ECE 376 - Homework #3

Binary Inputs, Binary Outputs, & LEDs - Due Monday, February 3rd

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 Inputs

Assume a thermistor has a resistance-temperature relationship of

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

1) Design a circuit which outputs

- 0V when T < 35C
- 5V when T > 35C

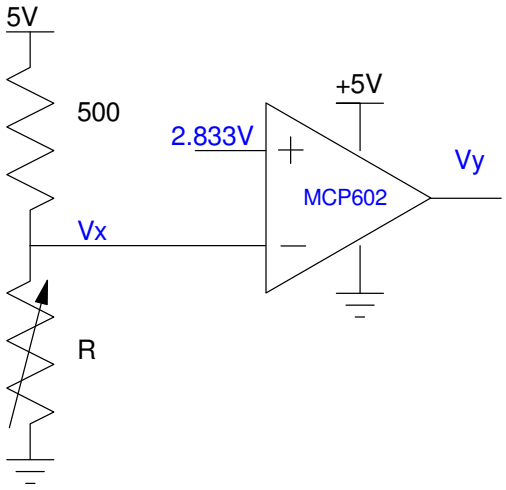
At 35C, R = 653.472 Ohms. Assuming a voltage divider with 500 Ohms

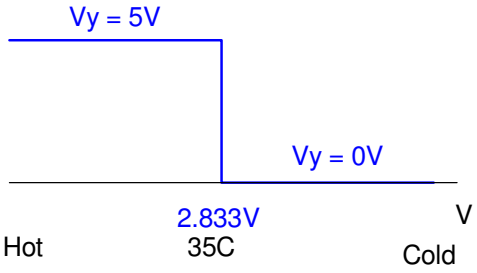
$$V_x = \left(\frac{653.472}{653.472+500}\right)5V = 2.833V$$

When temperature goes up

- Vx goes down
- Vy goes up

Connect to the minus input





2) Design a circuit which outputs

- 0V when T < 35C
- 5V when T > 40C

Use a Schmitt trigger for this circuit. Assume a 500 ohm resistor for the voltage divider

At 35C (Voff)

- R = 653.472 Ohms
- Vx = 2.833V
- Vy = 0V

At 40C (Von)

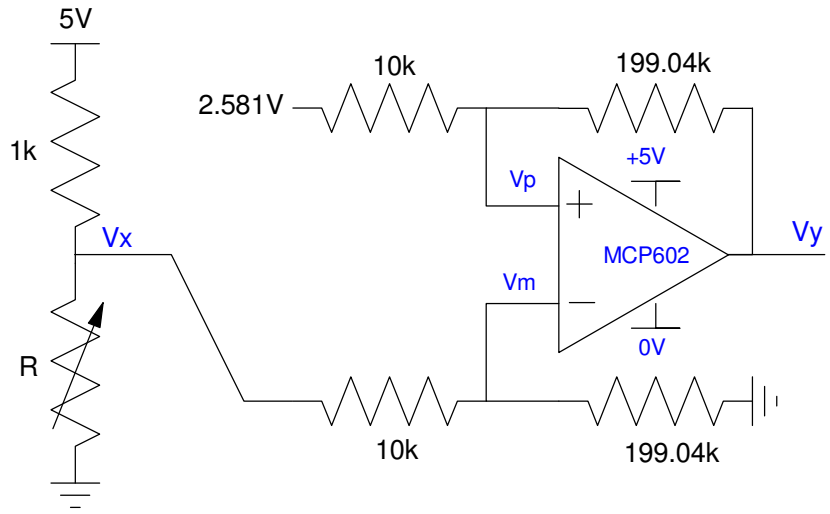
- R = 533.664 Ohms
- Vx = 2.581V
- Vy = 5V

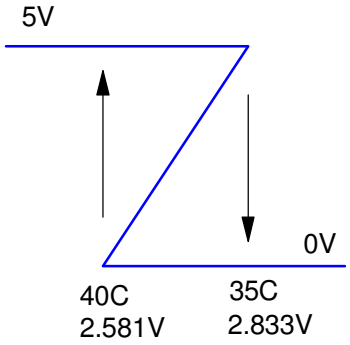
Since V(on) < V(off), connect to the minus input

Set the offset to V(on) = 2.581V

Make the gain

$$gain = \left(\frac{\text{change in output}}{\text{change in input}}\right) = \left(\frac{5V-0V}{2.833V-2.581V}\right) = 19.904$$





