Name : Faris Omar

Party Lights Documentation

Index

Goldsmiths
UNIVERSITY OF LONDON

Introduction

Party lights is similar to ambient lighting that is usually seen in big restaurants or shopping malls. But ambient lighting is becoming more and more popular with it being used for many other things. PC users are familiar with ambient lighting as it is a way of customizing a users PC setup with ambient lighting which makes it look better. Not only that, some big car companies such as Mercedes Benz even has RGB ambient lighting built into their car cabins which allows the driver to change the colour of the ambient light based on the preference of the driver. The only downside to ambient lighting today is that it is very costly to obtain. For example, for a PC user to experience ambient lighting on his PC setup, he has to purchase pre-made ambient lighting from companies such as NZXT , or Corsair which charge a lot for them. It goes the same for some televisions such as Philips televisions that come with ambient lighting which supposedly gives more immersion to the viewers when the movie and the ambient lighting is played together in coherence.

The decision to proceed with this project is not only for the purpose of using it on my own personal setup but also to challenge myself in producing something that i have been interested in doing.

Equipment and Software

- 1x Arduino UNO R3
- 1x Breadboard
- 1x RGB 30 LED Strip (15 is also usable)
    - And also independently lit RGB LED Strips are better as it will represent the colours and project it more accurately.
- 1x UNL2003 Transistor
    - Instead of using three MOSFET-N Transistor which would clutter the wiring on the breadboard compared to using the UNL2003 Transistor
- 1x 12V Power Adapter
    - This is needed to power the LED’s.
- Software : Arduino IDE and Processing 3
    - Processing is used for the purpose of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks.

Arduino Code

```
int red, green, blue; //red, green and blue values
int RedPin = 9; //Red pin 9 has a PWM
int GreenPin = 11; //Green pin 10 has a PWM
int BluePin = 10; //Blue pin 11 has a PWM
void setup()
{

Serial.begin(9600);
//initial values (no significance)
int red = 255;
int blue = 255;
int green = 255;
}

void loop()
{

//protocol expects data in format of 4 bytes
//(xff) as a marker to ensure proper synchronization always
//followed by red, green, blue bytes
if (Serial.available()>=4) {
if(Serial.read() == 0xff){
red = Serial.read();
green= Serial.read();
blue = Serial.read();
}
}
//finally control led brightness through pulse-width modulation
analogWrite (RedPin, red);
analogWrite (GreenPin, green);
analogWrite (BluePin, blue);
delay(10); //just to be safe
}
```

Processing Code

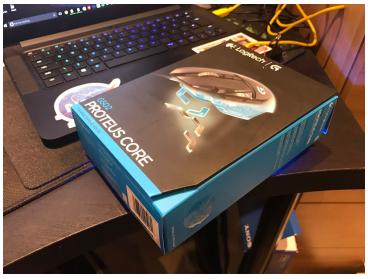
```
import java.awt.Robot;
import java.awt.AWTException;
import java.awt.event.InputEvent;
import java.awt.image.BufferedImage;
import java.awt.Rectangle;
import java.awt.Dimension;
import processing.serial.*;


Serial port;
Robot robby;

void setup()
{
  println(Serial.list());
port = new Serial(this, Serial.list()[0],9600); //set baud rate
size(100, 100); //window size (doesn't matter)
try //standard Robot class error check
{
robby = new Robot();
}
catch (AWTException e)
{
println("Robot class not supported by your system!");
exit();
}
}

void draw()
{
int pixel; //ARGB variable with 32 int bytes where
//sets of 8 bytes are: Alpha, Red, Green, Blue
float r=0;
float g=0;
float b=0;


//get screenshot into object "screenshot" of class BufferedImage
BufferedImage screenshot = robby.createScreenCapture(new Rectangle(new Dimension(1920,1080)));



int i=0;
int j=0;
//I skip every alternate pixel making my program 4 times faster
 for(i =0;i<1920; i=i+2){
    for(j=0; j<1080;j=j+2){
      pixel = screenshot.getRGB(i,j);
      r = r+(int)(255&(pixel>>16)); //add up reds
      g = g+(int)(255&(pixel>>8)); //add up greens
      b = b+(int)(255&(pixel)); //add up blues
}
}
r=r/(960*540); //average red
  g=g/(960*540); //average green
  b=b/(960*540); //average blue
```

```
//println(r+","+g+","+b);

// filter values to increase saturation
float maxColorInt;
float minColorInt;

maxColorInt = max(r,g,b);
if(maxColorInt == r){
  // red
  if(maxColorInt < (225-20)){
    r = maxColorInt + 20;
  }
}
else if (maxColorInt == g){
  //green
  if(maxColorInt < (225-20)){
    g = maxColorInt + 20;
  }
}
else {
   //blue
   if(maxColorInt < (225-20)){
    b = maxColorInt + 20;
  }
}

//minimise smallest
minColorInt = min(r,g,b);
if(minColorInt == r){
  // red
  if(minColorInt > 20){
    r = minColorInt - 20;
  }
}
else if (minColorInt == g){
  //green
  if(minColorInt > 20){
    g = minColorInt - 20;
```

