ECE 376 - Homework #3

Binary Outputs, and Timing. Due Monday, September 13th, 2021

Please make the subject "ECE 376 HW#3" if submitting homework electronically to Jacob_Glower@yahoo.com (or on blackboard)

Solder your PIC board (50pt)

Demonstrate that your PIC board works

- In person, video, de1mo during Zoom office hours
- 50pt: Board you built powers up & you're able to download code
- 25pt: Board you built is 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. You'll need a working board for the rest of the course.

Binary Outputs

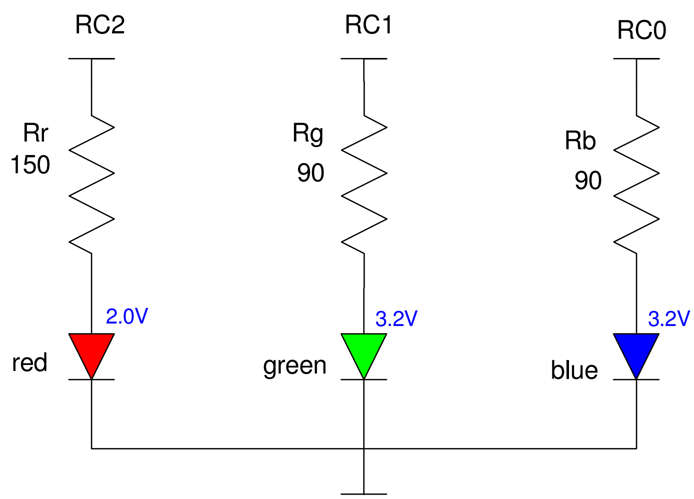
1) Design a circuit which allows your PIC board to turn on and off an RGB Piranah LED at 0mA (off) and 20mA (on). Assume the specifications for the LEDs are:

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

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

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

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



2) Design a circuit which allows your PIC board to turn on and off a 10W LED. The specs for the LED are:

- Vf = 10.0 - 11.0V
- Current = 700mA to 1000mA
- 550 - 650 Lumens (equivalent to a 60W light bulb).

Assume

- a 24V power supply
- 6144 NPN transistor
  - gain = 200
  - Vce(sat) = 0.36V

Rc (sets the current to 1000mA)

$$R_c = \left(\frac{24V - 11V - 0.36V}{1000mA}\right) = 12.6\Omega$$

Rb: Saturate the transistor.

The base current has to be at least

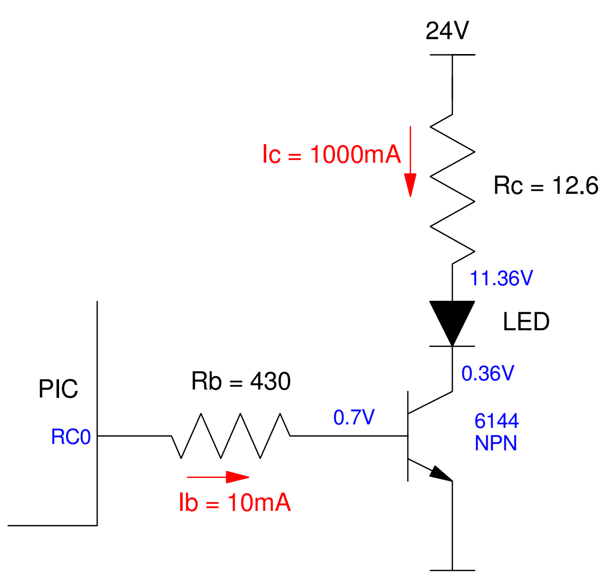
$$I_b > \frac{I_c}{\beta} = \left(\frac{1000mA}{200}\right) = 5mA$$

Pick something bigger than 5mA, less than 25mA (the most a PIC can outout). Let

Ib = 10mA

then

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



Assembler Coding

3) Determine the contents of the W register and memory locations A and B after each assembler command

Command	W	A	B
; Start	7	8	9
addwf A,W	**15**	8	9
subwf B,F	15	8	**250 (-6)**
incf A,W	**9**	8	250
incf B,F	9	8	**251**
movlw 23	**23**	8	251
andwf A,F	23	**0**	251
iorwf B,W	**255**	8	251

Timing:

4) Write a program which outputs the music note G2 (98.00 Hz)

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

98 Hz gives a wait loop of

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 51,020.04 \text{ clocks}$$

```
#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
   return

; Wait 51,020 clocks  (actual wait time is 51,260 clocks
Wait:
   movlw 51
   movwf CNT1
W1:
         movlw 100
         movwf CNT0
W0:
               nop
               nop
               nop
               nop
               nop
               nop
               nop
               decfsz CNT0, F
               goto W0
         decfsz CNT1, F
         goto W1
   return

   end
```

Using Pano Tuner, the actual freuqncy is 97.6Hz

Lab: 4 Key Sharp Piano

5) Requirements:

- Inputs: Buttons on RB0 / RB1 / RB2 / RB3
- Outputs: RC0
- Relationship: Output a square wave on RC0 based upon the button pressed:
  - RB0 F#3 185.00 Hz
  - RB1 G#3 207.65 Hz
  - RB2 A#3 223.08 Hz
  - RB3 C#4 277.18 Hz

6) 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 = 5AB + 5B + 5 + 13 (main routine = 13 clocks)

185Hz:

- N = 27,027
- A = 36, B = 146, N = 27028

207.65 Hz:

- N = 24,079
- A = 28, B = 166, N = 24088

233.08 Hz

- N = 21,452
- A = 31, B = 134, N = 21458

277.18 Hz

- N = 18,038
- A = 16, B = 212, N = 18,038

Code & Flow Chart

```
; Program
   org 0x800
   call Init
Loop:
        movlw 0
        cpfseq PORTB ; if any button is pressed
   btg   PORTC,0

   btfsc PORTB,0
    call  B0
   btfsc PORTB,1
    call  B1
   btfsc PORTB,2
   call  B2
   btfsc PORTB,3
   call  B3

   goto Loop

; --- Subroutines ---

Init:
   clrf TRISA         ;PORTA is output
    movlw 0xFF
   movwf TRISB        ;PORTB is input
   clrf TRISC         ;PORTC is output
   clrf TRISD         ;PORTD is output
   clrf TRISE         ;PORTE is output
   movlw 0x0F
   movwf ADCON1       ;everyone is binary
   return

B0:
   movlw 21
   movwf CNT1
B0a:
   movlw 164
   movwf CNT0
B0b:
   nop
   nop
   decfsz CNT0, F
   goto B0b
   decfsz CNT1, F
   goto B0a
   return

B1:
   movlw 28
   movwf CNT1
B1a:
   movlw 166
   movwf CNT0
B1b:
   nop
   nop
   decfsz CNT0, F
   goto B1b
   decfsz CNT1, F
   goto B1a
   return

B2:
   movlw 31
```

```
   movwf CNT1
B2a:
   movlw 134
   movwf CNT0
B2b:
   nop
   nop
   decfsz CNT0, F
   goto B2b
   decfsz CNT1, F
   goto B2a
   return

B3:
   movlw 16
   movwf CNT1
B3a:
   movlw 212
   movwf CNT07
B3b:
   nop
   nop
   decfsz CNT0, F
   goto B3b
   decfsz CNT1, F
   goto B3a
   return


   end
```

8) 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?

Frequency	Hz	Measured	Error (%)
F#3	185.00 Hz	185.3Hz	+0.162%
G#3	207.65 Hz	207.9Hz	+0.120%
A#3	223.08 Hz	233.4Hz	+0.137%
C#4	277.18 Hz	277.4Hz	+0.079%

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