Note: As of Ø1-Mar-86, customer support
for this product is provided by:
MISOSYS, Inc., P.O. Box 239
Sterling, VA 2217Ø-Ø239 <7Ø3-45Ø-4181>

# LED

LOGICAL SYSTEMS INC.

TRS-80 MODEL I/III

CATALOG # L-30-020

LOGICAL SYSTEMS INC.

MILWAUKEE, WI

LED - LDOS Text Editor

VERSION #

LED-51 S/N 30796

COPYRIGHT 1982

RIGHTS RESERVED

# LED - THE LDOS EDITOR

The LDOS EDitor is a general purpose screen oriented text editor designed to effect rapid generation and modification of ASCII and certain other text files (e.g. KSM). The syntax used to enter LED is as follows:

```
=========================================================
|                                                       |
| LED filespec (parm, parm, ...)                        |
|                                                       |
| filespec          This is the name of the file you    |
|                   are going to edit.                  |
|                                                       |
| ASCII             ASCII file, or at least it is       |
|                   supposed to be. Bit 7 of each       |
|                   character is reset on loading.      |
|                                                       |
| END=0             Specify a SAVE terminator for the   |
|                   file. Useful to create a SCRIPSIT   |
|                   compatible ASCII file.              |
|                                                       |
| SAVE="filespec"   Specify a filespec for saving the   |
|                   text to be different from the       |
|                   filespec that was loaded.           |
|                                                       |
| TABS              Expand tabs when loading a file     |
|                   with imbedded tabs. Tab stops used  |
|                   are every "8" character positions.  |
|                   Otherwise, tabs will appear as      |
|                   graphics blocks.                    |
|                                                       |
| WP                Specify a Word Processing text file |
|                   which ends with a hex 00 byte that  |
|                   will be retained. Otherwise, this   |
|                   byte is stripped and the file will  |
|                   be read until an end-of-file is     |
|                   detected.                           |
|                                                       |
| XLATE=X'aabb'     Perform a character translation on  |
|                   loading and saving the text file.   |
|                   "aa" is the character translated.   |
|                   "bb" is the value when in memory.   |
|                   This is used primarily to get 00's  |
|                   or FF's into a text file.           |
|                                                       |
| abbr: All parameters may be abbreviated to their      |
|       first character.                                |
|                                                       |
=========================================================
```

If you have used a screen oriented word processor before, you will be quite familiar with the operation of LED. If you have not, you will discover that LED is very easy to learn. Give yourself some time to play with its controls to become comfortable with it and to familiarize yourself with exactly what each command does.

## PARAMETER DETAILS

LED assumes that the file to be loaded is an ASCII file. LED is very clean on loading. The implication is that LED will load whatever is in the file until reaching the end-of-file marker. The ASCII parameter can be used to force LED to reset bit 7 of every byte loaded. This can be used, for instance, to convert a SCRIPSIT file to an ASCII SCRIPSIT file on loading. The reverse process cannot be performed. Once a file has been loaded with the ASCII parameter, you can not restore the bit 7 of any byte that had it removed.

LED does not add any special terminating character to the file when saving your text. Certain word processors use unique terminating characters to indicate the file's end rather than rely on the end-of-file pointer. If you are using LED to prepare text for downstream use in a word processor, then you may need to specify the END=Ø parameter. For instance, SCRIPSIT uses a terminator of X'ØØ'; therefore, if you are going to process the text file with SCRIPSIT, then you may want to specify the END terminator.

The SAVE parameter can be useful if you want to have the edited output written to a file different from that used for input. If you omit the SAVE parameter, the function can still be invoked by using the SAVE command in LED.

Certain files may have imbedded tab characters (X'Ø9'). If you wish to edit the file with the tabs expanded to normal modulo 8 column positions, use the TAB parameter. You can not compress the spaces introduced by the TAB parameter on saving the text.

LED will load your text file until the end-of-file pointer is reached. Some applications generate text files without a "true" end-of-file pointer. Where this is the case, a character is usually imbedded in the text to indicate the end of the file. This is the case with SCRIPSIT which uses a terminating character of X'ØØ'. Therefore, when reading a SCRIPSIT file, you may have to specify the WP parameter. If WP is not specified, then LED will discard any X'ØØ' byte on loading a text file and will continue to load until an end-of-file is reached. This, by the way, is a means for removing X'ØØ' bytes from a file.

