# ECE 376 - Homework #3

## Binary Inputs

A thermistor has the following temperature - resistance relationship:

$$R = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

where T is the temperature in degrees C.

1) Design a circuit which outputs
   - 0V when T < 5C
   - 5V when T > 5C

First, compute the resistance at 5C

$$R = 2566.99\Omega$$

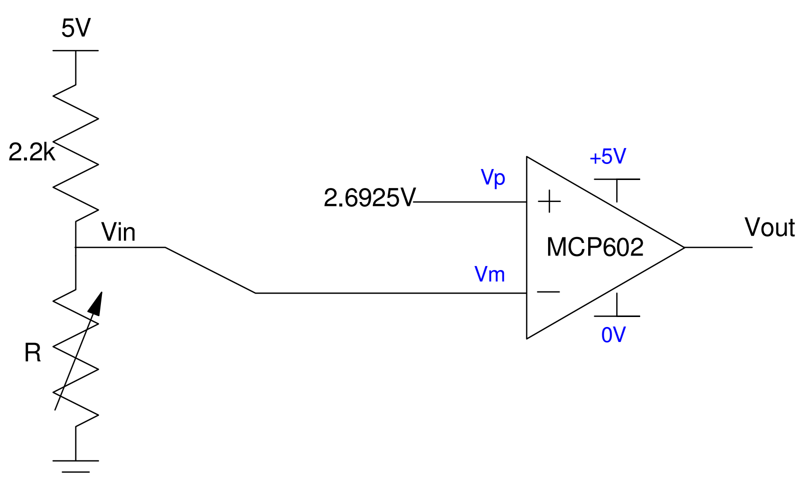
Next, pick your favorite voltage divider. Assume a 2.2k resistor and a 5V source

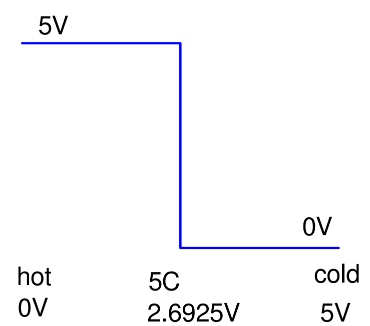
$$V_{in} = \left(\frac{R}{R+2000}\right)5V$$

$$V_{in} = 2.6925V$$

Finally, connect this to a comparitor. For the plus/minus inputs, pick whichever one results in T>5C (rather than T<5C). This works out to the minus input

- As T goes to infinity, R goes to zero
- Vin goes to zero
- Vout goes to 5V



2) Design a circuit which outputs

- 0V when T < 0C
- 5V when T > 5C
- No change for 0C < T < 5C

At 0C

- R = 3320.12 Ohms
- Vin = 3.0073V *V(off)*
- Vout = 0V

At 5C

- R = 2566.99 Ohms
- Vin = 2.6925V *V(on)*
- Vout = 5V

V(on) < V(off) so connect to the minus input

V(on) = 2.6925V

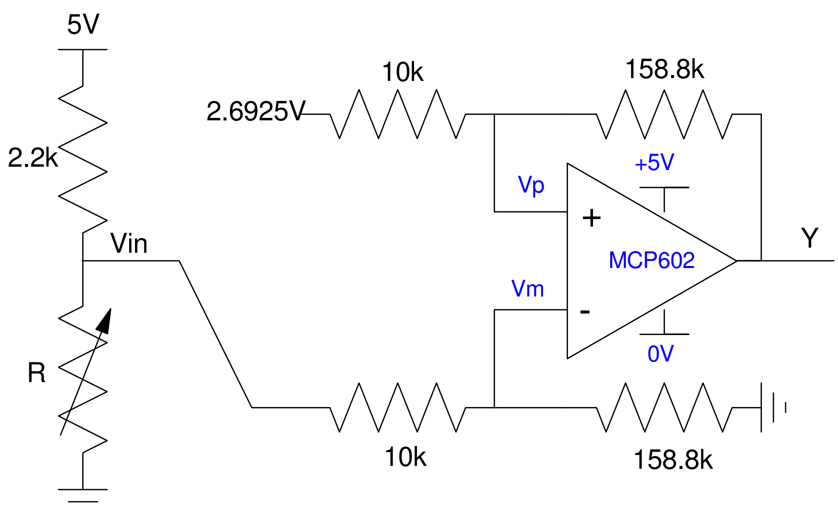
The gain required is

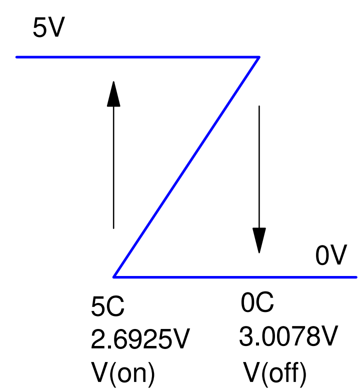
$$gain = \left(\frac{\text{change in output}}{\text{change in input}}\right)$$

