

ECE 376 - Homework #3

Binary Outputs, Timing, LEDs

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:

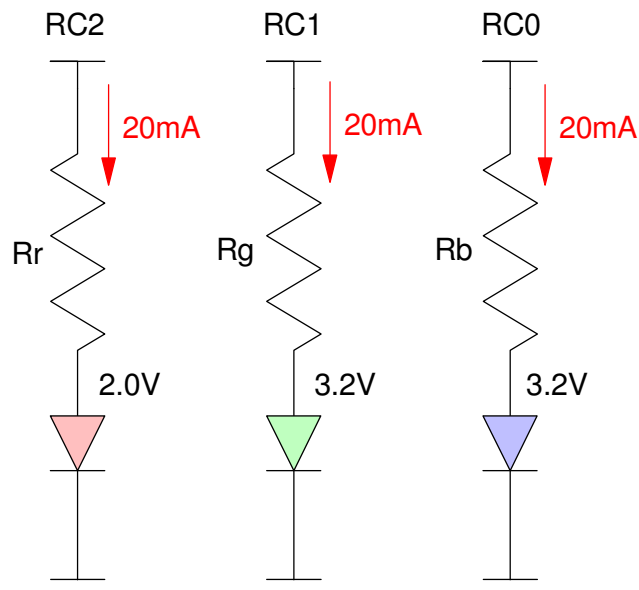
Color	Vf @ 20mA	mcd @ 20mA
red	2.0V	10,000
green	3.2V	10,000
blue	3.2V	10,000

Calculations

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

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

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



1W Star LED Flashlight

2) Write an assembler program which

- Turns on RC0 when button RB1 is pressed, and
- Turns off RC0 when button RB0 is pressed

Turn in

- The assembler code, and
- The compiled code size

Code:

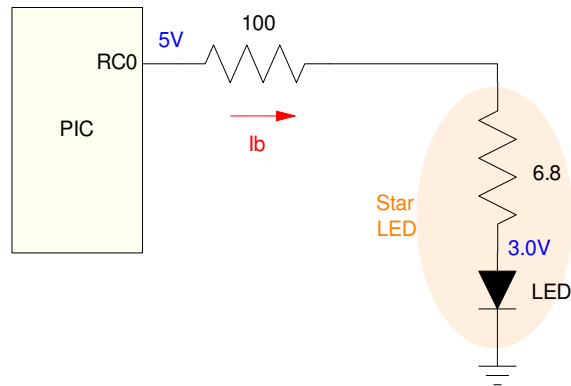
```
; --- Flashlight.asm ----  
  
#include <p18f4620.inc>  
  
; Program  
    org 0x800  
    movlw    0xFF  
    movwf    TRISB  
    clrf     TRISC  
    movlw    0x0F  
    movwf    ADCON1  
Loop:  
    movf     PORTC,W  
    btfsc    PORTB,0  
    movlw    0  
    btfsc    PORTB,1  
    movlw    0xFF  
    movwf    PORTC  
    goto     Loop  
  
end
```



Compilation Results

```
MPLINK 4.49, LinkerDevice Database Version 1.14  
Copyright (c) 1998-2011 Microchip Technology Inc.  
Errors      : 0Loaded C:\ECE376\HW3\OnOff.cof.  
-----  
Debug build of project `C:\ECE376\HW3\HW3.mcp' succeeded.  
Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49  
Preprocessor symbol `__DEBUG' is defined.  
Thu Sep 03 14:12:30 2026  
-----  
BUILD SUCCEEDED
```

- 3) The star LED in your lab kit is a 1W white LED in series with 6.8Ω .
- $V_f = 3.0V$ @ $330mA$
 - $200LM$ @ $330mA$
- a) Set up the following circuit so your PIC board can turn the LED on/off
- The 100Ω resistor limits the current
 - (the PIC can only output $25mA$)



