R,G,B LEDs

Three PWM outputs and *three* primary colors. Just screams to be made, doesn’t it?

Put back on the ProtoShield for this.
Use either the 220 or 330 ohm resistors in your kit, if you don’t have enough of one or the other
I have lots more 220 if you need them
Cut leads of resistors and LEDs to make for a more compact circuit. Also, less likely to short against itself.
RGB Color Fading

“dimmingLEDs”

Slow color fading and mixing

Also outputs the current color values to the serial port

This sketch is located in the handout. It just ramps up and down the red, green, & blue color values and writes them with analogWrite() from [http://www.arduino.cc/en/Tutorial/DimmingLEDs](http://www.arduino.cc/en/Tutorial/DimmingLEDs).
Mood Light

Diffuser made from piece of plastic scratched with sandpaper

Also, can use plastic wrap scrunched up to make an interesting diffuser.
Serial-controlled RGB

“serial_rgb_led”

Send color commands to Arduino e.g. “r200”, “g50”, “b0”

Sketch parses what you type, changes LEDs

This sketch is located in the handout.
Color command is two parts: colorCode and colorValue
- colorCode is a character, ‘r’, ‘g’, or ‘b’.
- colorValue is a number between 0–255.

Sketch shows rudimentary character string processing in Arduino
Reading Serial Strings

• New Serial function in last sketch:
  “Serial.available()”

• Can use it to read all available serial data from computer

• Great for reading strings of characters

• The “readSerialString()” function at right takes a character string and sticks available serial data into it

```
//read a string from the serial and store it in an array
//you must supply the array variable
void readSerialString (char *strArray) {
    int i = 0;
    if(!Serial.available()) {
        return;
    }
    while (Serial.available()) {
        strArray[i] = Serial.read();
        i++;
    }
}
```

Pay no attention to the pointer symbol (“*”)
Must be careful about calling readSerialString() too often or you’ll read partial strings
RGB LEDs

Normal LED

anode +

- cathode

RGB LED

anode +

red blue green

red cathode -
blue cathode -
green cathode -

actually 3 LEDs in one package

RGB LED, aka “tri-color LED”
Common-anode RGB LEDs are much more available than common-cathode. This is why we’re changing around the logic.
With just 3 LEDs you can make any* color

Mixing light is the additive color model
(paint is subtractive color, and can give you brown)

*besides the additive/subtractive color different, it’s hard to get the mix to be just right for a variety of annoying reasons:
- the physics of LEDs mean that different color LEDs put out different amounts of light
- our eyes respond non-linearly across the spectrum, i.e. we’re more sensitive to green than red
- the lenses in most RGB LEDs don’t focus each color to the same spot
Laying out RGB LED Circuit

- Slightly bend the longest lead and plug it into the +5v (red) bus.
- Plug remaining leads into rows (12, 14, & 16 here).
- Connect 220 (red-red-brown) resistors across middle to matching rows.
- Run wires from resistors to pins 9, 10, 11 of Arduino, can color-code if you want.

Ignore the green wire in the pictures, that's another circuit. Keep the pot from last circuit if you can.
RGB Color Fading

“RGBMoodLight”

Slow color fading and mixing

Also outputs the current color values to the serial port

This sketch is located in the handout. We’ll get to the serial port stuff in a minute.

It just ramps up and down the red, green, & blue color values and writes them with analogWrite() from [http://www.arduino.cc/en/Tutorial/DimmingLEDs](http://www.arduino.cc/en/Tutorial/DimmingLEDs)
Pot-controlled RGB

Arduino board

pin 1
gnd

pin 10
pin 9
220 (red, red, brown)
red green blue
+5V
common anode RGB LED

pin 2
+5V
gnd
50k pot
Pot-controlled RGB

“RGBPotMixer”

Use the pot from before to control the color mix

The code turns the single ranged input value into “sectors” where each sector is a color

Also see “RGBPotMixer2” for a variation.
How would you change it to adjust brightness?