$$gain = \left(\frac{5V-0V}{3.0073V-2.6925V}\right)$$

$$gain = 15.88$$

Set the resistor ratio to 15.88 : 1



### Binary Outputs

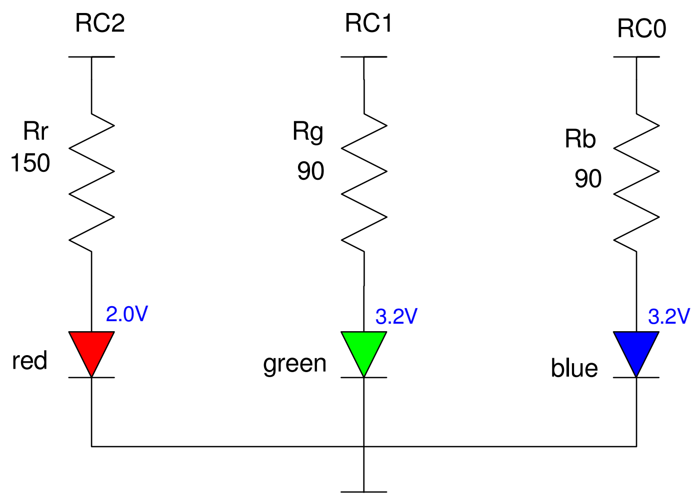
3) Design a circuit which allows your PIC board to turn on and off an RGB Piranah LED at 0mA (off) and 20mA (on). Assume the specifications for the LEDs are:

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

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

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

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



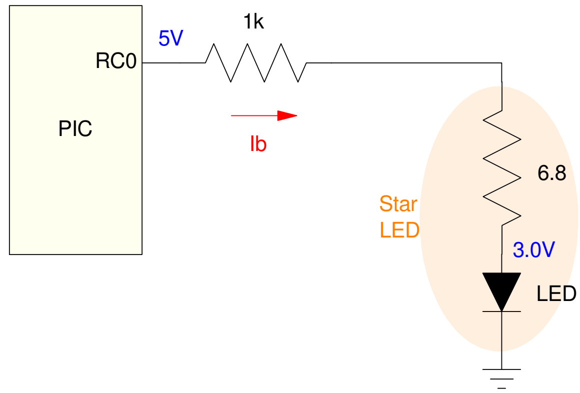
### 1W Star LED

4) The star LED in your lab kit is a 1W white LED in series with 6.8 Ohms.

- Vf = 3.0V @ 330mA
- 200LM @ 330mA

a) Set up the following circuit so your PIC board can turn the LED on/off

- The 200 Ohm resistor limits the current (200 to 1k works)




b) Compute the

- Current to the LED
- The brightness of the LED

$$I_b = \left(\frac{5V - 3.0V}{1k + 6.8}\right) = 1.986mA$$

$$Light = \left(\frac{1.986mA}{330mA}\right) 200LM = 1.2 \text{ lumens}$$

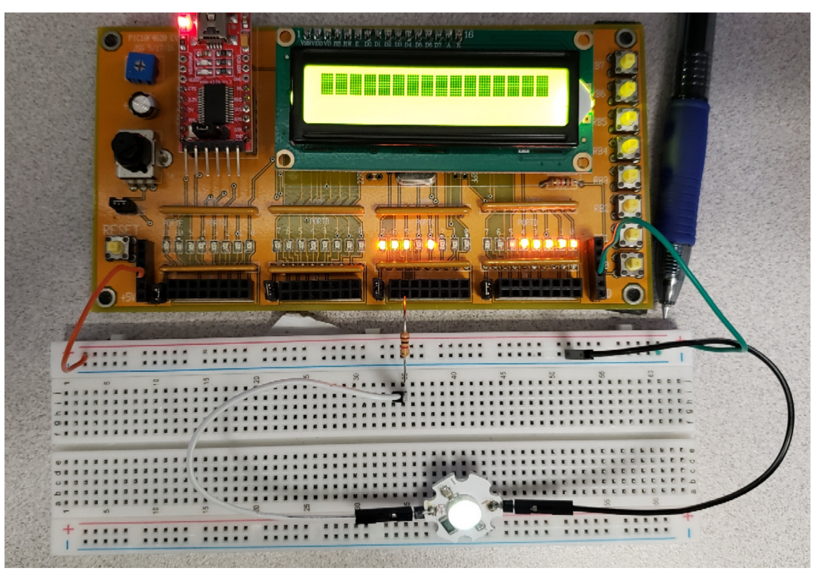
c) Build this circuit and measure

Id (the current through the LED)