LED does not load X'ØØ' bytes. LED also uses X'FF' bytes to indicate block markers. Therefore, if you have a text file which legitimately uses one or the other, then it presents a special case to edit with LED. This will require the use of the XLATE parameter. The X'ØØ' or X'FF' can be translated to some other unused byte while being edited. The reverse process will automatically be invoked. Any character translated on loading will be restored to its original value on saving. LED only provides a single-character translation.

## EXTENDED CURSOR MODE

In order to use LED, you must be running with the LDOS KI/DVR keyboard driver. LED is designed to also use the extended cursor mode of the LDOS KI/DVR released with versions 5.1.1 and later. If you are operating under an earlier release of LDOS, you will need to apply the ECM/FLT filter to the *KI device. This can be accomplished via the command:

```
FILTER *KI ECM
```

This is absolutely unnecessary under 5.1.1 or later releases.

## CHARACTER DISPLAY

One item about character display is important enough to be presented early in this documentation. Character values X'2Ø' through X'BF' will be displayed as their true character. Character values outside of this range will be displayed as X'84' on a Model I (small graphics block - left center) and X'FC' on a Model III (reverse image question mark). Imbedded tabs (X'Ø9' character value) will be displayed as X'84' on both a Model I and a Model III. A block marker will be displayed as a solid graphics block - character value X'BF' (191 decimal). The exact character value of every character position will be discernable from the prompting information line.

## MODES OF OPERATION

LED has four different modes of operation. "Standard" mode is defined as the mode in which characters entered will overtype any already at the cursor position. "Insert" mode inhibits overtyping of characters by shifting the entire text one character position as each "insert" character is entered. In "insert line" mode, a line of spaces will be inserted automatically into the text to provide a line buffer for text entry. Since the text buffer is shifted to provide space only once for each line, there is a perception of increased speed. The term, "open a line", will be used to imply a line insert. The last mode is "delete" mode. This mode deletes characters, spaces, lines, etc. The character used to indicate the cursor will be unique for each mode of operation.

## CURSOR DISPLAY

A flashing cursor will always be positioned on the screen at the location to receive the next character input. The cursor itself will be one of four characters dependent on the current mode of operation. In standard mode (overtype entry), the cursor will be an X'BØ' (176 decimal). This will appear as the graphic character displayed with the bottom left and right graphic blocks. In "insert" mode (character insertion), an X'B5' (181 decimal) will be displayed. The "insert" cursor appears as a graphics character with the left column and bottom right graphics blocks. The "insert-line" cursor is displayed as a graphics character with the right column and bottom left graphics blocks. That is the character X'BA' (186 decimal). In "delete" mode, all six graphics blocks will be used to indicate the cursor. This is character X'BF' (191 decimal). A block marker is also displayed as character X'BF'. Therefore, if you do not see the flashing cursor on the screen, it is positioned over a block marker.

## CURSOR POSITIONING

The cursor can be positioned either up, down, right, or left, by pressing one of the four arrow keys {<UP-ARROW>, <DOWN-ARROW>, <LEFT-ARROW>, <RIGHT-ARROW>}. If tab stops are present, then you can position the cursor at the next tab stop by depressing <SHIFT> and the <RIGHT ARROW> key. The cursor can be positioned to the previous tab stop by depressing <SHIFT> and the <LEFT ARROW> KEY. The beginning-of-line and end-of-line positions can be reached by depressing <CLEAR> and <LEFT-ARROW> or <RIGHT-ARROW> respectively. You can position the cursor to the top-of-text or the bottom-of-text by depressing <CLEAR> and either <UP-ARROW> or <DOWN-ARROW> respectively. Command keys are also available to position TOP and BOTTOM. Their use will be discussed later. Additional positioning can be invoked via certain control functions as described in the menu explanations.

## PROMPT LINE

When you first enter LED, a single line of prompting information will be displayed at the bottom of the screen. Immediately above this information will be a solid white line. The line will be used to split the screen into two windows. If the filespec entered in the command line identified a text file containing character data, the top window will display as much of your file as can be displayed. The prompting information line will look like this:

```
............................{Filename/EXT:Ø-N}( Ø):X'ØØ'|ØØØØØ
<-------------1------------><------2------><3><-4-><-5-><--6->
```

The information will be displayed in various fields. These fields will contain:

<1> Current status such as INSERT, DELETE, etc. This field will also be used to display prompting messages.

<2> Will contain the Filename/EXT of the current save file.

