Arduino: Potentiometer
Diagrams & Code

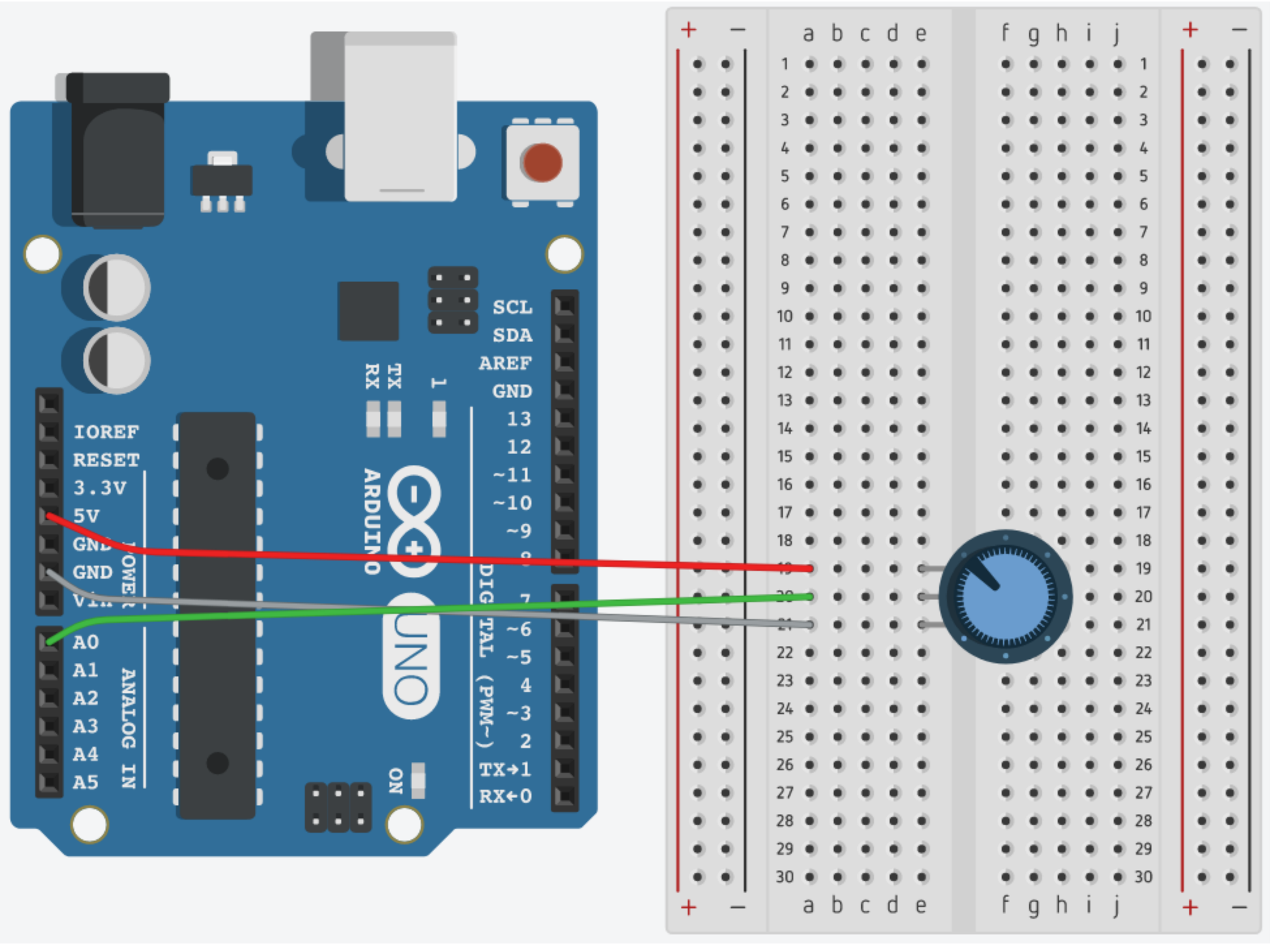
Brown County Library

Some projects require the use of the serial monitor in your Arduino IDE program (or whatever you are using to transfer code to the Arduino).