Binary Outputs

3) Design a circuit which allows your PIC board to turn on and off an RGB Piranah LED at 0mA (off) and 15mA (on). Assume the specifications for the LEDs are:

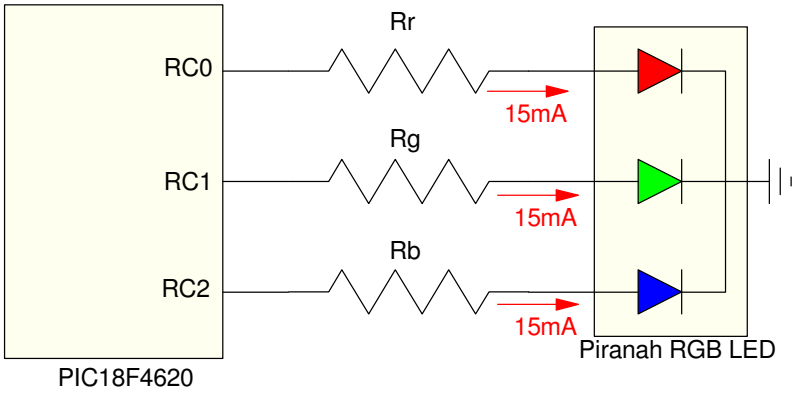
Color	Vf @ 20mA	mcd @ 20mA
red	2.0V	10,000
green	3.2V	10,000
blue	3.2V	10,000

To set the current to 15mA

$$R_r = \left(\frac{5V-2.0V}{15mA}\right) = 200\Omega$$

$$R_g = \left(\frac{5V-3.2V}{15mA}\right) = 120\Omega$$

$$R_b = \left(\frac{5V-3.2V}{15mA}\right) = 120\Omega$$



4) Design a circuit which allows your PIC board to turn on and off a 3W LED at 1000mA. The specs for the LED are:

- Vf = 3.0V @ 1000mA
- 100 Lumens @ 1000mA

Assume you have a 6144 NPN transistor:

- max continuous current = 3A
- current gain = 300
- Vbe = 0.7V, Vce(sat) = 0.2V

In this case, you need a transistor.

Rc: Pick Rc to set the on-current ot 1000mA

$$R_c = \left(\frac{5V-3.0V-0.2V}{1A}\right) = 1.8\Omega$$

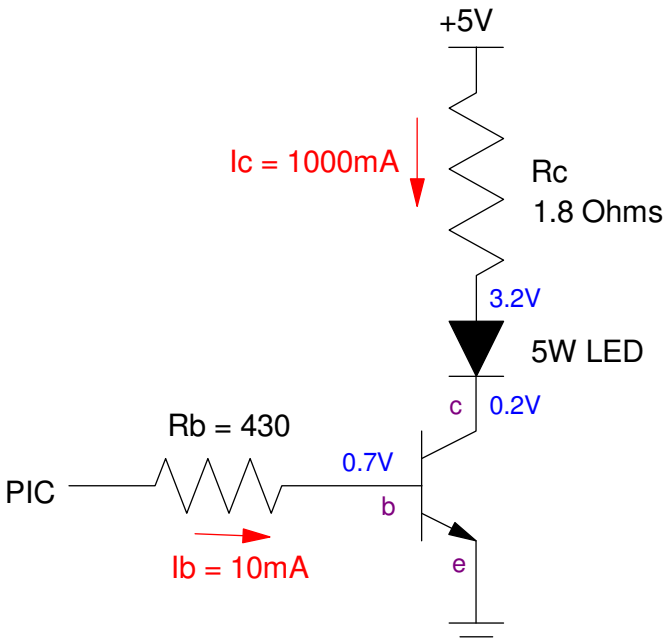
Rb: Pick Rb to saturate the transistor

$$\beta I_b > I_c$$

$$I_b > \frac{I_c}{\beta} = 3.33mA$$

Let Ib = 10mA (limit is 25mA: max a PIC can output)

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



Timing:

6) Write a program which outputs the music note A#2 (116.541 Hz)

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

The duration of the wait loop needs to be 42,903.356 clocks

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 42,903.356$$

One way to do this is to have three nested wait loops:

```
Wait:  movlw     A
; 4 clocks
                 movwf     CNT0
W1:    movlw     B
; 5 clocks * A
                 movwf     CNT1
W2:    nop
; 10 clocks * A * B
                 nop
                 nop
                 nop
                 nop
                 nop
                 nop
                 decfsz CNT1,F
                 goto W2
                 decfsz CNT0,F
                 goto W0
                 return
```

