CS 362 Lab 8 Spring 2024

Learning Goal:

- Learn how to program with interrupts
- Learn how to write Interrupt Service Routines: ISR

Lab: Due Monday 3/25/2025 at your registered lab time.

Materials

- 1x Arduino
- 1x Breadboard
- 1x 16x2 LCD
- 1x Potentiometer
- 1x 220 Ohm resistor
- 2x Pushbuttons
- 2x 10k Ohm resistors
- Wires

Write a program for the Arduino that utilizes Interrupt Service Routines, an "ISR". Interrupts allow a section of Arduino code to be called without writing code in the loop( ) function to explicitly check that code. Note that an ISR should be short and quick executing code that normally just change values of global variables (see link for discussion and restrictions).

The lab is to have you use 2 buttons and the 16x2 LCD display. Each button is to trigger an interrupt. The interrupts are to toggle between two states/messages displayed on the 16x2 LCD.

- When no button has been pressed your 16x2 display should say something like "System has been running for X seconds", where X starts at the value 0 and is updated as time goes on (i.e. don't display X, display the time in seconds!). **This is State 0.**
- When the first button is pressed, the display should display "Interrupt received! Press button 2 to continue". **This is State 1.**
- When the user presses the second button, if the machine is in State 1, restart the time for X to 0 seconds and have display goes back to displaying "System has been running for X seconds". Then go back to **State 0**. You may want to have your code first go to a 3rd State to perform the time reset before switching to State 0.

Note: that if the system is in State 0 (displaying the "running for X seconds" message), the time should not be reset if the second button is pressed! Pressing the second button WITHOUT being in State 1 is to do nothing.

Each button is to be attached to its own interrupt function. Thus, you are required to write two separate interrupt functions for this lab.

Information about how to tell the system to call your Interrupt Service Routine (ISR) can be found at:

http://arduino.cc/en/Reference/attachInterrupt

Important note: You must use pin 2 and pin 3 for this lab. Those two pins are specifically set up to work with interrupts.

CS 362 Lab 8 Spring 2024

Information about timer: https://www.arduino.cc/en/Reference/Millis

Your code must be submitted to Gradescope BEFORE you demo your lab!

Late Policy

- Lateness is determined by the time the lab is demonstrated, not when the .ino file is submitted.
- Labs that are not demonstrated get a score of 0.
  - -50% for no demonstration
  - -50% for being late
- Late Submission 1
  - Demonstrated before 11:59pm Thursday after Lab Due Date
  - 25% Penalty
- Late submission 2
  - Demonstrated between Friday and Monday after Lab Due Date
  - 50% Penalty
- Note this lab is due on Monday immediately at the end of Spring Break, March 25,2024

What should I include with my .ino Code File?
As with any code file, it should be written in Good Coding Style: in a manner that will help other people read and understand the intent, purpose, operation of the code. So your code must include:

- Name the .ino file with your NetId and Lab Number
  - I.E. something like: ptroy4Lab2.ino
- Header Comments (including the following)
  - // FirstName LastName, UIN and NetID
  - // Lab x - Title
  - // Description - what is this code supposed to do?
  - // Include any assumptions you may have made, what do you expect from the hardware, pinouts, particular arduino versions, etc.
  - // References - where did you find code snippets, ideas, inspirations? if no references used say: "no references used"
- Code is well documented/formatted with comments, indentations, and descriptive variable names
- Actual code - the functions in the cpp/ino file

Academic Integrity Guidelines:
You may use any resources linked from this lab, or posted by the professor or TAs on piazza/class web page/etc. You should not look at any other internet resources for this. This is an individual assignment, and should be completed on your own. You should not show anyone your code, or look at anyone else's code.