Projects 01 & 02: Reading a Potentiometer and Changing a Potentiometer Reading to a Percentage

Components needed:

- Arduino Uno board
- breadboard
- 3 jumper wires
- 10k potentiometer



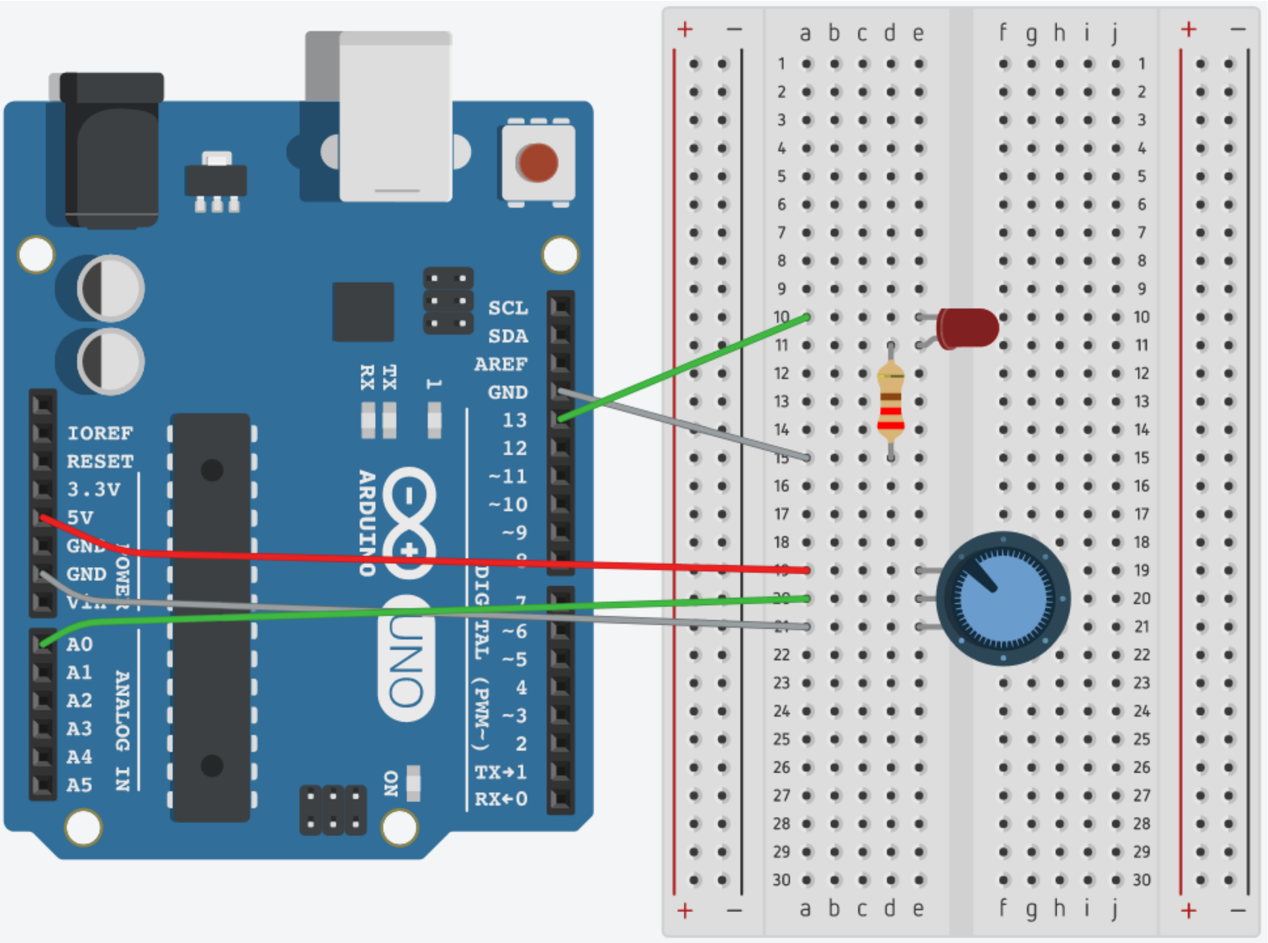
```
/*
Potentiometers 01 : Reading a Potentiometer
Source: Code adapted from Arduino.cc AnalogReadSerial
        (https://www.arduino.cc/en/Tutorial/AnalogReadSerial)
*/

void setup() {
  Serial.begin(9600);  // initialize the serial communication
  // Note: analog pins are automatically set as inputs
}

void loop() {
  int potValue = analogRead(A0);      // get a reading from the potentiometer on A0
  Serial.println(potValue);           // print out the value you read
  delay(100);                         // a delay makes values easier to read
}
```

```
/*
Potentiometers 02 : Changing Potentiometer Reading to a Percentage
Source: Code adapted from Jeremy Blum's Exploring Arduino
(http://www.exploringarduino.com/content/ch6/)
*/

int potPin = A0;   // potentiometer is connected to analog 0 pin
int potValue;   // variable used to store the value coming from the sensor
int percent;     // variable used to store the percentage value

void setup() {
  Serial.begin(9600);  // initialize the serial communication
  // Note: analog pins are automatically set as inputs
}

void loop() {
  potValue = analogRead(potPin);       // get a reading from the potentiometer, assign the name potValue
  percent = map(potValue, 0, 1023, 0, 100);   // convert reading to a percentage

  Serial.print("Analog Reading: ");       // print out the potentiometer reading
  Serial.print(potValue);
  Serial.print(",  Percentage: ");          // print out the percentage
  Serial.print(percent);
  Serial.println("%");

  delay(1000);              // wait a second
}
```

