Arduino: Serial Monitor
Diagrams & Code

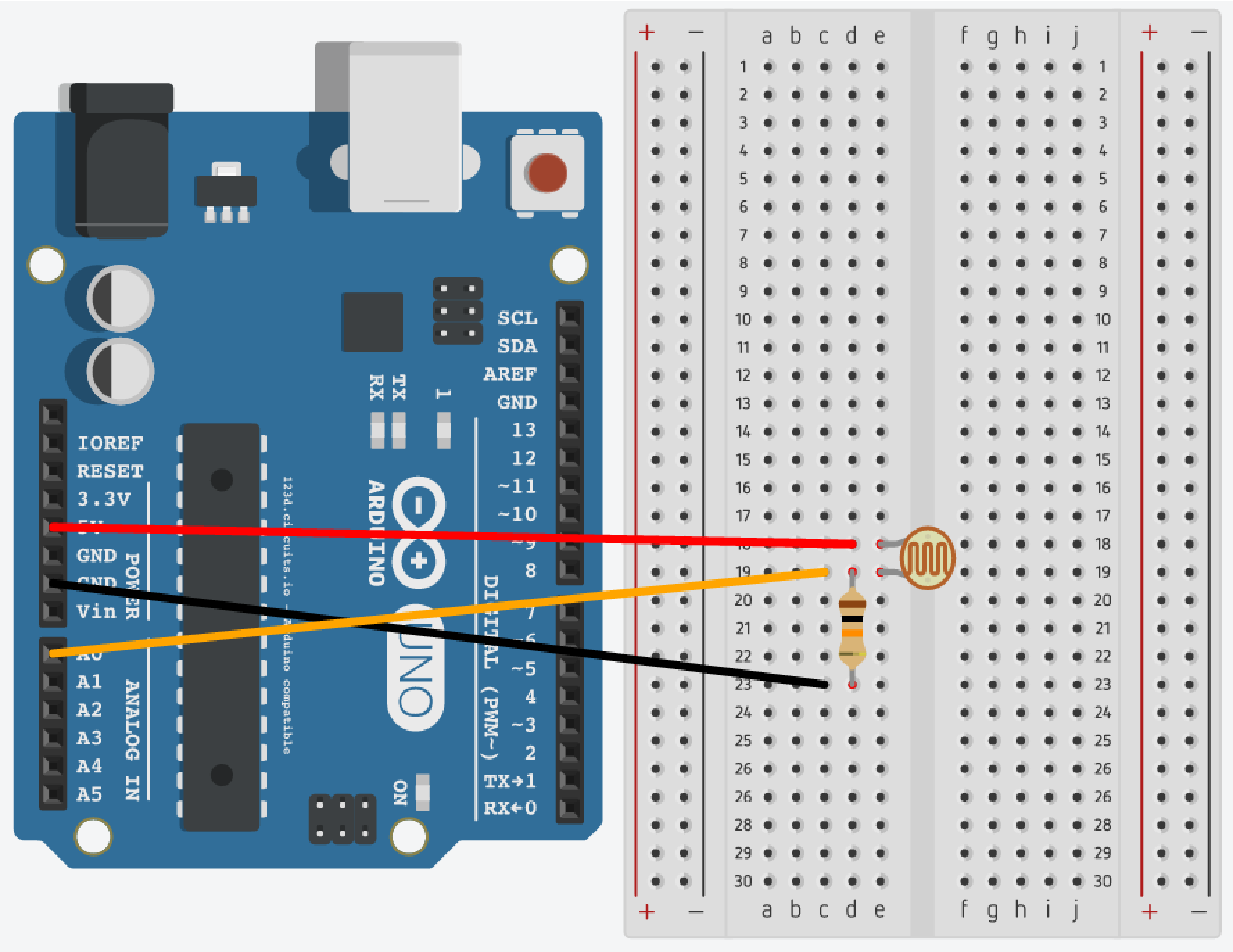
Brown County Library

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

Project 01: Monitor how much light is hitting a photoresistor

Components needed:

- Arduino Uno board
- breadboard
- photoresistor (light dependent resistor)
- 3 jumper wires
- 10k ohm resistor



