

# A Text Editor for the 6800

by Robert Hudson

A text editor is an important basic tool used to prepare the many types of files for everyday computer operations. It is a software program that allows one to easily create or modify text material. The text can include programs in Basic, Fortran, PL/1, or assembler language as well as job control language, data file, or magazine articles. The following text editor should be appealing to 6800 microcomputer owners due to its small size and powerful set of commands.

## The program

The actual program requires memory from 0100 hex to 04FD hex. Also, memory is required on page zero for two buffers of 72 characters each and variable storage of 22 bytes. The variables have been arranged to facilitate the printout of the text buffer size and limits; therefore, do not rearrange 'topbuf,' 'cpntr,' or 'endbuf.' The program is begun by executing 0100 hex (cold start). A hot start is provided by beginning execution at 0103 hex. The cold start automatically erases the text buffer. A jump table contains system links to 'mikbug' type routines (except for one). As can be seen in the listing, the command table follows the jump table. The system messages have been arranged after the command table. The actual software begins at 01AE hex. The other jump to a 'mikbug' routine is found in the 'bottom' routine where a call is made to the OUT4HS subroutine. This routine outputs a double byte word followed by a space and it increments the index register twice. The location of this call is 029E hex. The stack is initialized to A060 hex in location 01F8 hex.

## Text editor commands

**TOP** The pointer is set to the beginning of the text buffer.

**BOTTOM** The pointer is set to the end of the text buffer. Also an output is produced of three double byte hex numbers such as: 04FE 165C 43FE.  
04FE indicates the beginning of the text buffer.  
165C indicates the end of the current text in the buffer.

43FF indicates the current end or upper limit of the buffer.

These three numbers are produced whenever a 'bottom' command is executed.

**APPEND** 'Append' was implemented to tack information onto the end of the present text material. The program assumes the appended data is a cassette file. An automatic 'bottom' is executed and control falls through to the cassette load routine.

**NEXT nnn** This command moves the pointer through the text buffer. The number may be preceded by a minus sign (-) which causes the pointer to go back that many lines.

**PRINT nnn** Prints nnn lines from the text buffer. The output begins from the line pointer and continues for nnn lines. The line pointer is not changed.

**ERASE** This command clears the defined text buffer of any text which has been entered. All pointers are reset and can be checked by using the 'bottom' command. The size of the text buffer can be modified by the 'quantity' command.

**DELETE nnn** Beginning at the current position of the line pointer, nnn lines are deleted from the text buffer.

**INSERT** This command is used to enter text into the text buffer and can be terminated in several ways. The normal way to end an insert operation would be by pressing the ESC button. However, if the limits of the input buffer are exceeded, an error message will be printed and the program will revert back to the command mode. Also the backspace (5F hex) and line cancel (18 hex) are supported. The control characters can be modified to conform to individual systems.

**FIND/string/** This command will begin from the current line pointer, searching the text buffer for the string contained between the two de-

limiters (which may be any characters). If the string is not found, an error message will print. If the string is found, the line containing the string will print and the line pointer will be positioned at the beginning of the line. The 'R' command can be used to repeat this command. This means the 'find' argument does not have to be retyped to obtain a list of all occurrences of a given string.

#### CHANGE/string1/string2/

This command searches the text buffer from the current line pointer position looking for the first occurrence of 'string 1.' If 'string1' is found then 'string2' is inserted to take the place of 'string1.' If 'string1' is not found, an error message is printed and the line pointer remains at the current location. If 'string1' is found, the change is made and the line is printed and the current line pointer is changed to point to the beginning of the changed line. Also the 'R' (repeat) command can be used to execute a repeat change command. This allows one to change all occurrences of 'string1' to 'string2' easily. The two strings are defined by delimiters of any character, but the delimiter must be the same character for that operation.

#### QUANTITY nnn

This command allows one to respecify the size of the text buffer. For each nnn, the text buffer is increased by 256 bytes of memory. The new memory area is cleared beginning from the original buffer end location. This command can be executed while text is present in the text buffer. By using the 'bottom' command with its associated printout of the buffer limits, the text buffer size can be adjusted as required.

This command will load a file from cassette. The data read in from the cassette is stored beginning at the location pointed to by the current line pointer. The software could be changed to do an automatic 'top' command. It was decided not to do an automatic 'top' command to retain the power and flexibility of entering data at any convenient setting of the line pointer. The tape (or disk) load and save routine will be dependent upon the individual system. These routines have been placed at the end of the software in an attempt to minimize the rewrite impact of adding your own load and save routines.

This command will save data from the text buffer starting at the location pointed to by the line pointer to the end of the text stored in the buffer. Again an automatic 'top' was deliberately not performed for added power and flexibility. Also this routine, as for 'load,' is system dependent. Both of these routines must be rewritten for your system.

