

Terminal Editor
for C500C

UMI-R3-166



Terminal Editor

Revision	History	Date
001	First release.	99-05

Contents

Chapter 1

<u>Terminal Editor</u>	<u>1</u>
<u>Sample Program</u>	<u>1</u>
<u>Starting the Editor</u>	<u>2</u>
<u>Creating a New Program</u>	<u>2</u>
<u>Opening an Existing Program</u>	<u>2</u>
<u>Editing Programs</u>	<u>4</u>
<u>Inserting Lines</u>	<u>4</u>
<u>Listing Lines.....</u>	<u>5</u>
<u>Editing Characters of a Line</u>	<u>6</u>
<u>Deleting a Line</u>	<u>6</u>
<u>Replacing a Line.....</u>	<u>6</u>
<u>Finding a Line</u>	<u>8</u>
<u>Displaying Options and Help</u>	<u>9</u>
<u>Displaying the Main Menu</u>	<u>9</u>
<u>Displaying Help.....</u>	<u>9</u>
<u>Saving and Quitting.....</u>	<u>11</u>
<u>Saving the Program.....</u>	<u>11</u>
<u>Quitting the Editor</u>	<u>11</u>

CHAPTER 1

Terminal Editor

On a C500C controller, you can edit or create a program on the controller using the editor available through the terminal

When using the editor, you can:

- Open a program
 - Open a new program
 - Open an existing program
- List all the lines of a program
- Find a specific line
- Edit the characters in a line
- Modify an entire line at one time
 - Insert a line
 - Replace a line
 - Delete a line
- Save the program and exit from the editor
- Get help on editing features

Sample Program

The examples in the next sections use this sample program.

```
1:ready()  
2:move(a)  
3:move(b)  
4:finish()  
5:grip(1)  
6:finish()  
7:printf(" The program is finished running \n")
```

Starting the Editor

When you start the editor, you can create a completely new program or work on an existing program.

Creating a New Program

To create a new program file and start editing the program, you need to access the editor. You do this by entering the edit command and the program name.

```
>>edit  new_filename
>>edit  pick_place
>>edit  dispense
```

Since the program does not exist, the terminal responds with:

```
That file does not exist. Do you want to create it? (y/n) _
```

If you want to create the program, type y. The file name and the main menu are displayed.

File Name: *new filename*

```
'I'nsert
'L'ist
'E'dit
'D'elele
'R'eplace
'F'ind
'S'ave
'H'elp
'M'enu
'Q'uit
```

:

The main menu displays all of the possible editing commands. After this menu, the prompt changes from ">>" to ":" because the terminal is now in EDIT mode.

Opening an Existing Program

To open an existing program you need to start the terminal and name the existing program.

```
>>edit  old_filename
>>edit  pick_place
>>edit  dispense
```

The file name and the main menu are displayed.

File Name: *old_filename*

```
'I'nsert  
'L'ist  
'E'dit  
'D'eleete  
'R'eplace  
'F'ind  
'S'ave  
'H'elp  
'M'enu  
'Q'uit
```

:

The main menu displays all of the possible editing commands. After this menu, the prompt changes from ">>" to ":" because the terminal is now in EDIT mode.

Editing Programs

The main menu lists all of the available editing functions.

To perform an editing function,

- type the single letter for that function
- type the number of the line to perform that function on
- press the Enter key.

If you enter a non-existent command, you get the error message:

```
*** You entered a wrong char ***
```

If you enter a non-existent line number, you get the error message:

```
*** You entered a wrong LINE NO. ***
```

Inserting Lines

The Insert command is used to insert a new line or series of lines into the program.

When you insert a line, you specify a current line number. Your new line is inserted immediately before your current line with that number.

Inserting One or More New Lines

Starting with the sample program:

```
1:ready()  
2:move(a)  
3:move(b)  
4:finish()  
5:grip(1)  
6:finish()  
7:printf(" The program finish running \n")
```

To insert a new line 2, enter

```
i2
```

After you press the Enter key, the editor prompt displays the line number

```
2:
```

Type in the new line and press Enter.

```
speed (10)
```

The editor prompt displays the line number of the next line.

```
3:
```

You have two options:

- Continue inserting lines by typing another line and pressing Enter.
- Quit inserting lines by pressing Enter.

If you inserted only speed(10) as a new line 2, the program is now:

```
1:ready()  
2:speed(10)  
3:move(a)  
4:move(b)
```



```
5:finish()  
6:grip(1)  
7:finish()  
8:printf(" The program finish running \n")
```

Listing Lines

This command is used to list or display the contents of a line or a series of sequential lines.

Listing a Single Line

If you want to list line 3, use the list function, l, followed by the line number:

```
l3
```

Line 3 is displayed.

```
3:move(b)
```

Listing a Series of Lines

If you want to list lines 2 to 4, use the list function, l, followed by the line you want to start at (2), a comma, and the line you want to finish at (4).

```
l2,4
```

The lines are displayed:

```
2:move(a)  
3:move(b)  
4:finish()
```

Listing from the Beginning

If you want to list from the beginning of the program to a specific line, use the list function, followed by a comma (the editor assumes line 1), and then the line you want to finish at (3).

```
l,3
```

The lines are displayed:

```
1:ready()  
2:move(a)  
3:move(b)
```

Listing to the End

If you want to list from a specific line to the last line, use the list function and the line you want to start at.

```
l3
```

The lines are displayed:

```
3:move(b)  
4:finish()  
5:grip(1)  
6:finish()  
7:printf(" The program finish running \n")
```

Editing Characters of a Line

This command allows you to change the characters of a line.

