

# Arduino: RGB LEDs

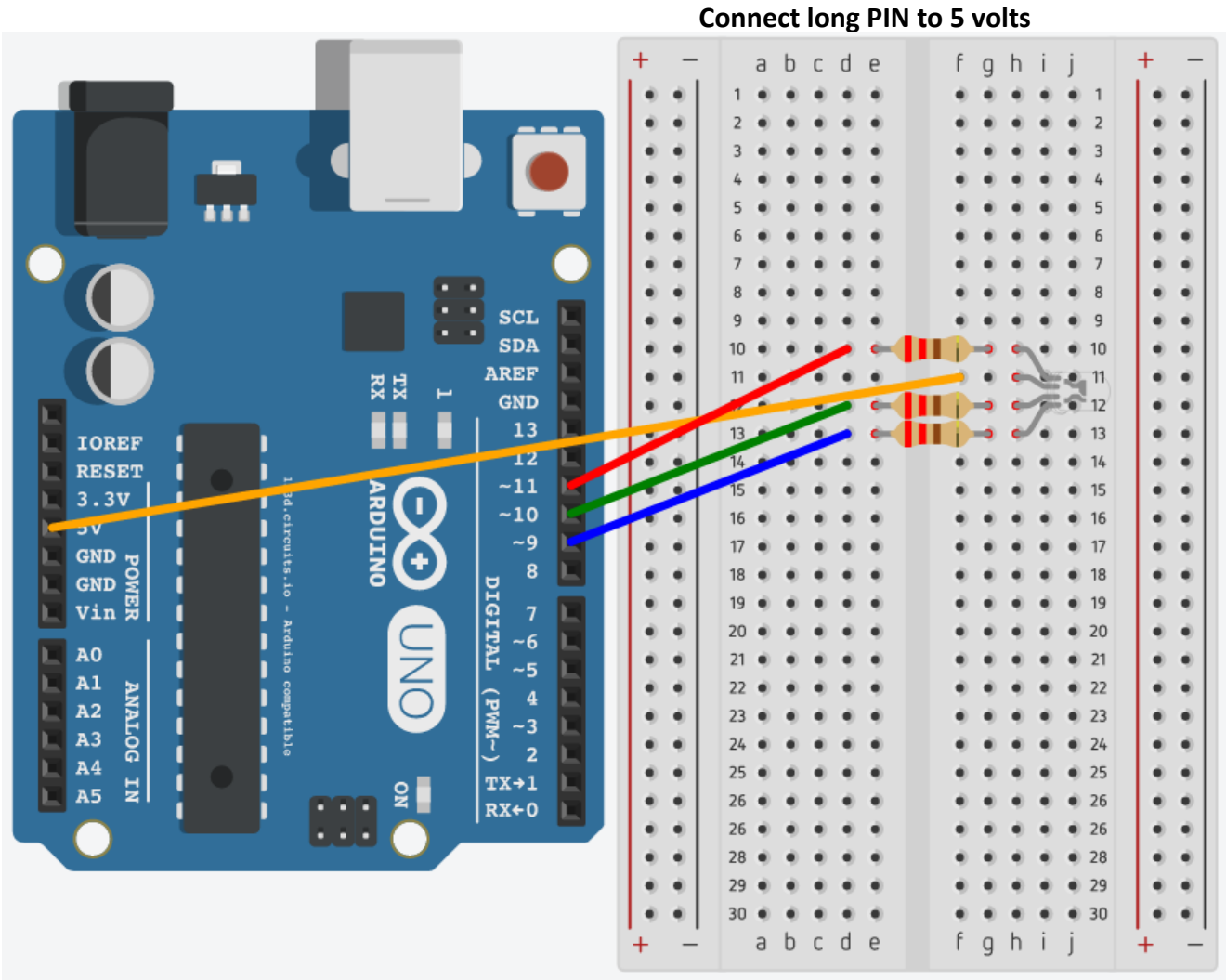
## Diagrams & Code

Brown County Library

### Projects 01 & 02: Blinking RGB LED & Smooth Transition

Components needed:

- Arduino Uno board
- breadboard
- RGB LED (common anode)
- 4 jumper wires
- 3 220 ohm resistors



```
/*  
RGB LED 01 : Blinking RGB LED  
Source: Code modified from Adafruit Arduino - Lesson 3. RGB LED  
*/
```

```
int redPin = 11;  
int greenPin = 10;  
int bluePin = 9;
```

```
void setup() {  
  pinMode(redPin, OUTPUT);  
  pinMode(greenPin, OUTPUT);  
  pinMode(bluePin, OUTPUT);  
}
```

```
void loop() {  
  setColor(255, 0, 0); // red  
  delay(1000);  
  setColor(0, 255, 0); // green  
  delay(1000);  
  setColor(0, 0, 255); // blue  
  delay(1000);  
}
```

```
void setColor(int red, int green, int blue) {  
  // our LEDs consider 255 to be all the way off and 0 to be all the way on  
  // since we're thinking about it the opposite way in our loop, subtract from 255  
  // if you are using a common cathode RGB LED, erase the next 3 lines of code  
  red = 255 - red;  
  green = 255 - green;  
  blue = 255 - blue;  
  
  // set the three pins  
  analogWrite(redPin, red);  
  analogWrite(greenPin, green);  
  analogWrite(bluePin, blue);  
}
```

## Experimenting with more colors

Try adding different colors to your blinking pattern. You can copy and paste the code below into the loop function of Project 01 to create a loop of six blinking colors. Try finding a RGB color picker online (such as this one: [http://www.rapidtables.com/web/color/RGB\\_Color.htm](http://www.rapidtables.com/web/color/RGB_Color.htm)) to create your own colors.

```
setColor(255, 0, 0); // red
delay(1000);
setColor(0, 255, 0); // green
delay(1000);
setColor(0, 0, 255); // blue
delay(1000);
setColor(255, 255, 0); // yellow
delay(1000);
setColor(255, 0, 255); // purple
delay(1000);
setColor(0, 255, 255); // aqua
delay(1000);
```

