

## Binary Outputs: LEDs

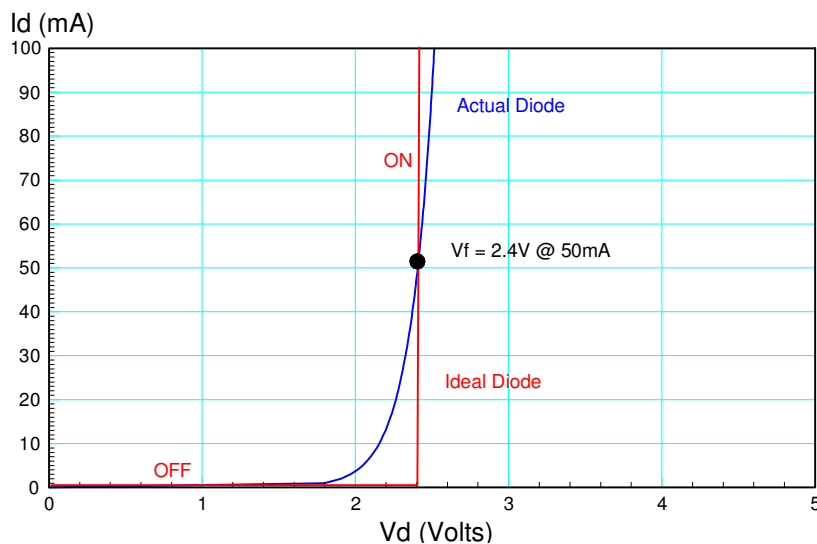


1W Star LED, 10mm 0.5W White LED, and Piranha RGB LED  
<http://www.ebaystores.com/Lighthouse-LEDs>

### Light Emitting Diodes ( LED's )

- Are diodes, allowing current to only flow in one direction,
- They convert current to light.(light is proportional to current flow), and
- They are *very* fast, capable of over 1000 flashes per second, and
- They are a simple way to output binary data (light on / light off)

Like all diodes, the V/I characteristics are exponential in nature - similar to the figure below.



Typical VI characteristic for an LED (blue line) and an ideal diode approximation (red line)

This nonlinear relationship makes diode circuits difficult to analyze. To simplify analysis, an ideal diode model is used

- When the diode is on, the voltage across the diode is assumed to be constant
- If the voltage drops below a threshold ( $V_f$ ), the diode is assumed to be off (current equals zero)

This is also how LED's are specified

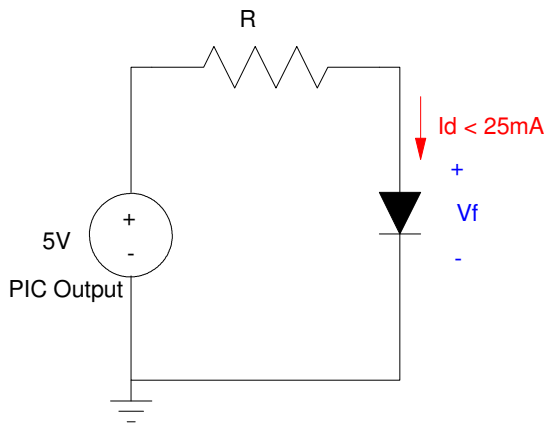
- $V_f$ : The voltage drop across the diode when on
- Typical mcd: The amount of light the LED outputs at a given current level
- Color: Kind of self evident
- Wavelength: A more accurate way of specifying the color of the LCD

For example, the diodes in room 211 have the following parameters:

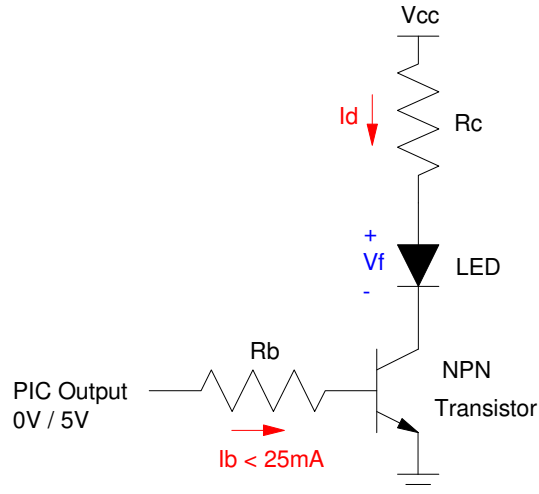
Part #	Color	Current	Typical $V_f$	Typical mcd	Wavelength (nm)	Price ea
1W White Star LED		350mA	3.4V	100 lm	n/a	\$1.55
0.5W 10mm White LED		100mA	3.3V	25 lm	n/a	\$0.30
Piranah RGB LED	Red	20mA	1.8V	8000 mcd	630 nm	\$0.31
	Green	20mA	3.0V	8000 mcd	525 nm	
	Blue	20mA	3.0V	8000 mcd	470 nm	

