

Haze detection and characterisation by light scattering

Yar Khine Phyo, Kingston Kuan Jun Xiang, Ahn Tae Gyu

NUS High School of Mathematics and Science
Singapore
h1110050@nushigh.edu.sg

Abstract—We aim to predict amount and type of haze by light scattering. We examined how scattering of light is affected by angle, size & amount of particles and wavelengths of light by building a haze chamber. Wooden blocks and incense sticks were burnt to produce haze, while light source switched between red, green and blue light. Light sensors placed at different angles showed that more haze particles results in higher intensity of scattered light. We also found that blue light is consistently scattered more than green and red. Wood smoke (PM3.0) scattered more light than incense stick smoke (PM0.6), implying bigger particle sizes scatter more light. Our results also followed the Mie scattering model than Rayleigh model more closely as seen from angle dependency. We conclude it is possible to find characteristics of particles with light sensors by measuring at varying angles under different wavelengths of light.

Keywords—haze; haze detection; haze characterisation; light; scattering; Rayleigh scattering; Mie scattering; wavelength dependency; angle dependency; wood; incense

I. INTRODUCTION

Note: Bigger versions of Fig. 1 to 9 including graphs available in Appendix

Haze contains many harmful substances including particulate matter (PM). For sensitive people, knowing level of air pollution is crucial as they have to take necessary precautions as soon as possible. In Singapore, the NEA releases average PSI readings hourly, by using equipment that allow particles smaller than a specific size through filters and taking their mass [1]. However different places in large areas may have a significant difference in haze levels, readings are thus deemed inaccurate by many people. By distinguishing haze properties through light scattering, we can tell the public, immediately and at the location closest to them, information about amount and size of PM in the air.

Visibility of smoke is mostly due to scattering of light by airborne particles into our eyes. This research aims to find out if it is possible to use optical detection as a tool for monitoring haze. There are two types of scattering for particles, Mie scattering and Rayleigh scattering. Rayleigh scattering can be defined as scattering in the small size parameter regime, $x \ll 1$, where x is diameter of particle divided by wavelength of light [2]. Mie Scattering is predominant in particles that are of about the same size as the wavelength of light [2]. By using these concepts, we will try to find out if light scattering

measurements at different angles and wavelengths can be used to quantify and distinguish different types of haze.

II. HYPOTHESIS OF THE RESEARCH

Our initial hypothesis was that from the characteristics of light scattering by particulate airborne contaminants, conclusions can be drawn about extent and nature of particulate pollution.

Thus, there are two parts to the experiment:

1. Building an inexpensive sensor system sensitive & responsive enough to distinguish scattered light
2. Contrasting light scattering at different angles & wavelengths by varying amount & size of haze particles

III. RESEARCH MATERIALS AND METHODS

A. Verifying plausibility of obtaining results

A key requirement for the experiment was a sufficiently sensitive light sensor. Photodiodes were chosen over phototransistors as although phototransistors have built-in amplification, photodiodes have larger surface area to capture more light and are thus more sensitive. However, as small amounts of light were scattered by haze particles, we had to ensure that a photodiode was sensitive enough to detect the light. A lock-in amplifier was used as it not only amplified signals from the photodiode but also filtered electrical noise effectively (**Fig. 1.1**). Signal received increased substantially after fog was produced, meaning the signal could be measured provided with high amplification & noise reduction (**Fig. 1.2**).

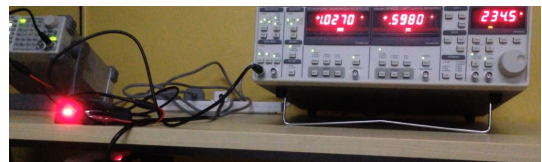


Figure 1.1: Photodiode with lock-in amplifier & red light



Figure 1.2: Light scattered in fog produced

B. Designing the light sensors

Without a lock-in amplifier, photodiodes produce weak signals so greater signal output was required or else the signal received could be even smaller than the electrical noise picked up. A high amplification circuit was created by converting the signal of photodiodes from current to voltage and then amplifying the voltage using operational amplifiers (**Fig. 2.1**). To reduce interference, a high-pass filter was used to ensure the circuit operates at a high frequency above a certain cut-off frequency [3]. Impedance from capacitors increases as frequency decreases thus eliminating noise received from lower frequencies, e.g. the 50Hz hum caused by appliances in Singapore. A different light sensor with decreased amplification was also created (**Fig. 2.2**) for direct exposure to the light source as a reference for the power of the light.

C. Designing the circuit for light source (RGB LED)

As different wavelengths and colours of light were required, a RGB LED was used. This was basically a single component with red, green & blue LEDs. The light sensor circuit was designed to take AC measurements from a blinking light source as this had advantages in noise removal. Other sources of interferences such as background light & electrical noises exist when either light is turned on or off. Our readings, having a constant frequency, will always add up with the right sign whereas source of interference will have different frequencies from our readings, thus adding up at a random phase. By finding the differences in the on & off values and averaging the differences sufficient numbers of times the interferences would be removed. As the light source needed to blink for this to work, an Arduino Uno, a microcontroller, was used to power the light source and receive data from the light sensors.

