	ADD. IS HELD. LOAD X WITH	
7A	08	AGAIN	INX	RETURN ADD.	
7B	A600		LDA A X	GET DATA TO BE PRINTED	
7D	81FF		CMP A #FF	LAST BYTE?	
7F	2704		BEQ END	YES - GO TO THE END	
81	808C		BSR PR	NO - PRINT BYTE	
83	20F5		BRA AGAIN		
85	31	END	INS	CLEAN UP STACK	
86	31		INS		
87	6E01		JMP X,01	RETURN TO CALLER	
89	80EB	GETADD.	BSR STRING	PRINT:	
8B	2053FF			(SPACE), S	
8E	8005		BSR RDX	INPUT 4 DIGIT HEX ADDRESS	
90	FFF0EF		STX START	STORE IT IN START	
93	80E1		BSR STRING	PRINT:	
95	2046FF			(SPACE), F	
98	80CB		BSR RDX	INPUT 4 DIGIT HEX ADDRESS	
9A	FFF0F1		STX STOP	STORE IT IN STOP	
9D	39		RTS	RETURN TO CALLER	
9E	16	ASCII	TAB	ACCB ← ACCA	



# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

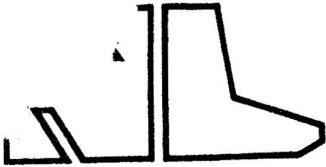
BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



HEXADECIMAL CODING FORM

PROGRAM <i>MINIMON</i>		VERSION	AUTHOR <i>ACH</i>	DATE <i>DEC '78</i>	PAGE <i>4</i> OF <i>14</i>
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
<i>FC9F</i>	<i>84F0</i>		<i>AND A #F0</i>		
<i>A1</i>	<i>44</i>		<i>LSR A</i>	<i>SHIFT MOST SIGNIFICANT</i>	
<i>A2</i>	<i>44</i>		<i>LSR A</i>	<i>4 BITS DOWN.</i>	
<i>A3</i>	<i>44</i>		<i>LSR A</i>		
<i>A4</i>	<i>44</i>		<i>LSR A</i>		
<i>A5</i>	<i>8109</i>		<i>CMP A #09</i>	<i>LETTER OR NUMBER?(MS)</i>	
<i>A7</i>	<i>220B</i>		<i>SHI SUBT</i>		
<i>A9</i>	<i>8B30</i>		<i>ADD A #30</i>	<i>MAKE IT A NUMBER(ASCII)</i>	
<i>A8</i>	<i>C40F</i>	<i>AND</i>	<i>AND B #0F</i>	<i>GET RID OF MS 4 BITS</i>	
<i>A0</i>	<i>C109</i>		<i>CMP B #09</i>	<i>LETTER OR NUMBER?(LS)</i>	
<i>AF</i>	<i>2207</i>		<i>SHI ADD</i>		
<i>B1</i>	<i>C830</i>		<i>ADD B #30</i>	<i>MAKE IT AN ASCII NUMBER</i>	
<i>B3</i>	<i>39</i>		<i>RTS</i>	<i>FINNISHED</i>	
<i>B4</i>	<i>8B37</i>	<i>SUBT</i>	<i>ADD A #37</i>	<i>MAKE IT AN ASCII LETTER</i>	
<i>B6</i>	<i>20F3</i>		<i>BRA AND</i>	<i>THEN DO LS CHARACTER</i>	
<i>B8</i>	<i>C837</i>	<i>ADD</i>	<i>ADD B #37</i>	<i>MAKE IT AN ASCII LETTER</i>	
<i>BA</i>	<i>39</i>		<i>RTS</i>	<i>RETURN</i>	
<i>BB</i>	<i>FFF0E0</i>	<i>PRX</i>	<i>STX TEMPX</i>	<i>STORE THE INDEX REG.</i>	
<i>BE</i>	<i>66F0ED</i>		<i>LDA A TEMPX<sub>H</sub></i>	<i>ACCA ← X<sub>H</sub></i>	
<i>C1</i>	<i>8006</i>		<i>BSR ZOUT</i>	<i>PRINT ACCA IN HEX</i>	
<i>C3</i>	<i>B6F0EE</i>		<i>LDA A TEMPX<sub>L</sub></i>	<i>ACCA ← X<sub>L</sub></i>	
<i>C6</i>	<i>8001</i>		<i>BSR ZOUT</i>	<i>PRINT ACCA IN HEX</i>	
<i>C8</i>	<i>39</i>		<i>RTS</i>	<i>RETURN</i>	
<i>C9</i>	<i>80D3</i>	<i>ZOUT</i>	<i>BSR ASCII</i>	<i>ACCA(BINARY) → ACCA,B(ASCII)</i>	
<i>CB</i>	<i>F7F0EC</i>		<i>STA B M</i>	<i>SAVE ACCB</i>	
<i>CE</i>	<i>BDFC0F</i>		<i>JSR PR</i>	<i>PRINT ACCA</i>	
<i>D1</i>	<i>86F0EC</i>		<i>LDA A M</i>	<i>RETRIEVE ACCB</i>	
<i>D4</i>	<i>BDFC0F</i>		<i>JSR PR</i>	<i>PRINT IT</i>	
<i>D7</i>	<i>39</i>		<i>RTS</i>	<i>RETURN</i>	
<i>D8</i>	<i>BDFC89</i>	<i>PUNCH</i>	<i>JSR GETAD<sub>H</sub></i>	<i>GET BOUNDARY ADDRESSES</i>	
<i>DB</i>	<i>CE0000</i>		<i>LOX #0000</i>	<i>WAIT APPROXIMATELY</i>	
<i>DE</i>	<i>80BE</i>	<i>LOOP</i>	<i>BSR ASCII</i>	<i>5 SECS.</i>	
<i>E0</i>	<i>08</i>		<i>INX</i>		
<i>E1</i>	<i>26FB</i>		<i>BNE LOOP</i>		
<i>E3</i>	<i>8D14</i>		<i>BSR BEGIN</i>	<i>START OF RECORD</i>	





# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

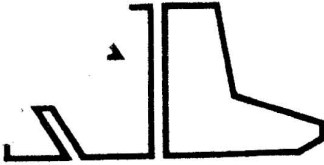
BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM MINIMON		VERSION	AUTHOR ACH	DATE DEC '78	PAGE 5 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
FC	ES FE F0 EF		LDX START	POINT X AT DATA TO	
	E8 09		DEX	BE PUNCHED.	
	E9 08	WRITE	INX		
	EA A6 00		LDA A X	GET BYTE	
	EC B0 FC 0F		JSR PR	PUNCH IT *	
	EF 0C F0 F1		CPX STOP	LAST BYTE?	
	F2 26 F5		BNE WRITE	NO - THEN REPEAT	
	F4 8D 35		BSR ROB	YES - WAIT FOR INPUT	
	F6 7E FF 8F		JMP START	FROM TERMINAL; RETURN	
	F9 86 53	BEGIN	LDA A #53	PUNCH TWO 'S' TO	
	FB 6D FC 0F		JSR PR	INDECTATE START OF	
	FE 8D FC 0F		JSR PR	DATA BLOCK ON TAPE	
	FD 01 86 11	BIN	LDA A #11	SET UP ACIA(a) FOR:	
→	03 67 F4 01		STA A CTRLA	8 BITS + 2 STOP BITS +	
	06 39		RTS	÷ 16 CLOCK.	
	07 6D FC 89	LOAD	JSR GETADD	GET BOUNDARY ADDRESSES	
	0A 8D 1F	NEXT	BSR ROB	BEGIN TO READ DATA	
	0C 81 53		CMP A #53	BUT DON'T STORE IT	
	0E 26 FA		BNE NEXT	UNTILL TWO 'S' (ASCII)	
	10 8D 19		BSR ROB	HAVE BEEN READ.	
	12 81 53		CMP A #53		
	14 26 F4		BNE NEXT		
	16 8D E9		BSR BIN	THEN SET UP ACIA(a)	
	18 FE F0 EF		LDX START	POINT X AT PROGRAM	
	1B 09		DEX	AREA	
	1C 08	READ	INX		
	1D 8D 04		BSR ROB	INPUT BYTE	
	1F A7 00		STA A X	STORE IT	
	21 0C F0 F1		CPX STOP	LAST BYTE?	
	24 26 F6		BNE READ	NO - CARRY ON	
	26 7E FF 8F		JMP START	YES - GOTO START OF MINIMON	
	29				
→	2B 66 F4 01	ROB	LDA A CTRLA	ACIA(a) RECIVED A CHAR.	
	2E 85 01		BIT A #01		
	30 27 F9		BEQ ROB	NO - TRY AGAIN	