Project 03: Blink LED Based on Potentiometer Reading

Components needed:

- Arduino Uno board
- breadboard
- 5 jumper wires
- 10k potentiometer
- 220 ohm resistor
- LED

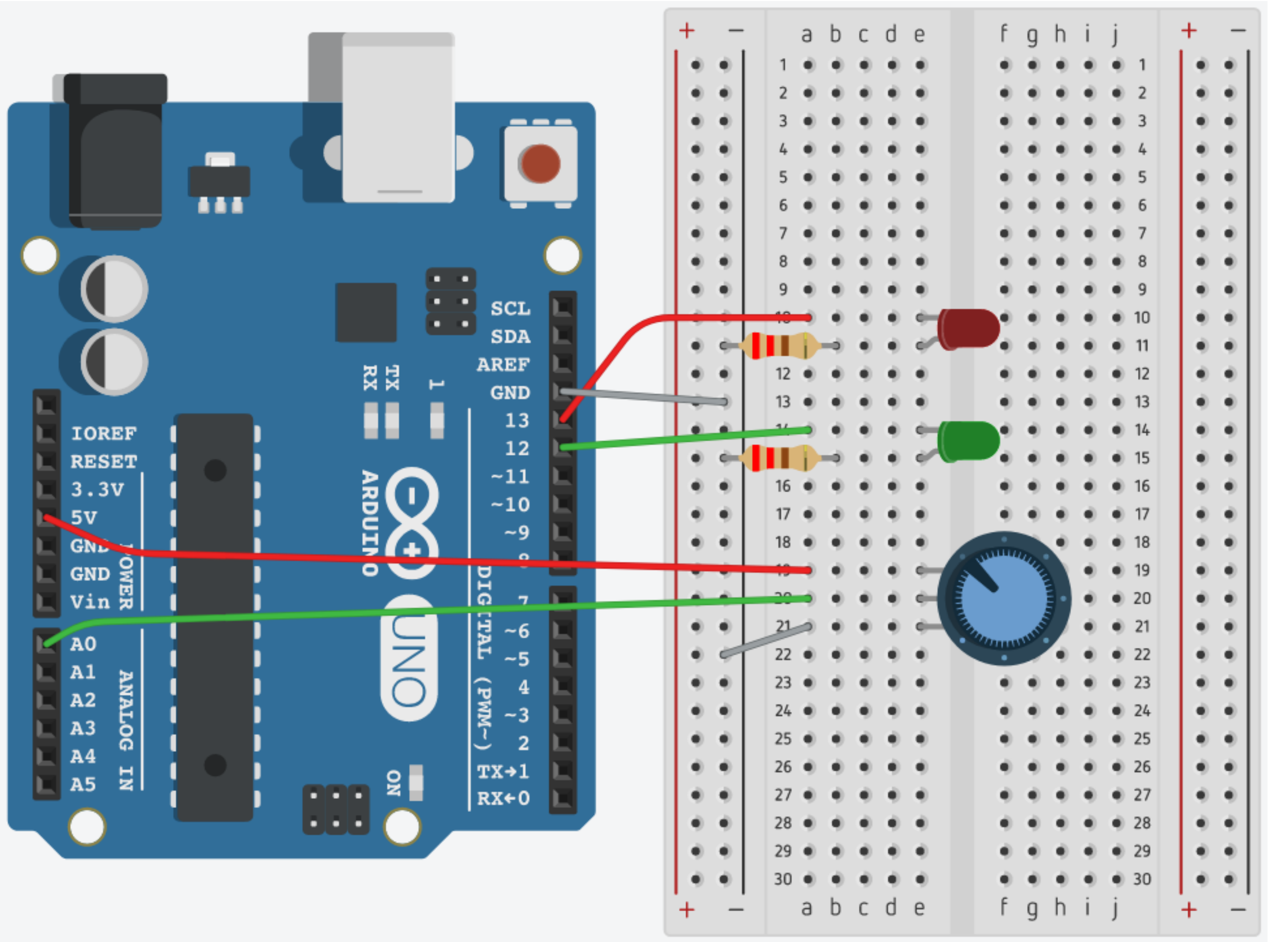


```
/*
Potentiometers 03 : Blink LED Based on Potentiometer Reading
Source: Code adapted from SparkFun Inventor's Kit Example Sketch 2
(https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-2-reading-a-potentiometer)
*/

int potPin = A0;          // potentiometer is connected to analog pin 0
int ledPin = 13;          // LED connected to digital PIN 13
int potValue;             // variable used to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT);  // LED is as an output
  Serial.begin(9600);       // initialize the serial communication
  // Note: analog pins are automatically set as inputs
}

void loop() {
  potValue = analogRead(potPin);    // read the value from the sensor and assign the name potValue
  Serial.println(potValue);         // print out the value you read

  digitalWrite(ledPin, HIGH);  // turn the LED on
  delay(potValue);             // pause for sensorValue in milliseconds
  digitalWrite(ledPin, LOW);   // turn the LED off
  delay(potValue);             // pause for sensorValue in milliseconds
}
```

Project 04: Control Two LEDs with a Potentiometer

Components needed:

- Arduino Uno board
- breadboard
- 6 jumper wires
- 10k potentiometer
- 2 x 220 ohm resistors
- 2 x LEDs (two different colors, if possible)

