ECE 376 - Final Exam: Name _____

Open-Book, Open Note, Calculators and Matlab permitted. Individual Effort.

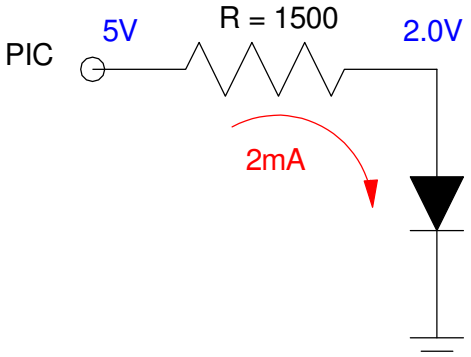
1) Binary Outputs: Assume a 6411 NPN transistor (if needed)

- Vbe = 0.7V
- Vce(sat) = 0.2V
- β = 300
- max(Ic) = 6A

1a) Give a circuit which allows a PIC to turn on and off a 40mW LED at 2mA

- Id = 20mA
- Vd = 2.0V

$$R = \left(\frac{5.0V - 2.0V}{2mA}\right) = 1500\Omega$$



1b) Give a circuit which allows a PIC to turn on and off 12V DC motor, which draws up to 3A @ 12V

- V(motor) = 12V
- Current draw < 3A

For a motor, you don't need (or want) to add a resistor:
the motor limit the current by itself to <3A.

Pick Rb to allow 3A to flow

$$\beta I_b > 3A$$

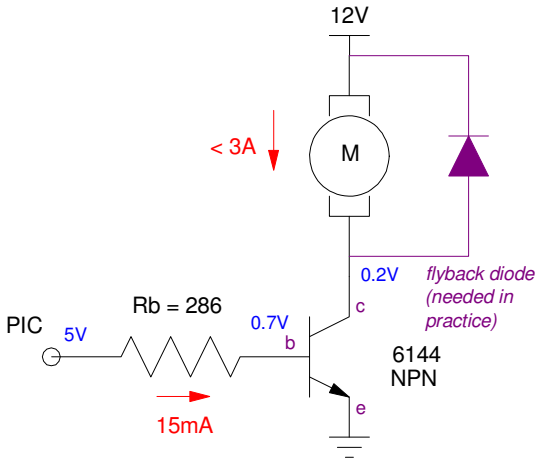
$$I_b > \left(\frac{3A}{300}\right) = 10mA$$

Let Ib = 15mA (max a PIC can output is 25mA)

$$R_b = \left(\frac{5V - 0.7V}{15mA}\right) = 286\Omega$$

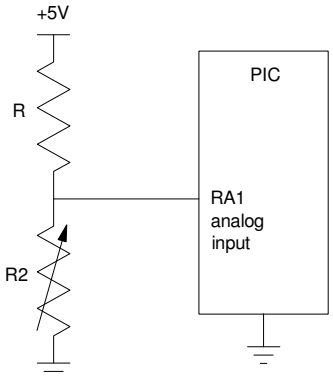
Any current in the range of 10mA .. 25mA works

Any resistance in the range of 430 .. 172 Ohms works



2) Analog Inputs: A CdS light sensor has the following resistance - lux (light intensity) relationship

$$R_2 = 1000 \cdot \left(\frac{10}{Lux}\right)^{0.6} \text{ Ohms}$$



If the A/D reading is 417, determine

- The voltage,
- The resistance,
- The light level in Lux, and
- The resolution (the smallest change in Lux you can detect)

with the following circuit. Assume

- R = 900 + 100*(your birth month) + (your birth date)

R 900 + 100*mo + day	A/D reading 0..1023	Voltage A/D = 712	R2 A/D = 712	Lux A/D = 712	Resolution smallest change in Lux you can measure
1,414	**712**	3.480 V	3237.196 Ohms	7.0841 Lux	0.0216 Lux

$$V = \left(\frac{712}{1023}\right)5V = 3.480V$$

$$R_2 = \left(\frac{V}{5-V}\right)1414 = 3237.196\Omega$$

$$Lux = 7.0841$$

If the A/D = 713 (smallest change you can measure)

