ECE 376 - Homework #3

Binary Outputs and Timing. Due Monday, September 12th

Solder your PIC board (50pt)

Demonstrate that your PIC board works

- In person, video, de1mo during Zoom office hours
- 50pt: Board your built powers up & you're able to download code
- 25pt: Board soldered but not working (swap for a working board)
- note: If your board doesn't work, we have working boards we can swap with you

Binary Outputs

1) Design a circuit which allows your PIC board to turn on and off an RGB Piranah LED at 0mA (off) and 20mA (on). Assume the specifications for the LEDs are:

Color	Vf @ 20mA	mcd @ 20mA
red	2.0V	10,000
green	3.2V	10,000
blue	3.2V	10,000

2) Design a circuit which allows your PIC board to turn on and off a 5W LED. The specs for the LED are:

- Vf = 6.0-7.0V
- Current = 700mA
- 500-600 Lumens (equivalent to a 60W light bulb).

https://www.ebay.com/itm/1W-3W-5W-10W-50W-100W-High-power-SMD-Chip-LED-COB-White-Blue-Red-Light-Beads/124011607823

Timing:

3) Write a program which outputs the music note E4 (329.63 Hz)

- Verify the frequency of the square wave you generate
- (Pano Tuner app on you cell phone works well for this)

Lab:

Problem 4-7) Design an embedded system with your PIC board which includes some timing. Some suggestions are

- Binary Clock: Have the PIC count 0..10 at a rate of once per second on PORTD. When PORTD reaches 10, it clears and PORTC increments.
- Electronic Dice: Generate random numbers based upon which button is pressed
  - RB0: 4 sided die (numbers 1..4)
  - RB1: 6 sided die (numbers 1..6)
  - RB2: 10 sided die (numbers 1..10)
  - Beep for 1 second at 220Hz each time a new number is generated
- Electronic Piano: Play notes B4 - B5 when you press buttons RB0 .. RB7
- Strobe Light: Generate a strobe light on RC0. The light should be
  - On for 1ms then Off for 99ms
  - repeat
- Other

4) Requirements: Specify

- The inputs
- The outputs
- How they relate (what your program does)

5) Analysis, Code, and Flow Chart. Give computations for resistor values (if any), timing, assembler code, and a flow chart for your code

6) Validation: Collect data in the lab to verify your code works.

- For a binary clock, is it counting once per second?
- For the dice, are the results random? Is the beep 220Hz? Is it 1 second?
- For the piano, is each note correct in frequency?

7) Demonstration: Demonstrate that your embedded system works (either in person or with a video)