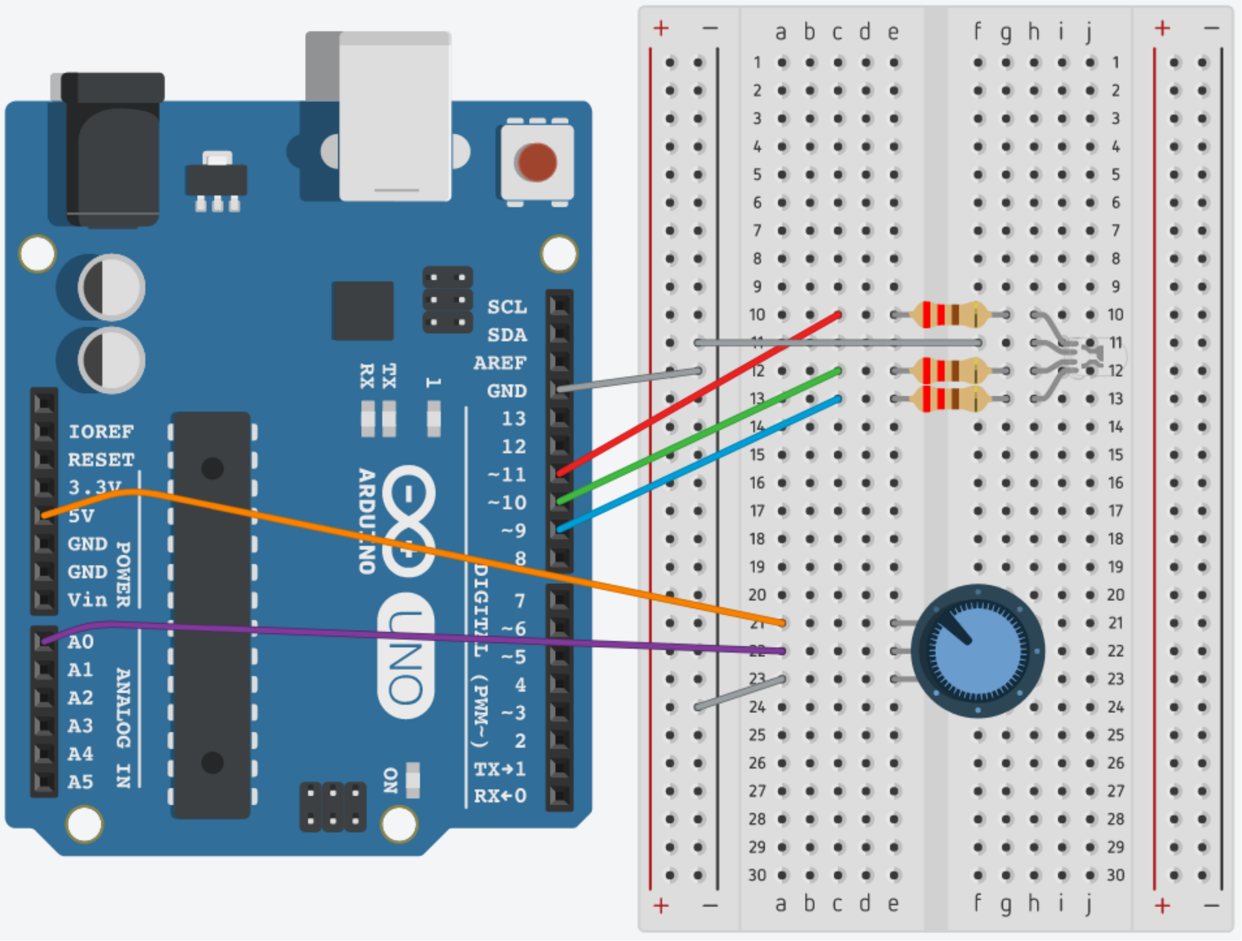


```
/*
Potentiometers 04 : Control Two LEDs with a Potentiometer
Source: Code adapted from Jeremy Blum's Exploring Arduino
(http://www.exploringarduino.com/content/ch6/)
and "Getting Started with Arduino" by Banzi/Shiloh (3rd ed.)
*/

int potPin = A0;          // potentiometer is connected to analog 0 pin
int redPin = 13;          // red LED connected to digital PIN 13
int greenPin = 12;        // red LED connected to digital PIN 12
int potValue;             // variable used to store the value coming from the sensor
int percent;              // variable used to store the percentage value

void setup() {
  pinMode(redPin, OUTPUT);              // red LED is as an output
  pinMode(greenPin, OUTPUT);            // green LED is as an output
  // Note: analog pins are automatically set as inputs
}

void loop() {
  potValue = analogRead(potPin);        // read the value from the potentiometer and assign the name potValue
  percent = map(potValue, 0, 1023, 0, 100);   // convert potentiometer reading to a percentage

  if (percent < 50) {                   //if the percentage is less than 50%...
    digitalWrite(redPin, HIGH);         // turn the red LED on
    digitalWrite(greenPin, LOW);        // turn the green LED off

  } else {                              //or else if it is more than 50%
    digitalWrite(redPin, LOW);          // turn the red LED off
    digitalWrite(greenPin, HIGH);       // turn the green LED on
  }
}
```

Project 05: Control a RGB LED with a Potentiometer

Components needed:

- Arduino Uno board
- breadboard
- 8 jumper wires
- 10k potentiometer
- 3 x 220 ohm resistors
- RGB LED (common cathode)