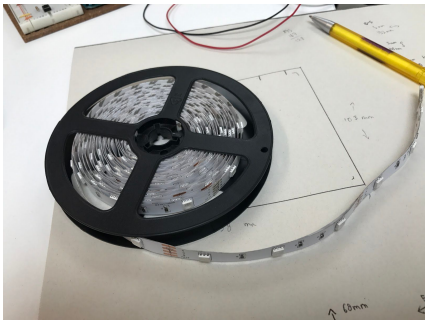
Enclosure

A mouse PC peripheral box was used as the enclosure as it has proper exit holes for the power lines, USB Cable, and also the RGB Strip. The box is also ideal as there is a cut through hole in the box which allows the user to reset the arduino if needed. There is also a compartment that is concealed behind the box which is perfect to stash the RGB strip. If the user needs a longer strip then the user can simply pull the strip from the hidden compartment of the box. The arduino and the breadboard is screwed onto a piece of wood that was cut from the laser cutting machine for added stability and also added security as this box can also be placed vertically as well.

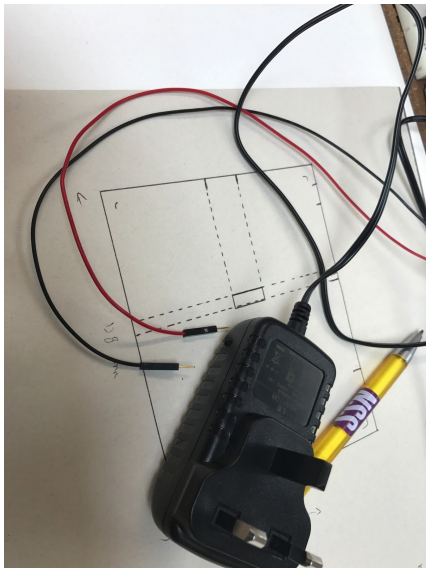
There is a cut out for the wires to reach their ports. (USB , AC power adapter.)



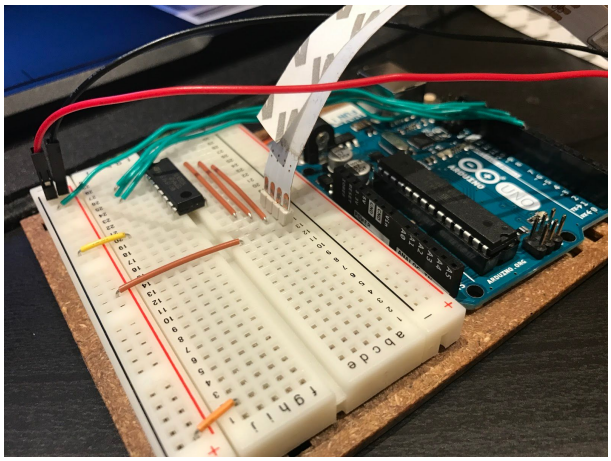
Equipment Images

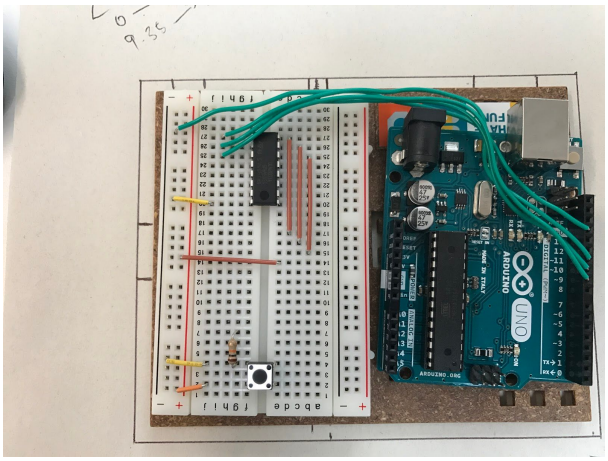


RGB Strip LED



Power Adapter 12V with the tips soldered on to the positive and negative End.





Problems and Future Development

Some of the major problems that was present were programming problems and also hardware compatibility. Instead of ordering the RGB Strips to follow the colours that are projected at the edge of the screen, i have only managed to program the RGB Strips to project the colours that are projected in the middle area of the screen. The program tends to crash whenever the code tells it to follow the the colours that are projected at the edge of the screen. As for the hardware compatibility is the RGB strip. There are only certain RGB strips that are compatible as some of them would not be able to present the colours accurately. The first problem that was encountered is when the RGB Strip did not project accurate colours due to the wrong type of RGB strip. Other problems that involve the first RGB strip used was the LEDs light blowing out due to the wrong type of power supply.

Future development that is possible for this project is instead of having these lights just for decorative purposes, they can also be usable. For instance, I hope to continue making progress on this project to integrate the RGB lights to work with alerts such as WhatsApp , Facebook notifications, Alarm clock , Temperature, and more as this would bring better function for the LED lights.

Conclusion

By the end of project , i had hoped to create more than just ambient lighting instead i was hoping to create notification lights and more but the instability of the program required me to pay more attention to the base before trying to achieve anything higher. Overall. Coming this far in the project and even making ambient lighting for my setup is considered a great achievement for me as it was challenging and i am definitely planning on continuing to work on this until i have achieved what i had planned to do from the beginning.