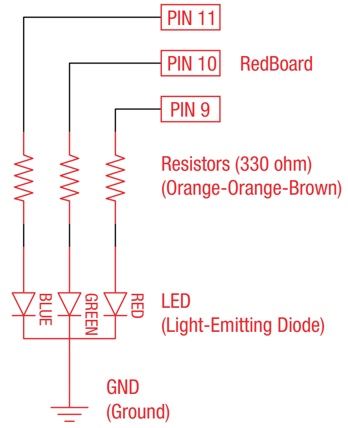
CIRCUIT #3

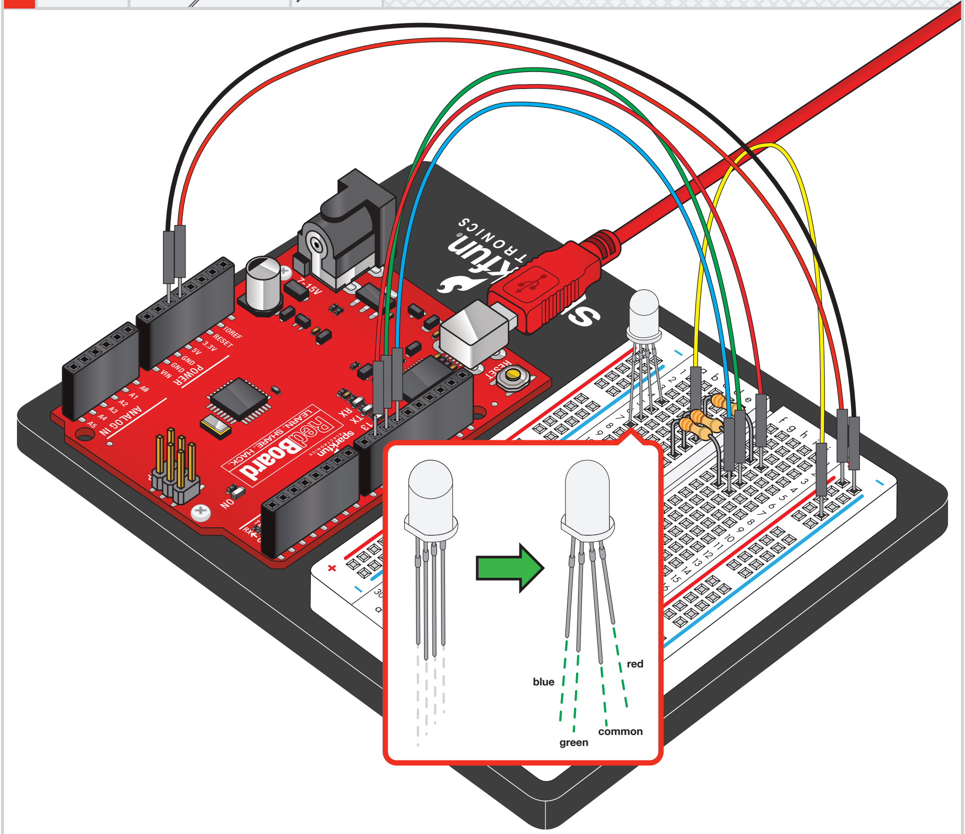
3



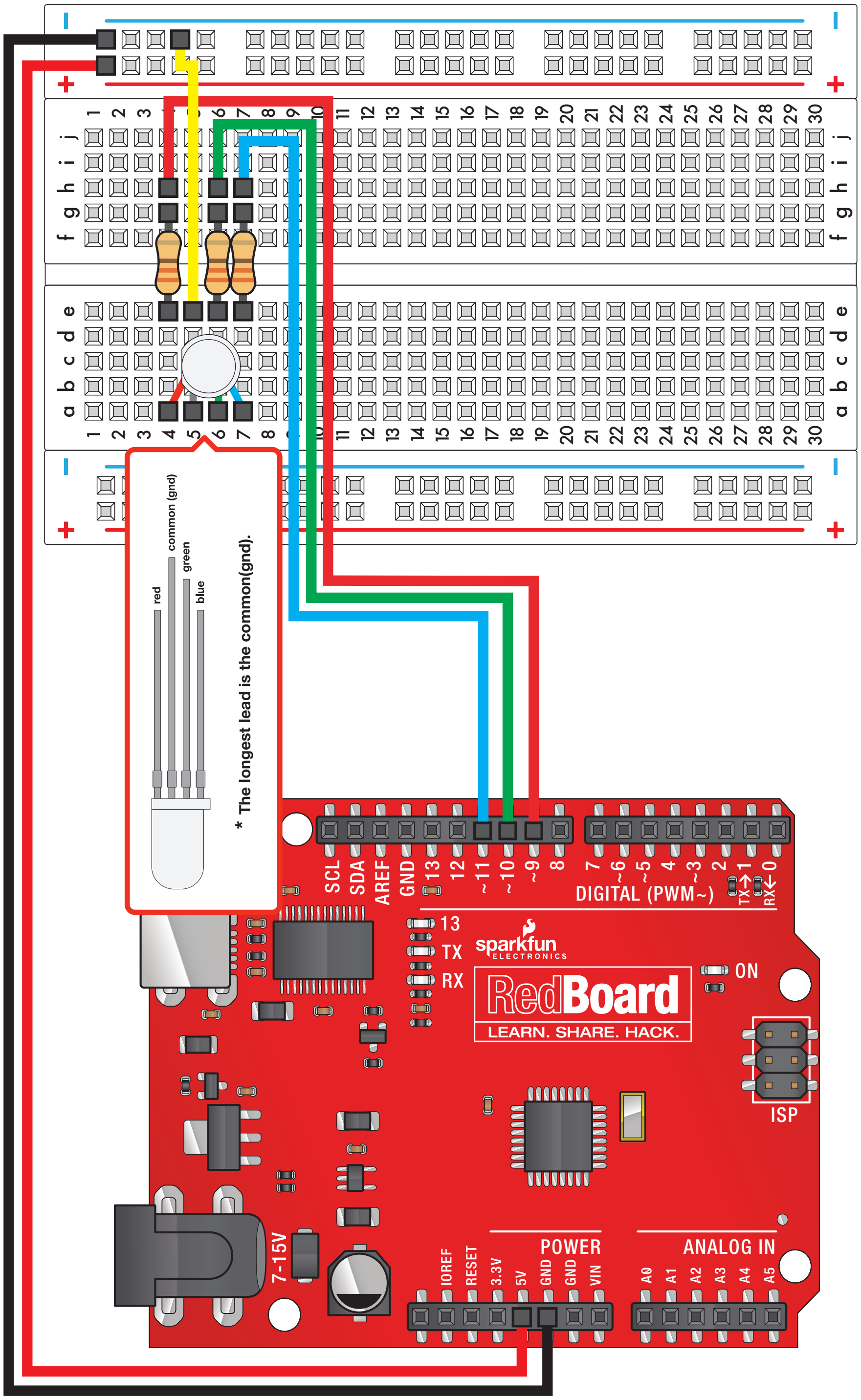
RGB LED

You know what's even more fun than a blinking LED? Changing colors with one LED. RGB, or red-green-blue, LEDs have three different color-emitting diodes that can be combined to create all sorts of colors. In this circuit, you'll learn how to use an RGB LED to create unique color combinations. Depending on how bright each diode is, nearly any color is possible!

PARTS:	LED	330Ω Resistor	Wire
