

day87 Arduino RGB

Connecting and using an RGB (Red-Green-Blue) LED

Tuesday 4/19/22 and Wednesday 4/20/22

Today you're going to connect up something called an RGB LED. RGB stands for Red-Green-Blue, which means that this single LED can be directed to make just about any color you want. Each partner group should have one of these LEDs. It's larger than the others, is clear or milky colored, and it has four leads.

Please look through your bags of parts and find this LED.

Get out the following parts:

- One RGB LED (the larger LED that has four leads).
- Three 270 ohm resistors:



For the RGB LED, the longest pin is the ground. Spread the pins apart, and put them into different rows on the breadboard (for example, how about 8e, 9e, 10e, and 11e). Pay attention to which row has the longest pin and connect this to GND on the Arduino. Connect each of the other three pins to a 270 ohm resistor, then to one each of the following pins: 9, 10, and 11 on the Arduino.

Please do NOT look up how to make this circuit on the Internet. I need you to make the circuit I have described above. If you need help, ask me, or ask a classmate.

You and your partner will together write a single program. Open a new sketch and do the following:

1) Above `setup()`, add three lines, one for each color as follows using pins 9, 10, and 11 and red, green, and blue:

```
int red = 9;
int green = 10;
int blue = 11;
```

2) In `setup()`, add three lines, one each for Red, Green, and Blue as follows:

```
pinMode(red, OUTPUT);
```

3) Initially, in `loop()`, add this **one** line:

```
analogWrite(red, 255);
```

Name your program "Arduino RGB". Just one program per partner group. One person turns it in, the other person can just mark the project "done" in the Google Classroom.

Debug and upload your program. The LED light should light up, either as red, green, or blue (I know we set red, but the truth is we don't know yet which pin is which color. If the light doesn't light up at all, recheck your circuit, try again, and/or get help to figure out how you wired it incorrectly.)

At this point, if the LED lights up as red, then by chance you got the red pin defined correctly. Change your line in loop() now to green or blue, and re-upload to see if you get the right color.

If you don't get the right color, but you do get one of the other colors, just change your **int lines at the top of the program** to be correct and re-Upload. You may have to upload the program several times to get the definitions correct. It's a process of trial and error.

For example, let's say you have the starter line I asked you to do with red in it, but the light lit up as blue. You would change the definition lines at the top of your program to the following:

```
int red = 11;  
int green = 10;  
int blue = 9;
```

It's not that we know that red will be 11 as much as we know that blue is 9. It's a simple process of elimination. You will need to test one color at a time, turn on that color, Upload, then see if it is correct, and then make changes if not. Ask for help if you're not clear on this.

4) Once you get the definitions correct (that is, when turning on "red", the light shows as red, blue turns on blue, etc., you're ready to control the LED to make any color that you want. We make colors by ordering up different levels of red, green, and blue. Believe it or not, you can get any color using these base colors (it's how LCD TVs and theatre lighting work, for example).

When you want to make the light be a certain color, you tell the LED in three lines how much of each color that you want. You provide values from 0 to 255. For example, the following three lines make the LED shine green:

```
analogWrite(red,0);  
analogWrite(green,255);  
analogWrite(blue,0);
```

The following three lines make the LED shine yellow (it turns out that a good yellow has some red and some green in it, when using this type of LED):

```
analogWrite(red,255);  
analogWrite(green,200);  
analogWrite(blue,0);
```

I found a link that shows you various colors and the RGB codes behind them that might help: http://www.rapidtables.com/web/color/RGB_Color.htm (scroll down to "RGB color table" a page or two down from the top).

Your task today:

Write code that will cycle your LED through the colors of the rainbow, one second for each color: Red, Orange, Yellow, Green, Blue, Purple

Some of the colors don't really look exactly like what you're going for; that's OK. This isn't a high-end HD TV after all. Do your best looking up the color values and programming them in, one second to each color in the rainbow.

After doing the rainbow, write code that fades through a bunch of different colors that uses at least one **for loop**. For example, you can make a for loop with a variable going from 0 to 255 and that will cycle through 256 colors; you'd want to have a short delay between color changes (for example "delay(10)"). Experiment until you get an interesting effect. You can change two color effects with a single for loop by making one use your variable, say *i*, for its power and making another one use $255 - i$. The highest value you can send is 255 and the lowest is 0.

When your light show is done, choose a single color and have it blink on and off twice with 500 millisecond delays so that we can all tell when the program is starting over.

You have lots of choices of how to do this. Feel free to look up code examples on the web, but please write your own code in the end. If you can't explain it to me, you didn't write it.

In summary, here is your coding assignment:

- Show the rainbow: red, orange, yellow, green, blue, purple, 1 second each
- Do a light show where you cycle through interesting colors for about 5 seconds
- Blink the light on and off twice clearly with 500 ms delays to indicate the end.
- Only one program per partner group. Please work together to both get credit.