

PART 4 The Assembler/Editor

13 USING THE EDITOR

The full screen editor developed for this text is simple to operate and yet powerful enough to enable assembler source code to be quickly and efficiently edited. All the example subroutines and programs given in the previous chapters have been prepared using this editor (and then assembled using the assembler described in the following chapter). In practice it is important to use this editor and not, for example, the word-processor package 'Quill', because the latter does not produce pure ASCII text files on the Microdrives. Pure text files are the only type of file which the complementary assembler can parse.

The editor is designed specifically for the creation of source (textual) programs. It allows up to 400 lines to be edited at any one time, with 72 characters per line. This is more than adequate for two major reasons. First, the size of program developed, at least in the early stages, is not likely to exceed this length, and second, the assembler will permit the inclusion of external 'library' files. If a large program is to be developed it is a simple case of creating one central program that will include as many external source file modules as it takes to produce the entire code.

13.1 Editor windows

On entry, the editor screen will appear as shown in Fig.13.1. Four windows exist in the display. Going from the top of the screen to the bottom the function of the windows is as follows.

At the very top there is a 'status' window. The contents of this window will show, in a continuous manner, the cursor line position, the total number of lines used, and the line position of a special marker which can be employed during an edit session.

Below the status window there is a 'tabulation ruler'. This particular window will never change. The tabulation ruler shows that nine tab-stops are available and that these exist at every eighth column position along a line. Each time the TABULATE key on the keyboard is pressed, the text cursor will move across to the next tab-stop. These tabulation positions are important in that the assembler, described in the next chapter, will tabulate its list-file in an identical way.

The third window in the display is the actual 'text window'. This is truly a window! It is a window with a view into your text. At any one time you can look at up to 17 lines of your program. The flashing cursor within the window enables you to edit your text program easily. When the editor is initially entered the text cursor will be in the top left-hand

corner of the text window, corresponding to column 1, line 0. Any text left in the editor buffer from a previous editing session will be displayed.

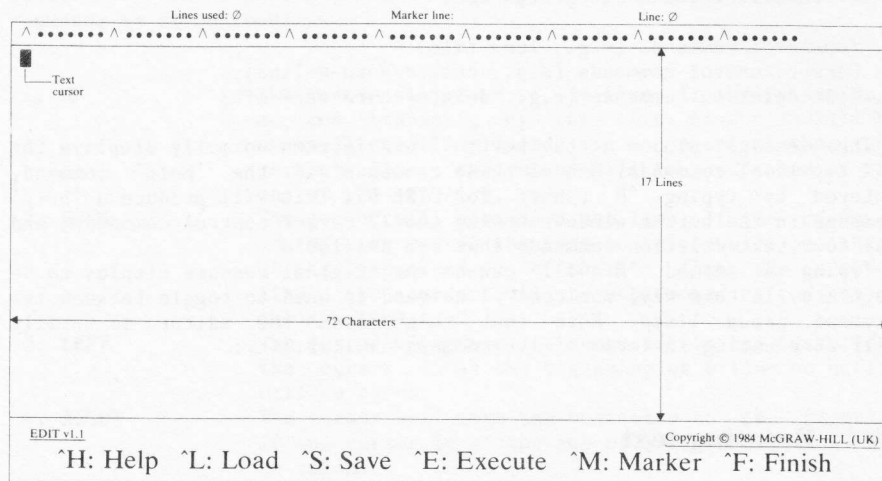


Figure 13.1 Editor screen layout

The last window, at the very bottom of the screen, is used as a message and prompt window. There will be a number of occasions when the contents of this window change. For example, when you request help, the 'help' message will appear in this bottom window.

13.2 Editor modes

When the editor is initially entered it will be in its command/edit mode. In this mode there are two major operations available. First, one of six top-level commands (listed in the bottom message window) may be executed. They are all control commands and are therefore entered by typing the appropriate control character (to do this hold the CTRL key down while typing the desired command character). Second, text may be entered and edited simply by typing the required characters, cursor control commands, and text deletion commands.

At first sight it may seem that there are a lot of options to learn. In practice, however, the editor is very simple to use and you can obtain a useful 'help' message, which lists all the cursor control and text deletion commands, should you need it!

13.3 Getting help

The editor has three basic groups of commands and a facility exists to enable you to view a list of the commands in each group, together with their function. The three groups are:

1. Top-level commands (e.g., load-file)
2. Cursor control commands (e.g., cursor-down-a-line)
3. Text deletion commands (e.g., delete-character-left)

The message window at the bottom of the screen normally displays the six top-level commands. One of these commands is the 'help' command, entered by typing ^H (short for CTRL-H). This will produce a 'help' message in the bottom window showing the 12 cursor control commands, and the four text deletion commands that are available.

