Authors:
NetIDs:
Date:
Project Name: Lab 6

 # Description
In this lab, you will work in your designated ** team ** within your lab section. Since this is the case, the code given to you is extremely minimal. You will work to interface with the adxl345 accelerometer using the SPI communication protocol. To show you have interfaced correctly, you will print the x, y, and z axis values to the monitor using Serial print statements. You will also have a buzzer alarm sound when the sensor values from the accelerometer reach a designated threshold level. A push button switch will silence the piezo alarm.

 # Instructions
You will need to create a circuit using your breadboard, jumper wires, accelerometer, piezo speaker and a push button switch.

 # 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/printing functionality using Serial.println.

In a file called spi.cpp

1. All communication with the accelerometer must be done over the SPI related pins.
2. It is suggested that you write the following SPI related functions to interface with the ADXL345 accelerometer:
   a. spi initialization or setup function.
   b. digitalwrite(pin, value) function (chip select line).
   c. spi_write function (output from master - MOSI)
   d. spi_read function (input from device - MISO).

   Note: If your code combines some of the functions described above, that is acceptable. The main goal is to demo the accelerometer working for all 3 axis using your code for the serial peripheral communication protocol.

main.cpp

1. Data should print out all 3 axis data points once every 1000 milliseconds.
2. If the accelerometer movement reaches a defined threshold value, a piezo alarm will trigger. Your group will determine the threshold value to trigger the piezo through experimentation.
3. Once the piezo alarm is triggered it will remain on until it is silenced by pressing a button switch.

timer.cpp

1. Implement a precise millisecond timer using timer 1.

pwm.cpp

1. Use a PWM output signal to change the frequency of the piezo in order to generate a chirping sound.

switch.cpp

1. Uses a switch to silence the audio chirping alarm.