Two circuits are used to drive an LED with a microcontroller:

- If the LED needs less than 5V and less than 25mA, you can drive it directly with the PIC using just a resistor.
- If you need more voltage or current, an NPN transistor is required



Case 1:  $I_d < 25\text{mA}$



Case 2:  $I_d > 25\text{mA}$

**Case 1:  $I_d < 25mA$** 

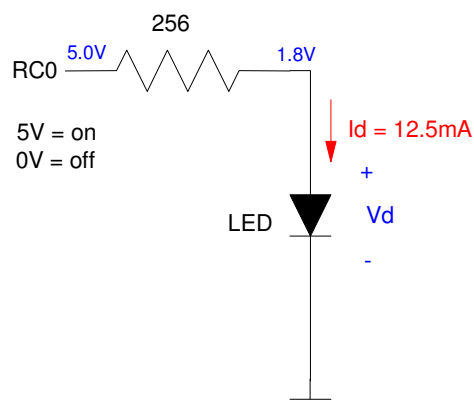
Example: Design a circuit so that a PIC processor can turn on and off the red LED on a piranah RGB LED at 5000 mcd

Solution: The current you need is

$$I_d = \left( \frac{5000mcd}{8000mcd} \right) \cdot 20mA = 12.5mA$$

The resistance you need to limit the current to 12.5mA is

$$R_r = \left( \frac{5V - 1.8V}{12.5mA} \right) = 256\Omega$$

**Case 2:  $I_d > 25mA$** 

Use a transistor as an electronic switch

# Binary Outputs: Transistors Used as a Switch

## Background

A PIC is able to output +5V at 25mA. Some devices require too much power for a PIC to drive these directly. These include:

- A stepper motor which draws 1A at 12V
- An LED array (i.e. a tail light), which draws +12V and 200mA
- A heater, which draws 2.4A at 25V.

If the device you're trying to drive requires more voltage or more current, you need a buffer circuit. This lecture covers three types of buffers: relays, transistors, and MOSFETs.

## Relays:

A simple way to allow a PIC to turn on and off a device is to use a relay. For example, the following is available from Marlin P. Jones ([www.mpja.com](http://www.mpja.com)).

- Input: 3 - 32V DC @ < 1mA
- Output: 40 - 480VDC @ 10A max

If you connect the input to a PIC, the PIC can turn on and off anything you power with up to 480VDC. With a similar module, you can turn on and off anything powered by 120VAC as well.

### 20A, 480VDC Solid State Relay



Stock No: 17288 RL

Review Average: ★★★★★  
Number of Reviews: 1

[View Reviews](#) | [Review this item](#)

**\$14.95**

**Special Pricing**

Buy 12 for \$12.95 each

Availability: **In Stock** Units: **6**

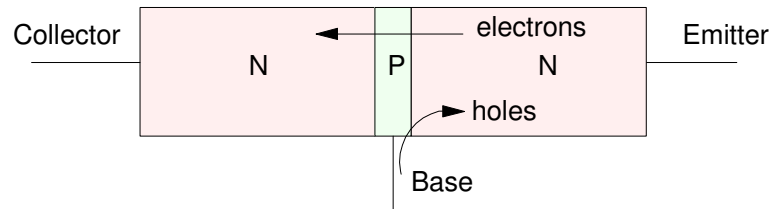
Solid State Relay from [www.mpja.com](http://www.mpja.com). AC versions capable of driving up to 480VAC at 10A are also available.

## Transistors

Transistors can also be used as a switch - only they're much cheaper and smaller than a relay. Transistors are three-terminal devices and can be of type NPN or PNP. The main one we will use in the NPN transistor.

The three terminals of an NPN transistor are called the collector, base, and emitter. Between the emitter and the collector is a reverse biased PN junction. Normally, this prevents current flow from the collector to the emitter (electrons, being negatively charged, flow the opposite direction). The switch is open.