#### MOVE

This command will store the line pointed to by the line pointer in the move buffer on page zero. Before the actual store is accomplished, the length of the line is determined. If the length is greater than 72 characters, the line will not be moved into the move buffer but instead an error message will print. If the move was successful, the prompt character will print and the line to move is stored. The line pointer is still pointing to the beginning of the original line that was duplicated in the move buffer. Therefore, by typing a D, the original line will be deleted. By using the 'next' command, the line pointer can be changed to point to a different line, specifically the line where you want to move the original line. Once the new location of the line has been determined, use the 'here' command to insert the line stored in the move buffer into the text buffer. Repeated 'here' commands will duplicate the line stored in the move buffer.

#### HERE

This is used with the move command to store the current line in the move buffer to the text buffer.

#### R

A single character command will repeat the previous command in the command buffer. It is used to obtain a repeat 'find' or 'change' command. (All other commands can be complete words or leading character, but the repeat command can only be 'R'.)

#### X (monitor)

This command will jump to your system monitor. To reenter the text editor without erasing the text buffer, begin execution at 0103 hex (hot start). This command is not implemented in the listing. To implement, enter 58 hex at location 013C hex followed by the location in hex of the monitor's entry point. In the assembler listing, this code would go where the 'spare' label is located.

#### General command syntax comments

The number argument is optional and, if not entered, the program assumes number one. Also the maximum number value is 255. All commands can be shortened to the first character and may be entered in either upper or lower case. The 'repeat' command is an exception and must be entered as a single upper case character (i.e. 'R').

The assembler listing is for a non-mikbug system. The following changes must be made for a mikbug system:

Location	Code	
0106	7E E1 AC	JMP INPUT
0109	7E E1 D1	JMP OUTPUT
010C	7E E0 7E	JMP PDATA
029E	BD E0 C8	JSR OUT4HS

The break routine at 027D hex is written for an ACIA port. If a different type of port (PIA) is utilized, the break routine must be changed to accommodate it. □

**TEXT EDITOR (Continued from Page 95)  
SAMPLE RUN**

```

.J 0100
EDITOR V 2
BUFFER SIZE? 5
>BOTTOM
04FE 04FE 09FE
>TOP
>PRINT
END OF TEXT REACHED
>NEXT
END-OF BUFFER REACHED
>INSERT
LOWER LIMIT TEST BACKSPACE THROUGH LEFT MARGIN
>I
UPPER LIMIT TEST GO THROUGH RIGHT MARGIN123456789012345678901234567890123
END OF BUFFER REACHED
>T
>P 10
END OF TEXT REACHED
>INSERT
this is line 1
this is lline 2
this is lline three
this is line 4
this is line 5
this is the last line
>top
>print 20
this is line 1
this is lline 2
this is lline three
this is line 4
this is line 5
this is the last line
END OF TEXT REACHED
>c /11/1/
this is line 2
>R
this is line three
>R
END OF TEXT REACHED
>c
>p 20
this is line 1
this is line 2
this is line three
this is line 4
this is line 5
this is the last line
END OF TEXT REACHED
>bottom
04FE 0563 09FE
>next -1

```

```

this is the last line
this is the last line
this is the last line
this is the last line
END OF TEXT REACHED
>p 256
ILLEGAL COMMAND
>c /the/not the/
this is not the last line
>R
this is not the last line
>R
this is not the last line
>R
this is not the last line
>R
this is not the last line
>R
this is not the last line
>R
this is not the last line
>R
this is not the last line
END OF TEXT REACHED
>NEXT -255
END OF BUFFER REACHED
>print 255
this is the first line
this is line 1
this is line 2
this is line three
this is line 4
this is line 5
this is not the last line
this is not the last line
this is not the last line
this is not the last line
this is not the last line
this is not the last line
this is not the last line
END OF TEXT REACHED
>BOTTOM
04FE 061A 0BFE
>HERE
>B
04FE 0630 0BFE
>top
>print 255
this is the first line
this is line 1
this is line 2
this is line three
this is line 4
this is line 5

```

this is not the last line  
this is not the last line  
this is not the last line  
this is not the last line  
this is not the last line  
this is not the last line  
this is not the last line  
this is not the last line  
this is the last line

END OF TEXT REACHED  
>f /not the/  
this is not the last line  
>c /not //  
this is the last line  
>delete 5

>b  
04FE 05AE 0BFE  
>top  
>print 255  
this is the first line  
this is line 1  
this is line 2  
this is line three  
this is line 4  
this is line 5  
this is not the last line  
this is not the last line  
this is the last line  
END OF TEXT REACHED  
>n 6

>p  
this is line 5  
>p  
this is not the last line  
>d  
>n -3  
>p  
this is line three  
>c /three/3/  
this is line 3  
>top