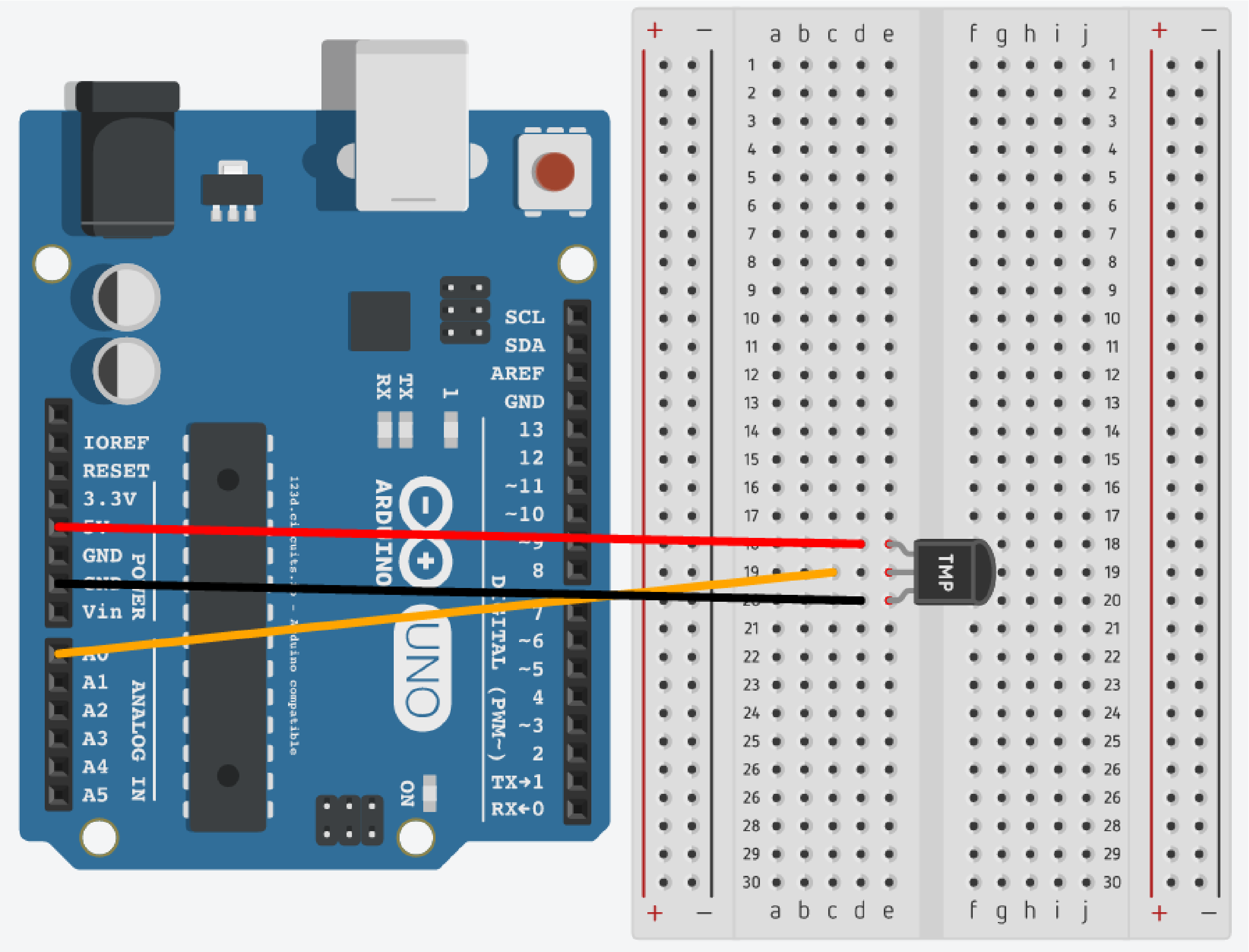
```
/*
Serial Monitor 01 : Monitor how much light is hitting a photoresistor
*/

const int sensorPin = A0;   // photoresistor (sensor) is connected to A0
int sensorVal = 0;     // variable used to store the value coming from the sensor

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

void loop() {
  sensorVal = analogRead(sensorPin);   // read the value from the sensor

  Serial.print("Sensor Value: ");
  Serial.println(sensorVal);

  delay(1000);
}
```

