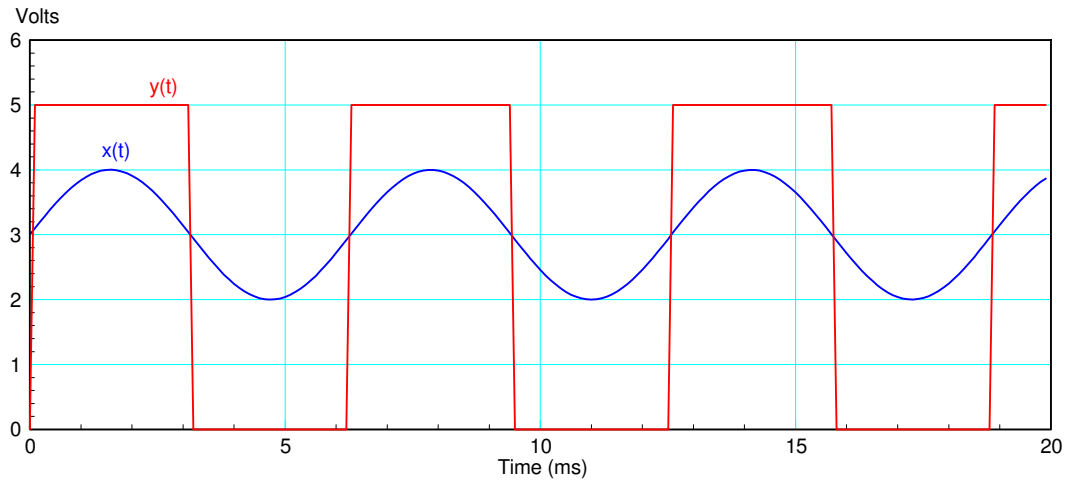# ECE 376 - Final Exam: Name _____

Open book. open notes

1) Hardware: Digital (0V/5V) signals are easy for a microcontroller to work with. Analog signals are more difficult. Design a circuit which converts a sine wave ranging from 2.0V to 4.0V into a TTL 0V/5V square wave at the same frequency as shown.

- Input: Analog sine wave going from 2.0V to 4.0V.
- Output: 0V/5V square wave

2) Hardware: Binary Outputs. A 5W yellow LED is to be used to create an electronic candle. Design a circuit which allows a PIC to turn on and off the LED
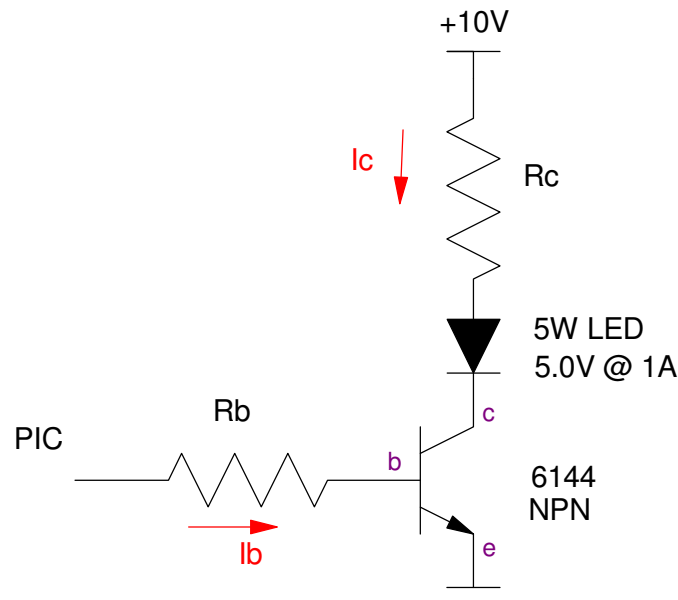
- Off: Id = 0mA
- On: Id = XmA where X is your birth date

Assume

- The LED has Vf = 5.0V @ 1A, Output = 5000 Lumens @ 1A
- A 6144 NPN transistor with β=300, Vce(sat) = 0.2V
- X = 900 + 100*(your birth month) + (your birth date). May 14th would give X = 1414mA.