>print 15  
this is the first line  
this is line 1  
this is line 2  
this is line 3  
this is line 4  
this is line 5  
this is not the last line  
this is the last line  
END OF TEXT REACHED  
>bottom  
04FE 0590 0BFE  
>x

>top  
>here  
>top  
>p 20  
this is the last line  
this is line 1  
this is line 2  
this is line three  
this is line 4  
this is line 5  
this is the last line

END OF TEXT REACHED  
>c /last/first/  
this is the first line  
>b  
04FE 057A 09FE  
>h  
>h  
>h  
>h  
>h  
>here  
>t

>p 30  
this is the first line  
this is line 1  
this is line 2  
this is line three  
this is line 4  
this is line 5  
this is the last line  
this is the last line  
this is the last line  
this is the last line  
this is the last line  
this is the last line  
this is the last line  
END OF TEXT REACHED  
>bottom  
04FE 05FE 09FE  
>quantity 2

>b  
04FE 05FE 0BFE  
>t  
>find /5/  
this is line 5  
>n  
>p 255  
this is the last line  
this is the last line

```

.J 0103
>P 255
END OF TEXT REACHED
>T
>P 255
this is the first line
this is line 1
this is line 2
this is line 3
this is line 4
this is line 5
this is not the last line
this is the last line
END OF TEXT REACHED
>U
ILLEGAL COMMAND
>X

```

**PROGRAM LISTING**

```

00001 NAM
00002 * ROBERT HUDSON
00003 * JANUARY 24, 1979
00004 OPT
00005 OUT4HS EQU S,NOG,P
00006 FCD9 EQU $FCD9
00007 F266 EQU $F266
00008 ECHO EQU $F3
00009 ACIA EQU $F000
00010 OD0A EQU $OD0A
00011 EOT EQU 4
00012 * VARIABLE STORAGE AND WORK AREA
00013 ORG $10
00014 TOPBUF RMB 2
00015 CPNTR RMB 2
00016 ENDRUF RMB 2
00017 TEMIDX RMB 2
00018 TEMBUF RMB 2
00019 O01A RMB 2
00020 O01D RMB 1
00021 O01E RMB 1
00022 O01F RMB 1
00023 O020 RMB 2
00024 O022 RMB 2
00025 O024 RMB 1
00026 O025 RMB 1
00027 O030 ORG
00028 * LINE BUFFER AREA
00029 LINEUF RMB 72
00030 * MOVE BUFFER AREA
00031 MOVBUF RMB 72
00032 * TEXT EDITOR PROGRAM
00033 ORG $0100
00034 O100 7E O1CA SETUP JMP INIT
00035 O103 7E O1F8 HOT JMP START
00036 O106 7E FCES INPUT JMP $FCES
00037 O109 7E FCES OUTPUT JMP $FCES
00038 O10C 7E F882 PDATA JMP $F882
00039 * COMMAND TABLE
00040 O10F 50 CRDTBL FCB 'P
00041 O110 258 FDB PRINT
00042 O112 4E FCB 'M
00043 O113 02B3 FDB NEXT

```

```

END OF TEXT REACHED
END OF TEXT REACHED
USED TO REARRANGE
USED TO PROVIDE BUFFER
LIMITS WITH BOTTOM COMMAND
JUMP TABLE FOR
SYSTEM LINKAGE
ENTER2-
SYNTAX
>P rdn
>M "-"rdn
NEXT

```

```

00112 O1CF D7 12
00113 O1D1 CE 0143
00114 O1D4 BD 010C
00115 O1D7 CE 0031
00116 O1DA 86 20
00117 O1DC A7 00
00118 O1DE 08
00119 O1DF BD 0106
00120 O1E2 A7 00
00121 O1E4 81 0D
00122 O1E6 26 F6
00123 O1E8 8D D2
00124 O1EA CE 0020
00125 O1ED 86 A0
00126 O1EF 6F 00
00127 O1F1 88
00128 O1F2 4A
00129 O1F3 26 FA
00130 O1F5 BD 02A5
00131 * RESTART
00132 O1F8 8E A060
00133 O1FB CE 013F
00134 O1FE BD 010C
00135 O201 CE 0030
00136 O204 BD 0106
00137 O207 81 5F
00138 O209 26 08
00139 O20B 09
00140 O20C 8C 002F
00141 O20F 27 6A
00142 O211 20 F1
00143 O213 81 18
00144 O215 27 E4
00145 * IMPLEMENTATION OF REPEAT FUNCTION.
00146 * SYNTAX: AFTER PROMPT CHARACTER PRESS "R"
00147 * EXAMPLE: >R
00148 * WILL REPEAT PREVIOUS COMMAND
00149 * FACILITATES A REPEAT FIND AND CHANGE COMMAND
00150 O217 81 52 REPEAT CMP A
00151 O219 26 10 BNE
00152 O21B 8C 0030 CPX
00153 O21E 26 08 BNE
00154 O220 8D 4E BSR
00155 O222 5F CLR B
00156 O223 D7 1D STA B
00157 O225 5C INC B
00158 O226 BD 02B5 JSR
00159 O229 20 0C REPT
00160 O22B A7 00 STA A
00161 O22D 08 INX
00162 O22E 8C 0078 CPX
00163 O231 27 48 BREQ
00164 O233 81 0D CMP A
00165 O235 26 CD BNE
00166 * BUFFER FILLED
00167 O237 CE 010F REPT LDA X
00168 O23A 86 5F LDA A
00169 O23C 94 30 AND A
00170 O23E A1 00 COMPAR CMP A
00171 O240 27 10 BEQ
00172 O242 08 INX
00173 O243 08 INX
00174 O244 08 INX
00175 O245 8C 013F CPX
00176 O248 26 F4 BNE
00177 O24A CE 0161 * ERROR
00178 O24D BD 016C * ERROR
00179 * ERROR MESSAGE

