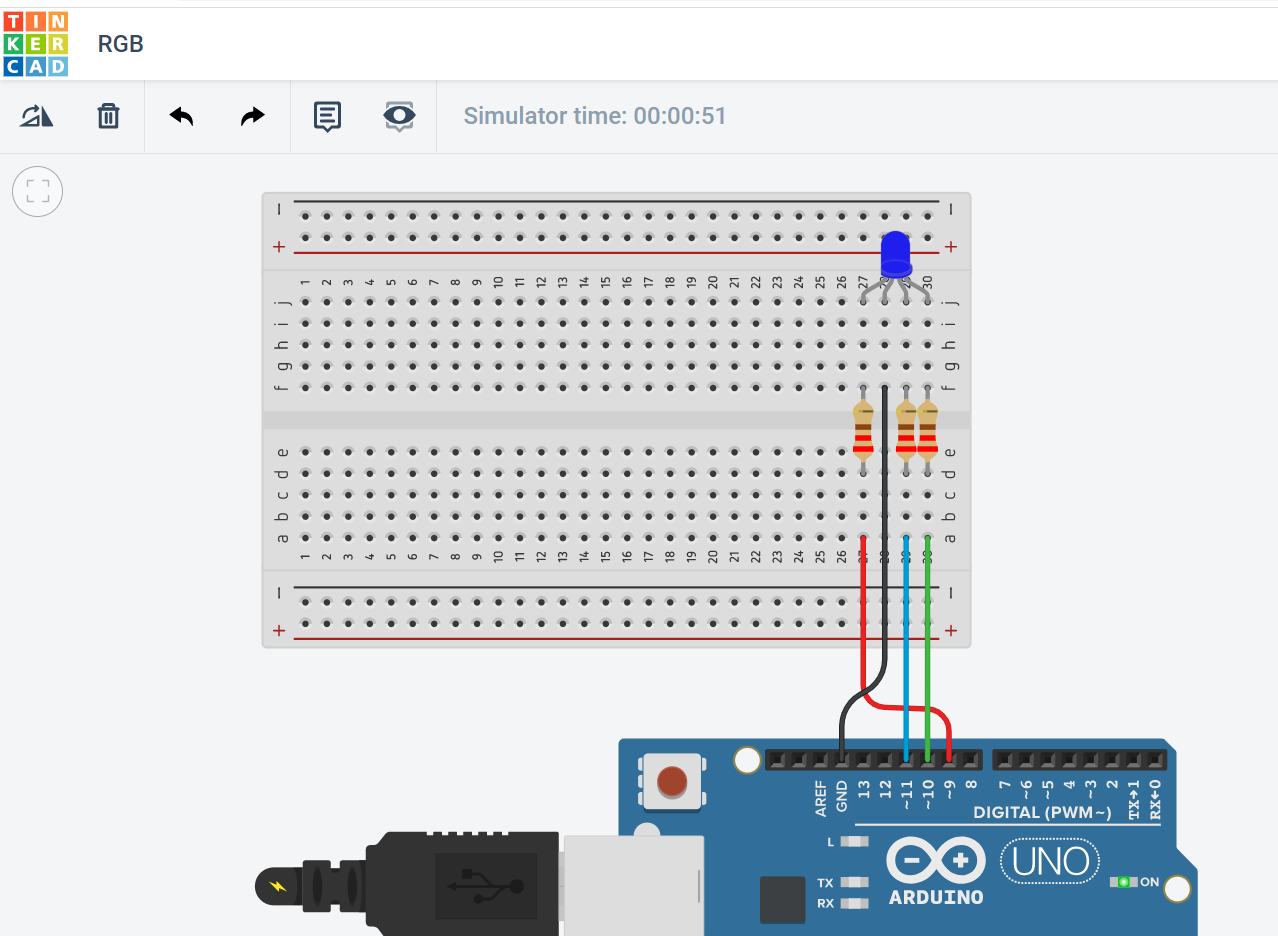
RGB LED Color Mixing

This project will be using an RGB LED to scroll through a variety of colors.  RGB stands for Red, Green and Blue and this LED has the ability to create nearly unlimited color combinations.