However, the Arduino cannot provide more than about 40mA current while the LED is rated at 350mA for full brightness, thus an NPN transistor was used to amplify the signal (**Fig. 3**). Furthermore, batteries supplied power to the circuit to prevent the large current drawn by the LED from causing supply voltage fluctuations in the Arduino circuit which could easily degrade the performance of the analog input channels used for sensors. As the RGB LED could be connected as separate LEDs, the amplification circuit (**Fig. 3**) could just be repeated 3 times. All circuits were tested extensively with oscilloscope and a prototype haze chamber to ensure sensitivity, amplification & noise reduction were sufficient for the experiment (**Fig. 4**).

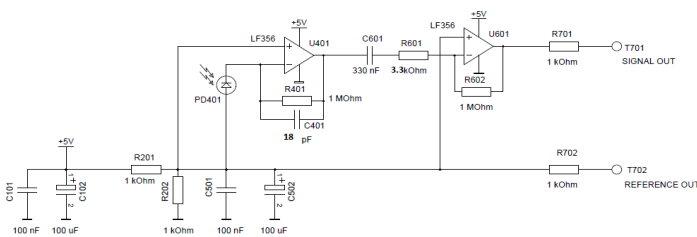


Figure 2.1: High amplification light sensor schematics

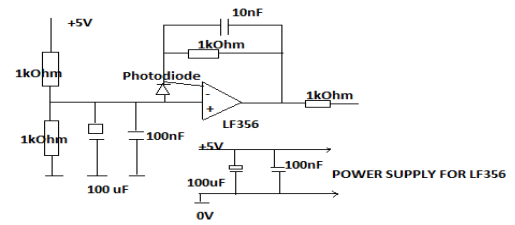


Figure 2.2: Low amplification light sensor schematics

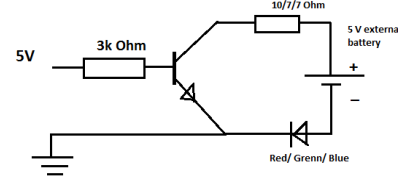


Figure 3: Light source amplification circuit



Figure 4: AC readings

D. Reducing electrical noise

The following steps were taken to further reduce electrical noise. The 2 op-amps (**Fig. 2**) were replaced with a 2-in-1 op-amp and circuit boards were cut to reduce circuit size and decrease potential for noise pickup. Wires from circuits to Arduino were braided to avoid EM induction. First paper, to avoid electrical contact, then aluminium foil, for electric shielding when grounded, was used to cover all circuits with holes cut to expose photodiodes to light.

E. Creating the experimental set-up

3 high amplification circuits (**Fig. 2.2**), 1 low amplification circuit (**Fig. 2.2**) and a RGB LED circuit (3 of **Fig. 3**) were soldered onto prototyping PCBs except RGB LED attached to wires and glued onto a heatsink to prevent damage from heat. The Arduino blinked the RGB LED and took readings from the light sensors when the light was switched on and off at 680 Hz to match the frequency response of the circuit. The program blinked a certain colour of light individually at first and then changed the colour after an adjustable amount of time. Using the Computer Aided Design (CAD) software LibreCAD, a haze chamber was designed to hold the set-up which included the light sensors, the light source, a biconvex lens and a ceramic cup (**Fig. 5**). A laser cutter was used to cut the chamber out of matt black acrylic sheets (blocked light & reduced reflection) and transparent acrylic sheets (in front of light sensors to prevent haze from disrupting the circuit). The light sensors were mounted onto the walls of the haze chamber using screws & nuts (Low amplification sensor as sensor 4 on Figure 5 and high amplification sensors as sensors 1, 2 & 3). Haze chamber was placed on top of fridge stands to place the ceramic cup in the hole to avoid melting the acrylic while also keeping the cup in place. (**Fig. 6**)

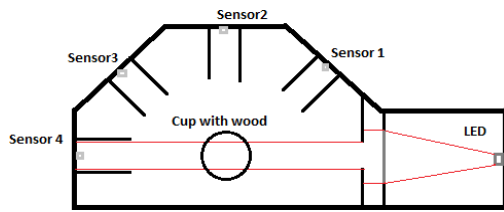


Figure 5: Extended haze chamber

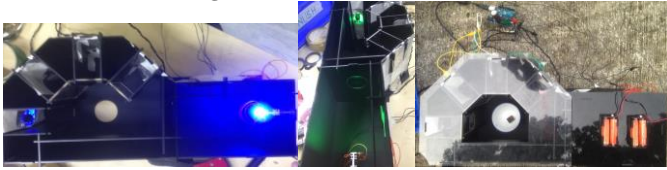


Figure 6: Final set-up of haze chamber with Arduino

F. Conducting the experiment

Note: Haze was produced from wood (PM 3.0) and incense sticks (PM 0.6) as particle size differed.

Step 1. A small length of incense stick was placed into the ceramic cup and put into the haze chamber.

Step 2. Incense was lit using the lighter and the haze chamber was covered with black paper.

Step 3. Data from the Arduino was sent to the computer via USB and recorded into a text file.

Step 4. When readings returned to a constant, the cup was taken out.

Step 5. Dust blower was used to get rid of any remaining haze particles inside the chamber.