	x 1	x 3	x 6

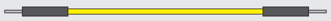
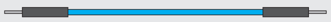
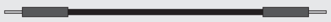


Circuit 3: RGB LED



* The longest lead is the common(gnd).

]

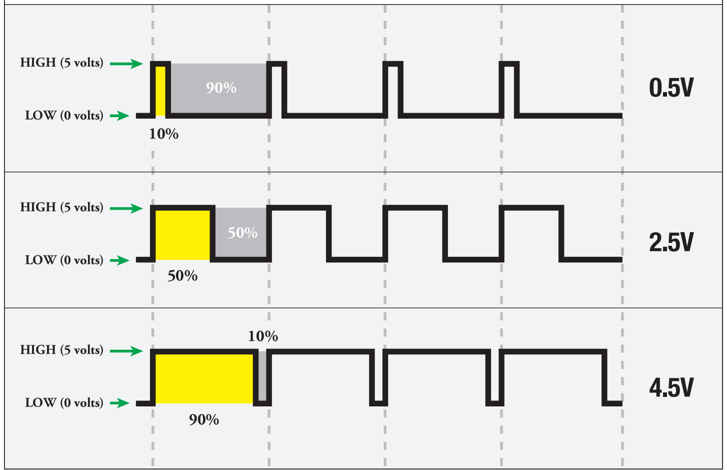
Component:	Image Reference:		
RGB LED (5mm) ⚠		▨	a4 a5 a6 a7 🟥 ⬜ 🟩 🟦
330Ω Resistor		▨	e4 - g4
330Ω Resistor		▨	e6 - g6
330Ω Resistor		▨	e7 - g7
Jumper Wire		Pin 9	h4
Jumper Wire		▨	e5 - -
Jumper Wire		Pin 10	h6
Jumper Wire		Pin 11	h7
Jumper Wire		5V	+
Jumper Wire		GND	-