For your circuit, determine the following:

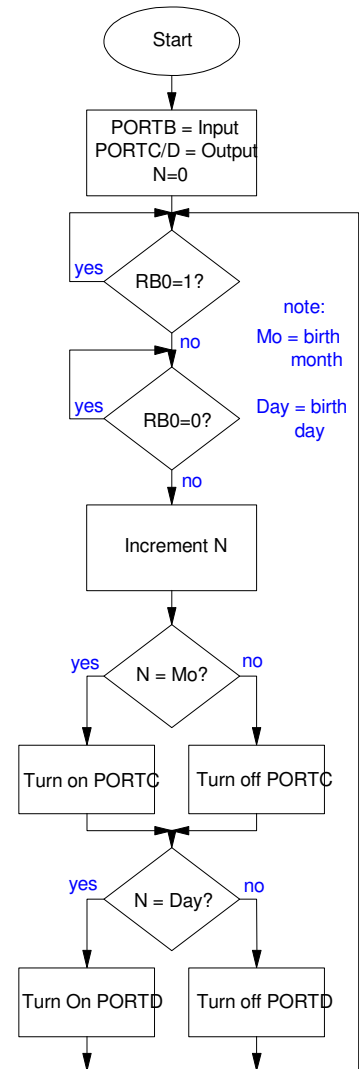| X (mA)	Lumens	Rc	Rb	Ib
900 + 100*mo + day	when LED is on			when the PIC outputs 5V

3) C Coding. The following flow chart counts the number of times button RB0 is pressed:

- N = the number of button presses
- When N matches your birth month (1..12), the lights on PORTC turn on
- When N matches your birth day (1..31), the lights on PORTD turn on

Write the corresponding C code

Start

PORTB = Input
PORTC/D = Output
N=0

RB0=1?   yes

no

note:
Mo = birth month

RB0=0?   yes

no

Day = birth day

Increment N

N = Mo?   yes   no

Turn on PORTC   Turn off PORTC

N = Day?   yes   no

Turn On PORTD   Turn off PORTD

4) Subroutines & C Coding: The signals a PIC sends to a sensor are as follows. Write a C program which outputs the data 001x to the sensor where x is either 1 or 0 depending upon what is passed to the subroutine. Assume RC0 / RC1 / RC2 are all output pins

RC0 (CS)

RC1 (CLK)

RC2 (DATA)   1   1   0   x

Subroutine Called   time   End of Subroutine
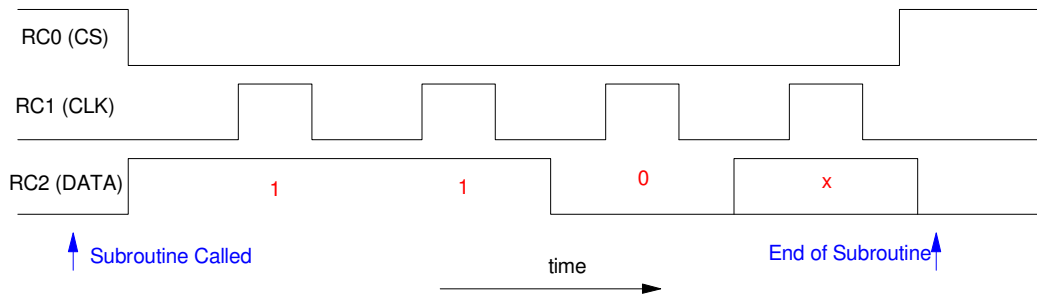
```
void Sensor(unsigned char x) {
```

5) Interrupts: Using interrupts

- Count how many times you press a button (detect rising edge on A)
- Clear the count when you press B
- When the count is equal to your birth month, play 100Hz for one second on RC0
- When the count is equal to yor birth date, play 200Hz for two seconds on RC1

Specify the interrupts used and how they're set up

Inputs		Outputs	
A count	B clear count	100Hz for 1 second	200Hz for 2 seconds

Write the corresponding interrupt service routines

Inputs		Outputs	
A count on A	B clear count on B	100Hz for 1 second when count equals birth month	200Hz for 2 seconds when count equals birth day

6) Interrupt Stoplight: Write the interrupt routines to drive a stoplight with four different modes:
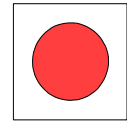
Input: Push Button

- Each time you press, you switch from mode: $0 \to 1 \to 2 \to 3 \to$ *repeat*
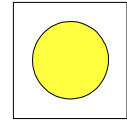
Output: LEDs

- Red = RD2
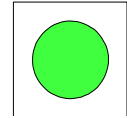- Yellow = RD1
- Green = RD0

Mode:

- 0: Normal Mode. Cycle from green (5s) to yellow (1s) to red (6s) and repeat
- 1: Stop: Solid red (green and yellow off0
- 2: Flashing Red: Turn the red light on and off every 1 second. (Y/G off)
- 3: Flashing Yellow: Turn the yellow light on and off every 1 second. (R/G off)

Specify the interrupts used (one or more)

Interrupt #1 Interrupt name, set-up (PS, edge, etc)	Interrupt #2 (optional) Interrupt name, set-up (PS, edge, etc)	Interrupt #3 (optional) Interrupt name, set-up (PS, edge, etc)

Specify the interrupt service routine(s). The main routine does nothing.

Main Routine	Interrupt #1	Interrupt #2 (optional)	Interrupt #3 (optional)
`while(1) {` `    }`			