What does RGB LED stand for?

Red-Green-Blue Light Emitting Diode!

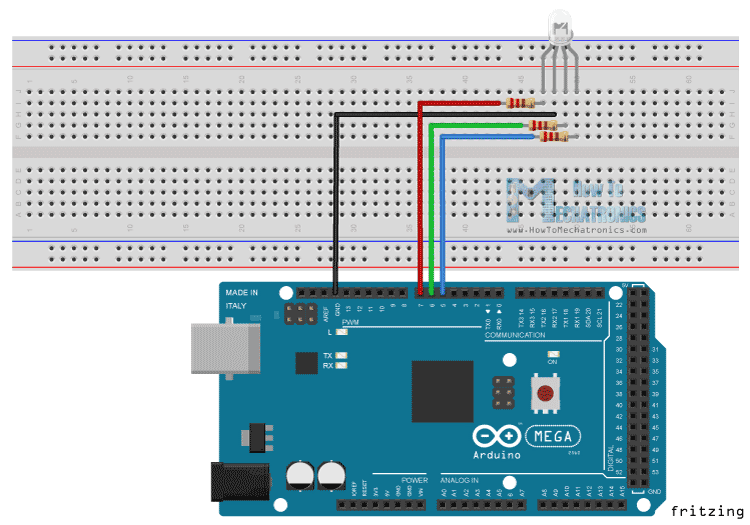
Purpose of today’s activity:

You are going to work with a partner to make your LED **blink a color of your choice.** Since there are two of you, you will have to help each other make the LED blink **two separate colors.**

Next, you will each work on creating a color sequence. The color sequence will be a list of colors that the LED blinks, and you can use any colors you like. You have to change the settings to change the length of the blinks between colors. **Later, you will upload your personal code to your very own detachable LED that you will get to take home!**

Arduino Uno Pinout:

Note: A **pinout** is an image that clearly labels what each plug point in a circuit board does. A **breadboard** is the board with lots of little holes; **each row of holes is connected** so we can connect the LED **pins** to the **resistors** which connect to the wires we finally plug into the Arduino.

This should look similar to the Fab Lights we did last week.

The red pin is connected to a 150 ohm resistor and the green and blue pins are connected to two 100 ohm resistors. Each pin is connected by wires to a separate, numbered plug point.

The ground pin (labeled in **black**, the second pin after the red pin) is connected to power on the Arduino, labeled “5 V” which stands for “5 volts.”

Understanding and Changing the Code

**NOTE: Every line of code ends with a semicolon which looks like this ; . If you forget to put a “ ; ” at the end of your line of code the code will NOT work!**

How do we know where the red, green, and blue pins should be connected on the Arduino? **Take a look at the code.** These are settings I put into the code:

int redPin = 11;

int greenPin = 10;

int bluePin = 9;

What does this mean? This tells the Arduino that the red pin is connected on the plug point labeled “11”, that the green pin is connected on 10, and the blue pin is connected on 9. Your Arduino should be connected like that. **Don’t change these settings, but know that you could if you wanted to.**

Look at the next line of code:

void setup()

{

pinMode(redPin, OUTPUT);