Project 02: Monitor the temperature

Components needed:

- Arduino Uno board
- breadboard
- TMP 36 temperature sensor
- 3 jumper wires

Warning: Temperature sensors can heat up quickly if they are not hooked up correctly! In the diagram below, the flat side of the temperature sensor should be facing the Arduino. For more information, go here: https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/circuit-4b-temperature-sensor



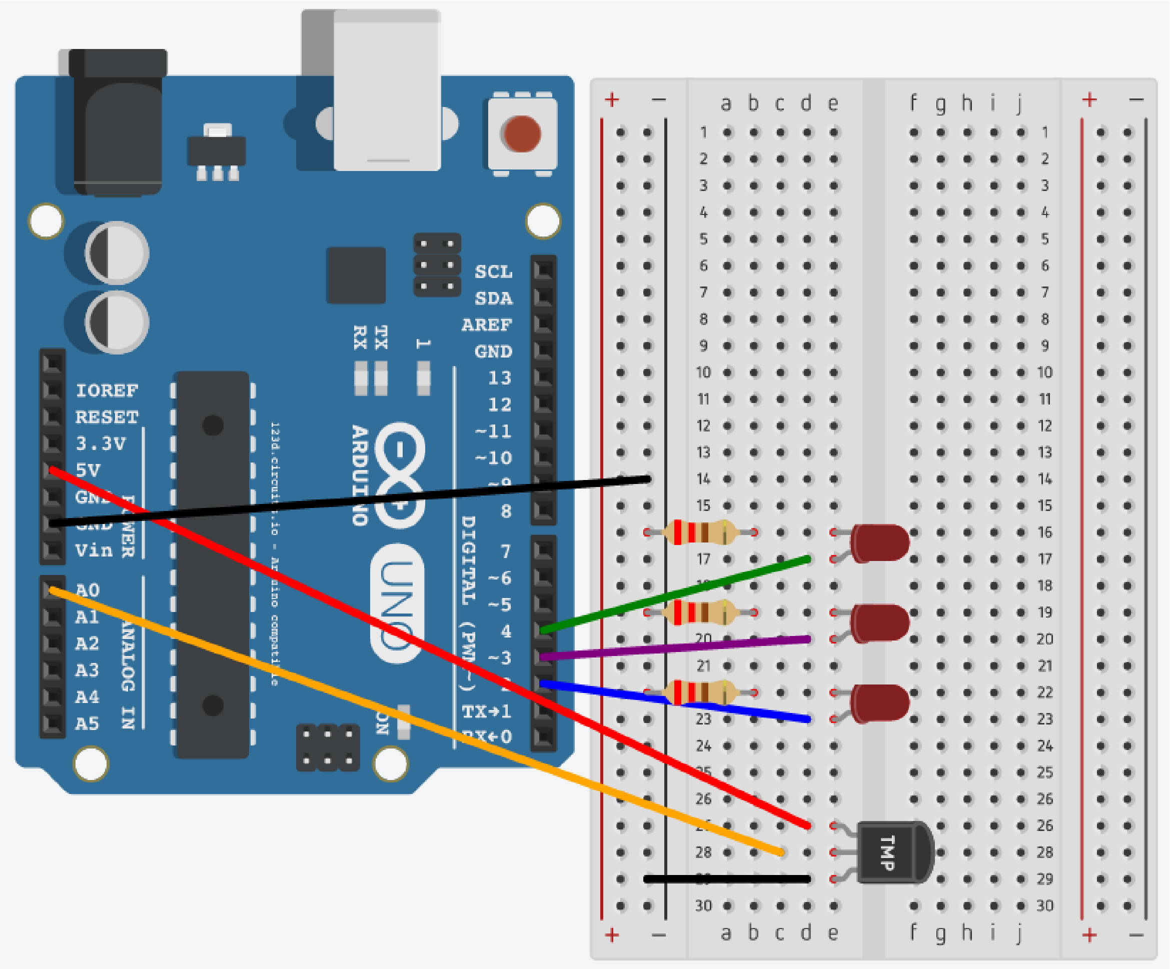
```
/*
Serial Monitor 02 : Monitor the temperature
Code partially adapted from the Arduino Projects Book (2012)
*/

const int sensorPin = A0;   // temperature sensor is connected to A0
int sensorVal = 0;     // variable used to store the value coming from the sensor

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

void loop() {
  sensorVal = analogRead(sensorPin);   // read the value from the sensor

  float voltage = (sensorVal/1024.0) * 5.0;  // convert the sensor reading to voltage
  float temperatureC = (voltage - .5) * 100;  // convert the voltage to temperature in Celsius
  float temperatureF = temperatureC * (9.0/5.0) + 32.0;  // convert the temperature in Celsius to Fahrenheit

  // send the readings to the serial monitor
  Serial.print("Sensor Value: ");  // send the sensor value to the serial monitor
  Serial.print(sensorVal);

  Serial.print(", Volts: ");  // send the voltage to the serial monitor
  Serial.print(voltage);

  Serial.print(", degrees C: ");  // send the Celsius reading to the serial monitor
  Serial.print(temperatureC);

  Serial.print(", degrees F: ");  // send the Fahrenheit reading to the serial monitor
  Serial.println(temperatureF);

delay(1000);
}
```