copy  
→



# New Bear Computing Store

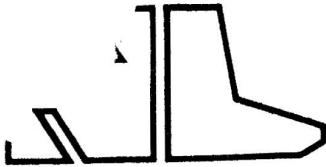
A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM MINIMON		VERSION	AUTHOR ACH	DATE DEC '78	PAGE 6 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
F032	66F400		LDA A DATAA	YES - GET IT	
35	39		RTS	AND RETURN	
36	8D15	REG.PRINT	BSR PR0	PRINT A SPACE	
38	FEF0FD		LDX PSTACK	POINT X REG. AT USERS STACK	
3B	8D14		BSR PR2	PRINT CC	
3D	8D1A		BSR PR2	" B	
3F	8D18		BSR PR2	" A	
41	8D10		BSR PR4	" X	
43	8D0E		BSR PR4	" PC	
45	CEF0FD		LDX #PSTACK	POINT X AT 'STACK POINTER'	
48	8D0A		BSR 4PR	PRINT SP	
4A	7EFF8F		JMP START	RETURN TO MINIMON	
4D	8620	PR0	LDA A #20	LOAD ACCA WITH AN ASCII	
4F	6DFC0F		JSR PR	SPACE AND PRINT IT	
52	39		RTS	RETURN	
53	08	PR4	INX	X ← X + 1	
54	A600	4PR	LDA A X	GET DATA(X)	
56	6DFCC9		JSR 2OUT	PRINT IT	
59	08	PR2	INX	X ← X + 1	
5A	A600		LDA A X	GET DATA(X)	
5C	6DFCC9		JSR 2OUT	PRINT IT	
5F	8DEC		BSR PR0	PRINT A SPACE	
61	39		KTS	RETURN	
62	8DFC89	BLOCKMONE	JSR GETADD	GET BLOCK LIMITS	
65	8DFC76		JSR STRING	PRINT:	
68	20544F			D TO	
6B	FF				
6C	8DFC65		JSR RDX	GET NEW START ADDRESS	
6F	FFF0EA		STX NEW	STORE IT	
72	FFF0E6		STX X	ALSO PUT IT IN X	
75	FEF0EF		LDX START	LOAD X REG WITH START	
78	FFF0E8		STX Y	TRANSFER IT TO Y	
7B	8D59		BSR SUB	DO: X - Y	
7D	2A23		BSPL DOWN	IF RESULT IS +VE THEN ITS	
7F	FEF0F1		LDX STOP	DOWN.	



# NewBear Computing Store

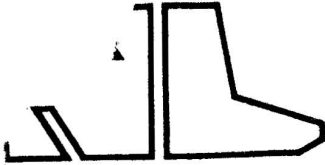
A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



HEXADECIMAL CODING FORM

PROGRAM		VERSION		AUTHOR	DATE	PAGE	OF
MINIMON				ACH	DEC '78	7	14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND		COMMENTS		
F082	08		INX		ADD 1 TO STOP		
83	FFFD F1		STX	STOP	AND STORE IT		
86	FEFD EF	DO	LDX	START	GET START		
89	BCFD F1		CPX	STOP	COMPARE IT WITH STOP		
8C	2603		BNE	CARRYON	FINNISHED?		
8E	7EFF8F	END	JMP	START	YES - RETURN TO MINIMON		
91	A600	CARRYON	LDA	A X	NO - GET DATA		
93	08		INX		INCREMENT START		
94	FFFD EF		STX	START	STORE IT		
97	FEFD EA		LDX	NEW	LOAD X REG WITH NEW		
9A	A700		STA	A X	STORE DATA		
9C	08		INX		INCREMENT NEW		
9D	FFFD EA		STX	NEW	STORE IT		
AD	20E4		BRA	DO	ROUND AGAIN		
A2	FEFD F1	DOWN	LDX	STOP	LOAD X REG WITH STOP		
A5	FFFD EG		STX	X	PUT IT IN X		
A8	8D2C		BSR	SUB	ACCA, B ← X - Y		
AA	BBFD EB		ADD	A NEWL			
AD	FBFD EA		ADD	B NEWH	NEW ← NEW + ACCA, B		
BD	B7FD EB		STA	A NEWL			
B3	F7FD EA		STA	B NEWH			
B6	FEFD EF		LDX	START	GET START		
B9	09		DEX		'DECREMENT IT		
BA	FFFD EF		STX	START	SAVE START		
BD	FEFD F1	DOIT	LDX	STOP	LOAD X REG WITH STOP		
C0	BCFD EF		CPX	START	COMPARE IT WITH START		
C3	22C9		BEG	END	IF EQUAL THEN END		
C5	A600		LDA	A X	ACC A ← X STOP		
C7	09		DEX		STOP ← STOP - 1		
C8	FFFD F1		STX	STOP			
CB	FEFD EA		LDX	NEW	LOAD X REG WITH NEW		
CE	A700		STA	A X	STORE DATA		
D0	09		DEX		NEW ← NEW - 1		
D1	FFFD EA		STX	NEW	STORE NEW		
D4	20E7		BRA	DOIT	AND AGAIN		



# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

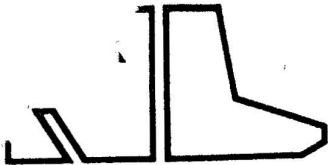
BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM		VERSION		AUTHOR	DATE	PAGE
MINIMON				ACH	DEC '78	8 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS		
F0,06	CE,F0,E6	SUB	LOX #X	POINT XREG AT DATA-		
D9	A6,01		LOA A XL	AREA. ACCA ← XL		
06	EG,00		LOA B XH	ACCB ← XH		
00	A0,03		SUB A YL	ACCA ← ACCA - YL		
0F	E2,02		SBC B YH	ACCB ← ACCB - YH - C		
E1	39		RTS	RETURN		
E2	8D,4F	Z	BSR DP	PRINT DP FOR PC		
E4	B0,FC,49		JSR ZIN	GET PC ADDRESS		
E7	B7,FO,ES		STA A P	SAVE IT		
EA	8D,4F		BSR DP	PRINT DP FOR DESTINATION		
EC	B0,FC,49		JSR ZIN	GET D ADDRESS		
EF	B0,FO,ES		SUB A P	D ← D - P		
F2	23,17		BLS BACK	BRANCHING BACK OR FORWARD?		
F4	80,02	FORWARD	SUB A #02	FORWARD: D ← D - 2		
F6	2B,17		BMI ERROR	IF D > 127 THEN ERROR!		
F8	B7,FO,ES	PRINT	STA A P	SAVE D		
F0	B0,FC,76		JSR STRING	PRINT:		
FE	20,52,3D			DR =		
FE,01	FF					
02	B6,FO,ES		LOA A P	RETRIEVE D		
05	B0,FC,49		JSR ZOUT	PRINT IT		
08	7E,FF,8F	END	JMP START	RETURN TO MINIMON		
00	80,02	BACK	SUB A #02	BACKWARDS: D ← D - 2		
00	2B,E9		BMI PRINT	IF D ≥ 128 THEN OK SO PRINT		
0F	B0,FC,76	ERROR	JSR STRING	PRINT:		
12	20,52,41					
15	4E,47,45			RANGE!		
18	21,FF					
1A	20,EC		BRA END	FINNISH		
1C	80,15	X	BSR DP	PRINT P FOR PL		
1E	B0,FC,65		BSR R0X	GET P (16 BIT)		
21	FF,FO,EB		STX Y	STORE P IN Y		
24	80,15		BSR DP	PRINT D FOR DESTINATION		
26	B0,FC,65		JSR R0X	GET D (16 BIT)		
29	FF,FO,EB		STX X	STORE D IN X		





# NewBear Computing Store

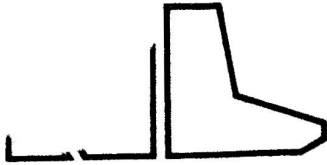
A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