<3> Will contain "-N" to indicate a NEW file or an "-R" to indicate a REPLACED file. A new file is the result of the command-line filespec identifying a non-existant file. If the command-line filespec identified an existing file, it would be reflected as "replaced".

<4> Indicates the current cursor column. It is useful for establishing tab stops or columnizing text.

<5> Displays the hex value of the character at the cursor position. It is useful for ascertaining the specific character value of control characters unused by LED. It is also useful when working with graphic characters.

<6> Displays the amount of memory available in characters. This will be constantly updated as characters are added or deleted in your text.

## COMMAND FUNCTIONS

All major command functions of LED are invoked by simultaneously depressing the <CLEAR> key and one of the special function keys on the top row of the keyboard. A menu which depicts the labeling of the keys can be presented by depressing <CLEAR-COLON>. This menu will be placed in the bottom window, thus shortening the top window by three lines. It is recommended that you keep the menu visible at all times until you become proficient at remembering the LED control keys.

The menu that appears in the bottom window will look like the following:

```
INDNT FIND CHANGE HEX  UNMRK DNP  UPP  ALL  AGN     NAME EXIT
 =1=   =2=   =3=   =4=   =5=  =6=  =7=  =8=  =9=     =:=  =-=
INSRT LIN   DEL    WRD   BLK  END  TOP  SPA  TAB     MENU SAVE
..........................{Filename/EXT:Ø-N}( Ø):X'ØØ'|ØØØØØ
```

The commands identified by the upper row are invoked by simultaneous depression of the <CLEAR><SHIFT> and respective key <1-9,:,-> whereas the commands identified by the bottom row are invoked by depression of the <CLEAR> and respective key.

## INSRT

This will place LED into "insert" mode. In this mode, characters are inserted instead of overtyped. All text to the right (and below the cursor where applicable) will be shifted by one character position to make room for the inserted character. Insert can also be used followed by "LIN" to place LED into "insert-line" mode. When in "insert" mode, the cursor character will be changed as identified under CURSOR DISPLAY. Certain actions will cause LED to exit "insert" mode. If the <BREAK> key is depressed or the <LEFT ARROW> is depressed, LED will return to "standard" mode. If DEL is depressed, LED will exit from "insert" mode and enter "delete" mode. All other cursor movements will maintain the "insert" mode. With this arrangement, you will be able to conveniently establish columnized text during "insert" mode. You will also find text insertion more easily accomplished with minimal danger of undesirable overtype with LED's "insert" mode. Since this behavior may be different from that used with other word processor editors, you should practice text entry with LED to become comfortable with text insertion.

## LIN

If LED is in "standard" mode, the LIN command will perform a positioning of the cursor to the beginning of the line on the screen. This will be video column zero as noted by the column counter in the bottom window.

If LED is in "insert" mode, the LIN command will open up a blank line for text insertion. Subsequent depressions of LIN will continue to open up additional blank lines for text insertion. There is no need to depress INSRT-LIN if LED is already in "insert" mode; LIN is sufficient. If LED is in "insert" line mode and the cursor advances to the first character of the subsequent text while you are inserting characters, LED will automatically open another blank line. As long as you stay in "insert" mode, LED will not overtype subsequent text but will open up additional blank lines as required. You will be pleasantly surprised of this behavior. If you are inserting a large amount of text and there is text already in the text buffer following your insertion, you may realize an apparent speed increase by opening up multiple lines prior to your text insertion. This only reduces the amount of text moves performed by LED while you are typing. With the TYPE AHEAD keyboard in LDOS, you will not lose any characters regardless. Depression of the <BREAK> key will close up any trailing spaces created by INSRT-LIN. If you had already pressed <BREAK> and still have extra spaces to remove due to cursor relocation, you may use the DEL-SPA command sequence to delete the extraneous spaces.

If LED is in "delete" mode, then LIN will delete a line of text for each depression of LIN. A line of text is denoted as all text from the present cursor position up to and including the last character displayed on the video line containing the cursor character.

## DEL

The DEL control will put LED into "delete" mode. The cursor character will be changed as discussed in CURSOR DISPLAY. "Delete" mode can be used to delete a character, a word, a series of contiguous spaces, a line, a block, or to the end of the text. Each depression of DEL will delete the character at the current cursor position.

If LED is in "delete" mode, then depression of the WRD control will delete the current word. A word is identified as all characters from the current cursor position up to and including the first space, comma, right parenthesis, or <ENTER>.

If LED is in "delete" mode, then depression of LIN will delete the current line. A line of text is denoted as all text from the present cursor position up to and including the last character displayed on the video line containing the cursor.