Typing a second ^H will cause the original message display to be regained. In this way, one control command is used to toggle between two command group lists. Note that this means the editor is totally self-documenting in terms of its command availability.

13.4 Entering text

Text is entered simply by typing in the characters required. On any one line a maximum of 72 characters may be entered. To move on to the next line press the ENTER key, and then continue as before. At any time, except at the very end of a line, the TABULATE key can be pressed, and the cursor will move to the next available tab-stop on the current line.

The real power of the editor is, of course, in the ability to change text, either because it is wrong or because you wish to delete some lines or add extra lines. To do this we need to be able to move the cursor to the appropriate place in the text, and then delete or enter text accordingly.

13.5 Moving the cursor

A total of 12 'immediate' cursor control commands exist. They are entered by using one of the four cursor control keys (up, down, left, or right) in one of three ways:

1. **NORMAL** - the keys are used on their own.
2. **SHIFT** - the keys are entered as 'shift' keys (i.e., the SHIFT key is held down while the cursor control key is pressed).
3. **ALTMODE** - the keys are entered as 'altmode' keys (i.e., the ALT key is held down while the cursor control key is pressed).

The cursor may be moved left or right along a line in a variety of ways. The cursor may also be moved up and down the text. You will notice if you type in more than eight lines that the cursor will stay in the middle of the text window and the text will scroll around it. If, at

some later stage, you position the cursor within the first eight lines you will again notice that the cursor moves up and down and the text stays still. This cursor operation is purpose designed to enable you to see the current cursor line in its true context. This in turn makes editing the text much easier. The function of each of the cursor control commands is as follows:

1. **UP** - The cursor will move up one line. If the cursor is at the beginning of the text no action will be taken. If the line moved to is shorter than the current line, the cursor will be positioned at the end of the new line.
2. **DOWN** - The cursor will move down one line. If the cursor is at the end of the text no action will be taken. If the line moved to is shorter than the current line, the cursor will be positioned at the end of the new line.
3. **LEFT** - The cursor will move one character to the left. If the cursor is at the beginning of a line no action will be taken.
4. **RIGHT** - The cursor will move one character to the right. If the cursor is at the end of a line no action will be taken.
5. **SHIFT-UP** - The cursor will move up one page, equivalent to 16 lines. Notice that, as the text window is 17 lines deep, there will always be an overlap of one line. This feature will help you to scan the text more easily. The cursor will always be positioned at the beginning of the new line.
6. **SHIFT-DOWN** - The cursor will move down one page, equivalent to 16 lines. Notice that, as the text window is 17 lines deep, there will always be an overlap of one line. This feature will help you to scan the text more easily. The cursor will always be positioned at the beginning of the new line.
7. **SHIFT-LEFT** - The cursor will move one word to the left. If the cursor is at the beginning of a line no action will be taken.
8. **SHIFT-RIGHT** - The cursor will move one word to the right. If the cursor is at the end of a line no action will be taken.
9. **ALTMODE-UP** - The cursor will move to the beginning of the text.
10. **ALTMODE-DOWN** - The cursor will move to the end of the text.
11. **ALTMODE-LEFT** - The cursor will move to the beginning of the current line.
12. **ALTMODE-RIGHT** - The cursor will move to the end of the current line.

If the cursor is moved to a position within a line, and then text entered in the usual way, the characters will be inserted immediately prior to the character under the cursor. The rest of the line will be

seen to pan to the right.

13.6 Deleting text

Only four immediate text deletion commands exist, and these in turn only provide three functions (because two of the commands perform the same task). The commands are entered by using the normal cursor control keys (up, down, left, and right) together with the CTRL key. The function of the commands are as follows:

1. **CTRL-UP** - The current cursor line will be deleted. The command will not be executed if a marker line exists.
2. **CTRL-DOWN** - (Same as CTRL-UP).
3. **CTRL-LEFT** - The character immediately to the left of the cursor will be deleted. The rest of the line will pan to the left. No action will be taken if the cursor is at the beginning of a line. If the character to be deleted is a space (single or as part of a tabulation) then spaces will continue to be deleted until the entire gap to the left of the cursor is erased.
4. **CTRL-RIGHT** - The character under the cursor will be deleted. The rest of the line will pan to the left. No action will be taken if the cursor is at the end of a line. If the character to be deleted is a space (single or as part of a tabulation) then spaces will continue to be deleted until the entire gap to the right of the cursor is erased.