```

CLEAR SOME VARIABLES PLUS  
THE TEXT AND MOVE BUFFERS

ESTABLISH STACK AREA

BACKSPACE?

LOWER LIMIT ERROR

CANCEL INPUT LINE

IMPLEMENTATION OF REPEAT FUNCTION.

REPEAT PREVIOUS COMMAND

REPEAT FIND AND CHANGE COMMAND

CLEAR NEGATIVE FLAG  
AND GO TO NEXT  
TEXT LINE

UPPER LIMIT ERROR

LOWER CASE  
MASK FOR LOWER CASE

UPPER OR LOWER CASE  
ALLOWS COMMANDS TO BE

PRINT ERROR MESSAGE

```

00046 0118 46      FCB      * F /string/
00047 0119 43      FCB      * C /string1/string2/
00048 0118 43      FCB      CHANGE
00049 011C 0465    FDB      * I
00050 011E 49      FCB      * D nnn
00051 011F 036D   FDB      * T
00052 0121 44      FCB      * E
00053 0122 0347   FDB      * B
00054 0124 54      FCB      * S
00055 0125 02AE   FDB      * L
00056 0127 45      FCB      * Q nnn
00057 0128 02A5   FDB      * N
00058 012A 42      FCB      * H
00059 012B 028E   FDB      * X" JUMP TO MONITOR
00060 012D 53      FCB      ADDRESS OF MONITOR
00061 012E 04CE   FDB      ENTRY POINT.
00062 0130 4C      FCB
00063 0131 04B1   FDB
00064 0133 51      FCB
00065 0134 01AE   FDB
00066 0136 4D      FCB
00067 0137 0482   FDB
00068 0139 48      FCB
00069 013A 049C   FDB
00070 013C 00      FCB
00071 013D 0000   FCB
*
* * R" REPEAT
00072
00073
00074
00075 013F 0D0A   CMDR FB
00076              * PROMPT CHARACTER
00077 0141 3E      FCB
00078 0142 04      FCB
00079              * SYSTEM MESSAGES
00080 0143 0D0A   SIZMSG FDB
00081 0145 45      FCC
00082 0151 0D0A   FDB
00083 0153 42      FCC
00084 0160 04      FCB
00085 0161 49      CMDR FB
00086 0170 04      FCB
00087 0171 0D0A   TEXTSG FDB
00088 0173 45      FCC
00089 0186 04      FCB
00090 0187 0D0A   ENDMSG FDB
00091 0189 45      FCC
00092 019E 04      FCB
00093 019F 53      SYMSG FCC
00094 01AB 0D0A   FDB
00095 01AD 04      FCB
00096              * PROGRAM BEGINS
00097 01AE DE 14   BUFSIZ LD
00098 01B0 DF 18   STX
00099 01B2 BD 02EB JSR
00100 01B5 BD 0C   BSR
00101 01B7 DE 18   LD
00102 01B9 7E 02A7 CLR01
00103 01BC BD 02EB JSR
00104 01BF DE 10   LD
00105 01C1 DF 14   STX
00106 01C3 7C 0014 INC512
00107 01C6 5A      DBC B
00108 01C7 26 FA   BNE
00109 01C9 39      RTS
00110 01CA CE 04FE INIT
00111 01CD DF 10   STX

* VALID CMD
GDCMD LD
JSR
BRA
* PRINT N LINES
PRINT JSR
CHARLD LDA A
REQ
INX
CMP A
REQ
BSR
BRA
DECLIN BSR
DEC B
BNE
RTS
CRFUNG PSH A
LDA A
BSR
LDA A
BSR
PUL A
RTS
ERRCBL BRA
* CHARACTER OUTPUT AND BREAK ROUTINE
CHOUT PSH A
LDA A
BSR A
PUL A
JMP
INTERP JMP
* POINTERS
FPTOER JMP
* BOTTOM
BOTTOM LD
LDA A
REQ
INX
BRA
* BOT1
BOT1 STX
LDX
LDA B
STORAG JSR
DSC B
BNE
RTS
* ERASE
ERASE LD
CLR
INX
CPX
BNE
* TOP
TOP LDX
STX
RTS
* NEXT
NEXT BSR
NEXTREP LDA A
LDX
TST
BEQ
CPX
BEQ
* LOOP2
LOOP2 CPX
BEQ
* NEXT ROUTINE NOW WILL
GO TO BOTH LIMITS OF
TEXT BUFFER EVEN IF NEXT
ARGUMENT IS GREATER OF
THAN THE NUMBER OF
LINES REQUIRED.
OVRNRM ALSO "CPWR" IS STORED