If LED is in "delete" mode, then depression of SPA will delete all contiguous spaces from the current cursor position to the first non-blank character found. If the cursor is not positioned over a space, then nothing will be deleted if SPA is depressed. The DEL-SPA is a convenient way of closing up blank spaces created by a series of INSRT-LIN operations when <BREAK> has already been used to exit "insert" mode.

All marked blocks can be deleted when in "delete" mode by depressing BLK. There is no way within LED to delete a single block if more than one block is marked. You will be prompted when entering the DEL-BLK command with the message: "Delete ALL blocks! You SURE?" A response of "Y" is required for the deletion to occur. When deleting a block, make sure that the cursor is positioned within a marked block since the DEL command will delete one character before BLK can be entered.

You can delete all of the text from the current cursor position to the end of the text by DEL-END. A message prompt: "Delete to end of text?" will be displayed to ensure that your request was desired. Respond with "Y" or "N".

Depression of any other key will exit LED from "delete" mode and return LED to "standard" mode.

## WRD

If LED is in "standard" mode then WRD will cause a skip to the next word. If LED is in "delete" mode, then depression of WRD will delete a word. A word is identified as all characters from the current cursor position up to and including the first space, comma, right parenthesis, or <ENTER>.

## BLK

BLK will insert a block marker if LED is in "standard" mode. A block marker will be displayed as a solid graphics block - character value X'BF' (191 decimal). Block markers surrounding a block in the text buffer are not saved when the text is written to disk, therefore, they need not be unmarked prior to saving the text. If you want to reposition the block markers surrounding a block of text, all you need to do is to delete the block marker or markers, position the cursor where you want to put the marker(s), then remark the block. It is not necessary to use UNMRK - block markers can be deleted.

If LED is in "insert" mode, then the first marked block found in the text starting from the TOP will be inserted in the text starting at the current cursor position. If the cursor is positioned within the first such marked block, no insertion will be performed and an appropriate error message will be displayed.

All marked blocks can be deleted when in "delete" mode by depressing BLK. There is no way within LED to delete a single block if more than one block is marked. You will be prompted when entering the DEL-BLK command with the message: "Delete ALL blocks! You SURE?" You need to respond with a "Y" to continue the delete. When deleting a block, make sure that the cursor is positioned within a marked block since the DEL command will delete one character before BLK is entered. Before the delete is performed, LED checks the entire text buffer to ensure that block markers are paired. If an uneven number of block markers are present, a warning message is issued and no block delete is executed.

## END

END is used to position to the end of the text if LED is in "standard" mode. A depression of <CLEAR> <DOWN-ARROW> will also position the cursor to the end of the text.

You can delete all of the text from the current cursor position to the end of the text by DEL-END. A message prompt: "Delete to end of text?" will be displayed to ensure that your request was desired. Respond with "Y" or "N".

## TOP

TOP is used to position the cursor to the top of the text. You can also use <CLEAR> <UP-ARROW> to perform the repositioning.

## SPA

When LED is in "standard" mode, the SPA command can be used to skip over contiguous spaces and advance the cursor to the first non-space character.

If LED is in "delete" mode, then depression of SPA will delete all contiguous spaces from the current cursor position to the first non-blank character found. If the cursor is not positioned over a space, then nothing will be deleted. The DEL-SPA is a convenient way of closing up blank spaces created by a series of INSRT-LIN operations when <BREAK> has already been used to exit "insert" mode.

## TAB

TAB is used to set or reset tab stops. When a tab stop is set, its position will be indicated in the window separating line by a break filled with a dot. When the cursor is positioned over a tab stop and TAB is depressed, the tab stop will be removed. All tab stops can be removed in one step by depressing <TAB> <ALL>.

## MENU

MENU will toggle the menu display on and off from its presentation in the bottom window. When the <BREAK> key is depressed, it will display the command for MENU: "CLR :".

## SAVE

The SAVE command will initiate a text save operation without leaving LED. Text will be saved to the file identified on the command line unless a SAVE parameter redefined the output filespec or a NAME command redefined the output filespec.

## INDNT

INDNT is used to establish an indentation position to be invoked after every depression of the <ENTER> key. Indentation will be established at the cursor position currently in effect when INDNT is commanded. The depression of INDNT will not change the current mode of LED. Once indentation is placed in effect, it can be deactivated by positioning the cursor at video position zero and then entering the INDNT command. If you wish to retain indentation but want to temporarily override its effects, just perform a cursor to beginning-of-line <CLEAR><LEFT-ARROW> after depressing an <ENTER>.