Step 6. Steps 1 to 5 were repeated using more incense sticks and small wooden blocks.

IV. INTERPRETATION OF DATA, RESULTS AND FINDINGS

Fig. 7 & Fig. 8 are found on the next page (page 4).

In every graph, Change in light intensity/% is the averaged reading from the light sensor in the presence of haze divided by the averaged reading from the light sensor in the absence of haze.

For each experiment in **Fig. 7.1 to 7.3**, the same size & no. of haze particles would have been inside the haze chamber when data was taken as it was taken during the same experiment. (e.g. Wood 1 in **Fig. 7.1 to 7.3** was data taken when during an experiment where haze particles were produced from wood) In experiments Wood 1 & Wood 2, haze was produced from burning wood and in experiments Incense 1 & Incense 2, haze was produced from burning incense.

In **Fig. 8.1** below, incense was burned for all experiments. Experiments Blue 1 & Blue 2 used blue light but Blue 2 had 3 times the no. of incense burned & produced compared to Blue 1. Experiments Green 1 & Green 2 used green light but Green 2 had 3 times the no. of incense burned & produced compared to Green 1 and experiments. Red 1 & Red 2 used red light but Red 2 had 3 times the no. of incense burned & produced compared to Red 1.

For **Fig. 8.2**, experiments Blue-Wood, Green-Wood & Red-Wood is the change in light intensity in the presence of burned wood as haze particles and experiments Blue-Incense,

Green-Incense & Red-Incense is the change in light intensity in the presence of burned incense as haze particles.

V. CONCLUSIONS AND RECOMMENDATIONS

From **Fig. 7.1 to 7.3**, it can be seen that the data showed the same type pattern. It can also be observed that when amount & size of haze particles were equal, the intensity of light increased the most in **Fig. 7.1** (blue light) followed by **Fig. 7.2** (green light) then **Fig. 7.3** (red light). From this, we can conclude that blue light scatters the most, then green, then red. Blue light has wavelength of range 450 to 495 nm whereas green light's range is 495 to 570 nm, and red light's range is 620 to 750 nm. Hence, the shorter the wavelength of light, the more light scattered. Therefore, to obtain less erroneous results, it is better to use the light with the shorter wavelength since it will scatter more and thus, with greater increase in light intensity, it will be easier to detect the presence of haze particles. This makes blue light the most suitable to determine the presence of haze particles.

From **Fig. 8.1**, it was observed that for Blue 2, Green 2 & Red 2, especially for sensor 3, there is a greater increase in light intensity compared to Blue 1, Green 1 & Red 1 respectively. This indicates that more scattering was observed in every second experiment, where 3 times as many incense sticks were burnt. Hence, we can conclude that as no. of particles increases, more light will be scattered.

From **Fig. 8.2**, it was observed that for Blue-Wood, Green-Wood & Red-Wood, especially for sensors 2 & 3, there is a greater increase in light intensity compared to Blue-Incense, Green-Incense & Red-Incense respectively. Since haze particles produced by burning wood (PM 3.0, particles have a diameter of 3 microns or less) has bigger particle size than that produced by burning incense (PM 0.6, particles have a diameter of 0.6 microns or less) [8], it implies that as particle size gets bigger, more light is scattered.

Where $\alpha = \pi D_p / \lambda$ (D_p is the circumference of a particle and λ is the wavelength of incident radiation), if $\alpha < 0.1$ then Rayleigh scattering applies (small particle compared to wavelength of light), if $0.1 < \alpha < 50$ then Mie scattering applies (particle about the same size as wavelength of light, valid only for spheres), if $\alpha > 50$ then Geometric scattering (particle much larger than wavelength of light)[9]. Calculating α with estimated incense and wood particles sizes produce the values for both within the Mie scattering range.

Regardless of α value, we wanted to find out whether or not Rayleigh scattering equation could still be used to obtain estimated values for intensity of light scattered. From the Rayleigh scattering equation, we can find that $I \propto I_0 (1 + \cos 2\theta)$ where I is intensity of light scattered, I_0 is intensity of unpolarised light and θ is angle scattered. According to this equation, if our results could be estimated with the Rayleigh scattering equation, our sensors 1 & 3, which had 45° & 135° angles from the beam of light, would return the same change in light intensity, indicating an equal intensity of light was scattered.