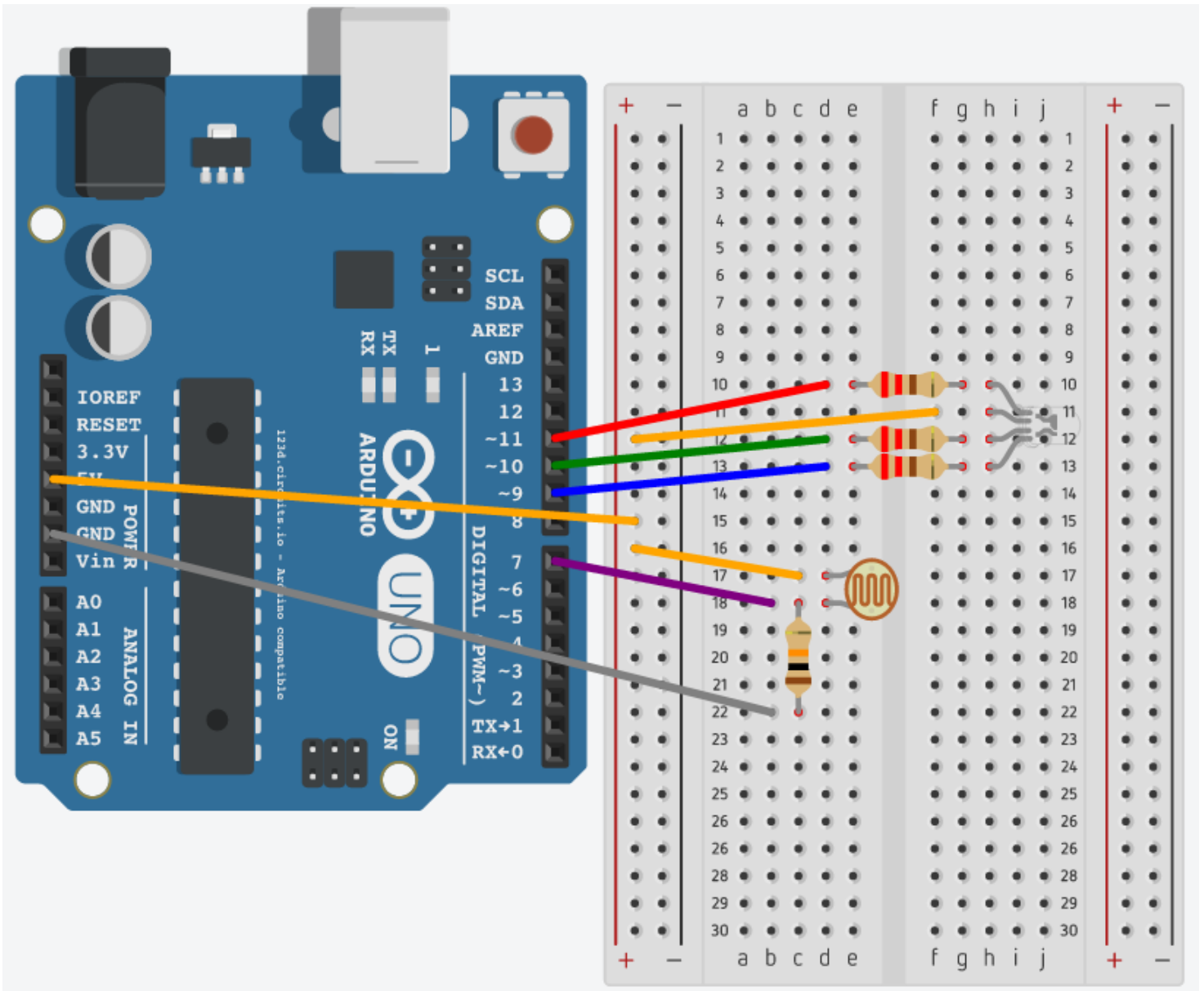
```
/*  
RGB LED 02 : Smooth color transition  
Source: Code adapted from project found here - forum.arduino.cc/index.php?topic=7933.0  
*/
```

```
int pins[] = {11, 10, 9};           // an array of pins! This is similar to int pin0 = 11; int pin1=10; int pin2=9;  
long values[3];                    // make an array of values but don't give them a value just yet  
long current_values[3];            // make an array of current values, but don't give them a value yet  
  
int short_delay;                   // time between transition  
  
void setup(){  
    randomSeed(analogRead(0));      // get some unpredictable value to start off our random number generator  
                                     // otherwise, we'd get the same random numbers each time (boring!)  
  
    for (int i=0; i <3; i++) {      // pins 0 to less than 3  
        values[i] = random(255);     // pick a random number between 0 and 255 for a pin (red, green, or blue)  
        current_values[i] = values[i]; // make our current value the same  
        analogWrite(pins[i], current_values[i]); // set the LED to our initial choice  
        values[i] = random(255);     // pick a new random number for our next color  
    }  
}  
  
void loop(){  
    short_delay = random(3, 9);  
  
    for (int i=0; i <3; i++) {  
        if (values[i] > current_values[i]){ // if our new color is a larger number than our current color ...  
            current_values[i]++;           // get just a little bit closer to the new color by adding 1  
            analogWrite(pins[i], current_values[i]); // set the LED to the new current color  
            delay(short_delay);            // wait a little bit  
        }  
  
        if (values[i] < current_values[i]){ // if our new color is a smaller number than our current color ...  
            current_values[i]--;           // get just a little bit closer to the new color by subtracting 1  
            analogWrite(pins[i], current_values[i]); // set the LED to the new current color  
            delay(short_delay);            // wait a little bit  
        }  
  
        if (values[i] == current_values[i]){ // if our new color and our current color are the same ...  
            analogWrite(pins[i], current_values[i]); // make sure the LED is set to the new color  
            values[i] = random(255);         // pick a new color  
        }  
    }  
}
```

