ECE 376 - Homework #3

Binary Inputs, Binary Outputs, & LEDs - Due Monday, January 29th

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 < 10C
- 5V when T > 10C

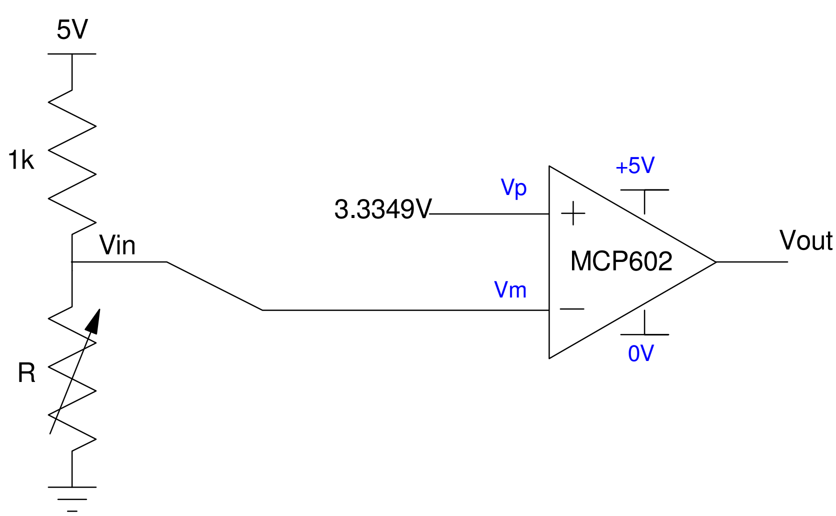
Assume a voltge divider with a 1k resistor. At 10C,

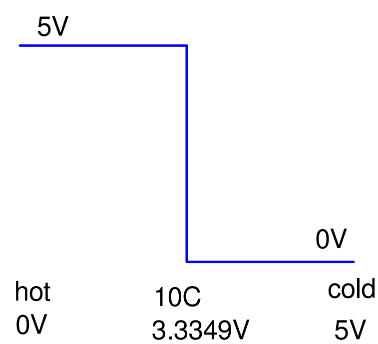
- R = 2002.817 Ohms
- Vin = 3.3349V

As temperature goes up

- R goes down
- Vin goes down
- Vout goes up

Connect to the minus input (negative correlation)





2) Design a circuit which outputs

- 0V when T < 10C
- 5V when T > 15C

This is a Schmitt trigger. Assume a voltage divider with a 1k resistor:

At 10C,

- R = 2002.817 Ohms
- Vin = 3.3349V
- Vout goes low

At 15C,

- R = 1576.1749 Ohms
- Vin = 3.0591V
- Vout goes high

Von < Voff

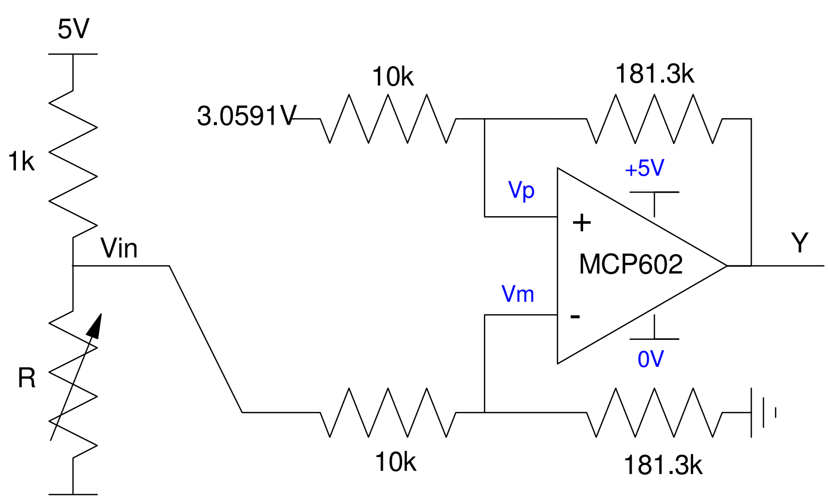
- Connect to the minus input

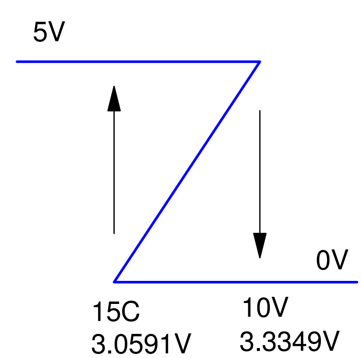
Von = 3.0591V

- set the offset to 3.0591V

Slope = gain

- $gain = \left(\frac{5V-0V}{3.3349V-3.0591V}\right) = 18.13$
- set the resistor ratio to 18.13