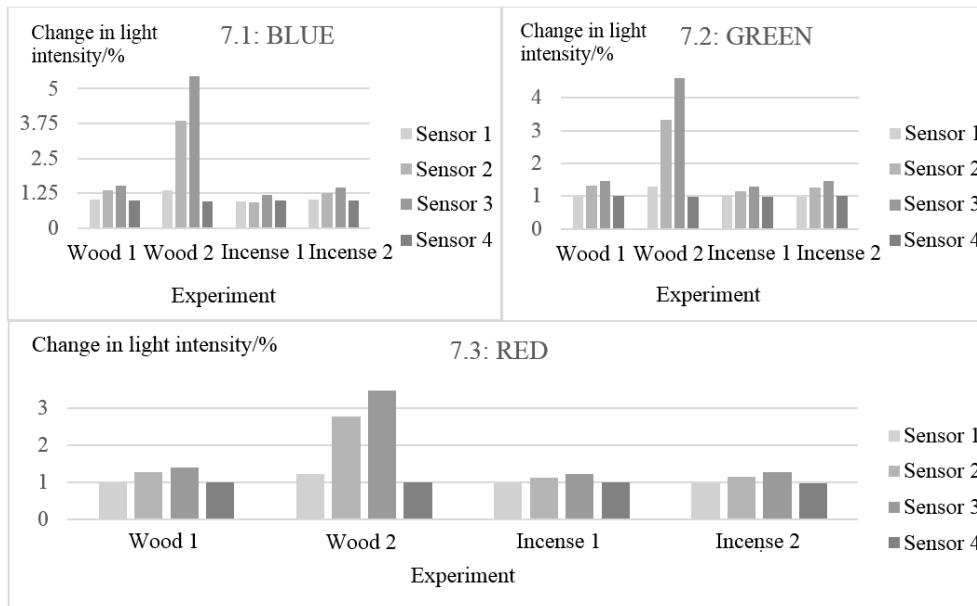


Figure 7: Graph of change in light intensity/% for experiments using (7.1) blue, (7.2) green & (7.3) red light

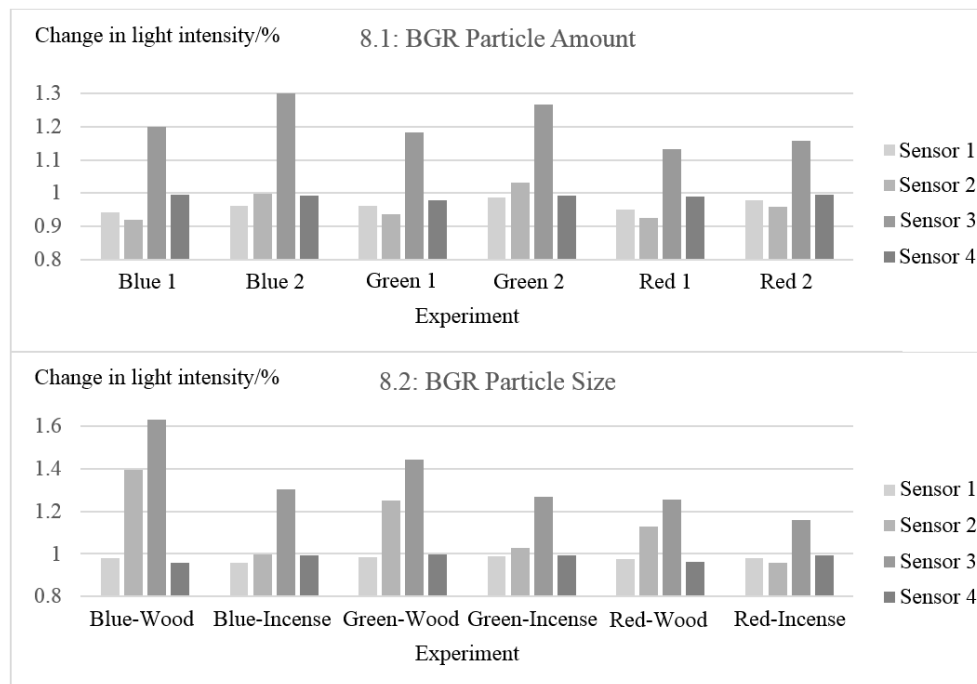


Figure 8: Graph of change in light intensity/% for blue (1 & 2), green (3 & 4) and red (5 & 6) light with (8.1) different amount of incense particles and (8.2) different haze particle sizes

However, our data (Fig. 7.1 to 8.2) clearly show that sensors 1 & 3 received an unequal amount of light and sensor 3 has a much greater increase in light intensity. Therefore, our results seem to suggest the use of mainly Mie scattering theory as it can be seen not only from α value but also from the fact that the sensor at 135° receives much more light than that at 45° . This is because Rayleigh scattering states that taking y axis as the center of the molecules, the light scattered will give a mirror image (Fig. 9) while Mie scattering states that taking y axis as the center of the molecules, the light scattered is

more in the direction of the incident light than in other directions.

From our figures, a consistent pattern where sensor 3 has the highest intensity in the presence of haze particles was observed. Furthermore, second highest intensity in data was only observed to be sensor 1 instead of 2 for data taken in the presence of incense particles (Fig. 7.1 Incense 1 & 2, Fig. 8.1 Blue 1, Green 1, Red 1 & 2 and Fig. 8.2 Red-Incense). Moreover, compared to the data taken in the presence of wood

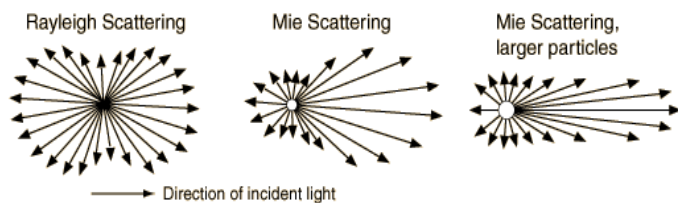


Figure 9: Mie and Rayleigh scattering [2]

particles, most data for incense have smaller difference in values of sensor 1 & sensor 3. Hence, when particle size is smaller, difference in values of sensor 1 & 3 decreases while values of sensor 2 tends to be lowest. With this pattern, we believe that through future work with different types of particles, values for sensor 1, 2, 3 and 4 which tell the estimated sizes & no. of particles can be found, allowing us to differentiate between different particle sizes and identify potentially harmful haze levels more quickly.

