**Project Report**

**Project 1 – Fancy Lighting**

**Zhong Zheng**

**Design Objective**

This fancy lighting is a lighting system controlled by two pushbuttons switch and a photo sensor. It has a pushbutton to controls its turning-on/off. It uses an RGB LED as the lighting. When light is dark, LED is turned on. When light is bright, LED is turned off. Meanwhile, there is a pushbutton controlling the color switch, including 6 colors defined as below.

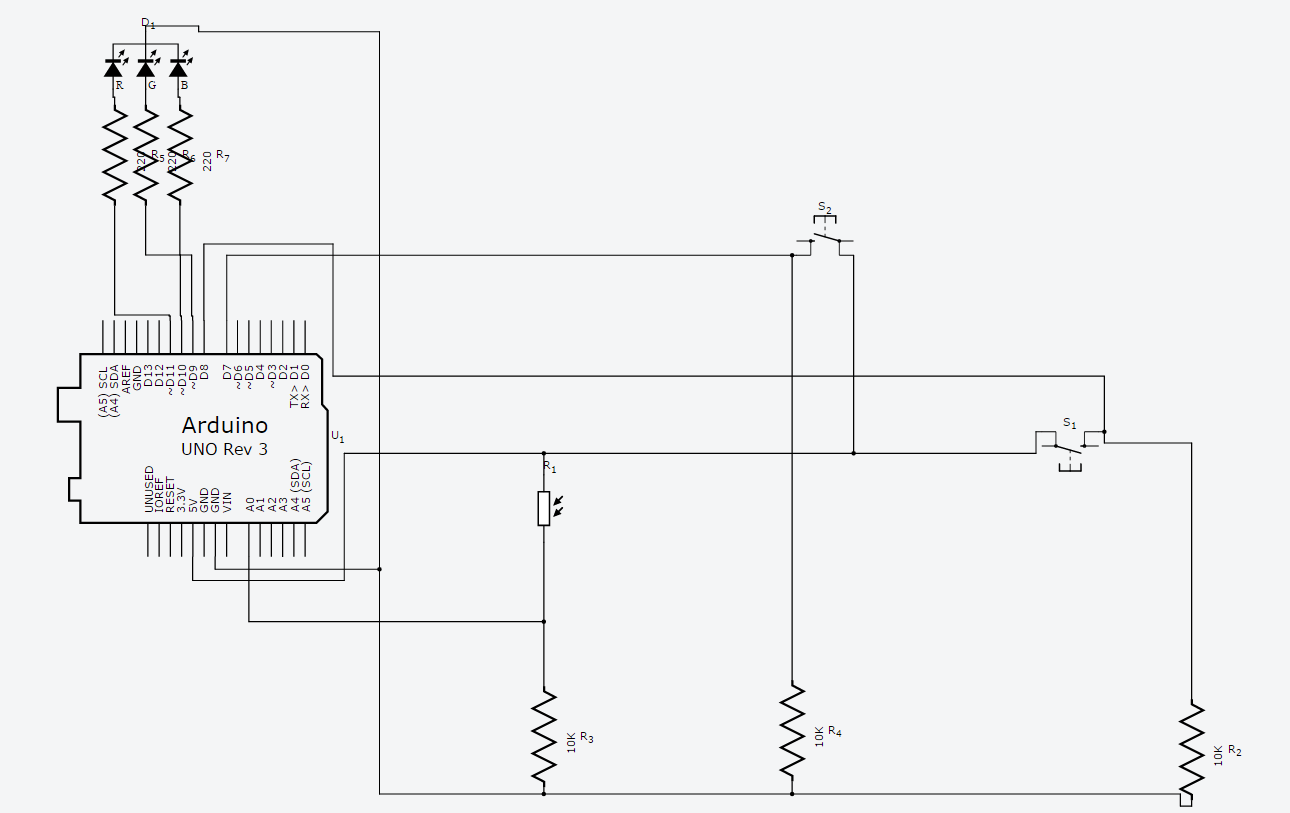
#define NUM\_COLORS 6

float colors[NUM\_COLORS][3] = {{1.0,0.0,0.0}, {1.0,1.0,0.0}, {0.0,1.0,0.0}, {0.0,1.0,1.0}, {0.0,0.0,1.0}, {1.0,0.0,1.0}};

The color switch is a smooth color changing. When changing from one color to another, it spend 2 seconds to change the color gradually and smoothly.

