**Readme File for Embedded Systems Lab Assignment 3 (Blinky\_With\_Console)**

* Features have been added to the program of Assignment 2 ( Blinky ) to make the switching of LED lights and changing of colours possible through user commands in addition to making it happen through switch 1 and switch 2 of TIVA board respectively.
* The console port was initialized with Baud Rate of 115200 bps.
* The characters sent from the PC has been echoed back in the terminal ( **MobaXterm** was used to set up a serial connection between the PC and TIVA board)
* The ASCII characters that are echoed back are stored in a global character array called **console\_cmd\_buffer**. The size of this buffer is of 30 bytes. If number of characters exceed this size, then they are not stored in the buffer as the max capacity would have been reached.
* Command is processed once the Enter key is pressed. Once the Enter key is pressed , all the contents of the buffer are considered for processing and once processing is done, the character array ( buffer ) is emptied so that new commands can be processed.
* **Pre-Processing the commands to remove extra space:**

1. Support has been added to the code to remove white spaces ( space bar and tabs )
2. The way in which this is achieved is once enter key is pressed, all the contents of console\_cmd\_buffer except tab and space bar characters are copied into another buffer called **console\_cmd\_buffer2**. And all the processing is done in this new buffer that does not include any space bar and tab entries. So, this will the program **space insensitive**. For example, if the user were to enter “**Color Green**” or “**Color Green**” or “**ColorGreen**”, all would mean the same thing.

* **Pre-Processing the commands to convert upper case characters to lower case characters:**

1. Support has been added in the code to make the commands **case insensitive.**
2. Once the enter key has been pressed, all the contents of the character array are converted to lower case using the “**tolower()**” function present in C. This converts all the characters in the command entered by the user into lower case characters. And this makes the program **case insensitive**.
3. That is, if the user were to give a command “**Color Green**” or “**coloR gREeN**”, all would mean the same thing.

* **Supporting Back Space Functionality:**

1. Back Space functionality support has also been added to the program.
2. If the user presses back space, the cursor moves back by one position ( one position to the left ).
3. Once it moves back by one position to the left, we enter “ **‘ ‘ character “** to make it look as though the character at that point was deleted.
4. Once **‘ ‘** is entered, the cursor would move one position to the right. So in order to bring it back to the position in which the character was replaced with ‘ ‘ , we move the cursor back by one position to the left one more time.

* **Commands supported by the application:**

1. Color [ color\_name]
2. Blink [blink\_rate]

* The above two commands are case insensitive and space insensitive. So, the user can enter commands in **upper case**, **lower case** , **mix of both upper case and lower case** , **with white spaces** and also **without white spaces** . All these will be supported by the application.
* If the user gives any command other than the ones mentioned above, then the following prompt is given to the user :

**Please enter any of the following commands:**

1. **Color <Color\_Name>**
2. **Blink <Blink\_Rate>**

**Valid numbers to be entered after 'blink' are 1,2,4,8,16,32**

* This program supports blink rates of 1,2,4,8,16 and 32. **The numbers indicate how many times the LED will blink in 2 seconds**. So the unit of the blink rates in **per 2 seconds**.
* Global variables **Color** and **Blink\_delay** are used in the program to maintain the state of the colour and the blink rate. This will make ensure continuity of functionality when the user switches between using the console to give commands or using the SW1 and SW2 of TIVA board to give commands.