In addition, we believe that instead of using the Mie scattering theory[4][5], which involves tedious calculation, the calculations could be simplified through future work to form an equation that gives predicted approximate values of change in percentage of intensity of light. We believe that this equation is a multi-variable function, $H(\lambda, N, A\lambda, L, \theta, Sp)$, involving wavelength of light, no. of particles, absorbance respect to wavelength, length of medium, angle of sensor respect to light source and size of the particles as variables. This could allow us to predict and classify the danger of particles in the medium easily and efficiently with low cost within a short time.

ACKNOWLEDGMENT

Special thanks to Dr Wulf Hofbauer for coming up with initial idea and mentoring us along the way, to staff and CRADLE Science centre for helping us with logistics, equipments and guidance, and to Mr Tang Chi Sin of NUS high for coordinating us.

REFERENCES

- [1] Sia, L. X. 2013. How does singapore measure haze levels. <https://sg.news.yahoo.com/how-does-singapore-measure-haze-levels--142610492.html>
- [2] Nave, R. Rayleigh and Mie scattering. <http://hyperphysics.phy-astr.gsu.edu/hbase/atmos/blusky.html#c2>
- [3] Tony R. Kuphaldt. 2007. High Pass Filters. http://www.allaboutcircuits.com/vol_2/chpt_8/3.html
- [4] David, W. H. 2009. Light Scattering Theory, pp.1-13. <http://plaza.ufl.edu/dwhahn/Rayleigh%20and%20Mie%20Light%20Scattering.pdf>
- [5] Ng, L. N. 2000. Manipulation of particles on optical waveguides Chapter 3 Mie Theory, pp.23-37. http://www.orc.soton.ac.uk/publications/theses/1460T_Inn/1460T_Inn_03.pdf
- [6] Scattering. <http://www2.ensc.sfu.ca/~glenn/e376/e37617.pdf>
- [7] Air Pollution: Measurement, Modelling and Mitigation, Third Edition, pp.345-347
- [8] Siao, W. S. Physical characteristics of nanoparticles emitted from incense smoke, pp.2, http://iopscience.iop.org/1468-6996/8/1-2/A06/pdf/STAM_8_1-2_A06.pdf

APPENDIX

1. Notes for schematics of light sensor

C101, C102 – These filter the supply voltage for the opamps. Combination of large electrolytic capacitor and small ceramic capacitor; exact values not critical.

R201, R202 – Voltage divider that generates an internal 2.5 V reference level. This is what should allow to use a single 5V power supply (from the Arduino).

C501, C502 – Filtering for the reference voltage. Combination of large electrolytic and smaller ceramic capacitor. Exact values not critical, but the presence of these capacitors is a must for the circuit to work properly.

PD401 – Photodiode

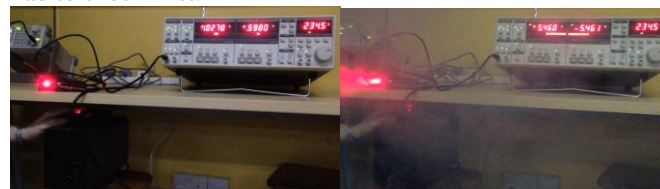
U401, R401, C401 – transimpedance amplifier (converts current from photodiode to voltage signal) with built-in low-pass for noise reduction. Gain as designed 1 V per microampere (1 MOhm). The product $R401 \cdot C401$ determines the upper frequency limit of the low pass $f = 1/(2 \cdot \pi \cdot R \cdot C)$, here set to approx. 1.6 kHz

U601, R601, C601, R602 – inverting voltage amplifier, gain 100 times, with high-pass characteristic. The gain is set by the ratio $R602/R601$. The lower cutoff frequency is set by $R601 \cdot C601$, $f = 1/(2 \cdot \pi \cdot R \cdot C)$, here set to approx. 50 Hz.

R701, R702 – not critical, but should help to improve stability and low noise performance of the amplifier.

2. Finding out if the experiment was even possible

Before we could even start on detecting haze, we were not even sure if it was even possible with our current tools, so we had to check first.

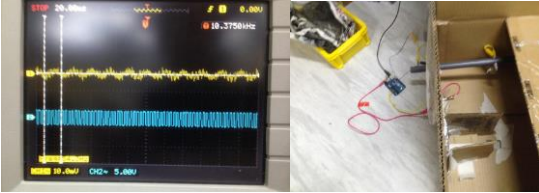


Checking out with lock-in amplifier if sensor can even detect the smoke. Photodiode is attached to lock-in amplifier before fog is produced. Photodiode and light source are separated by clay to prevent direct lighting. Signal increases after fog is produced, meaning signal can be measured as long as there is low noise and high amplification.