The total time spend in the wait loop is

N = 10AB + 5A + 4

Come up with integers which are in the range of (1..255) and the product is close to 42,903.356. In Matlab, you can find the best combination (not necessary - just showing off)

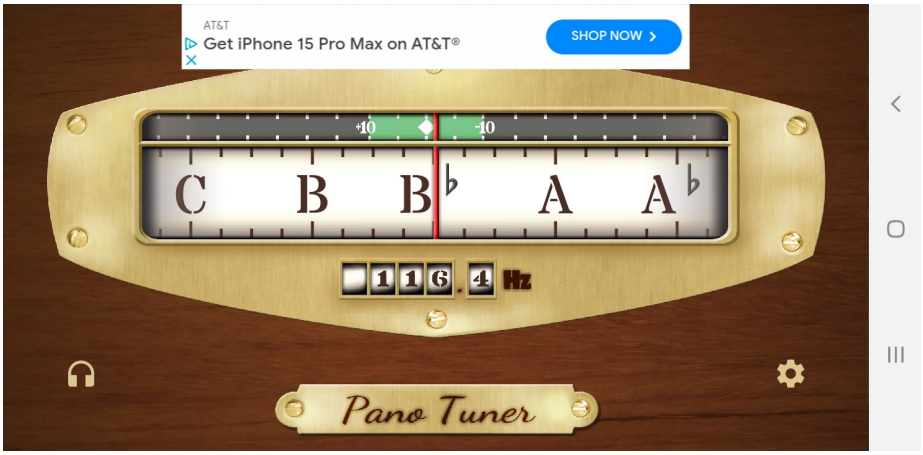
```
% Matlab Code
minE = 9999
for a = 1:255
    for b = 1:255
        N = 10*a*b + 5*a + 4;
        E = abs(42903.356 - N);
        if(E < minE)
            minE = E;
            A = a;
            B = b;
            [A,B,N]
         end
    end
end
```

This results in

- A = 20
- B = 214
- N = 42904

The resulting program is then

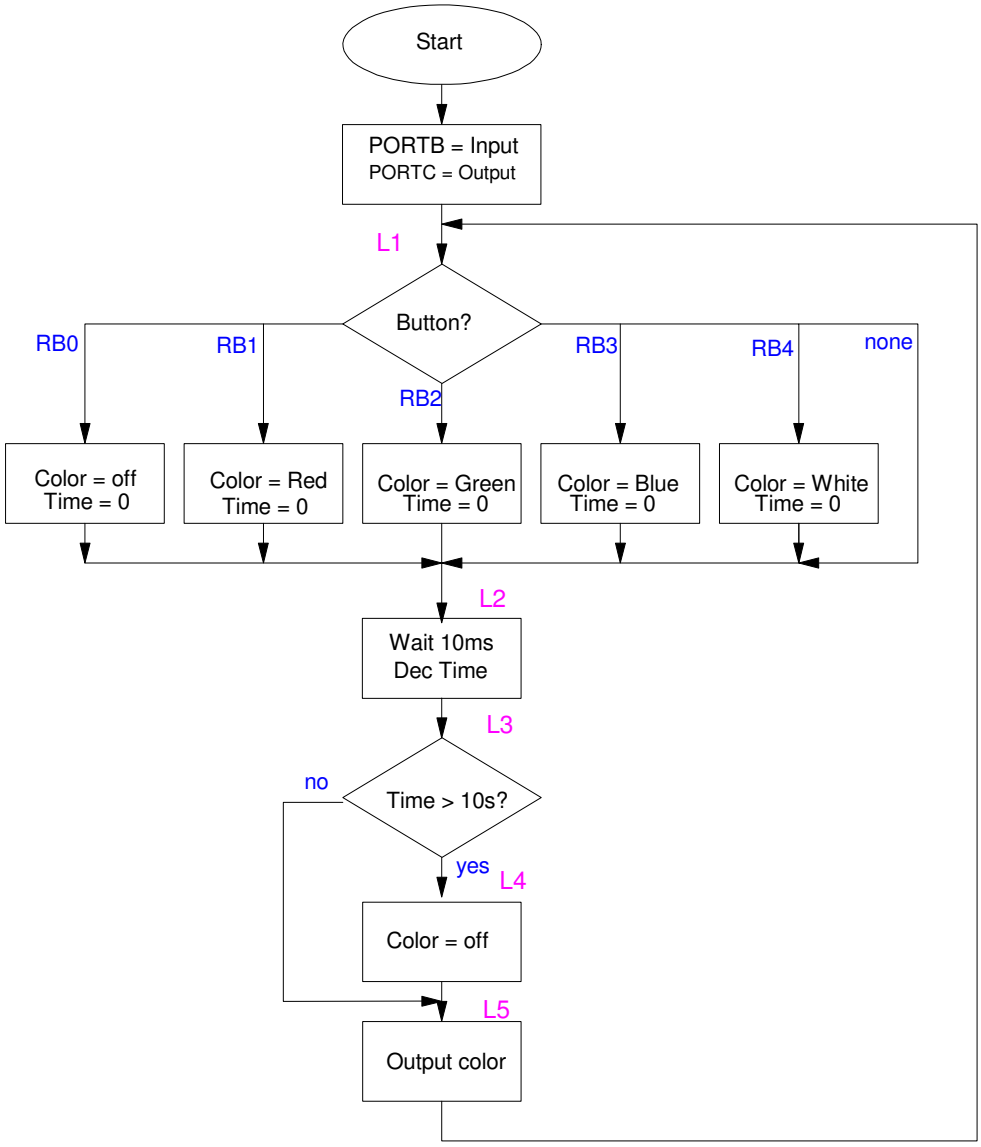
```
#include <p18f4620.inc>

; 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:
       movlw     20
       movwf     CNT0
W1:
       movlw     214
       movwf     CNT1
W2:
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       decfsz CNT1,F
       goto W2
       decfsz CNT0,F
       goto W0
       return

       end
```



