ECE 376 - Homework #3

Binary Inputs, Outputs, and Timing. Due Monday, January 31st

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

Assembler Coding

1) Determine the content of the W register and memory locations A and B after each operation:

Command	W	A	B
; Start	7	8	9
addwf B,F	7	8	16
incf A,W	9	8	16
subwf A,F	9	255	16
sublw 9	0	255	16
movlw 3	3	255	16
andwf A,F	3	3	16
iorwf B,W	19	3	16

Binary Inputs

Assume the resistance - votlage relationship for a thermistor is (T is temperature in Celsius)

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

2) Design a circuit that output

- 0V for temperatures less than 35C
- 5V for temperatures more than 35C

Assume a 1k resistor in a voltage divider

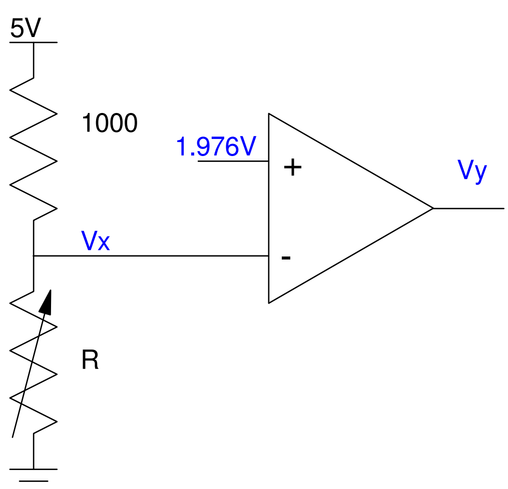
At 35C:

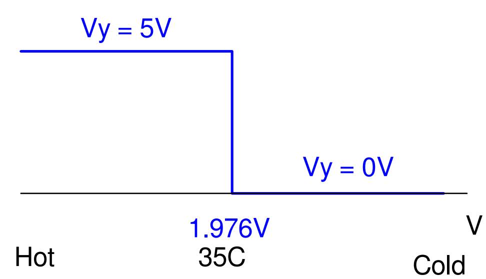
- R = 653 Ohms
- Vx = 1.976V

When the temperature goes up...

- R goes down
- Vx goes down
- Vy goes up to +5V

Connect to the minus input





3) Design a circuit with hysteresis that outputs

- 0V when the temperature is less than 35C
- 5V when the temperature is more than 40C
- No change (0V or 5V) for temperatures inbetween 35C and 40C

At 35C:

- R = 653 Ohms
- Vx = 1.976V
- Vy = 0V

At 40C

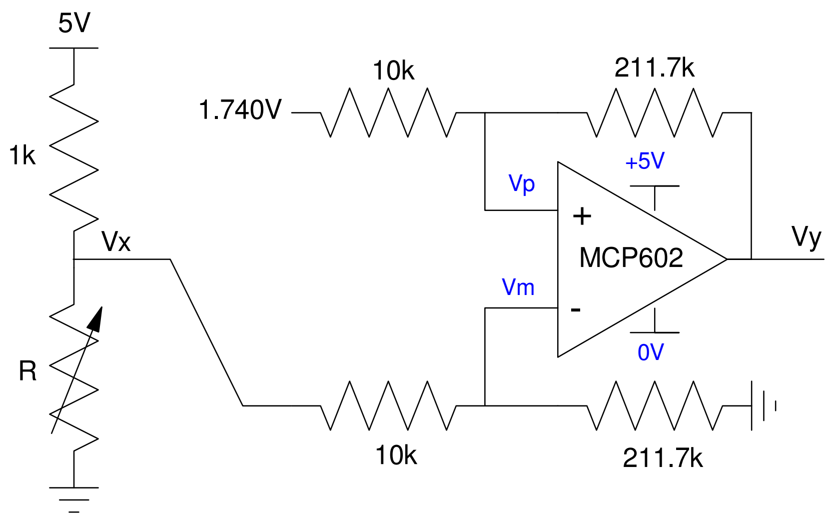
- R = 533.66 Ohms
- Vx = 1.740V
- Vy = 5V

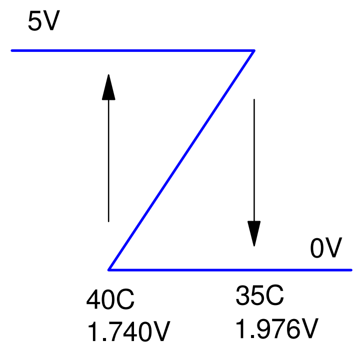
As Vx goes down, Vy goes up. Connect to the minus input.