3) Design a circuit which outputs

- 5V when 10C < T < 15C
- 0V otherwise

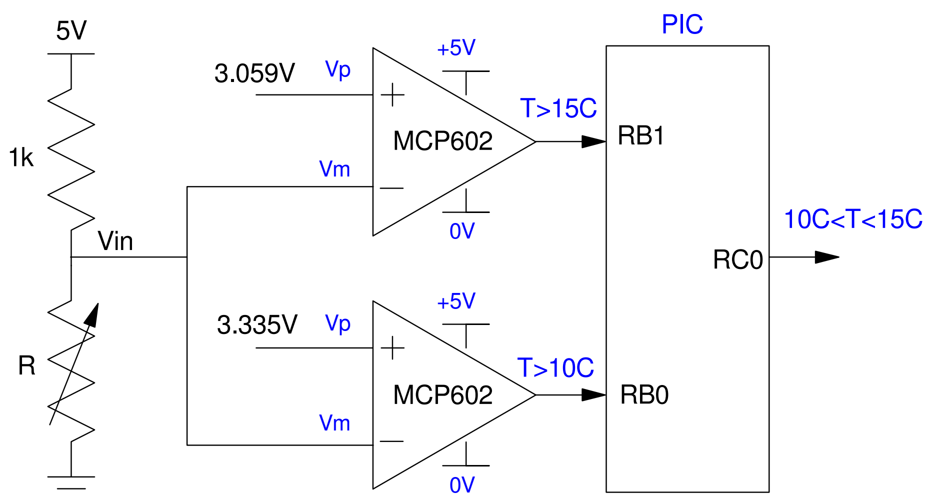
Option #1: Use two comparitors (problem #1)

- RB0: T > 10C
- RB1: T > 15C

In software, implement the logic

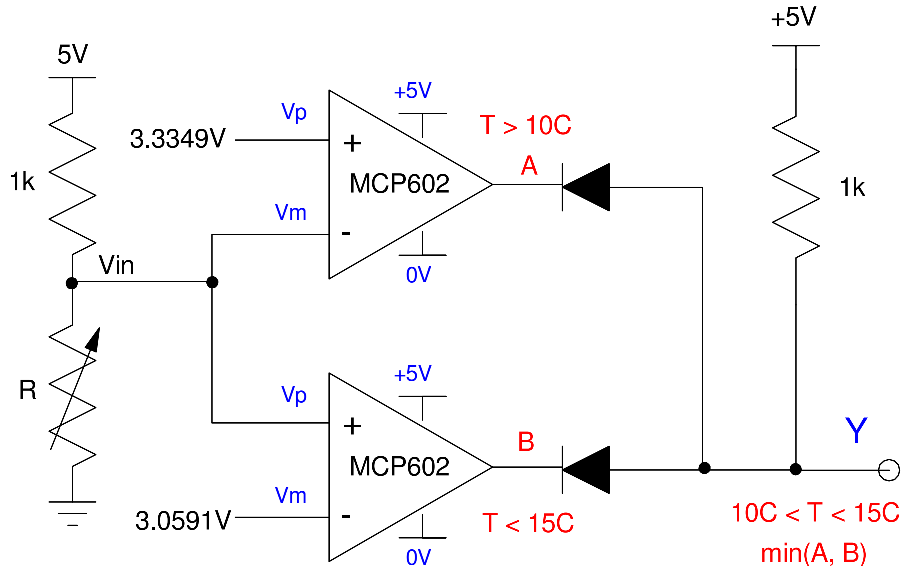
$$RC0 = RB0 \cdot \overline{RB1}$$

```
Main:
        btfsc     PORTB,1
        goto      Clear
        btfss     PORTB,0
        goto      Clear
Set:
        bsf       PORTC,0
        goto      Main
Clear:
        bcf       PORTC,0
        goto      Main
```



Option #2: Get a little tricky with diodes implementing a min function

- Y1: T > 10C
- Y2: T < 15C
- Y = min(Y1, Y2)



Binary Outputs

4) Design a circuit which allows your PIC board to turn on and off an RGB Piranah LED at 0mA (off) and 10mA (on). Assume the specifications for the LEDs are:

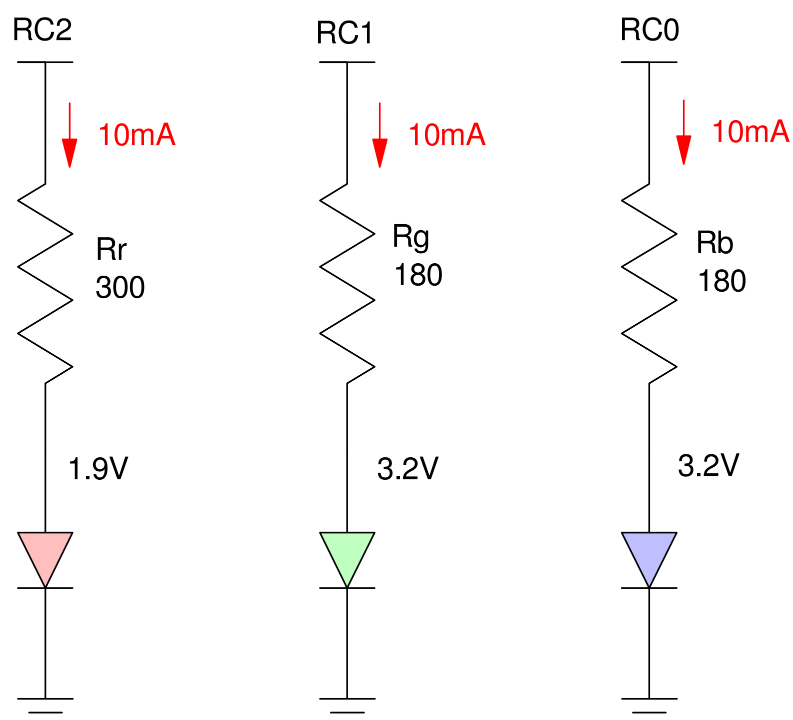
Color	Vf @ 20mA	mcd @ 20mA
red	2.0V	10,000
green	3.2V	10,000
blue	3.2V	10,000

Since this is less than 5V and 25mA, connect directly to a PIC using a resistor

$$R_r = \left(\frac{5V - 2.0V}{10mA}\right) = 300\Omega$$

$$R_g = \left(\frac{5V - 3.2V}{10mA}\right) = 180\Omega$$

$$R_b = \left(\frac{5V - 3.2V}{10mA}\right) = 180\Omega$$



5) Design a circuit which allows your PIC board to turn on and off a 5W LED at 250mA. The specs for the LED are:

- Vf = 6.0-7.0V
- Current = 700mA
- 500-600 Lumens (equivalent to a 60W 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

Assume you have a 6144 NPN transistor:

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

Since this is more than a PIC can output, use an NPN transistor as a buffer (switch)