**Hardware Design**

Here is the circuit diagram below (drawn by 123d.circuits.io )



The components include:

1 Arduino UNO board

1 RGB LED

2 Pushbutton switch

1 Photo sensor: Photoresistor (LDR)

3 10k ohm resistors and 3 220 ohm resistors

PIN usage:

PWM Pin 9, 10, 11 are used as output for RGB anode.

Pin 7, 8 are used as two pushbuttons.

Analog pin A0 is used as Photoresistor input.

**Software Design**

**Overall structure**

Button switch on/off has the highest priority, then the photo sensor, then the color switch button.

**Function list**

void setup()

void loop()

void lightUpRGB(int i, int s)

three Boolean

boolean button\_switch\_Pushed()

boolean button\_color\_Pushed()

boolean photo\_sensor()

**Goal Achievement**

1. Connect a RGB LED as the lighting

RGB anode connects to Pin 9, 10, 11, controlled by two pushbutton and one photoresistor

1. Connect a push button that controls its on/off

Debouncing pushbutton code handles this lights on/off feature and has the highest priority. Click it once to turn the system on, click it again to turn the system off. Related function and Boolean: void lightUpRGB(int i, int s), boolean button\_switch\_Pushed().

1. Connect a second button that changes its color

Debouncing pushbutton code handles this changing color feature. In a condition of dark light and switch button on, click this button to switch the color from one to another. Related function and Boolean: void lightUpRGB(int i, int s), boolean button\_color\_Pushed().

1. Connect a photo sensor that controls LED on/off

The data of the photo sensor comes from analog PIN A0. When the value is larger than a certain value, the light is off. When the value is smaller than this certain value, the light is turned on. The certain value is calculated by a calibrated lighting function during system setup in the first 5 second. Related function and Boolean: void setup(),void lightUpRGB(int i, int s), boolean photo\_sensor().

1. Smooth color changing

Three analog outputs write data changing from the last color to the current color, which has a slightly changes in each 20ms. This while() function repeats for 100 times, so the overall time for each color changing process lasts 2 seconds.

Related function and Boolean: void lightUpRGB(int i, int s), boolean button\_color\_Pushed().

**Special Considerations**

This project is a combination of all the devices and codes we have learned so far. I am using codebender.cc and 123D.circuit.io for this project. As 123D can be used for software simulation for circuit, it is very convenient to connect everything on this web and run my code. This shrinks the develop time and make debugging faster. However, I found the software simulation environment is too slow compared with the real environment. I am wondering a much more complicated project later on will make the simulation time too long to debug. And of course, that is the reason why we need hardware to prove our design. The other issue is that, software simulation environment is hard to debug the RGB light issue, especially for the smoothly color changing. As it is slow and 123D RGB model cannot do a good job in light changing simulation, I finally use my draft code on real board and improve the color changing feature.