Project 03: Lighting up LEDs based on the temperature

Components needed:

- Arduino Uno board
- breadboard
- TMP 36 temperature sensor
- 7 jumper wires
- 3 x 220 ohm resistors
- 3 x LEDs

Warning: Temperature sensors can heat up quickly if they are not hooked up correctly! In the diagram below, the flat side of the temperature sensor should be facing the Arduino. For more information, go here: https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/circuit-4b-temperature-sensor



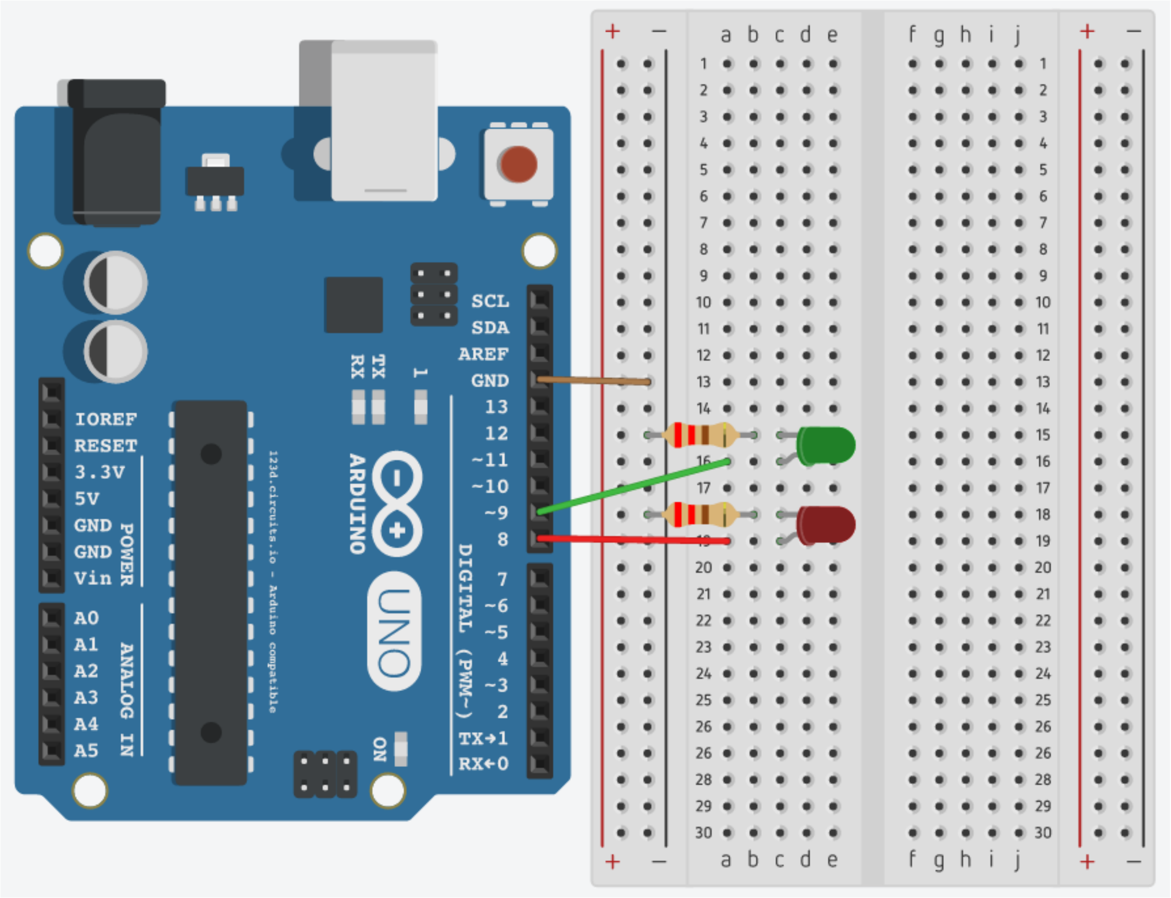
```
/*
Serial Monitor 03 : Lighting up LEDs based on the temperature
Code partially adapted from the Arduino Projects Book (2012)
*/

const int sensorPin = A0;   // temperature sensor is connected to A0
int sensorVal = 0;     // variable used to store the value coming from the sensor
const float baselineTemp = 68.0;  // baseline temperature (all LEDs are off at this temp)

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

  for(int pinNumber = 2; pinNumber<5; pinNumber++){  // pinNumber 2 and above but less than 5...
    pinMode(pinNumber,OUTPUT);                     // LEDs are set as outputs and...
    digitalWrite(pinNumber, LOW);                 // are turned off
  }
}

void loop() {
  sensorVal = analogRead(sensorPin);   // read the value from the sensor

  float voltage = (sensorVal/1024.0) * 5.0;  // convert the sensor reading to voltage
  float temperatureC = (voltage - .5) * 100;  // convert the voltage to temperature in Celsius
  float temperatureF = temperatureC * (9.0/5.0) + 32.0;  // convert the temperature in Celsius to Fahrenheit

  // send the readings to the serial monitor
  Serial.print("Sensor Value: ");  // send the sensor value to the serial monitor