- I(1k) = 2.438V
- Id = 2.439mA
- LED is on but dim

Vf (the voltage drop across the LED):

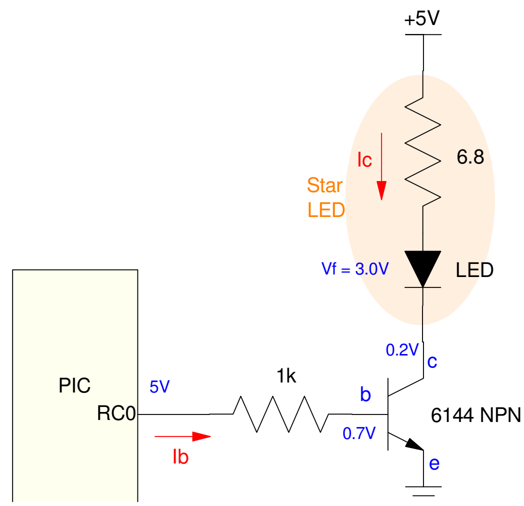
- Measured across the LED
- Vf = 2.623V
- (slightly less than the 3.00V @ 330mA from the data sheets. Vf changes slightly with current)



5) Use a 6144 NPN transistor as an electronic switch to turn on and off the LED (and amplify current).

- Ic(max) = 3A    (Ib(max) = 25mA - the max output of a PIC)
- hfe () > 200

a) Set up the following circuit so that your PIC board can turn on and off the LED




b) Compute

- The currents Ib and Ic and
- The brightness of the LED

$$I_b = \left(\frac{5V - 0.7V}{1k}\right) = 4.3mA$$

$$I_c = \left(\frac{5V - 3.0V - 0.2V}{6.8\Omega}\right) = 264.7mA$$

Is is saturated?

$$\beta I_b > I_c$$

$$200 \cdot 4.3mA > 264.7mA$$

$$860mA > 24.7mA$$

Yes, the transistor is saturated (meaning Vce = 0.2V (ish))

$$L = \left(\frac{264.4mA}{330mA}\right) 2000LM = 160.4 \text{ lumens}$$

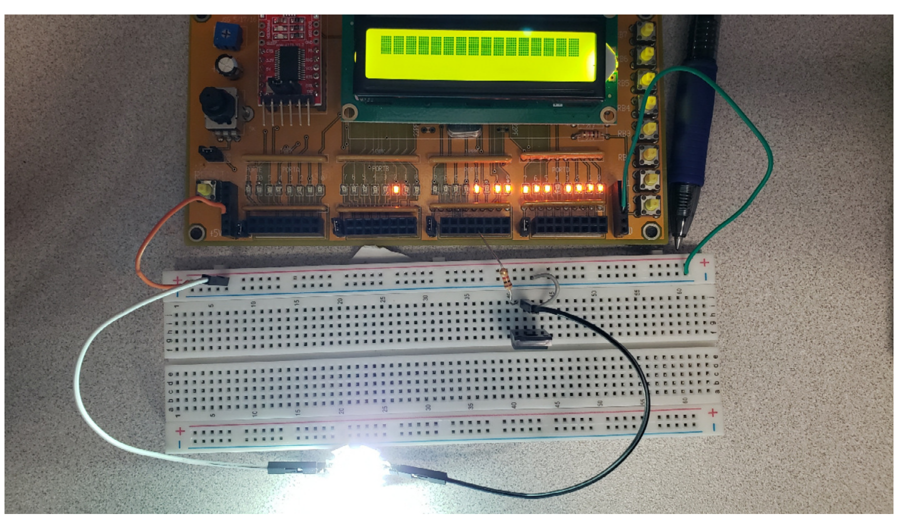
c) Build ths circuit and measure

Id (hint: measure the voltage across the 6.8 Ohm resistor on the LED)

- Vr = 1.566V
- Ir = Vr / 6.8 = 230.3mA  (really bright)

Vf (the voltage drop across the LED)

- 3.138V



**Timing:**

5) 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

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)