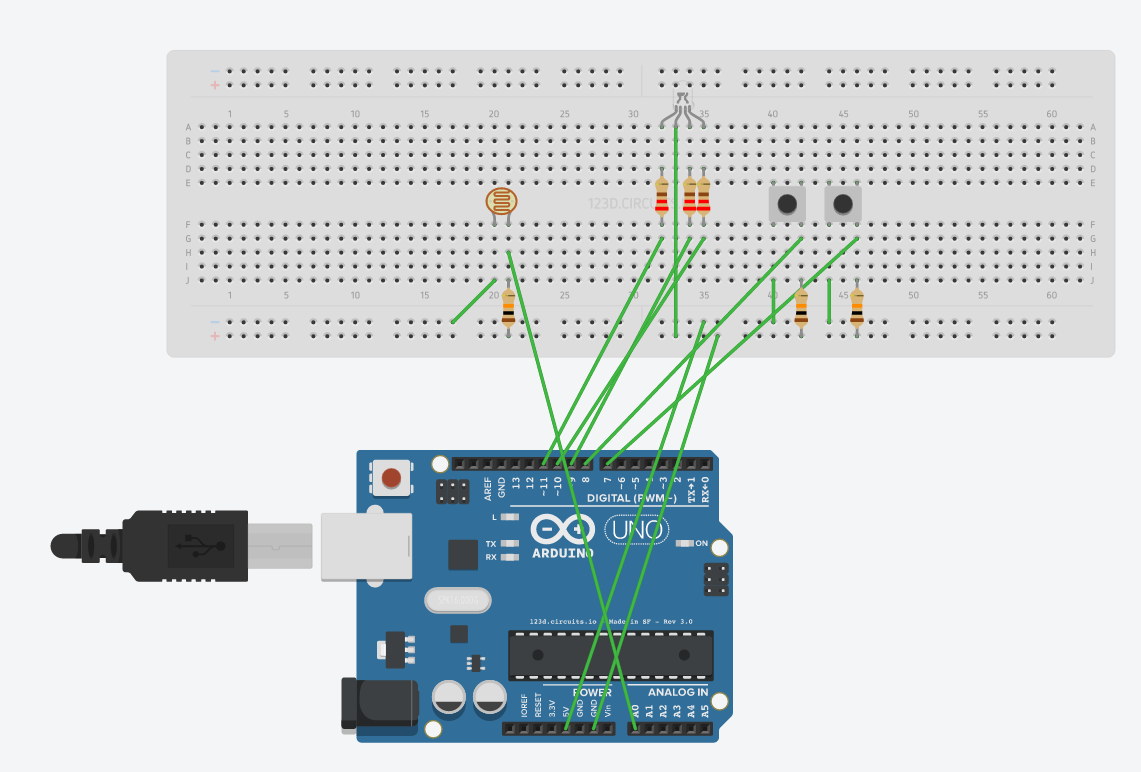
Another concern is that I write my draft code by using three functions to control the RGB LED light at the beginning. But later on, I optimize the code by combining them to one. Although this is OK for a small project to use a redundant code, when we have a much complex one, it will certainly waste memory to hold the program. As we are now coding for a hardware, we should take care of the memory and power after we achieve the basic function.

**Appendix**

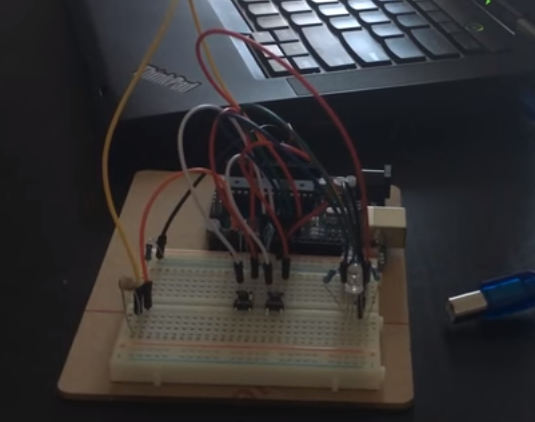
Sketch URL: <https://codebender.cc/sketch:237090>

Picture of hardware setup:

123d.circuits.io version



Real environment



A complete session which demonstrate all the features

Link to video clip on YouTube: <https://youtu.be/y__xxPwzAmk>

Corresponding serial console output of one complete run

Welcome! Zhong Zheng - Fancy Lighting

Press Button Switch to turn on light (highest priority)

Press Button Colors to change color (Totally 6 colors)

Photo sensor: light dark, turn on LED; light bright, turn off LED

RGB on/off changed from

0 to 1

Change color Button pushed!

Change color Button pushed!

Change color Button pushed!

Change color Button pushed!

Change color Button pushed!

Change color Button pushed!

RGB on/off changed from

1 to 0

Please email me if the sketch URL or YouTube link has any issue.

Thank you.

Zhong Zheng

[cathead@live.cn](mailto:cathead@live.cn)