

## 1. Introduction

### a. Statement of Purpose

Our goal is to design a color-changing lamp that can be activated by a multitude of sensors. The features that this product offers will provide it a competitive advantage in the interior ornament market.

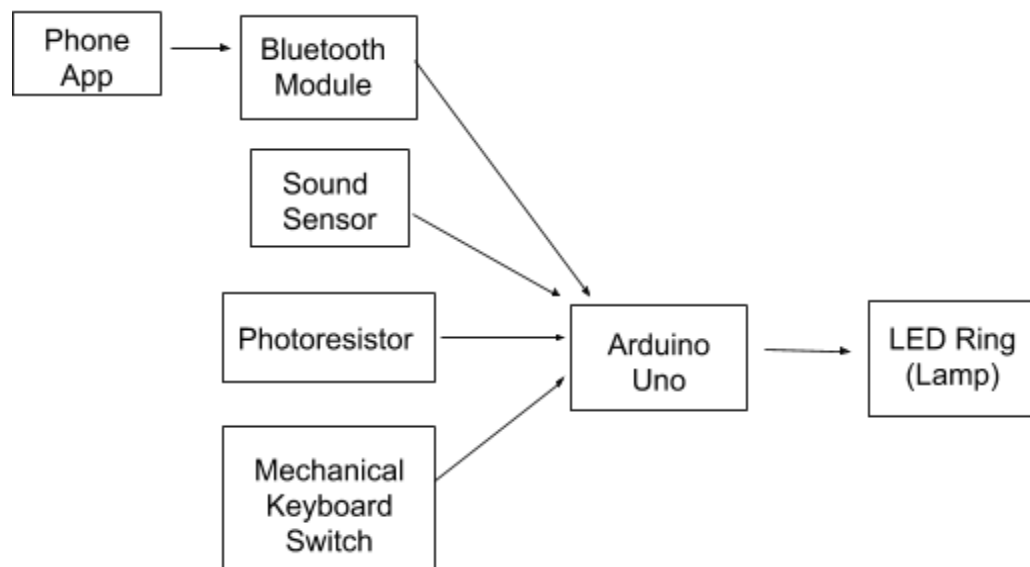
### b. Features and Benefits

The most outstanding feature of the Avalon lamp is the recognition of a multitude of sensory triggers, including light, sound, and touch activations and remote control. The lamp itself is a mechanical switch that would light up the LED ring inside when pressed down. The lamp could be triggered by sound intensity above a threshold level. When it is exposed to bright light, it would not light up. This feature would help conserve energy of the power source. We also incorporated a bluetooth module in our design so that an app on the phone could be used to control the activation of the lamp.

## 2. Design

### a. System Overview

#### Block Diagram



Our lamp has four different means of activation: remote control, sound control, light control, and touch control. Remote control is achieved by entering 1 through the phone app, which sends a signal to the bluetooth module. The sound sensor can detect a loud noise, such as a clap and send a signal. The photoresistor will vary its resistance based on the ambient light intensity. The mechanical keyboard switch acts like a button. The signals are processed by the Arduino program. The Arduino then outputs a voltage signal that would light up the LED ring.

### b. Design Details

Our lamp lights up when the LED ring receives an input signal from pin 5 in the Arduino, which turns active whenever one of the sensors sends a signal that is processed by the program to be high. Photoresistor is connected to the analog pin in the Arduino. When the photoresistor detects a low level of light intensity, a high input is sent to the Arduino. As long as the light intensity is low, the photoresistor will keep

sending signals to Arduino and keep the light on. And the threshold for photoresist is 600, which means that if the value returned from photoresist is higher than 600, then the signal will be active. The sound sensor is connected to the digital pin. One single active input from the sensor will keep the light on for 0.3 seconds. And this will create an effect that whenever there is a continuous sound input, the light will keep on and change its pattern every 0.3 seconds. The Bluetooth and the mechanical switch acts as to simply turn on the lamp for 6 seconds. When Bluetooth sends an ASCII value of "1", the Arduino will turn the light on for 6s ; when the mechanical switch is pressed and the current will flow through the pin which would create an active signal and this also turns the light on for 6s. Note, multiple inputs won't cumulatively delay the lighting time(it will all be 6s)

### 3. Results

	Activated	Inactivate
Photoresistor	1.8V	990mV
Sound Sensor	4.5V	990mV

Our project is tested with photoresistor, sound sensor, and Bluetooth module respectively. When we take the project to the dark room, where the light intensity is low(Photoresistor value > 600), the lamp automatically lights up; and when the background is bright again (Photoresistor value < 600 ) the light will automatically turn off. The sound sensor is used as a digital input and the active input triggers the light. In addition, while the light is turned on through the sound sensor, one more active input through the sound sensor will change the light pattern of the lamp.

