CS 454/654 Lab #1
LCD Module and Digital I/O

Spring 2024 (*Revision*: 2.0)

Demonstration due the week of Jan 31, 2024.

1 Overview

The purpose of this lab is to perform basic programming tasks on the user interface (UI) board, with specific attention on LCD, LEDs and joystick buttons.

You must write a (single) program that implements all of the following logic:

1. Display the name of each group member on the LCD. Each name should be displayed on a new line. (10 points).
2. Blink LED4 three times. The time between blinks should be exactly 1 second. The LED should be turned off at the end of the three blinks. (10 points).
3. LED1 should be on when joystick's button 1 (fire) is pressed and off when the button is not pressed (10 points).
4. LED2 should be on when joystick's button 2 (thumb) is pressed and off when the button is not pressed (10 points).
5. LED3 is off whenever buttons 1 and 2 both have the same state (both held down or both not held down). LED3 is on whenever buttons 1 and 2 have different states (10 points).
6. Keep an accurate (debounced) 8-bit unsigned count of the number of times that joystick's button 1 has been pressed. Display the counter's current value on the LCD in a fixed location as both a hexadecimal value and a decimal value (50 points).

2 Procedure

1. Before getting started, read sections 2, 3.1, 3.2, 4.1 - 4.7, 5 in the CS-454/654 Laboratory Manual (available on the course website, as well as on Piazza).
2. Setup for Lab1:
   (a) Follow the instructions in Lab0 to create a new project, or copy the HelloWorld project directly from within MPLAB X IDE. Either way, name the new project so that the resulting folder is "Lab1.X" and the name of the project is "Lab1".

(b) Download the auxiliary files for Lab1 from either Piazza or the course website.

(c) The auxiliary files archive contains only a template file, namely *joystick.h*. You will need to implement the macros declared but not implemented in the file.

3. Next, update *main.c* so that it fulfills the requirements specified in the Overview section. When doing so, please note:

   (a) If you want to achieve delays of predictable length in your code, use the functions `__delay_ms(...)` and `__delay_us(...)`, which insert a delay of the specified amount of milliseconds and microseconds, respectively. These functions are provided by the *libpic30.h* library.

   (b) Printing text to the LCD can be a relatively slow process. Therefore, static text (like the names of the team members) should be printed only once outside of the main program loop. In the provided demo of Lab 01, the first few lines of text are all static.

   (c) Debouncing the joystick trigger should be done using the algorithm proposed in section 4.6.3 of the Lab Manual.

   (d) Be careful when performing port operations and be sure to insert `Nop()` statements where appropriate.

   (e) Do not forget to appropriately initialize the direction of any I/O pin you will be using. Set pins controlling LEDs as output, and pins connected to the joystick's button as input.

At the start of Lab 2, each lab group will be asked to demonstrate their Lab 1 working setup to the TF.

The following questions are provided for your lab group to think about. No written response is required.

3 Questions to Ponder

1. When debouncing the trigger button, how many consecutive 0s do you read before considering the button pressed?

2. Suppose the microcontroller's CPU clock was slowed from 16 MHz to 8 MHz. How many consecutive 0s would you need to read to debounce the trigger?