ECE 376 - Homework #3

Binary Outputs and Timing. Due Monday, September 12th

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

Since the PIC is driving a load that needs

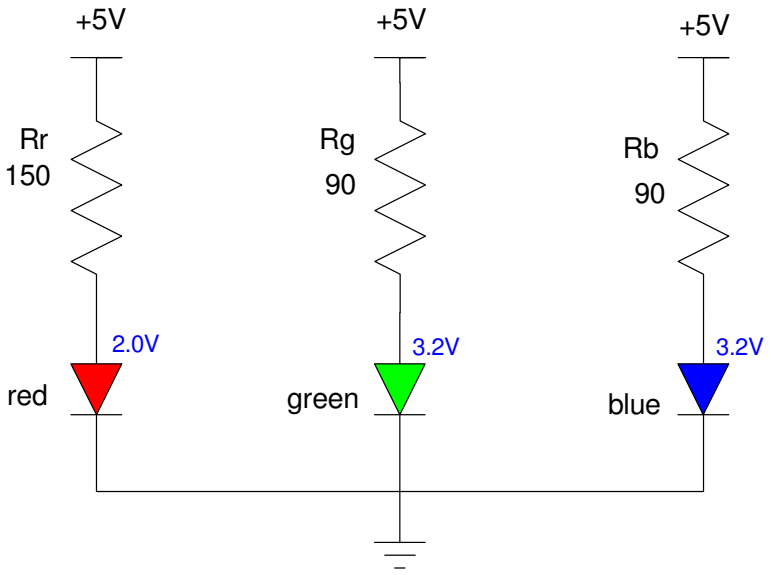
- Less than 5V and
- Less than 25mA

a PIC can drive the load directly using only a resistor to limit the current:

$$R_r = \left(\frac{5V - 2.0V}{20mA}\right) = 150\Omega$$

$$R_g = \left(\frac{5V - 3.2V}{20mA}\right) = 90\Omega$$

$$R_b = \left(\frac{5V - 3.2V}{20mA}\right) = 90\Omega$$



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

In this case, you need a transistor since the current is more than a PIC can output. Assume a 6144 NPN transistor.

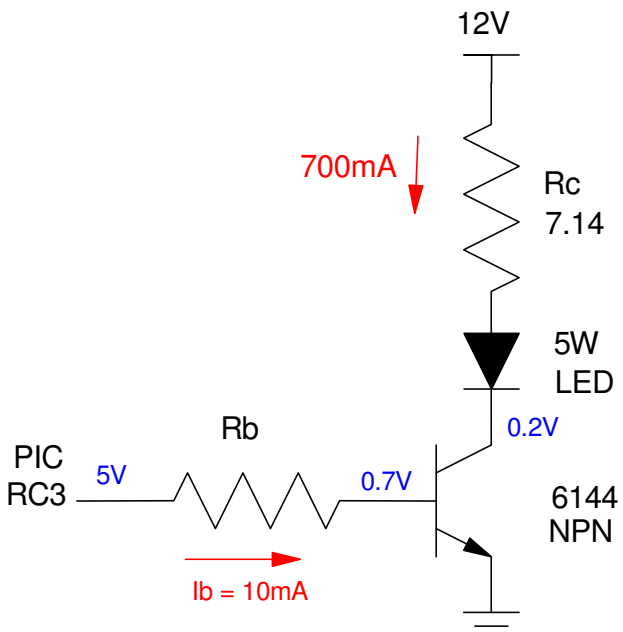
- $\beta = 100$ *worst case*
- $V_{ce}(sat) = 0.2V$

$$R_c = \left(\frac{12V-7V}{700mA}\right) = 7.14\Omega$$

$$I_b > \frac{I_c}{\beta} = \frac{700mA}{100} = 7mA$$

Let Ib = 10mA

$$R_b = \left(\frac{5V-0.7V}{10mA}\right) = 430\Omega$$



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)

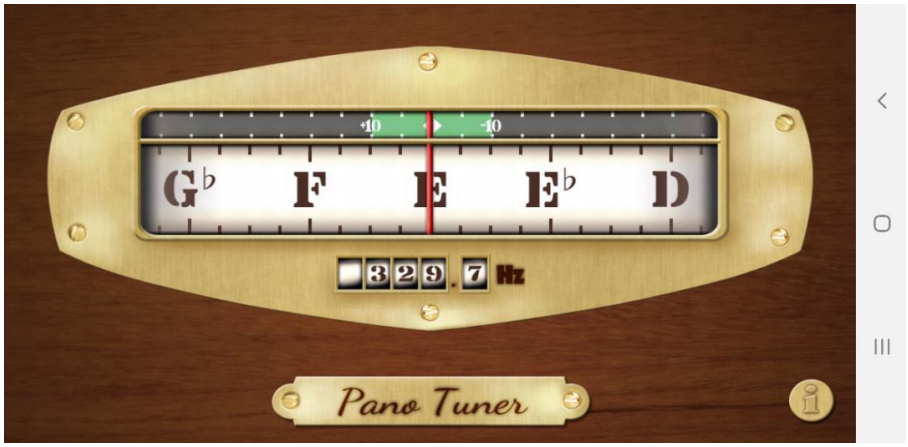
The number of clocks per toggle (the timing for the wait loop) is