## Projects 03: Mood Light

Components needed:

- Arduino Uno board
- breadboard
- RGB LED (common anode)
- 8 jumper wires
- 3 220 ohm resistors
- 10k ohm resistor
- light dependent resistor (sometimes called a *photoresistor*)



```
/*  
RGB LED 03 : Mood light with photoresistor  
Source: Code adapted from project found here - forum.arduino.cc/index.php?topic=7933.0  
*/
```

```
int pins[] = {11, 10, 9};      // an array of pins! This is similar to int pin0 = 11; int pin1=10; int pin2=9;  
long values[3];              // make an array of values but don't give them a value just yet  
long current_values[3];      // make an array of current values, but don't give them a value yet  
  
const int sensor = 7;        // the input pin where the sensor (photoresistor) is connected  

```

```

}

if (values[i] == current_values[i]){          // if our new color and our current color are the same ...
  analogWrite(pins[i], current_values[i]);    // make sure the LED is set to the new color
  values[i] = random(255);                    // pick a new color
}
}

} else {                                     // or if the input is HIGH (there is light)
  digitalWrite(11, HIGH);                    // set the three LEDs to HIGH (common anode RGB LEDs consider HIGH to be all
  // the way off and LOW to be all the way on)
  digitalWrite(10, HIGH);                    // so since we're thinking about it in the opposite way, setting the LEDs to HIGH
  // turns them off
  digitalWrite(9, HIGH);                     // if you are using a common cathode RGB LED, change these three lines to say
  // LOW
}
}
}

```

## Bonus Project: Using Hexadecimal Colors instead of RGB

You can also insert hexadecimal colors into your code instead of using RGB. Use a color picker (such as this one: [http://www.rapidtables.com/web/color/RGB\\_Color.htm](http://www.rapidtables.com/web/color/RGB_Color.htm)) and instead use the combination of numbers and letters (for example, the color “sienna” is A0522D). Look at the line of code for sienna below to see an example of how to plug it in. The Arduino knows that it is a hex number because of the “0x” in front of the letters and numbers, so make sure to leave that in!

```
/*
```

### RGB LED 01 with Hex Colors

Source: Code modified from Adafruit Arduino - Lesson 3. RGB LED

```
*/
```

```
int redPin = 11;
```

```
int greenPin = 10;
```

```
int bluePin = 9;
```

```
void setup() {
```

```
  pinMode(redPin, OUTPUT);
```

```
  pinMode(greenPin, OUTPUT);
```

```
  pinMode(bluePin, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  setColor(0x4B, 0x0, 0x82); // indigo
```

```
  delay(1000);
```

```
  setColor(0xA0, 0x52, 0x2D); // sienna
```

```
  delay(1000);
```

```
}
```

```
void setColor(int red, int green, int blue) {
```

```
  // our LEDs consider 255 to be all the way off and 0 to be all the way on
```

```
  // since we're thinking about it the opposite way in our loop, subtract from 255
```

```
  // if you are using a common cathode RGB LED, erase the next 3 lines of code
```

```
  red = 255 - red;
```

```
  green = 255 - green;
```

```
  blue = 255 - blue;
```

```
  // set the three pins
```

```
  analogWrite(redPin, red);
```

```
  analogWrite(greenPin, green);
```

```
  analogWrite(bluePin, blue);
```

```
}
```

## Learn More!

### Adafruit Arduino Lesson 3. RGB LEDs

<https://learn.adafruit.com/downloads/pdf/adafruit-arduino-lesson-3-rgb-leds.pdf>