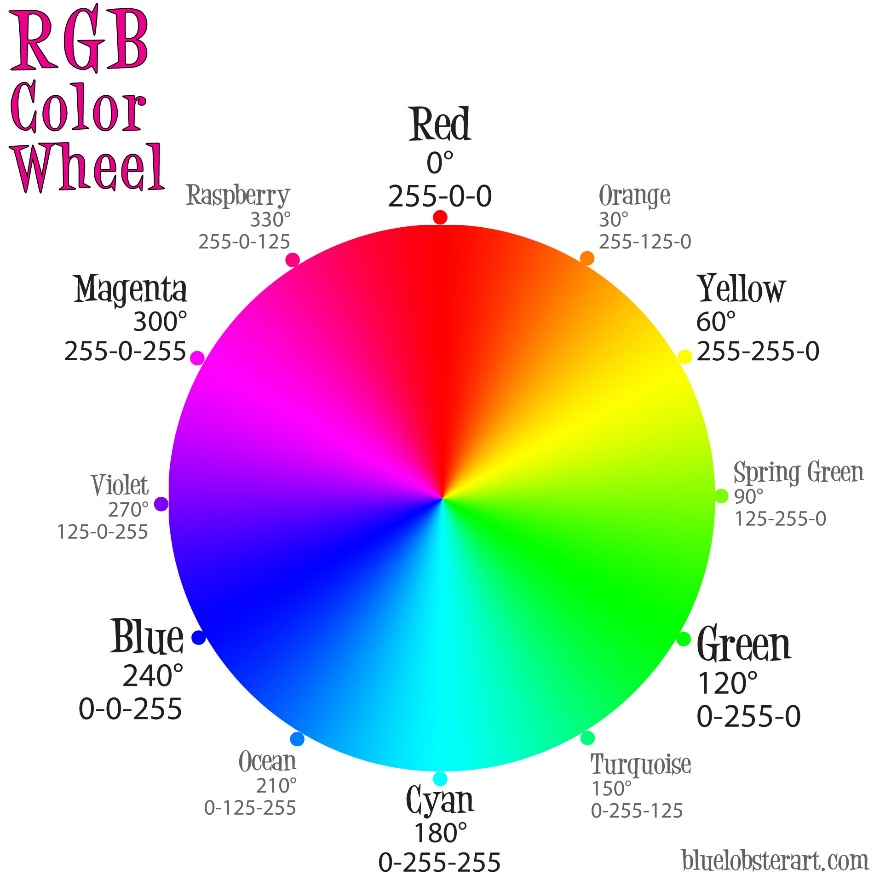
pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

}

void setup() is a function that runs once at the beginning before the Arduino runs your code. Here we are telling the Arduino that we have a red pin, a green pin, and a blue pin and all of these pins will have an output- in our case, that’s colored light.

**Let’s go to the important part where you have to set the LED’s color:**



void loop()

{

setColor( 0, 0, 0);

delay(1000);

blink();

}

void loop()is a part of the code that will continuously run over and over again (**it will run on a loop, like the name).**

What does the setColor ( ) function do? Think of the three numbers this way: setColor ( red, green, blue). You have to put in a **number** for each of the colors, so you have to put in **three numbers.** Right now **all the values are at 0** **which tells the LED to stay turned off.**

Remember, each value can be any number between **0** **(none of the color)** and **255 (maximum amount of color).** How do you know how much of each color to type in? **This is exactly like how you mixed the paint colors you wanted for your Halloween pumpkins.** Think about what two colors you need and how much of each color you need to add to make your final color.

Hint: You can use this **RGB calculator** at this URL to get an estimate of the correct numbers of red, green, and blue to get the color you want:

<https://www.w3schools.com/colors/colors_rgb.asp>

The next line says delay(1000); Why do we want to delay the LED? **The delay is the amount of time the LED will display the color.** This is important! If you want to see your color for 1 second, you need to have a delay of 1000 to which is 1000 milliseconds (equal to 1 second.) **You can change this part.** What happens when you make the number too low or too high?

The next line says blink(); which will blink the LED. How does it work? **Scroll down to the bottom of your code and take a look**.

void blink()

{

setColor(0, 0, 0);

delay(1000);

}

I wrote a function called blink(). I simply wrote what I want the function to do; this part of the code **will not run** unless I **call the function** in my void loop(). **DO NOT copy and paste this code in your void loop()when you want to blink your LED- instead simply type the line blink(); between each color!** As you saw on the last page, I did call the function in void loop() by writing blink();.

Look at the function void blink(). I set each of the color values to 0 which means the red, green, and blue pins have no output- I turned the LED off. Then I wrote a delay of 1000 milliseconds or 1 second. **This function tells the LED to turn off and STAY turned off for exactly 1 second.** That’s exactly what an LED has to do in order to blink between colors! **You can change the delay in this function to any number to change how fast the LED blinks between colors.**