Vy becomes 5V when Vx is 1.740V. Make the offset 1.740V

The gain needed is

$$gain = \left(\frac{\text{change in Vy}}{\text{change in Vx}}\right) = \left(\frac{5V-0V}{1.976V-1.740V}\right) = 21.17$$





Binary Outputs

4) 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

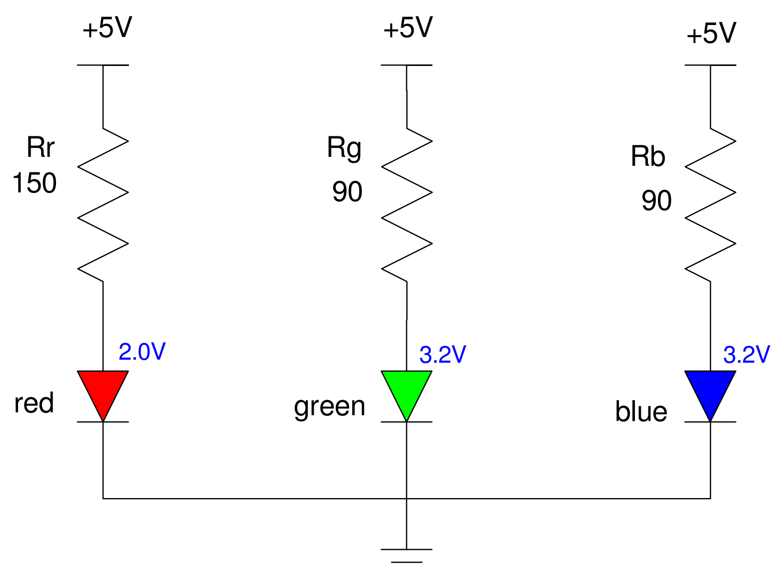
- Less than 5V and
- Less than 25mA

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

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

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

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



5) Design a circuit which allows your PIC board to turn on and off a 1W LED. The specs for the LED are:

- Vf = 3.2 - 3.6V
- Current = 350mA
- 100 Lumens (equivalent to a 10W 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.

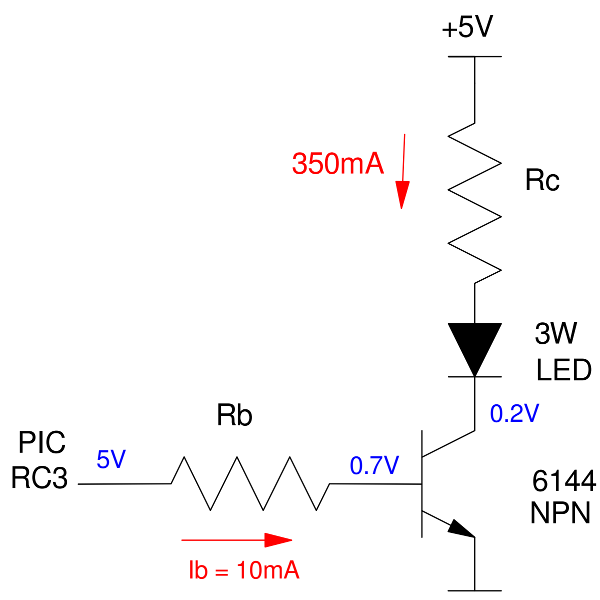
- $\beta = 100$ *worst case*
- $V_{ce}(sat) = 0.2V$

$$R_c = \left(\frac{5V - 3.4V}{350mA}\right) = 4.57\Omega$$

$$I_b > \frac{I_c}{\beta} = \frac{350mA}{100} = 3.5mA$$

Let Ib = 10mA

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



Timing:

6) Write a program which outputs the music note D3# (155.56 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

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 32,141.939$$

Come up with a wait loop that burns 32,141 clocks:

