Authors:
NetIDs:
Date:
Project Name: Lab 4

Description
In this lab, you will work in a ** team ** of 3-4 people. Since this
is the case, the code given to you is extremely minimal. You will
work together to control two motors using the L293D amplifier
supplied in every kit. Using the potentiometer, the speed of the
motors will be controlled.

Instructions
First, you will need to be able to capture an analog voltage from a
potentiometer that can swing from 5V to 0V.
The digital value captured from the potentiometer will control the
duty cycle of several PWM signals that control the motors. These PWMs
will be connected to the L293D amplifier and this amplifier will
connect to either side of the motors.

You will use two motors. The motor is controlled by a difference in
voltage between its two terminals. Therefore, to get it to spin
forward at full speed, it is only necessary to apply a 5V signal to
one side of the motor and 0V to the other side.

When the potentiometer is at its lowest point (~ 0V), only one motor
will be on at full power. When the potentiometer is at its highest
point (~ 5V), the other motor will be on at full power and the other
one will be off. When the potentiometer is in the middle (~ 2.5V),
both motors will be on at full power. This behavior emulates that of
wheels turning in a car with the potentiometer being the steering
wheel and the motors being the wheels.

You should be able to turn the entire system on and off by pressing a
debounced switch.

#Requirements
Overall
1. Code must be readable and well commented.
2. Every file must contain code related to a single device. For
example, the ADC and PWM code must be separated. The main function is
an exception.
3. There are no non-trivial SFR manipulations in the main function and
are wrapped in functions that have meaningful names
5. A state machine is used to implement the bulk of the functionality
of the program

ADC
1. A function exists to initialize the ADC
2. Uses the A0 pin as an input

PWM
1. Uses 2 PWMS on timer 3 and timer 4.
2. Has a changeDutyCycle function.

Timer
1. Uses a timer (0 or 1) to debounce states

Switch
1. Uses a switch to turn motors on and off.
2. Switch uses an ISR and is set up for debouncing