ECE 177 Lab 8 Spring 2022

Purpose: More programming practice, using functions, interfacing to a 4x4 keypad, using "`printf()`" with the LCD.

Assignment: We will use the 4x4 keypad along with the LCD display to make a simple decimal to hexadecimal converter. The user will enter a number on the keypad and then press the "#" key and the LCD will display the number and its hexadecimal equivalent. The program will then loop back for another number to be entered. For more information see class notes. Your code will include and use the following functions. Function prototypes are given with the "starter code" on the class web page.

`getkey()`: this function will "scan" the keypad and return the currently-pressed key or -1 if no key is currently pressed. The mapping of the GPIOB pins to the keypad is given with the comments on the "starter code". Bits 4,5,6, and 7 of GPIOB are connected to the keypad rows. They are shown as outputs, but you will actually "scan" the rows, making them outputs one at a time (and simultaneously making that line a logic "1"). When one row pin is an output, the other rows and all columns should be inputs. As each row is activated (made an output) you will check the column bits of GPIOB to see if any are logic 1, and if they are, the pressed key is given by 4*row + col. The keypad columns are hooked to bits 8,9,10 and 12 (not 11) of GPIOB. Use the HAL delay to delay 20ms before returning from the routine to take care of "key bounce."

`mapkey()`: this function will map a key from a key number to the actual digit or letter printed on the keypad key (shown in the starter code comments). For example, if 3 is input, then return 10, and if 12 is input then return **`'*'`** (ASCII code for asterisk). This function should store the mapping in an array and simply look up the value and return it. Return -1 if the input argument is outside the range of 0 to 15.

`getnum()`: this function will wait for the user to enter zero or more digits followed by the '#' key . It will return the number entered. The "*" key will be used as a backspace. Keys corresponding to letters will be ignored. This function can work as follows: declare "`result`" as an **`int`** and initialize it to zero. Then in a loop wait for the buttons to be released (**`getkey()`** returns -1), then wait for a button to be pressed (**`getkey()`** returns a non-negative number), then "map the key" using mapkey(), and if the mapped key is a digit, multiply `result` by 10 and add the new digit; if it is an asterisk, then divide the result by 10 (performs a backspace); if it is the "#"key, then break out of the loop and return the result. Ignore the letter keys. See starter code and class notes for more information.

The main program will simply initialize a few things (see starter code) and then in an infinite loop will call `getnum()` and then display the result as "decimal -> hex."For example entering 255 will cause **`"`**`255 -> 0xFF`**`"`** to be displayed.

Prelab: CodeLab problems have been created for this lab. In addition some test functions are given so you can work on and debug this code before entering it in CodeLab.

Notes: Do the above first for a "C" grade. For the "B " grade: in the "**`getnum()`**" routine, have the "A" key be a "change sign" key. Each press will negate the sign of the current number. For the "A" grade also do the following: Pressing the "B" key will display the number of times **`getnum()`** has been called so far, pressing the "C" key will display the numbers of time **`getnum()`** returned an even value and pressing the "D" key will display the average of the values returned so far.

When finished, have your circuit and correctly formatted and commented code checked off by the TA.