The shocking truth behind analogWrite():

We've seen that the Arduino can read analog voltages (voltages between 0 and 5 volts) using the **analogRead()** function. Is there a way for the RedBoard to output analog voltages as well?

The answer is no... and yes. The RedBoard does not have a true analog voltage output. But, because the RedBoard is so fast, it can fake it using something called **PWM ("Pulse-Width Modulation")**. The pins on the RedBoard with "~" next to them are PWM/Analog out compatible.

The RedBoard is so fast that it can blink a pin on and off almost 1000 times per second. PWM goes one step further by varying the amount of time that the blinking pin spends HIGH vs. the time it spends LOW. If it spends most of its time HIGH, a LED connected to that pin will appear bright. If it spends most of its time LOW, the LED will look dim. Because the pin is blinking much faster than your eye can detect, the RedBoard creates the illusion of a "true" analog output.



3

Arduino Code:

Open Arduino IDE // File > Examples > SIK Guide > Circuit # 3

Code to Note:

```
for (x = 0; x < 768; x++)
{}
```

A for() loop is used to step a number across a range, and repeatedly runs code within the brackets {}. Here the variable "x" starts a 0, ends at 767, and increases by one each time ("x++").

```
if (x <= 255)
{}
else
{}
```

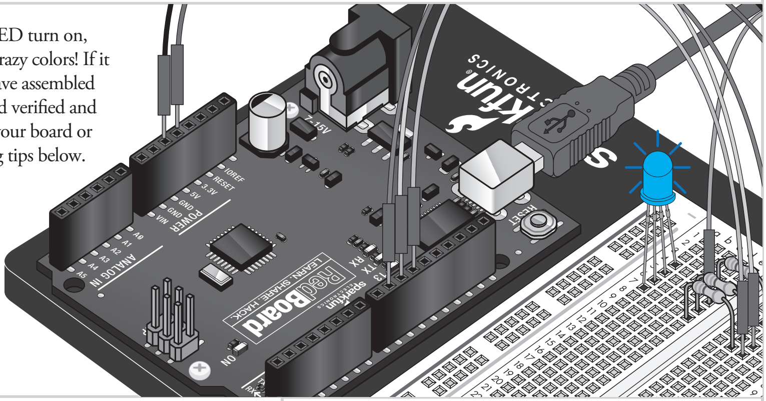
"If / else" statements are used to make choices in your programs. The statement within the parenthesis () is evaluated; if it's true, the code within the first brackets {} will run. If it's not true, the code within the second brackets {} will run.

```
delay(sensorValue);
```

The RedBoard is very very fast, capable of running thousands of lines of code each second. To slow it down so that we can see what it's doing, we'll often insert delays into the code. delay() counts in milliseconds; there are 1000 ms in one second.

What you Should See:

You should see your LED turn on, but this time in new, crazy colors! If it isn't, make sure you have assembled the circuit correctly and verified and uploaded the code to your board or see the troubleshooting tips below.



Troubleshooting:

LED Remains Dark or Shows Incorrect Color
With the four pins of the LED so close together, it's sometimes easy to misplace one. Double check each pin is where it should be.

Seeing Red
The red diode within the RGB LED may be a bit brighter than the other two. To make your colors more balanced, use a higher Ohm resistor. Or adjust in code.

```
analogWrite(RED_PIN, redIntensity);
```

to

```
analogWrite(RED_PIN, redIntensity/3);
```

Real World Application:

Many electronics such as videogame consoles use RGB LEDs to have the versatility to show different colors in the same area. Often times the diffent colors represent different states of working condition.