If you apply current to the base to the emitter, however (which is a forward biased PN junction), holes flow from the P-type base and electrons flow from the N-type emitter.

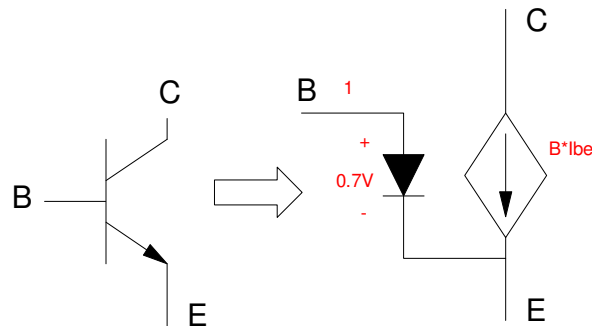


NPN Transistor: The base current controls the current from collector to emitter

If the base is very thin, most of the electrons pass through the base and appear at the collector, creating current flow from the emitter to the collector. (The switch is closed). If the doping on the emitter side is times higher than the doping in the P-type base, you get times more electrons than holes. This creates current amplification.

The model for a transistor reflects this:

- Between the base and the emitter is a diode. It takes 0.7V to turn on a Silicon diode.
- Between the collector and the emitter is a current-controlled current source. The current flow is times the base to emitter current.



Symbol and Circuit Model for an NPN Transistor: The arrow indicates a diode from base to emitter.

The transistors we use in lab have parameters given in the following table:

	3904	TIP112
Type	NPN	NPN
Current Gain	100	1,000
max(Ic)	200mA	4A
max(Vce)	40V	40V
Vce(sat)	300mV	900mV
Vbe	0.7V	1.4V
price	\$0.037	\$0.59

- The current gain tells you what  $\beta$  is.
- $\max(I_c)$ : When used as an "on" switch, this is the maximum current this transistor can conduct.
- $\max(V_{ce})$ : When used as an "off" switch, this is the maximum voltage the transistor can block.
- $V_{ce(sat)}$ : When used as an "on" switch, the voltage drop across the transistor.
- Price: The reason we use the 3904 and 3906 transistors in lab. We go through a lot of them.

Note that  $V_{ce}$  depends upon the base current and the collector current. Normally, we assume this is 0.2V for all transistors as a ballpark estimate.

## Using a Transistor as a Switch

To use an NPN transistor as a switch

- First, design a circuit which turns on your device including an extra 0.2V drop to model the losses in the transistor
- Next, break the path from power to ground, placing the NPN transistor in series with the emitter tied to ground.
- Finally, add a resistor to the base chosen so that

$$\beta I_b > I_c$$

Example: Design a circuit so that a PIC processor can turn on and off a 1W LED at 100mA. Assume the LED has the following specification:

- $V_f = 3.5V @ 350mA$
- 100 Lumens @ 350mA



1W Proflight 1W High Power LED: \$1.14 ea. <http://stores.ebay.com/TopBright-Led-Store>

Solution: First, design a circuit that drives the LED at 100mA along with a 0.2V drop. Assuming a 5V power supply,

$$R_c = \left( \frac{5V - 3.5V - 0.2V}{100mA} \right) = 13\Omega$$

Next, add an NPN transistor (3904) in series with the emitter tied to ground.

Finally, add  $R_b$  so that  $\beta I_b > I_c$ . When on,

$$I_c = 100mA$$

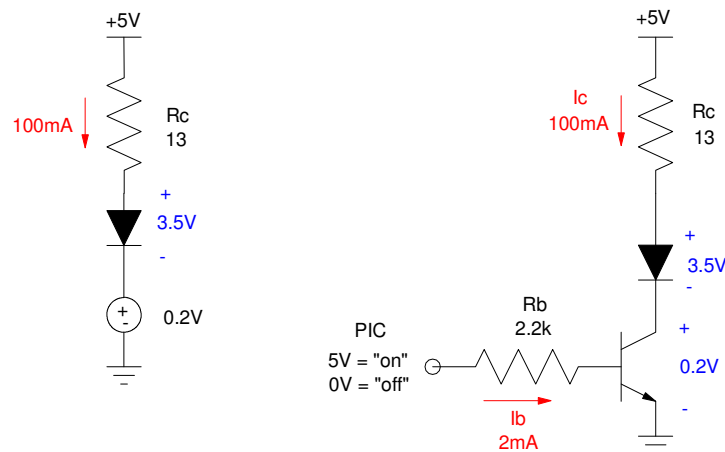
meaning

$$I_b > \frac{I_c}{\beta} = 1mA$$

Let  $I_b = 2mA$ .  $R_b$  is then

$$R_b = \left( \frac{5V - 0.7V}{2mA} \right) = 2150\Omega$$

The exact value of  $R_b$  isn't critical - so long as  $\beta I_b > I_c$ . Round  $R_b$  to 2.2k