Hardware Required

* Arduino Uno
* USB A-to-B Cable
* Breadboard
* RGB LED
* 220 Ω Resistors
* Jumper Wires

Circuit



Project Code

// Make an RGB LED display a rainbow of colors!

const int RED\_PIN = 9;

const int GREEN\_PIN = 10;

const int BLUE\_PIN = 11;

const int DISPLAY\_TIME = 1000;

void setup()

{

 pinMode(RED\_PIN, OUTPUT);

 pinMode(GREEN\_PIN, OUTPUT);

 pinMode(BLUE\_PIN, OUTPUT);

}

void loop()

{

 mainColors();        // Red, Green, Blue, Yellow, Cyan, Purple, White

 showSpectrum();    // Gradual fade from Red to Green to Blue to Red

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* void mainColors()

\* This function displays the eight "main" colors that the RGB LED

\* can produce. If you'd like to use one of these colors in your

\* own sketch, you can copy and paste that section into your code.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void mainColors()

{

 // all LEDs off

 digitalWrite(RED\_PIN, LOW);

 digitalWrite(GREEN\_PIN, LOW);

 digitalWrite(BLUE\_PIN, LOW);

 delay(DISPLAY\_TIME);

 // Red

 digitalWrite(RED\_PIN, HIGH);

 digitalWrite(GREEN\_PIN, LOW);

 digitalWrite(BLUE\_PIN, LOW);

 delay(DISPLAY\_TIME);

 // Green

 digitalWrite(RED\_PIN, LOW);

 digitalWrite(GREEN\_PIN, HIGH);

 digitalWrite(BLUE\_PIN, LOW);

 delay(DISPLAY\_TIME);

 // Blue

 digitalWrite(RED\_PIN, LOW);

 digitalWrite(GREEN\_PIN, LOW);

 digitalWrite(BLUE\_PIN, HIGH);

 delay(DISPLAY\_TIME);

 // Yellow (Red and Green)

 digitalWrite(RED\_PIN, HIGH);

 digitalWrite(GREEN\_PIN, HIGH);

 digitalWrite(BLUE\_PIN, LOW);

 delay(DISPLAY\_TIME);

 // Cyan (Green and Blue)

 digitalWrite(RED\_PIN, LOW);

 digitalWrite(GREEN\_PIN, HIGH);

 digitalWrite(BLUE\_PIN, HIGH);

 delay(DISPLAY\_TIME);

 // Purple (Red and Blue)

 digitalWrite(RED\_PIN, HIGH);

 digitalWrite(GREEN\_PIN, LOW);

 digitalWrite(BLUE\_PIN, HIGH);

 delay(DISPLAY\_TIME);

 // White (turn all the LEDs on)

 digitalWrite(RED\_PIN, HIGH);

 digitalWrite(GREEN\_PIN, HIGH);

 digitalWrite(BLUE\_PIN, HIGH);

 delay(DISPLAY\_TIME);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* void showSpectrum()

\*

\* Steps through all the colors of the RGB LED, displaying a rainbow.

\* showSpectrum() calls a function RGB(int color) that translates a number

\* from 0 to 767 where 0 = all RED, 767 = all RED

\*

\* Breaking down tasks down into individual functions like this

\* makes your code easier to follow, and it allows.

\* parts of your code to be re-used.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void showSpectrum()

{

 for (int x = 0; x <= 767; x++)

 {

   RGB(x);      // Increment x and call RGB() to progress through colors.

   delay(10)

 }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* void RGB(int color)

\*

\* RGB(###) displays a single color on the RGB LED.

\* Call RGB(###) with the number of a color you want

\* to display. For example, RGB(0) displays pure RED, RGB(255)

\* displays pure green.

\*

\* This function translates a number between 0 and 767 into a

\* specific color on the RGB LED. If you have this number count

\* through the whole range (0 to 767), the LED will smoothly

\* change color through the entire spectrum.

\*

\* The "base" numbers are:

\* 0   = pure red

\* 255 = pure green

\* 511 = pure blue

\* 767 = pure red (again)

\*

\* Numbers between the above colors will create blends. For

\* example, 640 is midway between 512 (pure blue) and 767

\* (pure red). It will give you a 50/50 mix of blue and red,

\* resulting in purple.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void RGB(int color)

{

 int redIntensity;

 int greenIntensity;

 int blueIntensity;

 color = constrain(color, 0, 767); // constrain the input value to a range of values from 0 to 767

   // if statement breaks down the "color" into three ranges:

 if (color <= 255)       // RANGE 1 (0 - 255) - red to green

 {

   redIntensity = 255 - color;    // red goes from on to off

   greenIntensity = color;        // green goes from off to on

   blueIntensity = 0;             // blue is always off

 }

 else if (color <= 511)  // RANGE 2 (256 - 511) - green to blue

 {

   redIntensity = 0;                     // red is always off

   greenIntensity = 511 - color;         // green on to off

   blueIntensity = color - 256;          // blue off to on

 }

 else                    // RANGE 3 ( >= 512)- blue to red

 {

   redIntensity = color - 512;         // red off to on

   greenIntensity = 0;                 // green is always off

   blueIntensity = 767 - color;        // blue on to off

 }

 // "send" intensity values to the Red, Green, Blue Pins using analogWrite()

 analogWrite(RED\_PIN, redIntensity);

 analogWrite(GREEN\_PIN, greenIntensity);

 analogWrite(BLUE\_PIN, blueIntensity);

}