- b) Compute the
- Current to the LED
 - The brightness of the LED

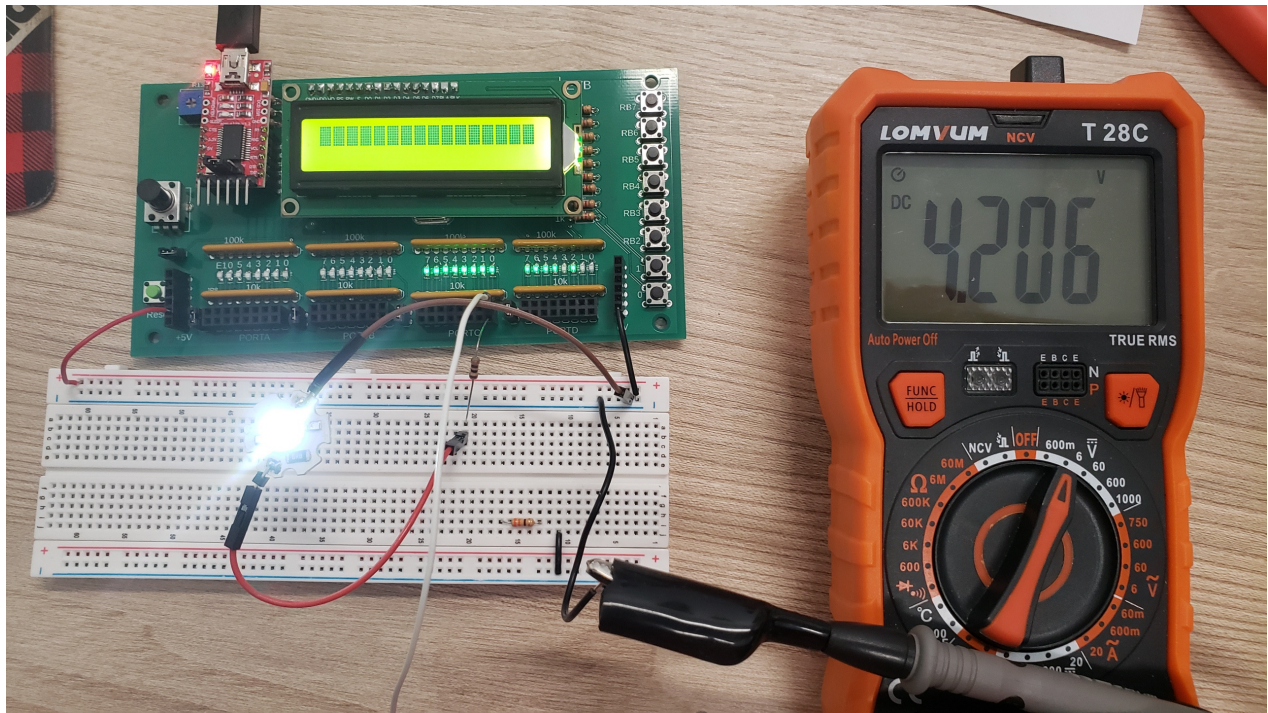
$$I_d = \left(\frac{5V - 3.0V}{106.8\Omega} \right) = 18.7mA$$

$$Light = \left(\frac{18.7A}{330mA} \right) 200LM = 11.35LM$$

c) Build this circuit and measure

- I_d (hint: measure V_r and compute the current)
- V_f (the voltage drop across the LED)

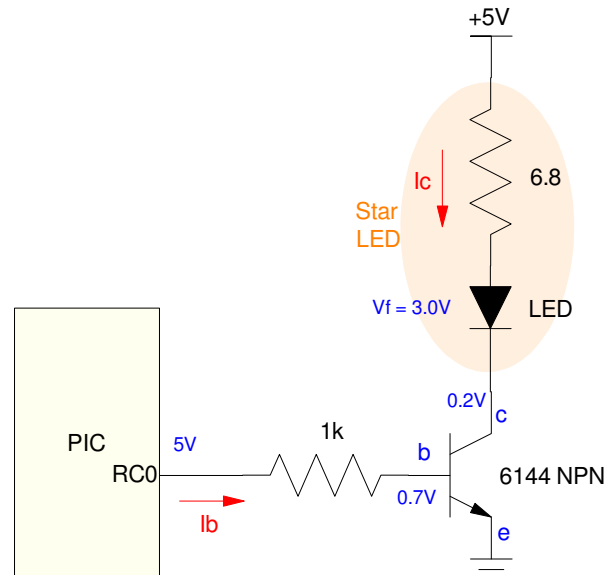
	V(RC0)	V(100)	V(LED)	I(LED)
Theory	5.00V	2.00V	3.00V	20.0mA
Measured	4.206V	1.405V	2.706V	14.05mA