Design of a Circuit to Drive 100mA Through the LED (left) and Final Circuit which Allows a PIC to Turn On and Off the LED (right)

One way to look at what's going on with this circuit is to look at the load-line relative to the transistor:

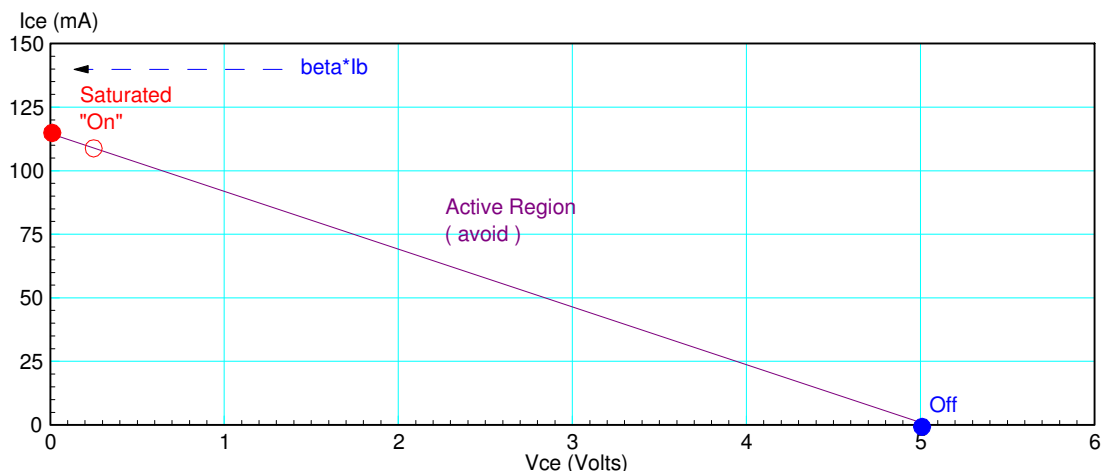
- When  $I_c = 0$ ,  $V_{ce} = 5V$
- When  $V_{ce} = 0V$ ,  $I_c = 115mA$

The base current,  $I_b$ , controls the collector current,  $I_c$  as

$$I_c = \beta I_b$$

When  $I_b = 0$ ,  $I_c = 0$  and the transistor is off. This is easily done by outputting 0V from the PIC processor.

If  $\beta I_b$  is more than 100mA, the current  $I_c$  clips at 100mA. This is called the "on" state or the transistor is saturated (it can't draw any more current).



Load Line for the NPN Transistor: Ideally you want to operate in the off or on (saturated) state.

Both of these states are what you want for an on/off switch:

- Off:  $I_c = 0$
- On:  $V_{ce} = 0$  (ideally) or 0.2V (in practice)

It also has the advantage that the power the transistor has to dissipate is zero at both of these states:

- The current is zero when off, meaning  $P = VI = 0$
- The voltage is zero when on, meaning  $P = VI = 0$  (approximately)

What you want to avoid is operating in-between these two points - where the power dissipated by the transistor is a maximum. This is why you design for  $\beta I_b > I_c$  : you want to make sure the transistor is saturated.

Example 2: Drive a 3W LED at 1A ( $V_f = 3.0V$ )

Solution: In this case, we need to use a TIP112 transistor (a 3904 can't handle 1A). The specs for a TIP112 transistor are:

- $|V_{be}| = 1.4V$
- $|V_{ce(sat)}| = 0.9V$
- $\beta = 1000$

First, design a circuit to turn the LED on at 1A. Assuming a 10V power supply

$$R_c = \left( \frac{10V - 3.0V - 0.9V}{1A} \right) = 6.1\Omega$$

To saturate the transistor

$$\beta I_b > I_c$$

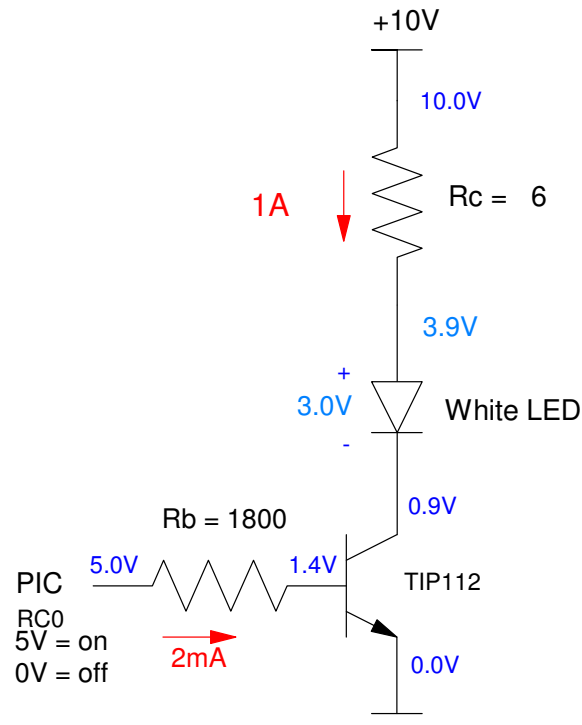
$$I_b > \frac{I_c}{1000} = \frac{1A}{1000} = 1mA$$



Let  $I_b = 2\text{mA}$ . Then

$$R_b = \left( \frac{5\text{V} - 1.4\text{V}}{2\text{mA}} \right) = 1800\Omega$$

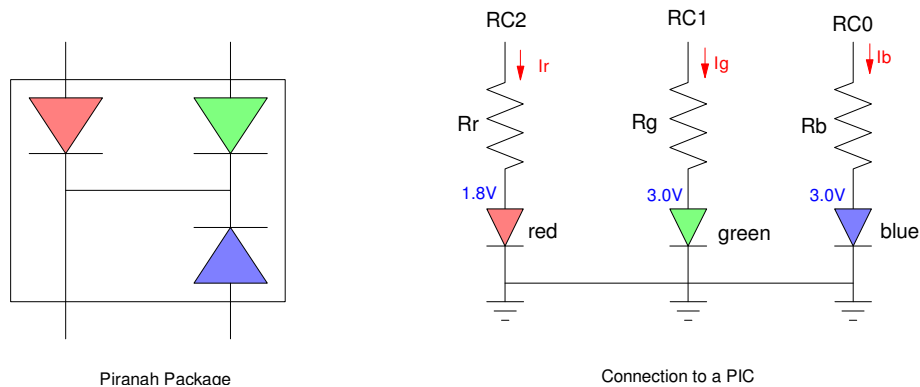
Again, the exact value of  $R_b$  isn't critical: as long as  $\beta I_b > I_c$  the transistor will saturate



## More Fun with RGB LEDs

If you mix a red, green, and blue LED, you can make other colors by mixing them. (RGB are the primary colors in light). If you can adjust the brightness of each one with  $N$  grey levels, you are able to output  $N^3$  combinations.

The Pirannah LEDs combine three LEDs in a single package as follows. Using three resistors, you can turn on and off each LED separately with a PIC processor.



Problem 1: LED Flashlight. Design a system which allows you to output 4 different colors based upon which button you press:

- RB0                      All lights off
- RB1                      Red light on
- RB2                      Green light on
- RB3                      Blue light on

Define "on" to be 20mA.

Hardware Solution: Connect the Pirannah LED to the PIC board as follows. Pick the three resistors so that 20mA is flowing when the output pin is 5V.

---

**Software Solution:**

```
; --- Flashlight.asm ----
; This program drives an RGB flashlight based upon the button pressed

COLOR      equ      0

#include <p18f4620.inc>

          org 0x800
          call Init

Loop:
    movf    COLOR,W
    btfsc   PORTB,0      ; if RB0 pressed, color = 000 (off)
    movlw   0
    btfsc   PORTB,1      ; if RB1 pressed, color = 001 (blue)
    movlw   1
    btfsc   PORTB,2      ; if RB2 pressed, color = 010 (green)
    movlw   1
    btfsc   PORTB,3      ; if RB3 pressed, color = 100 (red)
    movlw   4
    movwf   COLOR        ; send the color to PORTC
    movff   COLOR, PORTC

    goto   Loop

Init:
    movlw   0xFF
    movwf   TRISB
    clrf    TRISC
    movlw   0x0F
    movwf   ADCON1
    clrf    COLOR
    return

end
```

Problem #2: Make the LEDs only 20% on (1600 mcd each).