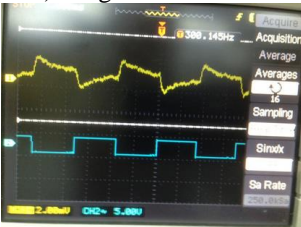


A cardboard box was used before acrylic box to see if changes in signal from photodiode can be seen in a dark box with and without smoke. Yes it is possible. On this oscilloscope, white line means voltage before smoke is produced and yellow line is the voltage from circuit when smoke is produced. One unit

represent 5 mV. (Photodiode is not directly in front of light source)



However, we can still see there is 50 Hz noise as although 280 Hz frequency was transmitted, signal received was combination of 50 Hz with higher amplitude and 280 Hz. So we wrapped the circuit with aluminium (with paper inside first) and grounded it



Now the signal is much clearer after grounding. One unit vertically now represent 2 mV, showing 50 Hz interference is gone and signal become clearer.

Code programmed for Arduino Uno:

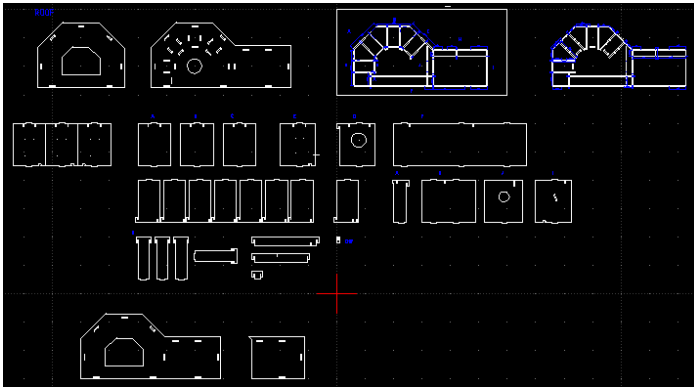
```
int n=1.0;
int m=30.0;
int t=200.0;//change t or m for frequency
void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
  pinMode(A3, INPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT); }
void loop() {
  float high, low, sumhigh, sumlow,high2, low2, sumhigh2,
  sumlow2,high3, low3, sumhigh3, sumlow3,high4, low4,
  sumhigh4, sumlow4;

  sumhigh=sumhigh2=sumhigh3=sumhigh4=sumlow=sumlow2
  =sumlow3=sumlow4=0;
  //Repeat m times
  for(int control=13;control>10;control--){
    for(int time=t;time>1;time--){
      for(int a=0; a<m; a++){

high=0;high2=0;high3=0;high4=0;low=0;low2=0;low3=0;low
4=0;
// Turn light on
digitalWrite(control, HIGH);
//Average n readings
for(int count=0; count<n; count++){
  high+=analogRead(A0);
  high2+=analogRead(A1);
```

```
high3+=analogRead(A2);
high4+=analogRead(A3);
delayMicroseconds(210); }
//Turn light off
digitalWrite(control, LOW);
//Average n readings
for(int count=0; count<n; count++){
  low+=analogRead(A0);
  low2+=analogRead(A1);
  low3+=analogRead(A2);
  low4+=analogRead(A3);
  delayMicroseconds(210); }
sumhigh += high;
sumlow += low;
sumhigh2 += high2;
sumlow2 += low2;
sumhigh3 += high3;
sumlow3 += low3;
sumhigh4 += high4;
sumlow4 += low4; }
//Average
sumhigh = 1000*sumhigh*5.0/(m*n*1023.0);
sumlow = 1000*sumlow*5.0/(m*n*1023.0);
sumhigh2 = 1000*sumhigh2*5.0/(m*n*1023.0);
sumlow2 = 1000*sumlow2*5.0/(m*n*1023.0);
sumhigh3 = 1000*sumhigh3*5.0/(m*n*1023.0);
sumlow3 = 1000*sumlow3*5.0/(m*n*1023.0);
sumhigh4 = 1000*sumhigh4*5.0/(m*n*1023.0);
sumlow4 = 1000*sumlow4*5.0/(m*n*1023.0);
Serial.print(sumhigh);
Serial.print(",");
Serial.print(sumlow);
Serial.print("\t");
Serial.print(sumhigh2);
Serial.print(",");
Serial.print(sumlow2);
Serial.print("\t");
Serial.print(sumhigh3);
Serial.print(",");
Serial.print(sumlow3);
Serial.print("\t");
Serial.print(sumhigh4);
Serial.print(",");
Serial.print(sumlow4);
Serial.print("\n");} } }
```

Design for haze chamber in LibreCAD:



Program was written to identify and classify red, blue and green light intensity (average of approx. last two cycles; Arduino was programmed to change colour of light and one cycle is completed when red, blue and green have taken turn to blink). Then, it was used to calculate the percentage change of light intensity for each colour.