```



SUBROUTINE CREATED TO ASSIST MOVE COMMAND

CPNTR LENGTH CLR A GETONE BEO DELTS #SOD LINEND LENGTH INX GETONE BEO DELTS BSR DLINEL TST A BSR BEC B RNE RTS ENDTXT BSR #SYMSG ERROT JMP DUTLIN LDX LDA A STA A LDA A STA A CPX RNE RTS \* INSERT

00318 0330 DE 12 DLINEL LDX  
00319 0331 CLR A GETONE LDA A  
00320 0332 AS 00 BEO  
00321 0333 27 00 DELTS  
00322 0334 81 00 #SOD  
00323 0335 27 06 BEO  
00324 0336 7C 001C LINEND  
00325 0340 08 INX  
00326 0341 20 F2 GETONE  
00327 0343 7C 001C LINEND INC  
00328 0346 39 DELTS RTS  
00329 0347 8D A2 BSR  
00330 0349 8D E5 DLINEL BSR  
00331 0348 4D TST A  
00332 034C 27 06 BEO  
00333 034E 8D 0C BSR  
00334 0350 5A BEC B  
00335 0351 26 F6 RNE  
00336 0353 39 RTS  
00337 0354 8D 06 ENDTXT BSR  
00338 0356 CE 0171 #SYMSG  
00339 0359 7E 024D ERROT JMP  
00340 035C DE 12 DUTLIN LDX  
00341 035E 9C 1C LDA A  
00342 0360 B7 0364 STA A  
00343 0363 A6 00 DLST LDA A  
00344 0365 A7 00 STA A  
00345 0367 08 INX  
00346 0368 9C 14 CPX  
00347 036A 26 F7 RNE  
00348 036C 39 RTS

\* MOVE: WILL STORE IN MOVE BUFFER THE LINE POINTED  
\* TO BY "CPNTR" - CURRENT LINE POINTER.  
\* ORIGINAL LINE CAN THEN BE DELETED BY "D" COMMAND.  
\* OR BY NOT DELETING "MOVE". CAN BE USED  
\* TO DUPLICATE LINES.  
\* SYNTAX: >MOVE  
\* USED WITH THE HERE COMMAND.  
MOVE LDX CPNTR FIND LINE LENGTH,  
IF > 72 THEN DO NOT MOVE

00455 0423 76 E6 RNE  
00456 0425 9E 1A LBS  
00457 0427 39 RTS  
00458 0428 DE 1D ENDSCH LDX  
00459 042A DF 12 STX  
00460 042C 9E 1A LBS  
00461 042E 7E 0356 JMP  
\* STRING  
00462 0431 DE 20 DEFBUF LDX  
00463 0433 08 INCO3 INX  
00464 0434 A6 00 LDA A  
00465 0436 81 20 CMP A  
00466 0438 27 06 BEO  
00467 043A 81 0D CMP A  
00468 043C 27 21 BEO  
00469 043E 20 F3 BRA  
00470 0440 08 INX  
00471 0441 A6 00 LDA A  
00472 0443 81 20 CMP A  
00473 0445 27 F9 BEO  
00474 0447 97 1F STA A  
00475 0449 DF 22 COENT STX  
00476 044B 5F CLR B  
00477 044C 08 INCO4 INX  
00478 044D A6 00 LDA A  
00479 044F 91 1F CMP A  
00480 0451 27 07 BEO  
00481 0453 81 0D CMP A  
00482 0455 27 08 BEO  
00483 0457 5C INC B  
00484 0458 20 F2 BRA  
00485 045A DF 20 ENDSCH STX  
00486 045C D7 1C STA B  
00487 045E 39 RTS  
00488 045F CE 019F SYNERR LDX  
00489 0462 7E 024D JMP  
\* CHANGE  
00490 0465 CE 0030 CHANGE LDX  
00491 0468 DF 20 STX  
00492 046A 8D C5 BSR  
00493 046B 8D C5 BSR  
00494 046C 8D 8F MATCH  
00495 046E BD 035C JSR  
00496 0471 DE 20 LDX  
00497 0473 8D D4 COENT  
00498 0475 BD 03AD JSR  
00499 0478 DE 22 LDX  
00500 047A 08 INX  
00501 047B DF 18 STX  
00502 047D 8D 2C JFBUF BSR  
00503 047F 7E 035C JMP BKMOV

ESC END OF INPUT?  
CANCEL CURRENT LINE?  
BACKSPACE?  
LOWER LIMIT ERROR  
HAS UPPER LIMIT OF INPUT BUFFER BEEN EXCEEDED?