Hardware Solution: Change the resistor.

Software Solution: Change the code so that each pin is high 20% of the time.

```

; --- PWM.asm ----
; This program drives an RGB flashlight based upon the button pressed

COLOR      equ      0
CNT1       equ      1

#include <p18f4620.inc>

                org 0x800
                call Init

Loop:
                clrfs    PORTC                ; Turn off the lights for four counts
                call    Wait
                call    Wait
                call    Wait
                call    Wait

                movf    COLOR,W
                btfsc   PORTB,0                ; if RB0 pressed, color = 000 (off)
                movlw   0
                btfsc   PORTB,1                ; if RB1 pressed, color = 001 (blue)
                movlw   1
                btfsc   PORTB,2                ; if RB2 pressed, color = 010 (green)
                movlw   2
                btfsc   PORTB,3                ; if RB3 pressed, color = 100 (red)
                movlw   4
                movwf   COLOR                  ; send the color to PORTC
                movff   COLOR, PORTC
                call    Wait                  ; Turn on for one count

                goto    Loop

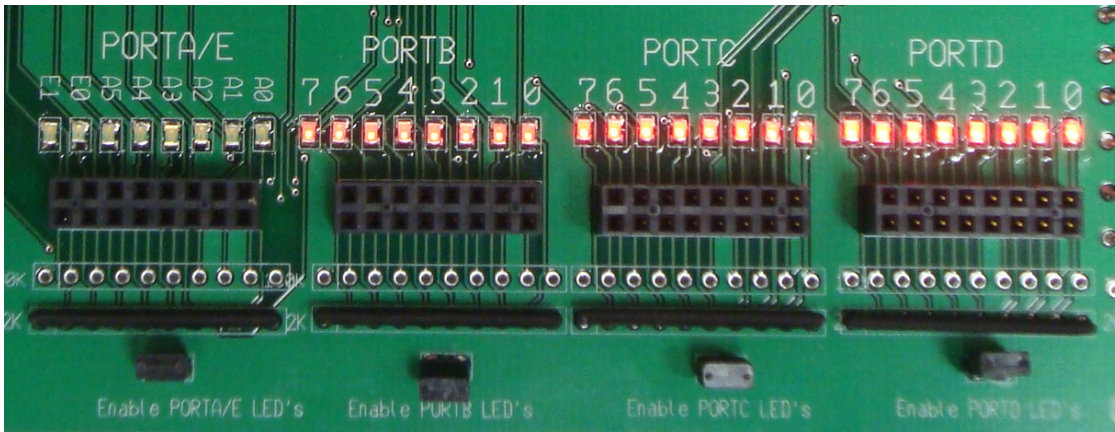
Init:
                movlw   0xFF
                movwf   TRISB
                clrfs   TRISC
                movlw   0x0F
                movwf   ADCON1
                clrfs   COLOR
                return

Wait:
                movlw   250                    ; waste 25,005 clocks (2.5ms)
                movwf   CNT1

L2:
                nop
                nop
                nop
                nop
                nop
                nop
                decfsz  CNT1,F
                goto    L2
                return
                end

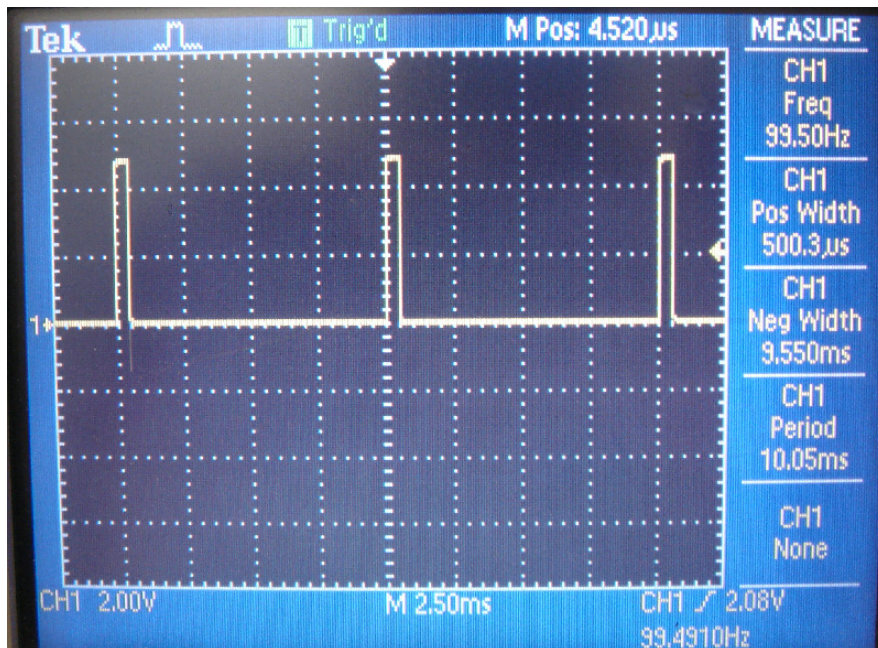
```