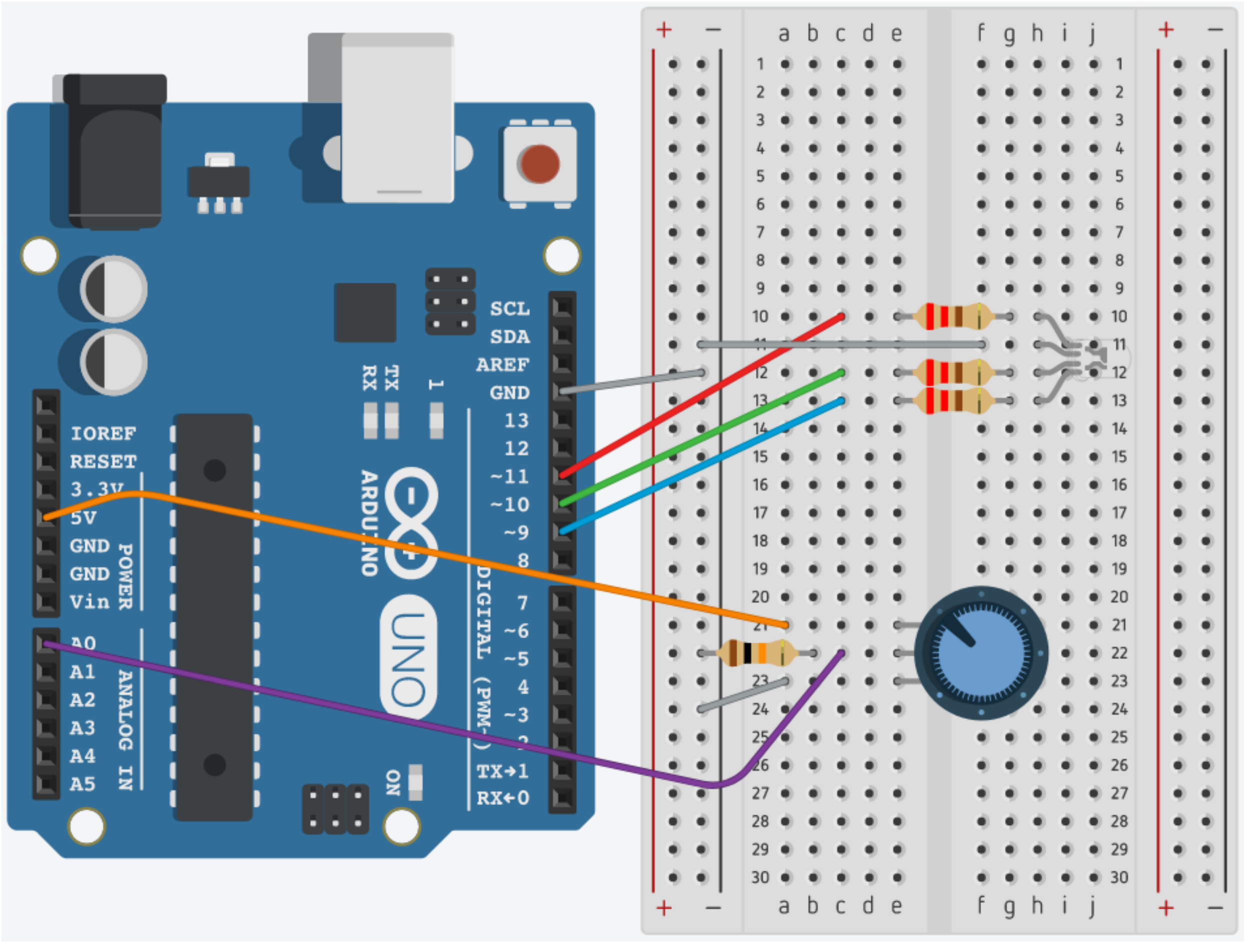
```
/*
Potentiometers 05 : Control a RGB LED with a Potentiometer
Source: Code adapted from SparkFun Inventor's Kit Example Sketch 10
(https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-10-reading-a-soft-potentiometer)
*/

int potPin = A0;          // potentiometer is connected to analog 0 pin
int redPin = 11;          // red pin is connected to 11
int greenPin = 10;        // green pin is connected to 10
int bluePin = 9;          // blue pin is connected to 9
int potValue;             // variable used to store the value coming from the sensor

// Global PWM brightness values for the RGB LED.
// These are global so both loop() and setRGB() can see them.
int redValue, greenValue, blueValue;


void setup() {
  // No need for any code here
  // Note: analog pins are automatically set as inputs
}

void loop() {
  potValue = analogRead(potPin);          // read the value from the potentiometer and assign the name potValue

  setRGB(potValue);      //Set a RGB LED to a position on the "rainbow" of all colors based on the potValue
}

void setRGB(int RGBposition) {            // a new function to make the "rainbow" of colors possible
  int mapRGB1, mapRGB2, constrained1, constrained2;   // define varibles that we need in this function

  mapRGB1 = map(RGBposition, 0, 341, 255, 0);  // the function maps each potentiometer value to a specifc color
  constrained1 = constrain(mapRGB1, 0, 255);   // combination of the three RGB lights

  mapRGB2 = map(RGBposition, 682, 1023, 0, 255);
  constrained2 = constrain(mapRGB2, 0, 255);

  redValue = constrained1 + constrained2;      //Create the red peak

  greenValue = constrain(map(RGBposition, 0, 341, 0, 255), 0, 255)         //Create the green peak
          - constrain(map(RGBposition, 341, 682, 0,255), 0, 255);

  blueValue = constrain(map(RGBposition, 341, 682, 0, 255), 0, 255)        //Create the blue peak
         - constrain(map(RGBposition, 682, 1023, 0, 255), 0, 255);

  analogWrite(redPin, redValue);            // Display the new computed "rainbow" color
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}
```

