Nathan Isaman ME 477: Lab 2 27 Jan 17

1. Description

The objective of this lab is the coding and testing of the *getchar_keypad()* function. This function captures a user's keypad entries and stores them into a buffer string prior to returning them one-by-one to the calling function (*fgets_keypad()*). Using a buffer string allows the user to delete erroneous entries in a straightforward manner. This function should do the following:

- Accurately return the user input to the calling function
- Not cause an error by buffer string underflow using DEL (i.e. pointing to position -1, etc)
- Correctly display the currently stored buffer string on the LCD (removing deleted chars)
- Correctly edit the buffer string (i.e. move pointer) when DEL is pressed

Hierarchy:

```
Main
|_MyRio_Open()                  // Opens a session with the MyRio
|_printf_lcd()                  // Prints formatted string to LCD screen
|_fgets_keypad()                // Captures keypad input by user and stores in a string
|        |_getchar_keypad()     // Creates buffer string from keypad inputs
|                 |_getkey()    // Captures key entry on keypad
|                 |_putchar_lcd()  // Prints passed character to the LCD screen
|_MyRio_Close()                 // Closes the session with the MyRio
```

Note: Some functions are called multiple times per hierarchy tier, but only one call is shown above.

2. Testing

The *getchar_keypad()* function was tested by calling the *fgets_keypad()* function in *main()* and entering the following (and other) entries into the keypad.

Keypad Input	**Returned Output fgets_keypad()**	**LCD Output**	**Desired Output**
5 5 DEL ENT	5	5	5
- 1 2 3 4 5 DEL DEL DEL ENT	-12	-12	-12
1 DEL DEL 3 4 5 ENT	345	345	345
DEL DEL DEL 1 ENT	1	1	1
DEL 1 DEL 1 ENT	1	1	1
0 . UP UP . DOWN DEL ENT	0.[[.	0.[[.	0.[[.
DEL DEL ENT	<blank>	<blank>	<blank>

3. Results

The function performed as desired and met all requirements laid out in the description section. The pointer and character counter were not decremented when the delete key was entered with a string length of zero, which prevented an undesirable buffer string underflow from occurring. The pointer and character counter were properly decremented when the delete key was entered with a string length greater than zero. The erroneous character was also removed from the LCD as desired. The LCD output and the *fgets_keypad()* output, which is a concatenation of *getchar_keypad()*'s output, matched the expected output from the user based on keypad entries.