If you change the code so that the lights are 5% on (PORTB), 20% (PORTC), and 90% (PORTD), it looks like the following. Note that it looks like the LEDs are variable brightness.

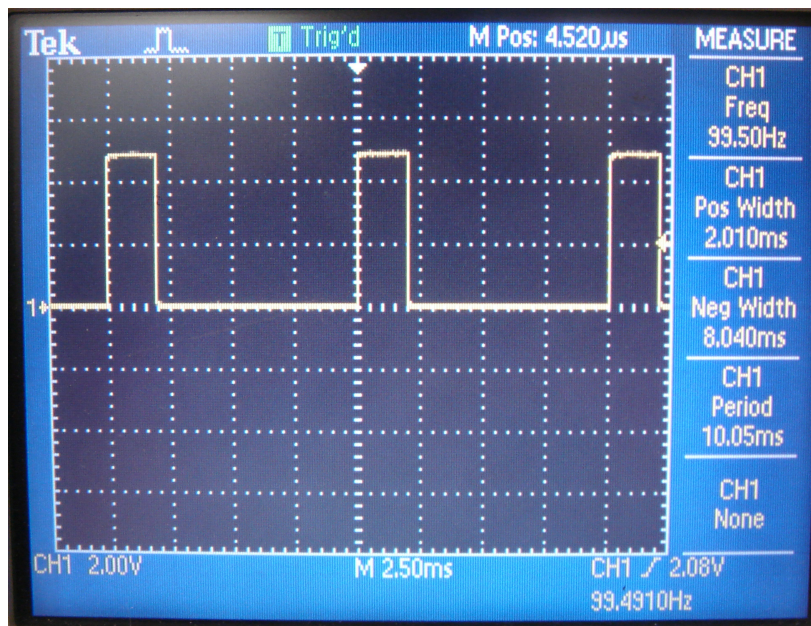


Assembler Code - Modified so that PORTB is on 5% of the time, PORTC is on 20% of the time, PORTD is on 90% of the time

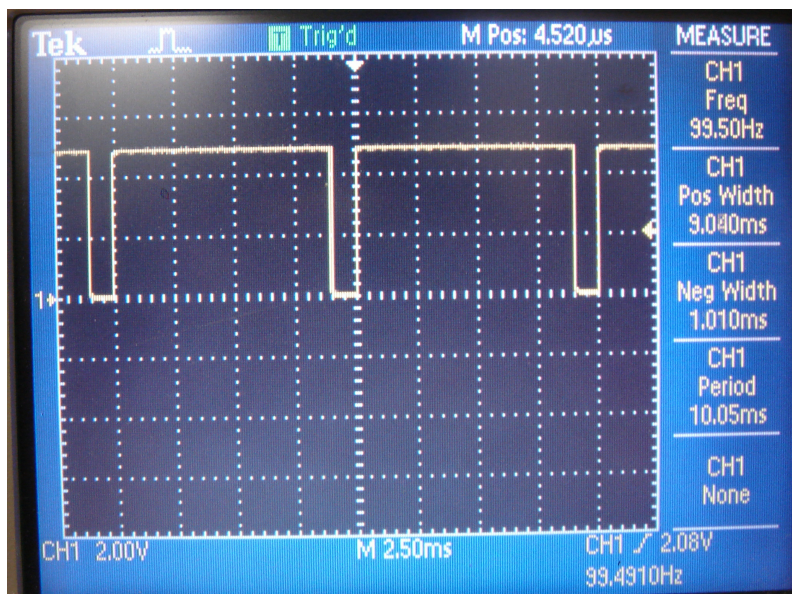
On an oscilloscope, you can see this better;



Signal for 5% Duty Cycle:



Signal for 20% Duty Cycle:



Signal for 90% Duty Cycle: