C

Project 2: Digital Inputs and Interrupts

Overview

For more information on digital inputs, interrupts, and filtering, please consult Chapter **??** and Example **??**

Make these parts and put it into the book

This project is designed to introduce you to concepts of filtering digital inputs and handling interrupts inside a microcontroller's program. It will also introduce a little bit about the Serial monitor. You will wire up three buttons to your Arduino and use them to turn on and off an LED. One button will not have any filtering or regulation at all; another will have a software-defined debounce filter; and the third will have a low-pass RC circuit filter. You will use an interrupt attached to each input pin to increment a counter and display the current press count to the Serial monitor. This will demonstrate how inputs to a program need to be considered as false readings can create undesired program behaviors. You should notice that the unregulated button has difficulty incrementing the counter by 1 every time you press it and may not always switch the light from off to on or vice versa. The software-regulated button input should be a little more accurate and reliable, but you may still notice some problems. The hardware-regulated button input should throw few, if any false positives - always incrementing by 1 with every press and switching the LED state consistently.

Special Consideration

For information on which pins to use and how to set up the interrupts, see `https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/`

The project as intended, requires three interrupt-capable digital input pins. The Arduino Uno and other ATmega328 devices can only handle two interrupt pins. The Arduino Mega and some other devices can handle more than two interrupt pins. If you have an Arduino Uno, you may request a Mega from the instructor or one of your class mates for this project. Please note that if you elect to loan a unit from the instructor, this project *will not* be graded until that loaner unit is returned.

Alternatively, you may demonstrate the project in two parts. In the first part, you will implement the unregulated and software-defined filtered button input. In the second part, you will implement the unregulated and hardware filtered button input. Please discuss this with your instructor for additional information.

Requirements

Reminder: The LED *must* be connected to the Arduino pins with resistors in series in order to protect the diodes. Please consult your Arduino kit manual for specific resistor values

For this project you will be required to demonstrate:

1. appropriately wiring three buttons and three LEDs to your Arduino and breadboard with all supporting components
2. reading in the three digital inputs and attaching appropriate interrupt service routines to the pins
3. incrementing counters for each button press and switching the corresponding LED on and off
4. appropriately printing the counters to the Serial monitor

Submission

You will be required to submit the following on Canvas:

1. a video of the project working with narration of what is occurring
2. a well-organized and documented schematic of the project setup
3. the source code file
4. a printout of the Serial monitor showing a counter increment when you press the corresponding button

Please package all of these items into a compressed (zipped) folder and upload them to Canvas.

Grading

You will be graded along the following criteria:

Criterion	Points
Efficacy	40
3 types of input filtering	20
Printout of Serial monitor	20
Schematic neatness	10
Code neatness	10
Extra credit	5

Extra Credit

An extra credit opportunity exists with this project! To earn the credits, you must program the buttons to increment their counters and change LED states *after* holding the button for a certain period of time. How you implement this is up to you. If you want to try and get the extra credit points, please let the instructor know in the submission and detail why you believe you earn the points.