Step 1: Set the current to 250mA

- Assume a 12V power supply

$$R_c = \left(\frac{12V - 6.5V - 0.2V}{250mA}\right) = 21.2\Omega$$

Pick Rc to saturate the transistor

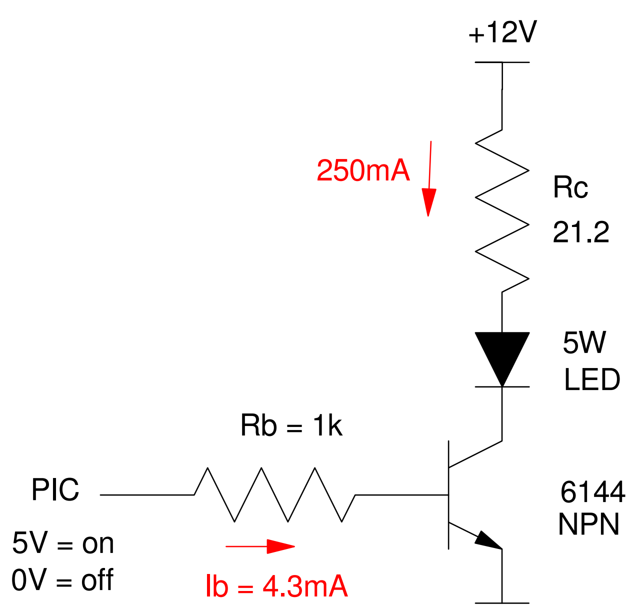
$$\beta I_b > I_c$$

$$300 I_b > 250mA$$

$$I_b > 0.833mA$$

Let Ib = 4.3mA (arbitrary: more than 0.833mA, less than 25mA)

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



Timing:

6) Write a program which outputs the music note E4 (329.63 Hz)

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

First, calculate the number of clocks between toggles

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz}\right) = 15,168.5223$$

Come up with a wait look that burns 15,168 clocks

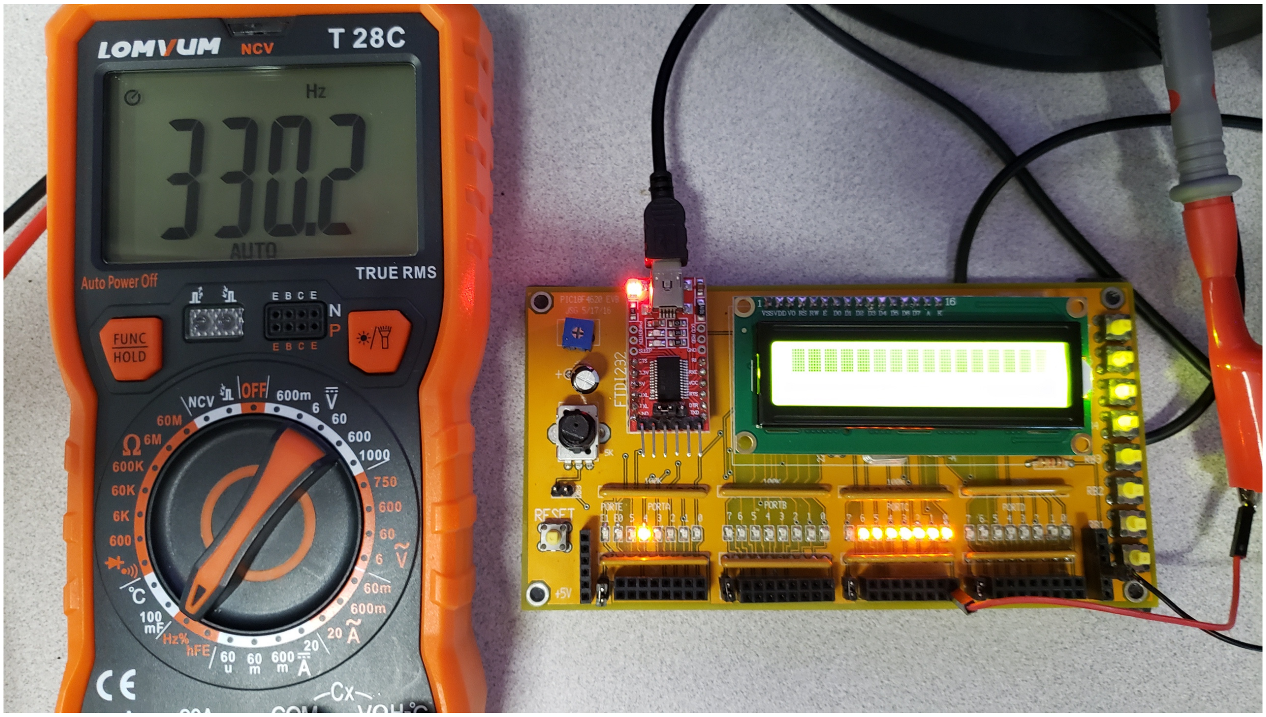
N = 10*A*B + 5*A + 9 = 15,168

A = 7, B = 216 (N = 15,164, 0.03% low)

```
#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 7        ; A
       movwf CNT1
W1:
          movlw 216    ; B
          movwf CNT0
W0:
             nop       ; 10 clocks
       nop
             nop
             nop
             nop
             nop
             nop
             decfsz CNT0, F
             goto W0
          decfsz CNT1, F
          goto W1
       return
```

Result = 330.2Hz

- +0.17% high



Lab: PIC Stoplight

7) Give the flow chart for a program to turn your PIC board into a stoplight

- PORTC = East/West
- PORTD = North/South

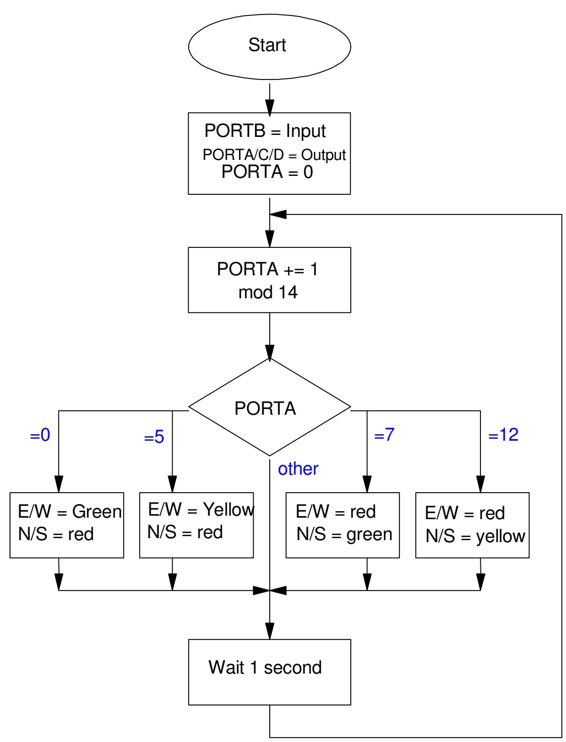
	7	6	5	4	3	2	1	0
PORTC (E/W)	-	-	R	R	Y	Y	G	G
PORTD (N/S)	-	-	R	R	Y	Y	G	G

The stoplight cycles every 14 seconds

Duration (seconds)	E/W	N/S
5s	G	R
2s	Y	R
5s	R	G
2s	R	Y

Counting mod 14

Count	E/W	N/S
0	0x03 (green)	0x30 (red)
5	0x0C (yellow)	0x30 (red)
7	0x30 (red)	0x03 (green)
12	0x30 (red)	0x0C (yellow)



8) Write the corresponding assembler code

- Include a routine which waits

```
; --- Stoplight.asm ----

#include <p18f4620.inc>

; Variables

SEC             equu    0
CNT0    equ     1
CNT1    equ     2
CNT2    equ     3
CNT3    equ     4

        org     0x800
        call    Init
L1:
        call    Count
        call    Lights
        call    Wait
        goto    L1

Init:
        clrf    TRISA
        clrf    TRISB
        clrf    TRISC
        clrf    TRISD
        movlw   0x0F
        movwf   ADCON1
        clrf    SEC
        return

Count:
        incf    SEC,F
        movlw   14
        cpfseq  SEC
        goto    L2
        clrf    SEC
L2:
        movff   SEC,PORTA
        return

Lights:
        movlw   0
        cpfseq  SEC
        goto    L3
        movlw   0x03
        movwf   PORTC
        movlw   0x30
        movwf   PORTD
        return
L3:
        movlw   5
        cpfseq  SEC
        goto    L4
        movlw   0x0C
        movwf   PORTC
        movlw   0x30
        movwf   PORTD
        return
L4:
        movlw   7
        cpfseq  SEC
        goto    L5
```

```
        movlw    0x30
        movwf    PORTC
        movlw    0x03
        movwf    PORTD
        return
L5:
        movlw    12
        cpfseq   SEC
        goto     L6
        movlw    0x30
        movwf    PORTC
        movlw    0x0C
        movwf    PORTD
        return
L6:
        return


; One second wait routine
; N = 10ABC + 5AB + 5A + 4
; N = 10,050,504


Wait:
        movlw    100        ; A
        movwf    CNT2
W2:
        movlw    100        ; B
        movwf    CNT1
W1:
        movlw    100
        movwf    CNT0       ; C
W0:
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        decfsz   CNT0,F
        goto     W0

        decfsz   CNT1,F
        goto     W1

        decfsz   CNT2,F
        goto     W2

        return

        end
```

9) Test your code.