### 4. Problems and Challenges

- For the electronic parts, the biggest obstacle we encountered was to deal with our sensors. One typical thing was our sound sensor. In our design, we planed to use the sound sensor to control the data delivered to the Arduino for turning on and off of the lamp. That is, if we make some loud noise besides the sound sensor, it is supposed to detect the noise and send the changes in data to the Arduino and the codes will be processed and make the light turning on or off. For this part, we did some researches and built our circuit based on some circuits online as a reference. There were, however, something wrong with the sensor and we spent a long period to debug that. One thing we found was that the threshold of the sensor should be adjust to make a better detection.
- Another main challenge we met was about the codes of the Arduino. We tried to use the Arduino as the main source of controlling mean of the led ring and that was harder than what we thought. At the beginning, we assumed only '0' and '1' are required to be received by the Arduino and take the control. Instead, the PWM required codes more complicated than that, and that took us some time to figure out the codes.

### 5. Future Plans

For what we have now, is a lamp that can be controlled by light intensity, human intensity and our phone using bluetooth, which has accomplished all our original goal. Based on what we have now, we think one thing we can do to improve our project is to add a button, for example, as a switch to control the turning on and off of our lamp without relating to the Arduino. That is, we can have separated switches for the lamp for both electronic based and software based.

## 6. References

- [1]"Arduino and HC-05 Bluetooth Module Tutorial | Android Smartphone & Laptop Control", *YouTube*, 2018. [Online]. Available: <https://www.youtube.com/watch?v=E-1w7dL3Cps>. [Accessed: 15- Dec- 2018].
- [2]"Sound Controlled Lights Circuit.", *ElectroSchematics.com*, 2015. [Online]. Available: <https://www.electroschematics.com/2903/sound-controlled-lights/>. [Accessed: 15- Dec- 2018].
- [3]"Ambient Light Sensor Using Photo Resistor and LED Lights!", *Hackster.io*, 2018. [Online]. Available: <https://www.hackster.io/DCamino/ambient-light-sensor-using-photo-resistor-and-led-lights-9b7f39>. [Accessed: 15- Dec- 2018].
- [4]"Photoresistor controlling LED | Arduino", *Arduino.sundh.com*, 2018. [Online]. Available: <http://arduino.sundh.com/2013/02/photoresistor-controlling-led/>. [Accessed: 15- Dec- 2018].
- [5]M. Slik, "Turn On and Off LED by LDR (Toggle LDR)", *Hackster.io*, 2018. [Online]. Available: <https://www.hackster.io/amar-slik/turn-on-and-off-led-by-ldr-toggle-ldr-2e3502>. [Accessed: 15- Dec- 2018].
- [6]R. Ranjan and S. Kumar, "Control LED By Clap Using Arduino and Sound Sensor", *Hackster.io*, 2018. [Online]. Available: <https://www.hackster.io/iotboys/control-led-by-clap-using-arduino-and-sound-sensor-e31809>. [Accessed: 15- Dec- 2018].

## Appendix (Code)

```
#include <Adafruit_NeoPixel.h>
Adafruit_NeoPixel strip=Adafruit_NeoPixel(24, 5, NEO_RGB + NEO_KHZ800);
int soundSensor = 2;
int LED = 4;
int Ring=5;
int cherry = 3;
char bluetoothData=0; //Variable for storing received data from Bluetooth
int brightness = 0;
boolean LEDStatus=false;

//setup code
void setup() {
  Serial.begin(9600);
  strip.begin();
  strip.show();
```

```

pinMode(soundSensor,INPUT);
pinMode(cherry,INPUT);
Serial.print("Type '1' to turn on Avalon Smart Lamp");
}

//function to turn on the 24 LEDs by setting up the RGB to be R=random, G=random, B=random
void turnOn(int r, int g, int b){
  for (int i=0; i<24; i++){
    strip.setPixelColor(i, r, g, b);
  }

  strip.show();
}

//function to turn off the 24 LEDs by setting up the RGB to be 0,0,0
void turnOff(){

  for(int i=0; i<24; i++){
    strip.setPixelColor(i,0,0,0);
  }
  strip.show();
}

void loop() {

  // put your main code here, to run repeatedly:
//chaser();

  //control LED by photoresistor; when the ambient dark level is above 600, turn on the LED, when
  lower than 600, turn off LED
  int light = analogRead(A5);
  //Serial.println(light);
  if(light > 600){
    turnOn(random(255),random(255),random(255));
    LEDStatus=true;
    delay(6000);
  }

  else{
    turnOff();
    LEDStatus=false;
  }
}

```

```

//control LED by bluetooth
if(Serial.available() > 0) // Send data only when you receive data:
{
  bluetoothData = Serial.read(); //Read the incoming data & store into data
  Serial.print(bluetoothData); //Print Value inside data in Serial monitor
  Serial.print("\n");
}

//control LED by soundsensor & switch; when detect an active soundlevel or active switch, change
the state of the LED
int SensorData=digitalRead(soundSensor);
int cherryData =digitalRead(cherry);
if( cherryData == 1 | bluetoothData == '1'){
  //change LEDStatus when input is high
  if(LEDStatus==false){
    LEDStatus=true;
    turnOn((255),random(255),random(255));
    delay(6000);
  }
  else{
    LEDStatus=false;
    turnOff();
  }
}

if(SensorData==1 ){
  //change LEDStatus when input is high
  if(LEDStatus==false){
    LEDStatus=true;
    turnOn((255),random(255),random(255));
    delay(300);
  }
  else{
    LEDStatus=false;
    turnOff();
  }
}
}

```