$$V = \left(\frac{713}{1023}\right)5V = 3.4848V$$

$$R_2 = \left(\frac{V}{5-V}\right)11414 = 3252.2\Omega$$

$$Lux = 7.1942$$

The differenice is the resolution

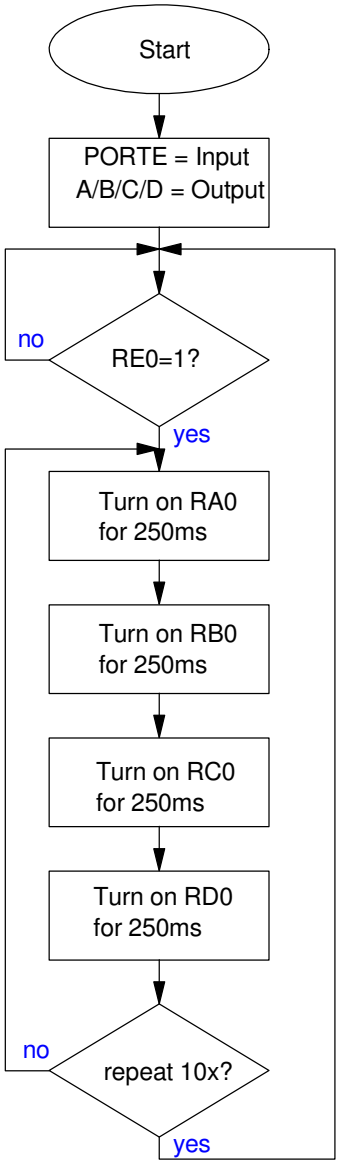
$$\delta Lux = 0.0216$$

3) C-Coding **without** interrupts: Write a C program for driving Christmas tree lights. Assume each output pin is connected to an LED. When pin RE0 goes high, the LEDs are to be turned on one at a time in the following sequence

- RA0 - RB0 - RC0 - RD0 - repeat ten times

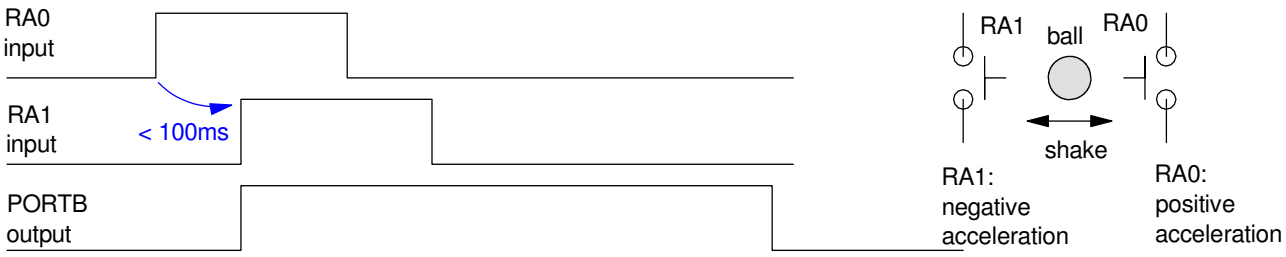
with a 250ms delay between each light. Write the corresponding C code

```
void main(void) {
   ADCON1 = 0x0F;

   TRISE = 0xFF;
   TRISA = 0;
   TRISB = 0;
   TRISC = 0;
   TRISD = 0;

   while(1) {

      while(!RE0);

      for(i=0; i<10; i++) {

         RA0 = 1;
         Wait_ms(250);

         RA0 = 0;
         RB0 = 1;
         Wait_ms(250);

         RBV0 = 0;
         RC0 = 1;
         Wait_ms(250);

         RC0 = 0;
         RD0 = 1;
         Wait_ms(250);

         RD0 = 0;
         }

      }
```



4) C Coding without interrupts: Christmas Present: Write a C program that turns on lights if the PIC is shaken back and forth in less than 100ms.

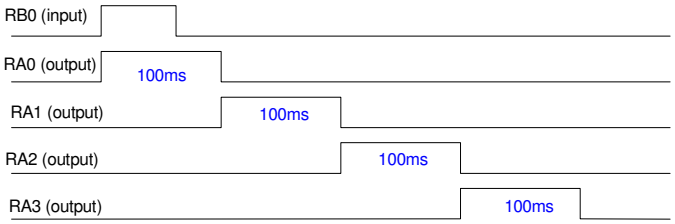
- RA0: 5V if the PIC board experiences positive acceleration
- RA1: 5V if the PIC board experiences negative acceneration
- PORTB: Connected to eight LEDs




```
void main(void) {
   unsigned int TIME;
   ADCON1 = 0x0F;

   TRISA = 0xFF;
   TRISB = 0;

   while(1) {

      if(TIME) TIME = TIME - 1;   // decrement 100ms timer to zero, stop at zero

      if(TIME == 0) {             // start a 100ms timer when RA0 goes high
         if(RA0) TIME = 100;
         }

      if(TIME > 0) {
         if(RA1) {                // if - shake within 100ms or a + shake
            PORTB = 0xFF;         // turn on PORTB for 1000ms
            Wait_ms(1000);
            PORTB = 0;
            TIME = 0;
            }
         }

       Wait_ms(1);                // set the loop time to 1ms
      }
   }
```