The above commands enable local text to be deleted. It is often useful to delete whole blocks of program text, and this can be done by using one of the editor 'execute' command options.

13.7 The ENTER key

When you have entered a reasonable number of program lines into the editor and moved the cursor around the text, you will undoubtedly notice that the ENTER key performs different functions at different times. Its functions may be defined as follows.

If the cursor is at the very end of the text then the ENTER key will move the cursor to the beginning of a newly created line, directly after the previous line. This makes the initial entry of text, and the appending of text, very simple. If the cursor is at the beginning of a line, but that line is not the last line of the text, then the ENTER key will create a new blank line at that point and move the rest of the text down. This enables new lines to be inserted within some current text very easily. If the cursor is within a line then the ENTER key will simply move the cursor down to the beginning of the next line.

13.8 Editor 'execute' command options

These commands are entered initially by using the top-level command ^E (CTRL-E). Four commands are available and they will be listed in the bottom message display window when the ^E command is given. To execute any one of the options simply press the character key corresponding to the first letter of the option. For example, to execute the 'find string' option you would simply enter F. The operation of the commands is as follows.

FIND STRING COMMAND

A prompt will be given in the bottom window asking for the textual string to search for. Simply type in your search string and press the ENTER key. The editor will search for the string, beginning at the start of the current cursor line. If the string is found then the cursor will be moved to the beginning of the string, and the text window updated accordingly.

If the search string is not within the text searched, the cursor will remain in its previous position. Note that the case of the text is not relevant. For example, 'This' is exactly the same as 'this', as far as the editor search option is concerned.

It may be that the editor search option places the cursor at a match, which is not the particular one you were looking for. If this is so then remember to move the cursor down to the beginning of the next line, or the search option will simply find the same one again.

DELETE BLOCK COMMAND

This is the one command that requires the use of the special marker, so let us look at this first. A marker symbol (shown as a right-sided comilla - double angled bracket character) can be entered, by using the ^M (CTRL-M) top-level command, to mark any particular line in the text. The marker will always be entered and shown at the beginning of the current line, regardless of the current cursor position, and the cursor moved to the first character in the line. It is not possible to mark a line which is completely full, and neither is it possible to mark more than one line.

Assuming a marked line is available, the 'delete-block' command will delete all lines between the marked line and the current cursor line inclusive. The command will issue an error message if no marker is present (type any character to continue after the error message is printed).

MOVE TO LINE COMMAND

This command lets you move to an absolute line within the text. A prompt will be given requesting the number of the line to which you wish to move. Enter the appropriate number and press the ENTER key. The cursor will be moved to the beginning of the corresponding line and the text window updated.

If a line number less than zero is entered, the cursor will be moved to the beginning of the text. Conversely, if a line number greater than the total number of lines available is entered, the cursor will be moved to the end of the text.

INCLUDE FILE COMMAND

When editing program text it is very useful to be able to merge in other bits of program text from another file. This command will enable a text file (produced by this editor) to be included in the current source text at the current cursor position. A prompt will be given for the device and file name of the external file, which must be on a Microdrive. Simply enter the appropriate information and the file will be included in the current text. While the text inclusion is taking place a series of '+' markers will be displayed in the bottom window to act as an indicator. Without such an indicator the editor could appear to lock-up, whereas in fact it is simply doing some internal shuffling.

Care must be taken over this operation. If the requested file does not exist, the editor will report a fatal error and cease running! You are advised to save a copy of your current text on a Microdrive before executing this command. The command will issue an error message if a marker line is present (type any character to continue after the error message is printed). An error message will also be given if the editor runs out of storage in the process of trying to merge in the external file.

When specifying the device and file name of the external file, the extension may or may not be given. If it is left off, the default extension '_ASM' will be used.

13.9 Loading a text file

A new source file can be loaded into the editor from a Microdrive by using the ^L (CTRL-L) top-level command. The file must have been created previously using the editor. Any current text will be erased from the memory of the editor and the cursor will be returned to the beginning of the new text.

When specifying the device and file name of the file, the extension may or may not be given. If it is left off, the default extension '_ASM' will be used.

13.10 Saving the current text

The current contents of the editor buffer can be saved on to a Microdrive by using the ^S (CTRL-S) top-level command. The contents of the editor will not be erased and therefore the 'save' command can be used any number of times during an editing session for safety backup purposes.

When specifying the device and file name of the file, the extension may or may not be given. If it is left off, the default extension '_ASM' will be used. It is not possible to save text which has a marker line in it, and under such a condition an error message will be issued (type any character to continue after the error message is printed).