00504 0504  
00505 0505  
00506 0506  
00507 0507  
00508 0508  
00509 0509  
00510 0510  
00511 0482 DE 12  
00512 0484 DF 18 STX  
00513 0486 BD 0330 JSR  
00514 0489 D6 1C LDA B  
00515 048B BD 02DF JSR  
00516 048E D7 24 STA B  
00517 0490 CE 0078 LDX  
00518 0493 DF 12 STX  
00519 0495 8D 14 BSR  
00520 0497 DE 18 LDX  
00521 0499 DF 12 STX



00522 049B 39

```

RTS
* HERE: COMMAND USED TO INSERT CURRENT LINE OF
* TEXT IN "MOVE" BUFFER TO LINE IN TEXT
* BUFFER POINTED TO BY "CPNTR".
* SYNTAX: AFTER "MOVE" COMMAND USE "TOP", "BOTTOM"
* OR "NEXT" COMMAND TO POSITION
* CURRENT LINE POINTER TO DESIRED LINE TO INSERT
* TEXT STORED IN "MOVE" BUFFER.
* >MOVE :STORE LINE IN MOVE BUFFER
* >DELETE :DELETE EXISTING LINE
* >NEXT -5 :GO BACK FIVE LINES
* >HERE :INSERT LINE STORED IN "MOVE" BUFFER.
          :INSERT LINE STORED IN "MOVE" BUFFER.
HERE LDA B
      MBUFL
      LEND
      STA B
      MOVLIN
      LDX
      STX
      JPBUP JMP
      * APPEND
00541 04AE BD 028E APPEND JSR
00542 04AE BD 028E APPEND JSR
      * LOAD
00543 04AE BD 028E APPEND JSR
      * LOAD
00544 04B1 DE 12 LDX
      *FF LDA A
00545 04B3 86 FF LDA A
00546 04B5 97 F3 STA A
00547 04B7 BD 0106 LOAD1 JSR
00548 04BA 4D TST A
00549 04BB 27 FA BEQ
00550 04BD 81 03 CMP A
00551 04BF 27 00 BEQ
00552 04C1 A7 00 STA A
00553 04C3 08 INX
00554 04C4 9C 14 CPX
00555 04C6 26 EF BNE
00556 04C8 7E 00F3 LEND CLR
00557 04CB 7E 02AE JMP
      * SAVE
00558 04CE 7F F266 CLR
00559 04CE 7F F266 SAVE
00560 04D1 8D 20 BSR
00561 04D3 DE 12 LDX
00562 04D5 09 DEX
00563 04D6 08 MORS INX
00564 04D7 A6 00 LDA A
00565 04D9 27 09 BEQ
00566 04DB 9C 14 CPX
00567 04DD 27 05 BEQ
00568 04DF BD 0109 JSR
00569 04E2 20 F2 BSR
00570 04E4 8D 0D ENDS
00571 04E6 86 03 LDA A
00572 04E8 BD 0109 JSR
00573 04EB 86 80 LDA A
00574 04ED B7 F266 STA A
00575 04F0 7E 02AE JMP
00576 04F3 C6 32 NULLS LDA B
00577 04F5 4F CLR A
00578 04F6 BD 0109 MNULLS JSR
00579 04F9 5A DEC B
00580 04FA 26 FA BNE
00581 04FC 39 RTS
00582 04FD 0D PGEND FCB
00583 04FE 0D END