  Serial.print(sensorVal);

  Serial.print(", Volts: ");  // send the voltage to the serial monitor
  Serial.print(voltage);

  Serial.print(", degrees C: ");  // send the Celsius reading to the serial monitor
  Serial.print(temperatureC);

  Serial.print(", degrees F: ");  // send the Fahrenheit reading to the serial monitor
  Serial.println(temperatureF);

  if(temperatureF < baselineTemp){  // if the temperature is below the baseline temp...
    digitalWrite(2, LOW);         // all the LEDs are off
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);

  }else if(temperatureF >= baselineTemp+1 &&  // if the temperature is more than 1 degree F but...
    temperatureF < baselineTemp+2){          // less than 2 degrees above the baseline temp...
    digitalWrite(2, HIGH);                 // the first LED is on and the others are off
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }else if(temperatureF >= baselineTemp+2 &&  // if the temperature is more than 2 degrees F but...
    temperatureF < baselineTemp+3){          // less than 3 degrees above the baseline temp...
    digitalWrite(2, HIGH);                    // the first 2 LEDs are on and the last is off
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);

  }else if(temperatureF >= baselineTemp+3){  // if the temperature is more than 3 degrees F above the baseline temp...
    digitalWrite(2, HIGH);                    // all of the LEDs are on
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }

  delay(1000);
}
```

Project 04: Receive data from a computer

Components needed:

- Arduino Uno board
- breadboard
- 3 jumper wires
- 2 x 220 ohm resistors
- 2 x LEDs

