Laboratory Worksheet #10
Keypad/LCD Input Exercise

This worksheet is an activity to learn how to use the functions associated with reading from the keypad and writing to the LCD display. In completing the requested C-program you will develop a segment of code that can be integrated into your Lab 4, 5 and 6 exercises. This will permit you to input multi-digit values from the LCD keypad while ignoring multiple inputs of the same key because the user didn't release the button quickly enough. This is very similar to the pushbutton switch debouncing that you have already performed for the Lab 2 exercise.

Exercise 1: Inputting a single keypad character

1) Download the `kpdlcdtestPCA.c` code.

2) Change XBR0 to be consistent with your lab 3 code.

3) Connect the LCD/Keypad to your protoboard, using the header pins to connect to power, ground, SDA and SCL.

    When you put the LCD/Keypad away, make sure the header pin is connected to the ribbon cable, not left on your protoboard.

4) Verify the hardware and software are setup correctly by running the `kpdlcdtestPCA.c` code and checking the output on both Putty and the LCD.

    `Note:  You should see multiple lines being printed.`

5) Change the code so that a single push and release of a keypad button results in a single read and print (on both Putty and the LCD).

    This implementation is very similar to reading a pushbutton press in Lab 2, with different indicators for press and release.

6) Add code that converts numeric characters to their decimal value. Add a print statement that prints the decimal value, using %u typecasting.

Report the results of your print statement.

Button that was pressed

_____

Print output for ASCII value

Character: _____   Hex: _____

Print output of decimal value

Decimal: _____

Exercise 2: Inputting a multdigit number

1) Change the Exercise 1 code to read a two digit number. The first key input should be the 10s digit and the second key input should be the ones digit. Print the number, using %u typecasting to verify your code is working correctly.

   Single digit values may be entered by pressing '0' for the first digit.

2) change your program to use the function `kpd_input(char mode)` to accept a multi-digit unsigned integer using the keypad.. Your program should take the unsigned integer value returned by the function and print it on the SecureCRT terminal (not the LCD panel). Try both modes [`kpd_input(0)` and `kpd_input(1)`] and note the differences in the way they work.

3) Note what happens when a value is entered outside the range of 0 − 65535. Enter 65536 and 65537 to see what value is actually returned.

4) Find the equation that gives the actual value returned by the function when the input value is outside its allowed range.

5) Predict what will be returned when 99999 is entered.

   Result of entering 65536

   _____

   Result of entering 65537

   _____

   Equation for actual value when any 5 digit number greater than 65535 is input

   _____

   Result of entering 99999

   _____

There are some critical issues with the use of kpd_input() that may cause the 8051 to freeze. It seems if the I2C bus communication sequence is interrupted (a PCA0 interrupt) in the middle of a transaction the processor locks up waiting for something that will never occur. The best way to avoid this problem, other than repeatedly disabling interrupts just before using kpdinput(), is to keep your PCA ISR short and efficient. Also make sure you are NOT using any Timer 0 interrupts. They are no longer necessary for anything. When printing values received from kpdinput() remember to use %u rather than %d since they must be declared as unsigned int.

This program should be completed BEFORE integrating together the two parts of Lab 3 into a single program to control both the steering and speed of the car in Lab 4.

When complete, include Worksheet 10 with your Laboratory 4 Pre-lab submission.