- Compile and program your PIC board
- Verify each button's operation

Step #1: Test the wait routine

```
        org       0x800
        call      Init
   L1:
        incf      PORTC,F
   ;    call      Count
   ;    call      Lights
        call      Wait
        goto      L1
```

After fixing the wait routine, PORTC counts every second

Step #2: Test the count mod-12 routine

```
        org       0x800
        call      Init
   L1:
        call      Count
   ;    call      Lights
        call      Wait
        goto      L1
```

After fixing more bugs, PORTA counts 0..13 (mod 12)

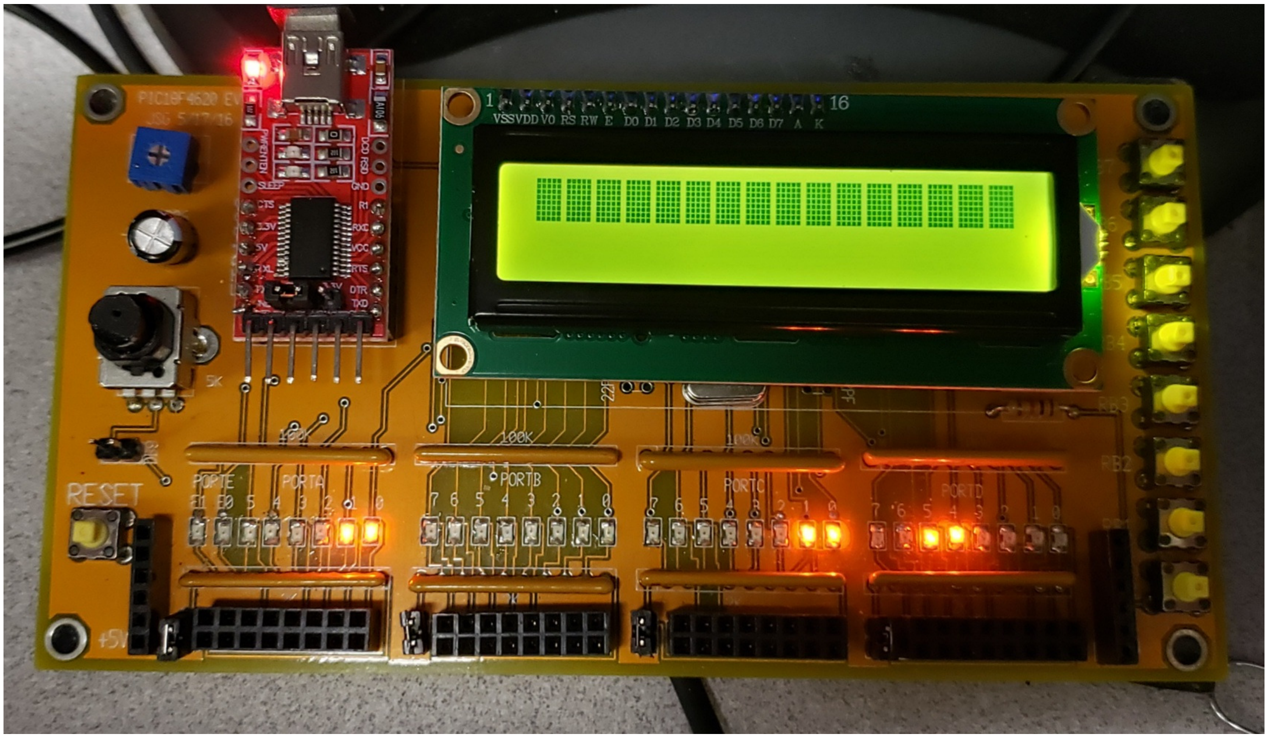
Step #3: Test the lights routine

```
        org       0x800
        call      Init
   L1:
        call      Count
        call      Lights
        call      Wait
        goto      L1
```

After a few more bugs, the lights work correctly

10) (20 points) Demonstration

- In-person of with a video



Stoplight Code:

- PORTA = Count (currently at 3 seconds)
- PORTC = E/W (currently green light)
- PORTD = N/S (currently red light)