## FIND

The FIND command is used to perform a search for a specified string of characters. The search string will be prompted for. The string entered is case dependent. When you enter the desired string, a dollar-sign character ($) appearing within the string is considered a wildcard character and will match up with any character.

Once a FIND is in effect, depression of AGN will invoke a search for the next occurance of the specified string. The ALL command can be used to find every occurance of the specified string; however the display will scroll so rapidly, each occurance will just flash by before your eyes. The FIND-ALL command sequence is, however, useful to locate the last appearance of the specified string.

## CHANGE

The CHANGE command is used to replace text strings identified with the FIND command. You will be prompted to enter the replacement string. Change will perform a FIND operation starting at one position following the current cursor position. Therefore, if you have just performed a FIND and want to replace that appearance of the string, then you must backspace the cursor one position prior to commanding the CHANGE operation. Once a string has been changed, the next appearance of the original string can be changed to the replacement string by depression of the AGN command.

If you desire to change all appearances of a string to a replacement string, first perform a FIND of the string. Next, backspace one character position to prepare for the FIND automatically performed by CHANGE. Perform a CHANGE command and enter the replacement string, then depress the ALL command.

## HEX

This command will place LED into HEX input mode. This is the mode that can be used to enter any character value in its hexadecimal form. Once you enter HEX input mode, the only way of exiting the mode and returning to standard input mode is by depressing the <BREAK> key. When in HEX input mode, you will be prompted to enter two hexadecimal digits {Ø-9,A-F}. All other entries will be ignored. HEX input mode may be entered while in "standard" mode (overtype), "insert" mode, or in "insert line" mode.

## UNMRK

The UNMRK command is used to remove all block markers. The blocks themselves are not disturbed. This command should be used when duplicating blocks of text by first surrounding a text block with markers and then performing INSRT-BLK commands at the appropriate text positions. UNMRK will then remove the unneeded block markers surrounding the marked text.

## DNP

This LED command will scroll the text down one page of the display. The bottom line of the present page is displayed at the top of the next page to present visual continuity of text. This method of scrolling is useful when reviewing text during final editing. A page of text is considered to be the screen area in the top window.

## UPP

This LED command will scroll the text up one page of the display. The top line of the present page is displayed at the bottom of the next page to present visual continuity of text. This method of scrolling is also useful when reviewing text during final editing. A page of text is considered to be the screen area in the top window.

## ALL

The ALL command is used with the FIND, CHANGE, and TAB commands to select global operations. Consult the information on these commands to learn the use of the ALL operator.

## AGN

The AGN command is another operator similar to ALL. It will find the next occurance of a specified string when immediately preceded by a FIND command. If AGN follows a CHANGE command, the next occurance of the specified string will be found and replaced.

## NAME

If you want to save the text buffer under a name different than that which was loaded, then you can depress the NAME command key. A prompt message, "Save f:", will appear in the information line of the bottom window to prompt for the entry of the filespec. Please note that this does not invoke a SAVE operation. The NAME command will just schedule the use of a new name for the next SAVE or EXIT (and save) operation. The filespec identified in field <2> of the information line will be updated only after the text is saved by a subsequent SAVE operation.

## EXIT

This command key is used to exit from LED. If no changes have been made to the text since the last SAVE operation, LED will return to LDOS Ready. If changes have been made, LED will issue the query, "Changes made. Save?". If you wish to save any of the changes made to the text, respond with a "Y", otherwise, none of the changes made since the previous SAVE will appear in the text file.

## KSM FILE EDITING

LED has been designed to support editing of KSM files. When the text filespec is one with a file extension of "KSM", LED will add a letter character followed by a right angle bracket (>) to the beginning of each line during loading and surround each such sequence with block markers. The letter will correspond to the KSM key for the phrase. For example, the first line will be prefixed with "A>", the second with "B>", and so forth.

Editing of the text can proceed as with any ASCII text file. When you want to save the edited file, position the cursor to the interior of one of the blocks and perform a DEL-BLK command. This will remove all of the prefixed markers and prepare the file for saving.

If you want to obtain a labeled listing of the KSM phrases, you may find it convenient to save the text to a different file (use NAME) without deleting the block markers. Since block markers are not written to an output file, the file will maintain the letter labels only. Then use the LDOS LIST command with the PRINT parameter and you will have a labeled listing of your KSM phrases.