To edit the characters of a line, you need to enter the edit command (e) and the number of the line to edit.

Example

If you want to edit line 5, enter:

```
e5
```

Line 5 is displayed:

```
5:grip (1)
```

Use the right and left arrow keys to move the cursor along the line. Type new characters. Delete existing characters.

Deleting a Line

This command deletes a line. Once deleted, a line cannot be restored.

To delete a line, enter the delete command, followed by the number of the line to be deleted.

```
d <line number>
```

Notice that no comma is required between the d and line number.

Example

Start with the sample program with a speed setting at line 2:

```
1:ready()  
2:speed(10)  
3:move(a)  
4:move(b)  
5:finish()  
6:grip(1)  
7:finish()  
8:printf(" The program finish running \n")
```

To delete line 2

```
d2
```

The program is now:

```
1:ready()  
2:move(a)  
3:move(b)  
4:finish()  
5:grip(1)  
6:finish()  
7:printf(" The program finish running \n")
```

Replacing a Line

Use this command when you want to delete an existing line and insert a new line all in one step. You can use this command to change more than one line at a time.

To replace a line, use the replace command (r) and the number of the line being replaced. There is no comma between the command and the number.

```
r line_number
```

Replacing a Single Line

If you want to replace line 2:

```
r2
```

The editor prompt changes to the line number:

```
2:
```

Type the new line. When you press Enter, it takes the place of line 2.

Replacing a Series of Lines

Start with the following sample program.

```
1:ready()  
2:move(a)  
3:move(b)  
4:finish()  
5:grip(1)  
6:finish()  
7:printf("The program is finished running. \n")
```

If you want to delete lines 4 to 6 and insert a new line or new lines, you begin inserting at line 4 and continue until you are finished.

```
r4,6
```

The editor prompt changes to the earliest line number:

```
4:
```

Type a line, such as

```
calrdy()
```

and press Enter to insert a new line 4. The editor prompt changes to the next line number.

```
5:
```

To insert a new line 5, type the line, such as

```
ready()
```

and press Enter to insert a new line 5.

The editor prompt changes to the next line number.

```
6:
```

To stop inserting new lines, press Enter.

The program now has the following lines.

```
1:ready()  
2:move(a)  
3:move(b)  
4:calrdy()  
5:ready()  
6:printf("The program is finished running. \n")
```

Finding a Line

There are two ways of using this command:

The format for finding a line is:

```
f
```

Note: The Find command is case sensitive for lowercase and UPPERCASE letters.

Finding a Specific Line

If you want to find the line, `printf ("The program is finished running. \n")`, enter the find command:

```
f
```

The editor prompts you for the string of characters of the line.

```
Please enter your string:
```

Enter the string

```
printf ("The program is finish running. \n")
```

The editor displays the line with its number.

```
8: printf ("The program is finish running. \n")
```

Finding Similar Lines

If you want to find all occurrences of "move", you can also use the find command. Enter:

```
f
```

The editor prompts you for the string of characters of the line.

```
Please enter your string:
```

Enter

```
move
```

The editor displays:

```
2: move (a)
```

```
3: move (b)
```

If no match to the string is found, the editor displays the message:

```
No match for this string ...
```

Displaying Options and Help

Displaying the Main Menu

This command displays all of the command options available, but in less detail than the help command.

m

The editor displays:

File Name: *filename*

```
'I'nsert
'L'ist
'E'dit
'D'elete
'R'eplace
'F'ind
'S'ave
'H'elp
'M'enu
'Q'uit
```

Displaying Help

You use the help command to display all of the editor options. The format is:

h

The editor displays:

Inset <LINE #>

Exp: I3 'Insert before Line#3'

List <FROM>,<TO>

Exp: L 'List all'

L1,3 'List from Line#1 to Line#3'

L1 'List Line#1 only'

L1, 'List from Line#1 to end of file'

L,3 'List from Line#1 to Line#3'

Delete <FROM>,<TO>

Exp: D3,6 'Delete from Line#3 to end of Line#6'

F'ind 'list lines containing a string'

E'dit <LINE #>

Exp: E3 'Display and Edit the characters in line #3'

R'eplace <FROM>,<TO>

Exp: D3,6 'Overwrite Line#3 to Line#6 And insert Line#3'
'S'ave 'to Save the file'
'M'enu' 'to show main menu again'
'Q'uit 'to Quit from Edit'
Exp: 'Y' 'to Quit with saving'
Exp: 'N' 'to Quit without saving'

Saving and Quitting

Saving the Program

Use this function to save changes of the program. The save format is:

```
s
```

If there were no changes to save, the editor responds with:

```
No changes made on this file !!!
```

Quitting the Editor

This command exits the editor.

```
q
```

Example

After all editing is complete, enter the Quit command:

```
q
```

The editor responds with:

```
Do you want to Save ('Y'es / 'N'o) ?
```

If you enter **Y**, the program is saved, the editor is exited, and control is passed back to the command line interpreter

```
> for ash  
$ for operating system
```

If you enter **N**, the editor is exited without saving changes to the program.

Any other input returns you back to the editor prompt, **:**.