Once you complete the setup and transfer the code, open the serial monitor and enter any number between one and four.



```
/*
Serial Monitor 04 : Receive data or a command from a computer
Code partially adapted from Physical Pixel Tutorial (arduino.cc/en/Tutorial/PhysicalPixel)
and Adafruit Arduino Lesson 5 (learn.adafruit.com/adafruit-arduino-lesson-5-the-serial-
monitor)
*/

int redPin = 8;     // red LED is connected to pin 8
int greenPin = 9;  // green LED is connected to pin 9
int incomingByte;   // a variable to read incoming serial data into

void setup() {
  Serial.begin(9600);        // initialize the serial communication
  pinMode(redPin, OUTPUT);    // red pin is an output
  pinMode(greenPin, OUTPUT);  // green pin is an output
}

void loop() {
  // Check for incoming serial data
  if (Serial.available() > 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();

    // Turn on the red LED:
    if (incomingByte == '1') {
      digitalWrite(redPin, HIGH);
      Serial.println("Red turned on");
    }

    // Turn off the red LED:
    if (incomingByte == '2') {
      digitalWrite(redPin, LOW);
      Serial.println("Red turned off");
    }

    // Turn on the green LED:
    if (incomingByte == '3') {
      digitalWrite(greenPin, HIGH);
      Serial.println("Green turned on");
    }

    // Turn off the green LED:
    if (incomingByte == '4') {
      digitalWrite(greenPin, LOW);
      Serial.println("Green turned off");
    }
  }
}
```

Project 05: Control a RGB LED with the serial monitor

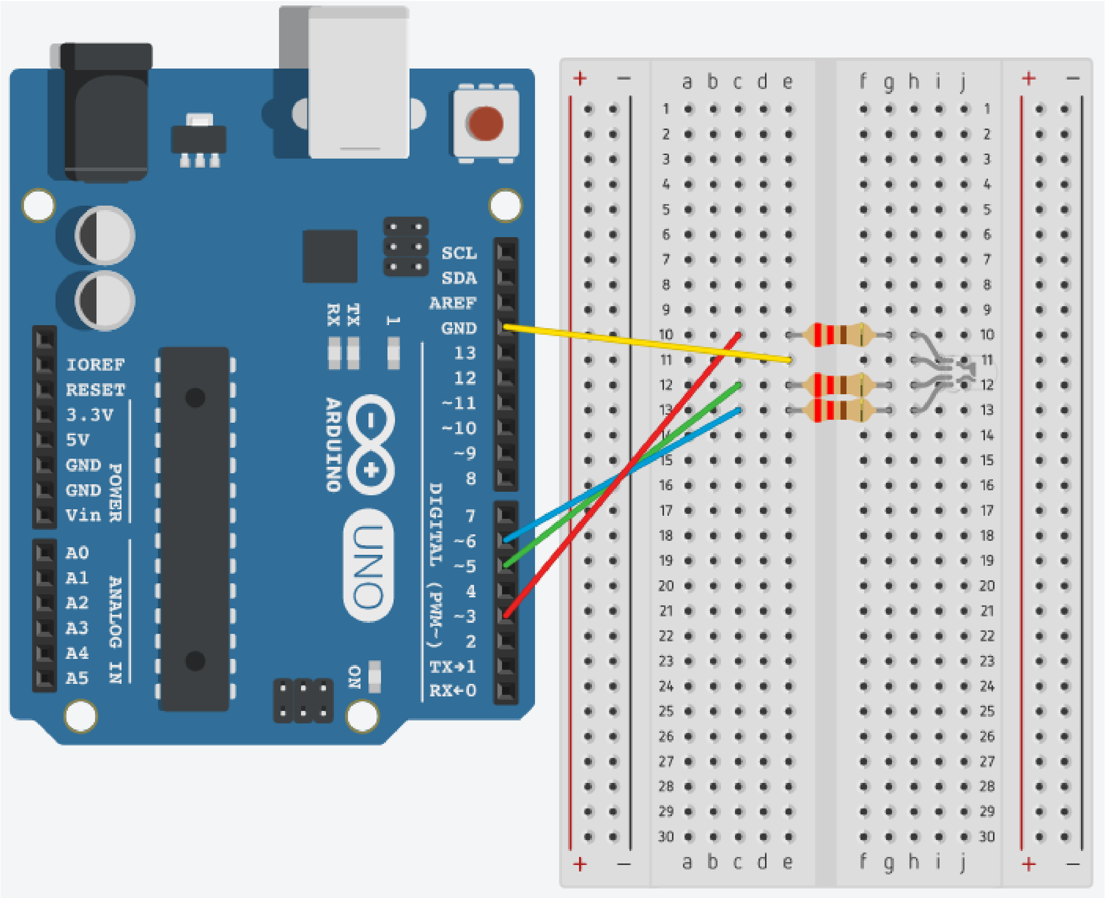
Components needed:

- Arduino Uno board
- breadboard
- 4 jumper wires
- 3 x 220 ohm resistors
- Common cathode RGB LED
    - If you have a common anode RGB LED, connect the long pin to 5V instead of ground, and use the code found on page 12

Once you complete the setup and transfer the code, open the serial monitor and enter RGB color codes to set the LED to a specific color. For example, to turn on just the red light, enter: 255,0,0

Depending on your serial monitor, you may need to choose "new line" from a dropdown menu.