Project 05a: Control a RGB LED with a Soft Potentiometer

Components needed:

- Arduino Uno board
- breadboard
- 8 jumper wires
- Soft potentiometer (example)
- 3 x 220 ohm resistors
- 1 x 10k ohm resistor
- RGB LED (common cathode)

Use the same code from Project 05, but adjust your setup as follows. Then replace the regular potentiometer with a soft potentiometer (plugging the legs into the same lines on the breadboard).



Ideas to Build On

Learn more about how the soft potentiometer works by running some simple code to see a graph of where you finger is on the slider:
https://learn.sparkfun.com/tutorials/softpot-hookup-guide#example-circuit

Adjust the code from Project 4:

- Add a yellow LED to create a stoplight
- What if you wanted both lights to be on for some of the time (for example, when the reading is less than 350 the red is on, between 350 & 650 both lights are on, and above 650 just the green is on)
- **Hint:** you have to add an additional else if statement (check out the code for Serial Monitor Project 3 for reference).

Control the three RGB colors with three potentiometers:

- https://create.arduino.cc/projecthub/FIELDING/arduino-rgb-color-mixer-b81863
- A slightly more complex version: https://create.arduino.cc/projecthub/shakram02/arduino-color-mixer-d6264a

Learn More

Want to learn more about how potentiometers work? Try these resources:

Arduino – Analog Read Serial.
https://www.arduino.cc/en/Tutorial/AnalogReadSerial

Random Nerd Tutorials: How a Potentiometer Works.
https://randomnerdtutorials.com/electronics-basics-how-a-potentiometer-works/

Resistor Guide: Potentiometer.
http://www.resistorguide.com/potentiometer/

Sparkfun SIK Experiment Guide for Arduino V4.0 – Circuit 1B: Potentiometer.
https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/circuit-1b-potentiometer

Sparkfun SIK Experiment Guide for Arduino V3.3 – Experiment 10: Reading a Soft Potentiometer.
https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-10-reading-a-soft-potentiometer

Sparkfun Tutorials: Analog to Digital Conversion.
https://learn.sparkfun.com/tutorials/analog-to-digital-conversion

Sparkfun Tutorials: SoftPot Hookup Guide.
https://learn.sparkfun.com/tutorials/softpot-hookup-guide

Sparkfun Tutorials: Voltage Dividers.
https://learn.sparkfun.com/tutorials/voltage-dividers