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

- $I_c(\text{max}) = 3\text{A}$ ($I_b(\text{max}) = 25\text{mA}$ - the max output of a PIC)
- $h_{fe} (\beta) > 200$

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



b) Compute

- The currents I_b and I_c and
- The brightness of the LED

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

$$I_c = \left(\frac{5V - 0.2V - 3V}{6.8\Omega} \right) = 265mA$$

$$Light = \left(\frac{265mA}{330mA} \right) 100M = 80.1LM$$

c) Measure

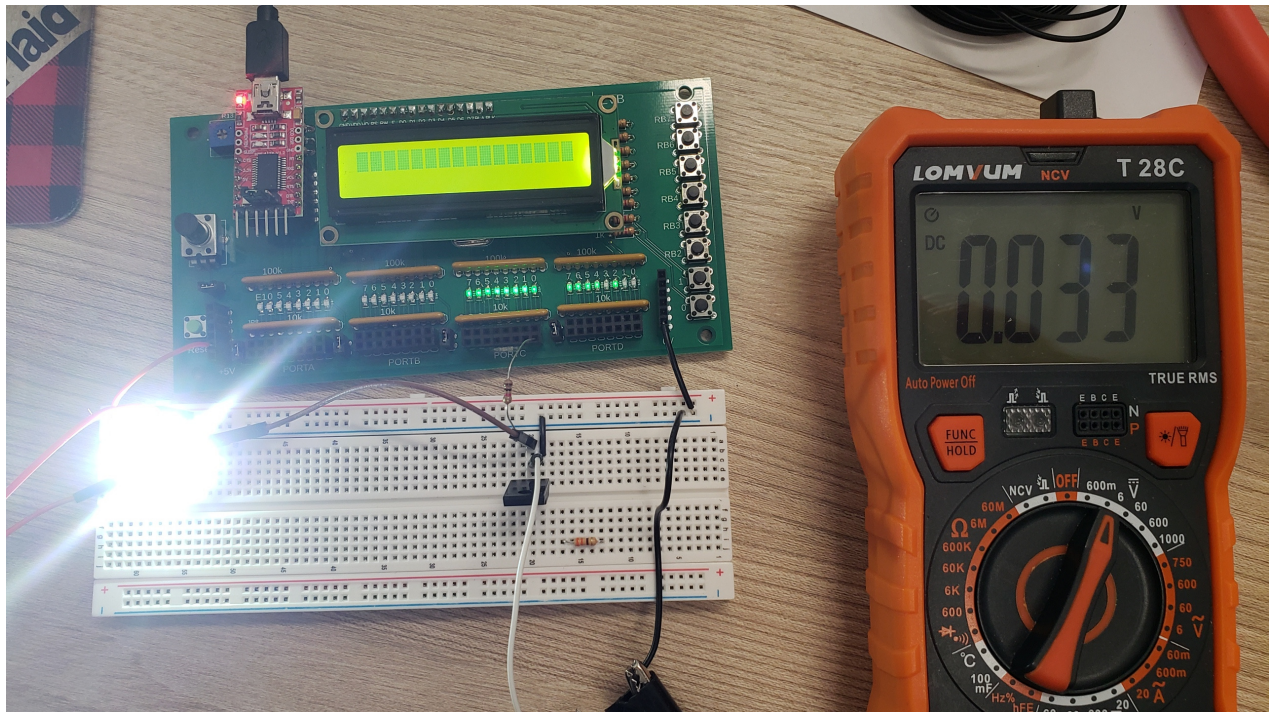
- The currents I_b and I_c
- The voltages V_b , V_c , and $V(\text{LED})$

	$V(RC0)$	V_b	V_c	$V(\text{LED})$	$V(6.8\ \Omega)$
Theory	5.00V	0.7V	0.2V	3V	1.8V
Measured	4.517V	0.696V	33mV	3.02V	1.658V

	I_b	I_c		
Theory	4.3mA	265mA		
Calculated	3.821mA	243.8mA		

$$I_b = \left(\frac{4.517V - 696mV}{1k} \right) = 3.821mA$$

$$I_c = \left(\frac{1.658V}{6.8\Omega} \right) = 243.8mA$$



Timing:

- 5) Write a program which outputs the music note F3 (174.61 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 between edges should be

$$N = \left(\frac{10,000,000}{2 \cdot Hz} \right) = 28,635.2443$$

For the following code, the wait routine takes

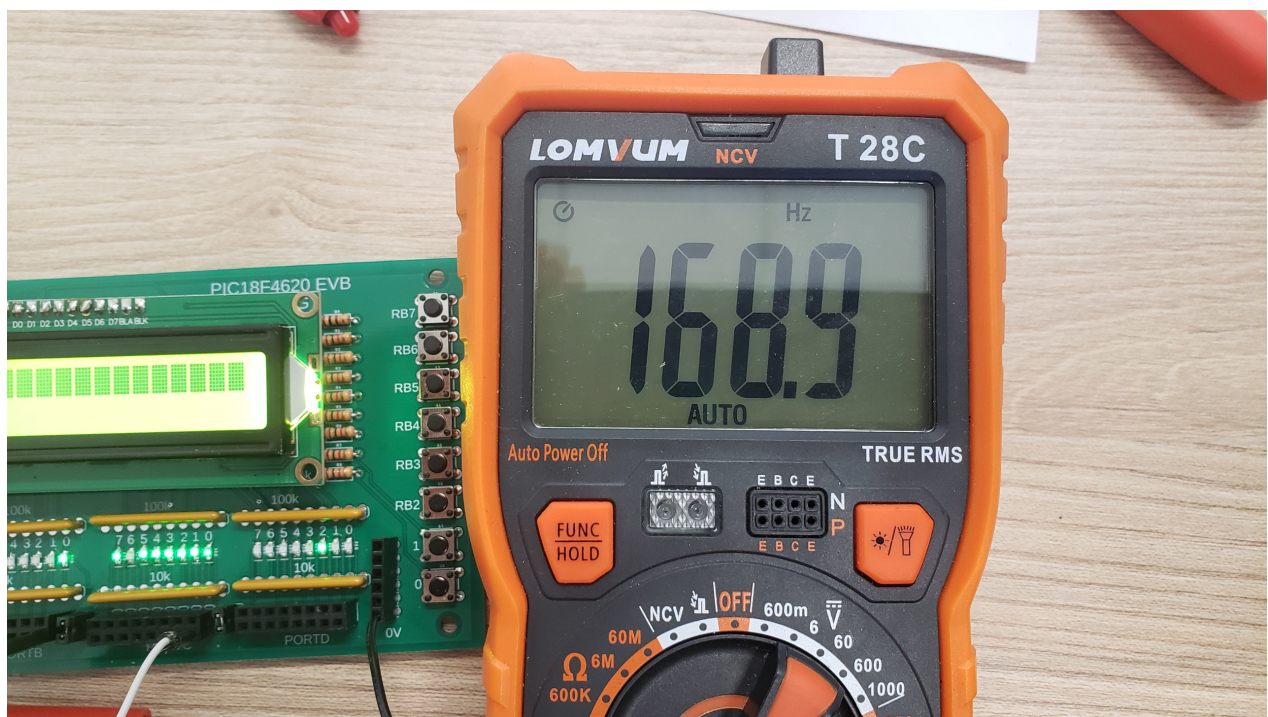
$$N = 10AB + 5A + 4$$

clocks. Playing with {A, B, C} subject to them being integers in the range of [1, 255] gives

- A = 204
- B = 14
- N = 28634

Result:

- F = 168.9Hz



Code:

```
#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 204      ; A
    movwf CNT0
W1:
    movlw 14       ; B
    movwf CNT1
W2:
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    decfsz CNT1,F
    goto W2
    decfsz CNT0,F
    goto W1
    return

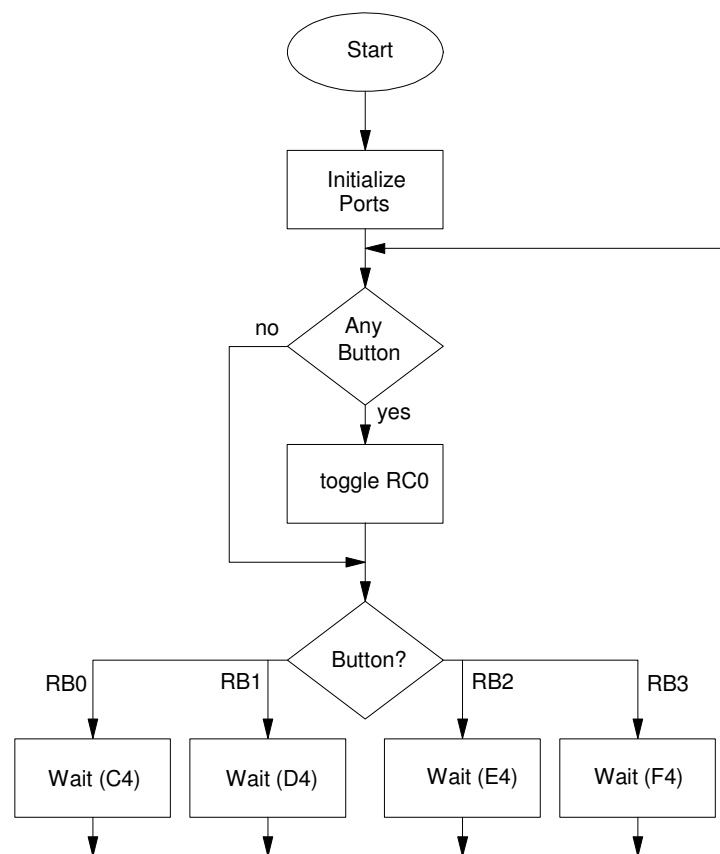
end
```

4-Key Piano:

Design an embedded system to play four notes:

- Inputs: Buttons on RB0 / RB1 / RB2 / RB3
- Outputs: RC0
- Relationship: Output a square wave on RC0 based upon the button pressed:
 - RB0 F3 174.61 Hz
 - RB1 G3 196.00 Hz
 - RB2 A3 220.00 Hz
 - RB3 B3 246.94 Hz

6) Give a flow chart for your program



7) Give an assembler program for the 4-key piano.

- Include computations for the wait loops

- RB0 F3 174.61 Hz
- RB1 G3 196.00 Hz
- RB2 A3 220.00 Hz
- RB3 B3 246.94 Hz

Freq	174.61 Hz	196 Hz	220 Hz	246.94 Hz
N (ideal)	28,635	25,510	22,727	20,248
A	204	12	22	11
B	14	204	101	176
N	28,634	25504	22,729	20,244
Measured (problem 8)				

Code:

```
; --- Piano4.asm ----

#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   F3
    btfsc PORTB,1
    call   G3
    btfsc PORTB,2
    call   A3
    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
```

```

        clrfs TRISD           ;PORTD is output
        clrfs TRISE          ;PORTE is output
        movlw 0x0F
        movwf ADCON1         ;everyone is binary
        return

F3:
        movlw 204
        movwf CNT1
L1a:
        movlw 14
        movwf CNT0
L1b:
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        decfsz CNT0, F
        goto L1b
        decfsz CNT1, F
        goto L1a
        return

G3:
        movlw 12
        movwf CNT1
L2a:
        movlw 204
        movwf CNT0
L2b:
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        decfsz CNT0, F
        goto L2b
        decfsz CNT1, F
        goto L2a
        return

A3:
        movlw 22
        movwf CNT1
L3a:
        movlw 101
        movwf CNT0
L3b:
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        decfsz CNT0, F
        goto L3b
        decfsz CNT1, F
        goto L3a
        return

```

```
B3:
    movlw 11
    movwf CNT1
L4a:
    movlw 176
    movwf CNT0
L4b:
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    decfsz CNT0, F
    goto L4b
    decfsz CNT1, F
    goto L4a
    return

end
```

8) Validation: Collect data in the lab to verify your code works.

- Are the frequencies correct?
 - Within 4%

Freq	174.61 Hz	196 Hz	220 Hz	246.94 Hz
N (ideal)	28,635	25,510	22,727	20,248
A	204	12	22	11
B	14	204	101	176
N	28,634	25,504	22,729	20,244
Measured (problem 8)				
% Error	-3.27%	+3.88%	+1.72%	+4.19%

9) Demonstration: Demonstrate that your 4-key piano works

- either in person or with a video
- corresponding dance routine optional

