Course Name: ECE372A
Author:
NetID:
Date:
Project Name: Lab3

Description

1. In this lab, you will interface with an LCD to display various characters. Using bit shifting and masking you will send commands to the LCD. You will need to create a microsecond delay for the LCD command delays that will work up to at least 5000 microseconds.

2. You will also create a millisecond delay to control the rate at which several LEDs blink. These LEDs will represent a binary number and will count up continuously to 15. A switch will toggle the rate at which the binary number counts up between an original rate and twice the original rate. The switch will do so using an interrupt service routine. (Note: This part is essentially Lab 2 which you have already completed. In lab 3 we are adding an LCD display to Lab 2).

3. The LCD display will show the current counting speed, fast or slow, of the LED counter.

a. Display on the first line “Blinking rate = ”

b. Display on the second line “Fast” or “Slow” based on the speed of the LED counting rate.

Instructions

Examine comments in lcd.cpp, main.cpp, timer.cpp, switch.cpp, and led.cpp and complete the described functions. You will need to create a circuit using your breadboard, jumper wires, potentiometer, resistors, LEDs, and LCD.

Requirements

Overall

1. The project must follow good coding practices and be well commented.

2. Arduino libraries are not allowed at all for this lab with the exception of debug functionality using Serial.println.

3. The LCD must display the proper information on two lines.

lcd.cpp

1. All functions provided must be used in the implementation of the LCD interface according to their descriptions. Please read comments within the blank Lab 3 file

2. PORTA0, PORTA1, PORTA2, and PORTA3 must be used for the data pins on the LCD with PORTA0 corresponding to the least significant bit

3. PORTB4 and PORTB6 must be used for the enable pin and RS pin, respectively.

4. When four bits need to be assigned to PORTA, it must be done in **one line of code**. There are times where you need to send four bits twice, in this case, you need two lines of code.

switch.cpp

1. A function implementation that returns void and has no parameters called initSwitchPB3 must be present and is used in the main function to initialize the switch on the pin named *PB3*.

2. PB3 must be initialized with an input pull-up resistor

3. Pin-Change Interrupts must be enabled for pin PB3

led.cpp

1. A function implementation that returns void and has no parameters called initLED must be present and is used in the main function to initialize all LED pins as outputs. Use PORTD for the led function.

2. Pins named PD0, PD1, PD2, and PD3 must be used to control the LEDs.

3. A function implementation called turnOnLEDWithChar that returns void and has a parameter called num of type unsigned unsigned char must be present.

4. The turnOnLEDWithChar function must be **one line of code.** See LED control section for more details.

timer.cpp

1. Implement a precise microsecond timer using any timer module that you wish.

2. implement a precise millisecond timer using any timer module that you wish.

main.cpp

1. Display the following message on the LCD to receive full credit.
Display on the first line “Current count rate is: ”
Display on the second line “Fast” or “Slow” based on the speed of the LED counting rate .