```

```

OUT4HS FCD9
TVMASK F266
ECHO 00F3
ACIA F000
SERIAL 900A

```

```

LCRTS 02DS
LEACK 02DF
OVREUN 02E5
GETNUM 02EB
NUM1 02F1
NUM2 0306
NUM3 0309
NUM4 030D
MULL0 0315
ADDRA 031C
OUTONE 032B
NUMRS 032C
GTORER 032D
DLINE 0330
GETOME 0335
LINDM 0343
DELRTS 0346
DELETE 0347
DLINEL 0349
ENDTST 0354
RT01 0356
DUTLIN 035C
DLET 0363
INSERT 036D
INS01 0373
INS02 037F
INS03 0388
INS04 0392
INS05 039D
INRTS 03AC
MOVLIN 03AD
MOV02 03B6
MOV03 03C2
MOV04 03C5
PUTBUF 03CC
PULL1 03D7
PUTRET 03E0
FIND 03E3
BEMOV 03EC
DECR 03F0
MATCH 03FD
GETLEN 0407
PULLCH 040B
FNDONE 041B
FNDTWO 0421
FOUND 0425
ENDSCH 0428
DEFBUF 0431
INCO3 0433
SPACE1 0440
COENT 0449
INC04 044C
ENDSTR 045A
SYNERR 045F
CHANGE 0465
MOVE 0482
HERE 049C
JPBUF 04AB
APPEND 04AE
LOAD1 04B1
LEND 04B7
MORS 04C8
MORS 04D6
ENDS 04E4
NULLS 04F3
NULLS 04FE

```

TEMBUF 0014  
 TEMBUF 0016  
 TEMBUF 0018  
 STKSTR 001A  
 LENGTH 001C  
 NEGLG 001D  
 FLAGO1 001E  
 DELIM 001F  
 TBUF1 0020  
 TBUF2 0022  
 MBUFL 0024  
 ADDCNT 0025  
 LINBUF 0030  
 MOVBUF 0078  
 SETUP 0100  
 HOT 0103  
 INPUT 0106  
 OUTPUT 0109  
 PDATA 010C  
 CMDTBL 010F  
 SPARE 013C  
 CMDHDR 013F  
 SIZMSG 0143  
 CMDERR 0161  
 TXTMSG 0171  
 ENDMSG 0187  
 SYNMSG 019F  
 BUFSIZ 01AE  
 SIZE 01BC  
 INCSIZ 01C3  
 INIT 01CA  
 SIZE1 01DE  
 CLREND 01EF  
 START 01F8  
 BEGIN 01FB  
 GETCHR 0204  
 CONT1 0213  
 REPEAT 0217  
 CONT2 022B  
 REPT 0237  
 COMPAR 023E  
 ERROR 024A  
 ERROUT 024D  
 GDCMD 0252  
 PRINT 0258  
 CHARLD 025D  
 DECLIN 026A  
 CRFUNC 0270  
 ERRCEL 027B  
 CHOUT 027D  
 INTERP 0288  
 PTOER 028B  
 BOTTOM 028E  
 BLOAD 0290  
 ROT1 0297  
 STORAG 029E  
 CHERT 02A4  
 ERASE 02A5  
 CLR01 02A7  
 TOP 02AE  
 NEXT 02B3  
 NXTREP 02B5  
 LOOP2 02BE  
 LOOP0 02C3  
 LOOP1 02CF  
 LOOP3 02D0