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 15,168.52$$

Come up with a wait loop that burns 15,168 clocks:

N = 10*A*B + 5*A + 9 = 15,168

A = 7, B = 216 results in N = 15,164 (off by 0.03%)




```
#include <p18f4620.inc>

; Variables
CNT0 EQU 1
CNT1 EQU 2

; Program
   org 0x800
   call Init
Loop:
   incf  PORTC,F
   call Wait
   goto Loop

; --- Subroutines ---

Init:
   clrf TRISA
   clrf TRISB
   clrf TRISC
   clrf TRISD
   clrf TRISE
   movlw 0x0F
   movwf ADCON1 ;everyone is binary
   return

; Wait 15,168 clocks  (actual wait = 15,164)
Wait:
   movlw 7         ; A
   movwf CNT1
W1:
      movlw 216    ; B
      movwf CNT0
W0:
         nop       ; 10 clocks
         nop
         nop
         nop
         nop
         nop
         nop
         decfsz CNT0, F
         goto W0
      decfsz CNT1, F
      goto W1
   return
```

Lab:

4) Requirements:

- Inputs: Buttons on RB0 / RB1 / RB2 / RB3
- Outputs: RC0
- Relationship: Output a square wave on RC0 based upon the button pressed:
  - RB0: 185.00 Hz (F#3)
  - RB1: 207.65 Hz (G#3)
  - RB2: 233.08 Hz (A#3)
  - RB3: 277.18 Hz (C#4)

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

The number of clocks needed for each note are:

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right)$$

N is created using a series of loops:

N = 10AB + 5B + 5

185Hz:

- N = 27,027.027
- A = 245, B = 11

207.65 Hz:

- N = 24,078.979
- A = 240, B = 10

233.08 Hz

- N = 21,451.86
- A = 238, B = 9

277.18 Hz

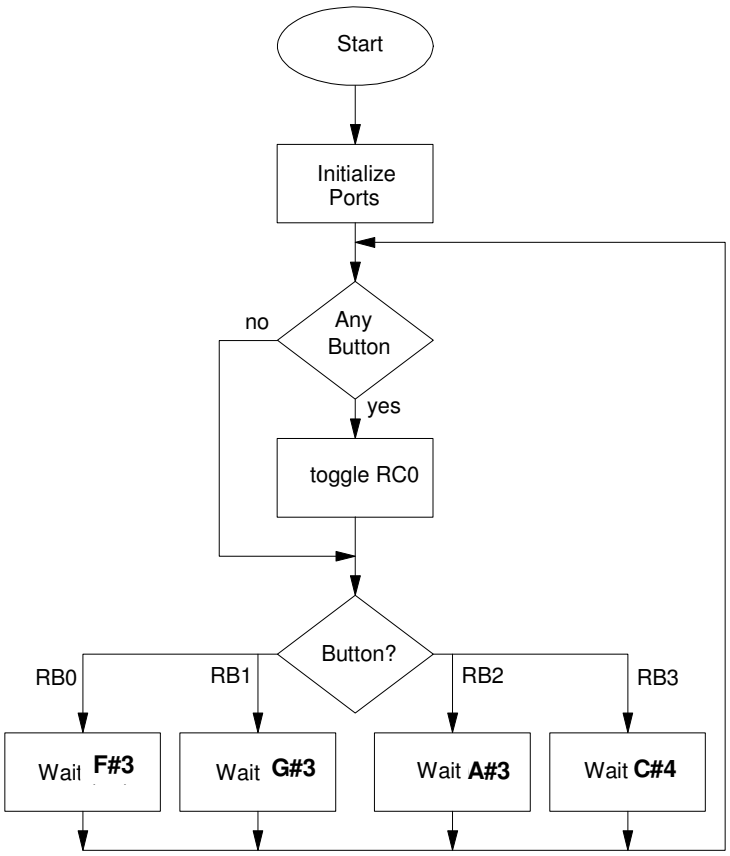
- N = 18,038.819

A = 225, B = 8

```
; --- Piano4.asm ----
; When you press button RB0..RB3, you play a note
; on RC0:
; RB0:  185.00 Hz (F#3)
; RB1:  207.65 Hz (G#3)
; RB2:  233.08 Hz (A#3)
; RB3:  277.18 Hz (C#4)

#include <p18f4620.inc>


:
:
:
assembler 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?

RB0: 185.00 Hz

RB1: 207.65 Hz

RB2: 233.08Hz

RB3: 277.18Hz

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