Lab: RGB Flashlight

7) Give the flow chart for a program to turn your PIC board into an RGB Flashlight

- On power up, the LEDs are off
- Each button sets the color of the flashlight
  - RB0: Off
  - RB1: Red
  - RB2: Green
  - RB3: Blue
  - RB4: White
- If a button isn't pressed for 10 seconds, the lights turn off



8) Write the corresponding assembler code

```
; --- Flashlight.asm ----

#include <p18f4620.inc>

; Variables
CNT0 EQU 1
CNT1 EQU 2
CNT2 EQU 3
MODE EQU 4
COLOR equ 5
TIME equ 6


; Program
       org 0x800
       call Init
L1:
       btfsc    PORTB,0
       call    Off
       btfsc    PORTB,1
       call    Red
       btfsc    PORTB,2
       call    Green
       btfsc    PORTB,3
       call    Blue
       btfsc   PORTB,4
       call    White

       call    Wait

       dcfsnz  TIME
       clrf    COLOR

       movff   COLOR, PORTC
       movff    TIME, PORTD
       goto    L1

Off:
       clrf     COLOR
       movlw    100
       movwf    TIME
    return
Red:
       movlw    1
       movwf    COLOR
       movlw    100
       movwf    TIME
       return
Green:
       movlw    2
       movwf    COLOR
       movlw    100
       movwf    TIME
       return
Blue:
       movlw    4
       movwf    COLOR
       movlw    100
       movwf    TIME
       return
White:
       movlw    7
       movwf    COLOR
       movlw    100
       movwf    TIME
```

```
        return

Wait:
        movlw   10
        movwf    CNT2
W2:
        movlw    100
        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

        decfsz  CNT2,F
        goto     W2

        return

Init:
        clrf TRISA
        movlw 0xFF
        movwf    TRISB
        clrf TRISC
        clrf TRISD
        clrf TRISE
        movlw 15
        movwf ADCON1
        clrf PORTA
        clrf COLOR
        clrf TIME
        return


        end
```

9) Test your code.

- Compile and program your PIC board
- Verify each button's operation
- Verify the lights turn off after 10 seconds

Comments:

- On power up, all LEDs are off
- Pressing RB1 turns on RC0
- Pressing RB2 turns on RC1
- Pressing RB3 turns on RC2
- Pressing RB3 turns on RC0 / RC1 / RC2
- Pressing RB0 turns off all LEDs
- After 10 seconds, all LEDs turn off automatically as well

10) (20 points) Demonstration

- In-person of with a video