HEXADECIMAL CODING FORM

PROGRAM MINIMON		VERSION	AUTHOR ACH	DATE DEC '78	PAGE 9 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
F E 2 C	60 F 0 0 6		JSR SUB	ACC ← X - Y	
2 F	2 B 0 A		BMI BACK	BACK OR FORWARD?	
3 1	2 0 C 1		BRA FORWARD		
3 3	60 F C 7 6	VP	JSR STRING	PRINT:	
3 6	2 0 5 0 2 0			VP	
3 9	F F				
3 A	3 9		RTS	RETURN	
3 B	60 F C 7 6	VD	JSR STRING	PRINT:	
3 E	2 0 4 4 2 0			VD	
4 1	F F				
4 2	3 9		RTS	RETURN	
4 3	60 F C 6 5	MODIFY	JSR K 0 X	GET ADDRESS	
4 6	60 F 0 4 0	STRT	JSR A R	PRINT A SPACE	
4 9	7 F F 0 F 5	<del>STRT</del>	CLR R	R ← 0	
4 C	A 6 0 0	PR	LDA A X	GET DATA	
4 E	60 F C C 9		JSR Z 0 V T	PRINT IT	
5 1	60 F C 0 0	WAITBIT	JSR K 0	GET COMMAND	
5 4	8 1 2 0	WHAT	CMP A # 2 0	IS IT A SPACE?	
5 6	2 6 0 C		BNE /?	NO - WHAT THEN	
5 8	0 8	CARRYON	INX	YES - INC X	
5 9	6 6 F 0 F 5		LDA A R	ACC A ← R	
5 C	8 1 1 6		CMP A # 1 6	AT END OF THE LINE?	
5 E	2 F E C		BLE PR	NO - CARRY ON	
6 0	8 0 3 5	NEW	JSR NEWLINE	YES - PRINT A NEW LINE	
6 2	2 0 E 2		BRA <del>STRT</del>	CARRY ON	
6 4	8 1 2 F	/?	CMP A # 2 F	IS IT A SLASH?	
6 6	2 6 0 3		BNE CR?	NO - WHAT THEN?	
6 8	0 9		DEX	YES - X ← X - 1	
6 9	2 0 F 5		BRA NEW	CARRY ON	
6 B	8 1 0 0	CR?	CMP A # 0 0	IS IT A CARRIAGE RETURN	
6 D	2 6 0 3		BNE DATA?	NO - WHAT THEN?	
6 F	0 8		INX	YES - X ← X + 1	
7 0	2 0 E E		BRA NEW	CARRY ON	
7 2	60 F C 1 0	DATA?	JSR VHEX	IS IT A VALID HEX CHAR.	
7 5	2 5 1 9		BCC ?	NO - PRINT ?	



# NewBear Computing Store

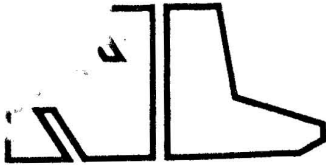
A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



HEXADECIMAL CODING FORM

PROGRAM MINIMON		VERSION	AUTHOR ALH	DATE DEC '78	PAGE/DOF/14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
FE,77	B7,F0,F6		STA A Q	YES - SO SAVE IT	
7A	BD,FC,00		JSR R0	GET SECOND CHARACTER	
7D	BD,FC,10		JSR VHEX	IS IT HEX ?	
80	25,02		KCS WHAT	NO - THEN WHAT IS IT	
82	16		TAB	YES - ARRANGE FOR	
83	B6,F0,F6		LDA A Q	CONVERSION TO BINARY	
86	BD,FC,31		JSR BINARY	CONVERT	
89	A7,00		STA A X	STORE IT AT THE RIGHT	
8B	BD,FD,4D		JSR PR	PLACE, PRINT A SPACE	
8E	20,C8		BRA CARRYON	CARRY ON	
90	86,3F	?	LDA A #3F	ACCA ← ASCII ?	
92	BD,FC,0F		JSR PR	PRINT ACCA	
95	20,BA		BRA WAIT	CARRY ON,	
97	FF,F0,E3	NEWLINE	STX SAVE	SAVE X REG	
9A	BD,FC,76		JSR STRING	PRINT:	
9D	00,0A,00			CR, LF, NULL	
A0	FF				
A1	FE,F0,E3		LDX SAVE	RESTORE X REG	
A4	BD,FC,BB		JSR PRX	PRINT IT	
A7	39		RTS	RETURN	
A8	BD,FC,65	ALTER	JSR R0X	GET START ADDRESS	
AB	7F,F0,FS	NEW	CLR R	R ← 0	
AE	8D,E7		JSR NEWLINE	PRINT 'NEWLINE'	
B0	BD,F0,4D		JSR PR	PRINT SPACE	
B3	BD,FC,49	GET	JSR ZIN	GET DATA	
B6	A7,00		STA A X	STORE IT	
B8	08		INX	X ← X + 1	
B9	B6,F0,FS		LDA A R	ACCA ← R	
BC	81,26		CMP A #26	AT END OF LINE ?	
BE	2F,F3		BLE GET	NO - CARRY ON	
C0	20,E9		BRA NEW	YES - PRINT A NEW LINE	
C2	BD,FC,65	GO	JSR R0X	GET PROGRAM START ADD.	
C5	BD,FC,00		JSR RJ	RIGHT ADDRESS ?	
C8	B6,F0,F4		LDA A ZL	ACC ← ADDRESS	
CA	F6,F0,F3		LDA B ZH		