Remember that some colors show up better than others. Consult a RGB Color Picker for color combinations, such as: webpagefx.com/web-design/color-picker/



```
/*
Serial Monitor 05 : Control a RGB LED with the serial monitor
Source: Code modified from Arduino.cc Read ASCII Tutorial
(arduino.cc/en/Tutorial/ReadASCIIString)
For use with a common cathode RGB LED
*/

const int redPin = 3;    // red pin is connected to digital pin 3
const int greenPin = 5;  // green pin is connected to digital pin 5
const int bluePin = 6;    // blue pin is connected to digital pin 6

void setup()
{
  Serial.begin(9600); // initialize the serial communication

  // set pins as outputs
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop()
{
  while (Serial.available() > 0)  // check for incoming serial data
  {
    // look for 3 valid integers (numbers) separated by commas - this will set the RGB values (ex. R,G,B)
    int red = Serial.parseInt();  // first valid integer
    int green = Serial.parseInt();  // second valid integer
    int blue = Serial.parseInt();  // third valid integer

    if (Serial.read() == '\n') // once the transmission is finished:
    {
      // this constrains the numbers so they are between 0 and 255
      red = constrain(red, 0, 255);
      green = constrain(green, 0, 255);
      blue = constrain(blue, 0, 255);

      // then set the color of the RGB LED
      analogWrite(redPin, red);
      analogWrite(greenPin, green);
      analogWrite(bluePin, blue);
    }
  }
}
```

```
/*
Serial Monitor 05 : Control a RGB LED with the serial monitor
Source: Code modified from Arduino.cc Read ASCII Tutorial
(arduino.cc/en/Tutorial/ReadASCIIString)
For use with a common anode RGB LED
*/

const int redPin = 3;    // red pin is connected to digital pin 3
const int greenPin = 5;  // green pin is connected to digital pin 5
const int bluePin = 6;    // blue pin is connected to digital pin 6

void setup()
{
  Serial.begin(9600); // initialize the serial communication

  // set pins as outputs
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop()
{
  while (Serial.available() > 0)  // check for incoming serial data
  {
    // look for 3 valid integers (numbers) separated by commas - this will set the RGB values (ex. R,G,B)
    int red = Serial.parseInt();  // first valid integer
    int green = Serial.parseInt();  // second valid integer
    int blue = Serial.parseInt();  // third valid integer

    if (Serial.read() == '\n') // once the transmission is finished:
    {
      // first we do a math problem so the values will make sense with a common anode RGB LED
      // this constrains the numbers so they are between 0 and 255 and inverts them
      red = 255 - constrain(red, 0, 255);
      green = 255 - constrain(green, 0, 255);
      blue = 255 - constrain(blue, 0, 255);

      // then set the color of the RGB LED
      analogWrite(redPin, red);
      analogWrite(greenPin, green);
      analogWrite(bluePin, blue);
    }
  }
}
```

Ideas to Build On

What happens when the baud rates don't match between the code and the serial monitor?

Create a baseball scoreboard:
Using three LEDs, create a display of the current number of balls that the batter has in a baseball game, or with three greens and two reds, you could do balls and strikes.

Create a temperature alert:
Use a RGB LED with a temperature sensor to create a visual alert when a room gets too hot or too cold. Hint: You'll want to combine the Serial Monitor 03 setup/code with the RGB LED 01 setup/code.

Learn More

Want to learn more about how the serial monitor works? Try these resources:

Adafruit Arduino Lesson 5. Serial Monitor
https://learn.adafruit.com/adafruit-arduino-lesson-5-the-serial-monitor

Arduino.cc Physical Pixel Tutorial:
https://www.arduino.cc/en/Tutorial/PhysicalPixel

Arduino.cc Read ASCII Tutorial:
https://www.arduino.cc/en/Tutorial/ReadASCIIString