5) C-Coding **with** interrupts: Write a C which uses interrupts to turn on the lighs for a Christmas tree:

- When pin RB0 goes high (INT0 interrupt)
- Pins RA0, RA1, RA2, then RA3 go high sequentially
- Each pin goes high for 100ms (controlled by Timer0 interrupt)
- Once RA3 turns off, the process stops until the next INT0 interrupt



INT0 initialization rising or falling edge	Timer0 Initialization Pre-scalar = ?
rising	PS = 256

INT0 Interrupt Service Routine
start the light show when RB0 goes high

```
if(INT0IF) {

// reset the counter
   N = 0;

// turn on RA0
   PORTA = 1;

// for 100ms
   TMR0 = -3906;

   INT0IF = 0;
   }
```

On a rising edge of RB0, reset a counter (N) to zero, set up a Timer0 interrupt in 100ms, and set RA0

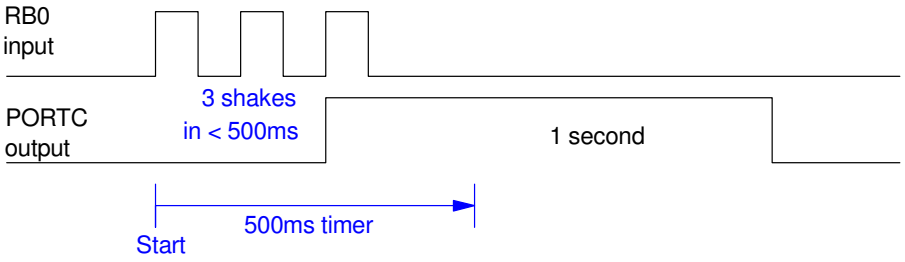
Timer0 Interrupt Service Routine
Trigger every 100ms
Turn on RA0 then RA1 then RA2 then RA3
three times then stop (until next INT0 iinterrupt)

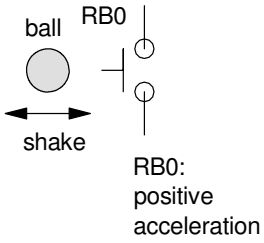
```
if(TMR0IF) {

// every 100ms
   TMR0 = -3906;

// count up to five, stop at 5
   if(N < 5) N = N + 1;

// at 100ms, set RA1
   if(N == 1) PORTA = 2;

// at 200ms, set RA2
   if(N == 2) PORTA = 4;

// at 300ms, set RA3
   if(N == 3) PORTA = 8;

// past 300ms, turn off all lights
   if(N > 3) PORTA = 0;

   TMR0IF = 0;
   }
```

Every 100ms, increment a counter and turn on the next light. After four interrupts, turn off PORTA

6) C-Coding **with** interrupts. Write a C program which uses interrupts to detect if a Christmas present is being shaken.

- If RB0 goes high three times in less than 500ms
- PORTC goes high (0xFF) for one second





INT0 rising / falling edge	Timer2 N for 1ms	Timer2 A	Timer2 B	Timer2 C
rising	10,000	10	250	4

Main Loop if needed	INT0 Interrupt Count edges On 1st edge, start 500ms counter If 3 edges in < 500ms, turn on PORTC for 1sec	Timer2 every 1ms

`while(1) {`	`if(INT0IF) {`
	
	`// if first shake, start 500ms timer`
	`if(TIME == 0) {`
	`N = 1;`
	`TIME = 500;`
	`}`
	
	`// if 2+ shakes, count`
	`else {`
	`N = N + 1;`
	`}`
	
	`// if 3rd shake, turn on lighs`
	`if(N == 3) {`
	`PORTC = 0xFF;`
	`TIME2 = 1000;`
	`}`
	`INT0IF = 0;`
	`}`
	
	On a rising edge on RB0 (INT0IF) if the timer is zero, start a 500ms timer and set a counter (N) to one.
	
	Else, increment the counter each rising edge.
	
	If you detect a 3rd rising edge in 500ms, start a 1000ms timer to turn on PORTC
		`if(TMR2IF) {`
		
		`//decrement 500ms counter to zero`
		`if(TIME) TIME -= 1;`
		
		`//decrement 1000ms counter to zero`
		`if(TIME2) TIME2 -= 1;`
		
		`//turn off lights after 1000ms`
		`else PORTC = 0;`
		
		`TMR2IF = 0;`
		`}`
		
		every 1ms, decrement each timer to zero, stop at zero.
		
		Once the 1000ms timer goes to zero (TIME2), clear PORTC