PROGRAMME:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
int main()
{ FILE *fp,*fp2;
  double sensor[4][10000],check[4],k;
  int x,y,z[3],zR,condition[3],option;//zR is replacement for z
  double
  colour[3][4][10000]={0},Caverag[3][4],CRaverag[3][4],Tcolour;
  int determine=0,counter[3];
  char fname[20]="";
  int rate=199;// Removing 21 in corners
  //Open File .csv rgb gbr rgb
  fp=fopen("WithoutHAZE.csv","r");
  fp2=fopen("WithHAZE.csv","r");
  if(!fp){
    printf("File failed to open\n");
    system("pause");
    exit(1); }
  printf("-----Initial-----\n");
  for(x=0;x<3;x++){
    for(y=0;y<4;y++){
      Caverag[x][y]=0;}
    counter[x]=0;
    z[x]=0; }
  for(x=0;fscanf(fp,"%lf,%lf,%lf,%lf\n",&check[0],&check[1],
  &check[2],&check[3])!=EOF;x++){
    if(x!=0 && x%rate==0){
      determine+=1;
      determine=determine%3; }
    if((x%rate)>10 && (x%rate)<189){
      for(y=0;y<4;y++){
        sensor[y][x]=check[y];
        colour[determine][y][z[determine]]=check[y]; }
      z[determine]+=1;
```

```
counter[determine]+=1;
  //To print DATA Values
  //printf("%lf,%lf,%lf,%lf\n",sensor[0][x],sensor[1][x],sensor[2]
  ][x],sensor[3][x]); } }
  //printf("*-----*\n");
  for(x=0;x<3;x++){
    printf("\nCounter:%d\n",counter[x],z[x],z[x]-178*2);
    for(y=0;y<4;y++){
      if(counter[x]!=0){
        Tcolour=0;
        if((z[x]+1)>(178*2)){
          for(zR=z[x]-178*2;zR<z[x];zR++){
            Tcolour+=colour[x][y][zR];
            Caverag[x][y]=Tcolour/(178*2);
            printf("Average of colour[%d][%d]:
            %lf\n",x,y,Caverag[x][y]); }
          else{ for(zR=0;zR<z[x];zR++){
            Tcolour+=colour[x][y][zR];
            Caverag[x][y]=Tcolour/counter[x];
            printf("Average of colour[%d][%d]:
            %lf\n",x,y,Caverag[x][y]); } } }
    fclose(fp);
    //Relative
    if(!fp2){
      printf("File failed to open\n");
      system("pause");
      exit(1);}
    printf("\n-----Output-----\n");
    //initialize
    determine=0;
    for(x=0;x<3;x++){
      for(y=0;y<4;y++){
        CRaverag[x][y]=0; }
      counter[x]=0;
      z[x]=0; }
    for(x=0;fscanf(fp2,"%lf,%lf,%lf,%lf\n",&check[0],&check[1],
    &check[2],&check[3])!=EOF;x++){
      if(x!=0 && x%rate==0){
        determine+=1;
        determine=determine%3; }
      if((x%rate)>10 && (x%rate)<189){
        for(y=0;y<4;y++){
          sensor[y][x]=check[y];
          colour[determine][y][z[determine]]=check[y]; }
        z[determine]+=1;
        counter[determine]+=1;
        //To print DATA Values
        //printf("%lf,%lf,%lf,%lf\n",sensor[0][x],sensor[1][x],sensor[2]
        ][x],sensor[3][x]); } }
      //printf("*-----*\n");
      for(x=0;x<3;x++){
        printf("\nCounter:%d\n",counter[x],z[x],z[x]-178*2);
        for(y=0;y<4;y++){
          if(counter[x]!=0){
            Tcolour=0;
```

```

if((z[x]+1)>(178*2)){
    for(zR=z[x]-178*2;zR<z[x];zR++)
        Tcolour+=colour[x][y][zR];
    CRaverag[x][y]=Tcolour/(178*2);
    printf("Average of colour[%d][%d]:
    %lf\n",x,y,CRaverag[x][y]);
    else{for(zR=0;zR<z[x];zR++)
        Tcolour+=colour[x][y][zR];
        CRaverag[x][y]=Tcolour/counter[x];
        printf("Average of colour[%d][%d]: %lf\n
        ",x,y,CRaverag[x][y]); } } }
printf("\n-----Change in Intensity in Constant-----
-----\n");
//Change in Intensity = Output - Control
for(x=0;x<3;x++){
    for(y=0;y<4;y++){
        printf("Change in Intensity [%d][%d]:
        %lf\n",x,y,CRaverag[x][y]/Caverag[x][y]); }

```