N = 10*A*B + 5*A + 9 = 32,141

A = 13, B = 247 results in N = 32,184 (off by 0.131%)

```
#include <p18f4620.inc>

; Variables
CNT0 EQU 1
CNT1 EQU 2

; Program
   org 0x800
   call Init
Loop:
   incf  PORTC,F
   call Wait       ; Play note D#2
   goto Loop

; --- Subroutines ---

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

; Wait 32,141 clocks  (actual wait = 32,184)
Wait:
   movlw 13        ; A
   movwf CNT1
W1:
      movlw 247    ; 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:

7) Requirements:

- Inputs: Buttons on RB0 / RB1 / RB2 / RB3
- Outputs: RC0
- Relationship: Output a square wave on RC0 based upon the button pressed:
  - RB0: 261 Hz (C4)
  - RB1: 293 Hz (D4)
  - RB2: 329 Hz (E4)
  - RB3: 349 Hz (F4)

8) 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 = 100AB + 5B + 5

261Hz:

- N = 19,157
- A = 100, B = 19

293 Hz:

- N = 17,026
- A = 100, B = 17

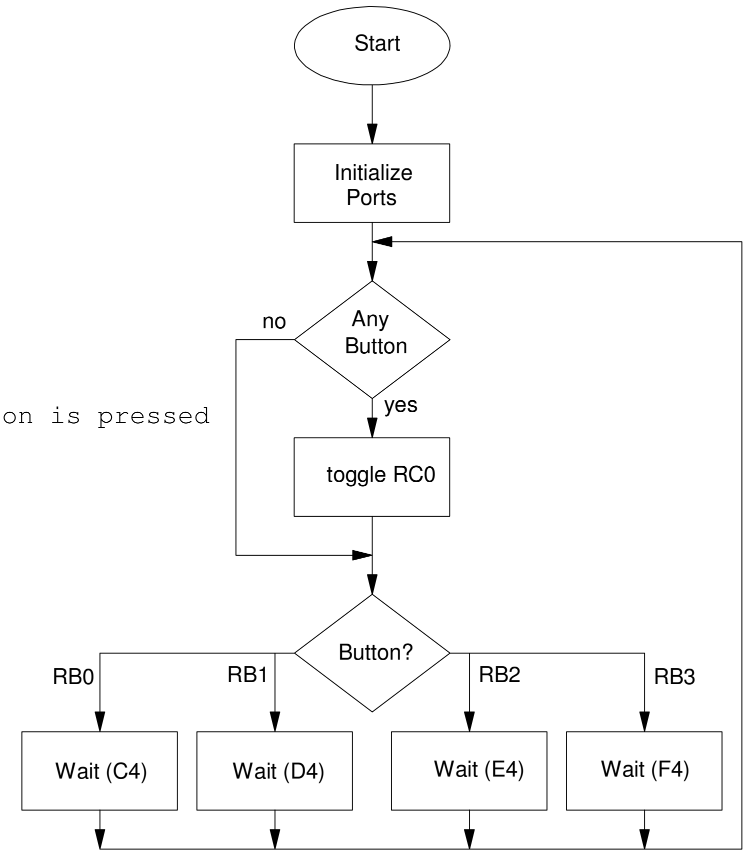
329 Hz

- N = 15,167
- A = 100, B = 15

349 Hz

- N = 14,317
- A = 100, B = 14

```
; --- Piano4.asm ----
; When you press button RB0..RB3, you play a note
; on RC0:
;   RB0:  291 Hz (C4)
;   RB1:  293 Hz (D4)
;   RB2:  329 Hz (E4)
;   RB3:  349 Hz (F4)

#include <p18f4620.inc>

; Variables
CNT0 EQU 1
CNT1 EQU 2

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

   btfsc PORTB,0
    call  C4
   btfsc PORTB,1
    call  D4
   btfsc PORTB,2
   call  E4
   btfsc PORTB,3
   call  F4

   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 15
   movwf ADCON1       ;everyone is binary
   return

C4:   ; 261Hz = 19,157 clocks
   movlw 19
   movwf CNT1
C4a:
   movlw 100
   movwf CNT0
C4b:
   nop
   nop
   nop
   nop
   nop
   nop
   nop
   decfsz CNT0, F
   goto C4b
   decfsz CNT1, F
   goto C4a
   return
```



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?

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