# NewBear Computing Store

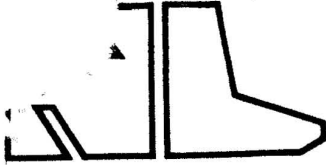
A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



HEXADECIMAL CODING FORM

PROGRAM		VERSION	AUTHOR	DATE	PAGE
MINIMON			ACH	DEC '78	11 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
FECE	FEF0FD	GOTO	LDX PSTACK	LOAD XREG WITH PSTACK	
01	A707		STA A X07	PUT NEW RETURN	
03	E706		STA B X06	ADDRESS ON STACK	
05	BEF0FD		LDS PSTACK	SET UP PROGRAMS STACK	
08	3B		RTI	GO TO USER PROGRAM	
09	BDFC00	CONTINUE	JSR RD	1 OF 2	
0C	8131		CMP A #31	IS IT A 1?	
0E	2610		BNE 2?	NO-2 THEN	
E0	FEF0F7		LDX A BYTE1	YES-GET ADDRESS OF BR.PT.	
E3	86F0FB		LDA A BYTE1	GET SAVED OPP. CODE	
E6	A700		STA A X	PUT IT BACK IN THE PROG.	
E8	86F0FB		LDA A A BYTE1	GET ADDRESS AT WHICH	
E0	F6F0F7		LDA B A BYTE1	TO RE-ENTER PROGRAM	
EE	20DE		BRA GOTO	GO TO USERS PROGRAM	
F0	8132	2?	CMP A #32	IS IT A 2?	
F2	2703		BEQ 2	YES-OK	
F4	7EFF8F		SWP START	NO-RETURN TO MINIMON	
F7	FEF0F9	2	LDX A BYTE2	GET ADDRESS OF BR. PT 2	
FA	86F0FC		LDA A BYTE2	GET OLD OPP. CODE	
F0	A700		STA A X	PUT IT BACK	
FF	86F0FA		LDA A A BYTE2	LOAD ADDRESS TO START	
FF02	F6F0F9		LDA B A BYTE2	RUNNING PROGRAM AT	
05	20C7		BRA GOTO	RUN USER PROGRAM.	
07	8DFC89	DUMP	JSR GETADD.	GET BOUNDARY ADDRESSES	
0A	FEF0EF		LDX START		
0D	80FE97	NEW	JSR NEWLINE	PRINT A NEW LINE	
10	8608		LDA A #08		
12	87F0F6		STA A Q	Q ← 8	
15	09		DEX		
16	08	NEXT	INX		
17	80FD40		JSR PRG	PRINT A SPACE	
1A	A600		LDA A X	GET DATA	
1C	80FC09		JSR ZOUT	PRINT IT	
1F	8CF0F1		CPX STOP	FINNISHED ?	
22	2606		BNE CARRYON	NO-CARRY ON	



# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

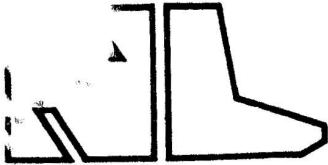
BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM MINIMON		VERSION	AUTHOR ACH	DATE DEC '79	PAGE 2 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
FF24	B6.F400	END	LDA A DATAA		
27	7EFF8F		JMP START	RETURN TO MINIMON	
2A	B6.F401	CARRYON	LDA A CTRLA		
2D	8501		BIT A #01	STOP DUMP?	
2F	26.F3		BNE END	YES-OK	
31	7AF0F6		DEC Q	NO-CARRY ON, NEW LINE?	
34	26E0		BNE NEXT	NO-OK	
36	08		INX	YES-PRINT NEXT BYTE	
37	20D4		BRA NEW	ON A NEW LINE	
39	BDFC00	BR.PT.SET	JSR RD	IS IT A 1 OR A 2?	
3C	8131		CMP A #31		
3E	2612		BNE 2?		
40	BDFC65		JSR RDX	ITS A 1 SO GET THE ADD.	
43	A600		LDA A X	GET THE OPP CODE	
45	B7F0FB		STA A BYTE1	SAVE IT	
48	FF0F7		STX A BYTE1	ALSO SAVE THE ADDRESS	
4B	863F	SAME	LDA A #3F	SET THE	
4D	A700		STA A X	BREAK POINT	
4F	7EFF8F	END	JMP START	FINNISHED	
52	8132	2?	CMP A #32	IS IT A 2?	
54	26F9		BNE END	NO-THEN ITS RUBBISH	
56	BDFC65		JSR RDX	YES-GET ADDRESS	
59	A600		LDA A X	GET OPP.CODE	
5B	B7F0FB		STA A BYTE2	SAVE IT	
5E	FF0F9		STX A BYTE2	SAVE ADDRESS	
61	20E8		BRA SAME	SET THE BREAK POINT	
63	BDFC76	HEADER	JSR STRING	PRINT THE FOLLOWING	
66	204343			ASCII STRING WHICH	
69	204220			IS TERMINATED WITH	
6C	204120			HEX FF:	
6F	202058				
72	202020			CC B A X PC SP	
75	205043				
78	202020				
7B	5350FF				





# NewBear Computing Store

A Division of NEWBURY LABORATORIES LTD.

BONE LANE NEWBURY BERKSHIRE RG14 5SH Tel: 0635 46898



## HEXADECIMAL CODING FORM

PROGRAM		VERSION	AUTHOR	DATE	PAGE
MINIMON			ACH	DEC '78	13 OF 14
ADDRESS	MACHINE CODE	LABEL	OPERATOR & OPERAND	COMMENTS	
FF,7E	7E,FF,8F		JMP START	RETURN TO MINIMON	
81	8E,FD,00	RESET	LDS #STACK	SET UP STACK FOR MINIMON	
84	86,03		LDA A #03	RESET:	
86	B7,F4,01		STA A CTRLA	ACIA (a)	
89	B7,F4,03		STA A CTRLB	ACIA (b)	
8C	0F,FD,FD	SWI	STS PSTACK	SAVE PROGS. STACK PTR.	
8F	8E,FD,00	START	LDS #STACK	SET UP MINIMONS STACK	
92	86,01		LDA A #01	SET UP	
94	B7,F4,01		STA A CTRLA	ACIA (a)	
97	BD,FC,76		JSR STRING	PRINT :	
9A	0D,0A,00			CR, LF, NULL, *	
9D	2A,FF				
9F	BD,FC,00		JSR RD	GET COMMAND	
A2	81,53		CMP A #53		
A4	2,7,93		BEQ BR.PTSET	IS IT S	
A6	8,1,48		CMP A #48		
A8	2,7,69		BEQ HEADER	" H	
AA	8,1,50		CMP A #50		
AC	26,03		BNE 1	" P	
AE	7E,FC,08		JMP PUNCH		
B1	8,1,4C	1	CMP A #4C		
B3	26,03		BNE 2	" L	
B5	7E,FD,07		JMP LOAD		
B8	8,1,52	2	CMP A #52		
BA	26,03		BNE 3	" R	
BC	7E,FD,36		JMP REGPR		
BF	8,1,42	3	CMP A #42		
C1	26,03		BNE 4	" B	
C3	7E,FD,62		JMP BLOCKM		
C6	8,1,5A	4	CMP A #5A		
C8	26,03		BNE 5	" Z	
CA	7E,FD,E2		JMP Z		
CD	8,1,58	5	CMP A #58		
CF	26,03		BNE 6	" X	
D1	7E,FE,1C		JMP XX		

\* >