0100 7E 01 CA 7E 01 F8 7E FC EB 7E FC E3 7E F8 82 50  
 0110 02 58 4E 02 B3 41 04 AE 46 03 E3 43 04 65 49 03  
 0120 6D 44 03 47 54 02 AE 45 02 A5 42 02 8E 53 04 CE  
 0130 4C 04 B1 51 01 AE 4D 04 82 48 04 9C 00 00 00 0D  
 0140 0A 3E 04 0D 0A 45 44 49 54 4F 52 20 56 20 32 20  
 0150 20 0D 0A 42 55 46 46 45 52 20 53 49 5A 45 3F 20  
 0160 04 49 4C 4C 45 47 41 4C 20 43 4F 4D 41 4E 44  
 0170 04 0D 0A 45 4E 44 20 4F 46 20 54 45 58 54 20 52  
 0180 45 41 43 48 45 44 04 0D 0A 45 4E 44 20 4F 46 20  
 0190 42 55 46 46 45 52 20 52 45 41 43 48 45 44 04 53  
 01A0 59 4E 54 41 58 20 45 52 4F 52 4F 52 0D 0A 04 DE 14  
 01B0 DF 18 BD 02 EB 8D 0C DE 18 7E 02 A7 BD 02 EB DE  
 01C0 12 CE 01 43 BD 01 0C CE 00 31 86 20 A7 00 08 BD  
 01E0 01 06 A7 00 81 0D 26 F6 8D D2 CE 00 20 86 A0 6F  
 01F0 00 08 4A 26 FA BD 02 A5 8E A0 60 CE 01 3F BD 01  
 0200 0C CE 00 30 BD 01 06 81 5F 26 08 09 8C 00 2F 27  
 0210 6A 20 F1 81 18 27 E4 81 52 26 10 8C 00 30 26 0B  
 0220 8D 4E 5F D7 1D 5C BD 02 B5 20 0C A7 00 08 8C 00  
 0230 78 27 48 81 0D 26 CD CE 01 0F 86 5F 94 30 A1 00  
 0240 27 10 08 08 08 8C 01 3F 26 F4 CE 01 61 BD 01 0C  
 0250 20 A6 EE 01 AD 00 20 A3 BD 02 EB DE 12 A6 00 27  
 0260 2A 08 81 0D 27 04 8D 15 20 F3 8D 04 5A 26 EE 39  
 0270 36 86 0D 8D 08 86 0A 8D 04 32 39 20 68 36 B6 F0  
 0280 00 47 32 25 03 7E 01 09 7E 01 F8 7E 03 56 DE 12  
 0290 A6 00 27 03 08 20 F9 DF 12 CE 00 10 C6 03 BD FC  
 02A0 D9 5A 26 FA 39 DE 10 6F 00 08 9C 14 26 F9 DE 10  
 02B0 DF 12 39 8D 36 DE 12 86 0D 7D 00 1D 27 12 9C 10  
 02C0 27 23 09 09 A1 00 26 FB 08 DF 12 5A 26 F0 39 08  
 02D0 9C 14 27 11 A1 00 26 F7 08 DF 12 5A 26 F2 39 27  
 02E0 04 C1 48 23 F9 CE 01 87 7E 02 4D CE 00 30 5F D7  
 02F0 1D 08 A6 00 81 0D 27 33 81 20 26 F5 08 A6 00 81  
 0300 2D 26 0A 7C 00 1D 08 A6 00 81 0D 27 1F 80 30 2B  
 0310 1C 81 09 2E 18 36 86 09 97 25 17 0C 1B 25 0E 7A  
 0320 00 25 26 F8 33 1B 25 05 16 20 DB 5C 39 7E 02 4A  
 0330 DE 12 7F 00 1C A6 00 27 0D 81 0D 27 06 7C 00 1C  
 0340 08 20 F2 7C 00 1C 39 8D A2 8D E5 4D 27 06 8D 0C  
 0350 5A 26 F6 39 8D 06 CE 01 71 7E 02 4D DE 12 96 1C  
 0360 B7 03 64 A6 00 A7 00 08 9C 14 26 F7 39 CE 00 30  
 0370 5F D7 1E BD 01 06 81 18 26 05 7C 00 1E 20 1E 81  
 0380 18 26 05 BD 02 70 20 E5 81 5F 26 06 5A 2B 1D 09  
 0390 20 E1 5C BD 02 DF A7 00 08 81 0D 26 D6 D7 1C 8D  
 03A0 0C CE 00 30 DF 18 8D 24 96 1E 27 C1 39 96 14 D6  
 03B0 15 D0 1C 24 01 4A 97 16 D7 17 DE 16 08 D6 1C F7  
 03C0 03 C6 09 A6 00 A7 00 9C 12 26 F7 39 9F 1A 9E 18  
 03D0 34 D6 1C 27 0B DE 12 32 A7 00 08 5A 26 F9 DF 12  
 03E0 9E 1A 39 CE 00 30 DF 20 8D 47 8D 11 DE 12 86 0D  
 03F0 09 A1 00 26 FB 08 DF 12 C6 01 7E 02 5D 9F 1A DE  
 0400 22 08 9E 12 9F 1D 34 D6 1C 9F 16 32 4D 27 19 A1  
 0410 00 27 08 9E 16 DE 22 31 08 20 EC D1 1C 26 02 9F  
 0420 12 08 5A 26 E6 9E 1A 39 DE 1D DF 12 9E 1A 7E 03  
 0430 56 DE 20 08 A6 00 81 20 27 06 81 0D 27 21 20 F3  
 0440 08 A6 00 81 20 27 F9 97 1F DF 22 5F 08 A6 00 91  
 0450 1F 27 07 81 0D 27 08 5C 20 F2 DF 20 D7 1C 39 CE  
 0460 01 9F 7E 02 4D CE 00 30 DF 20 8D C5 8D 8F BD 03  
 0470 5C DE 20 8D D4 BD 03 AD DE 22 08 DF 18 8D 2C 7E  
 0480 03 EC DE 12 DF 18 BD 03 30 D6 1C BD 02 DF D7 24  
 0490 CE 00 78 DF 12 8D 14 DE 18 DF 12 39 D6 24 BD 02  
 04A0 DF 1C BD 03 AD CE 00 78 DF 18 7E 03 CC BD 02  
 04B0 8E DE 12 86 FF 97 F3 BD 01 06 4D 27 FA 81 03 27  
 04C0 07 A7 00 08 9C 14 26 EF 7F 00 F3 7E 02 AE 7F F2  
 04D0 66 8D 20 DE 12 09 08 A6 00 27 09 9C 14 27 05 BD  
 04E0 01 09 20 F2 8D 0D 86 03 BD 01 09 86 80 B7 F2 66  
 04F0 7E 02 AE C6 32 4F BD 01 09 5A 26 FA 39 0D 00 00