```

    printf("\n"); }
fclose(fp2);
//Wrting DATA
printf("Enter 0 to SAVE DATA, 1 to END:\n");
scanf("%d",&option);
if(option==0){
    printf("Type a Valid File Name to SAVE Data:\n");
    scanf("%s",&fname);
    fp=fopen(fname,"w");
    for(x=0;x<3;x++){
        for(y=0;y<4;y++){
            fprintf(fp,"%lf",CRaverag[x][y]/Caverag[x][y]); }
            fprintf(fp,"\n"); }
    fclose(fp);
    printf("FILE Created\n"); }
else{
    printf("Nothing To Be Done\n"); }
system("pause");
return 0; \}

```



Figure 1.1



Figure 1.2

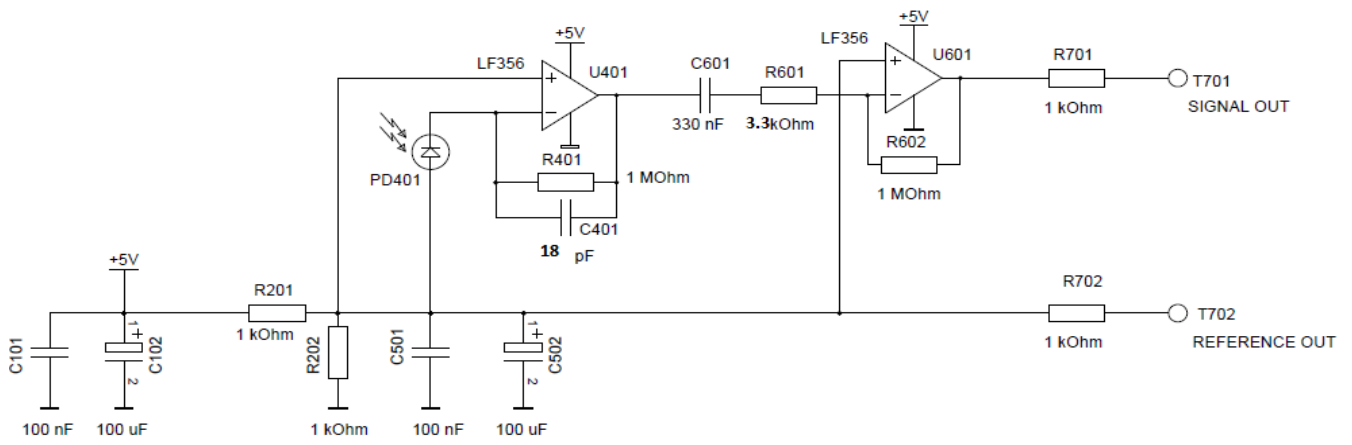


Figure 2.1

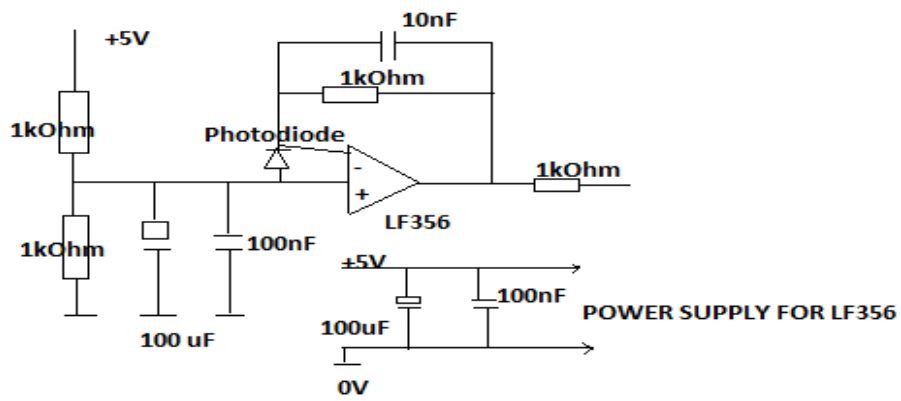


Figure 2.2

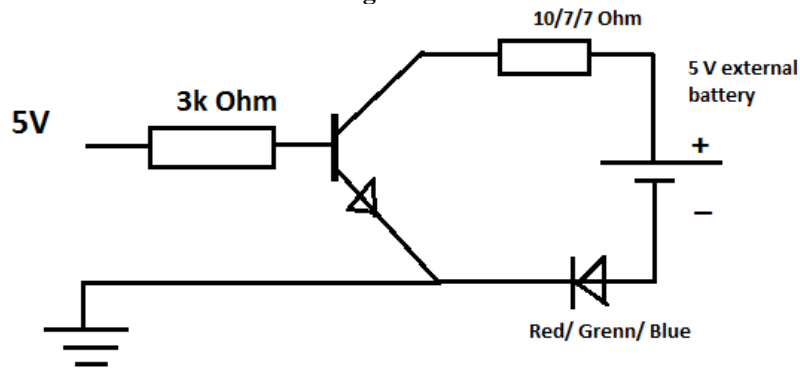


Figure 3



Figure 4

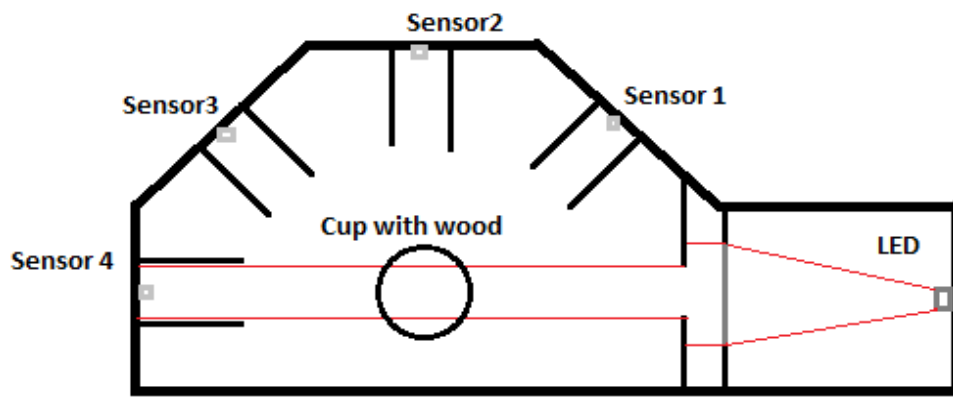


Figure 5

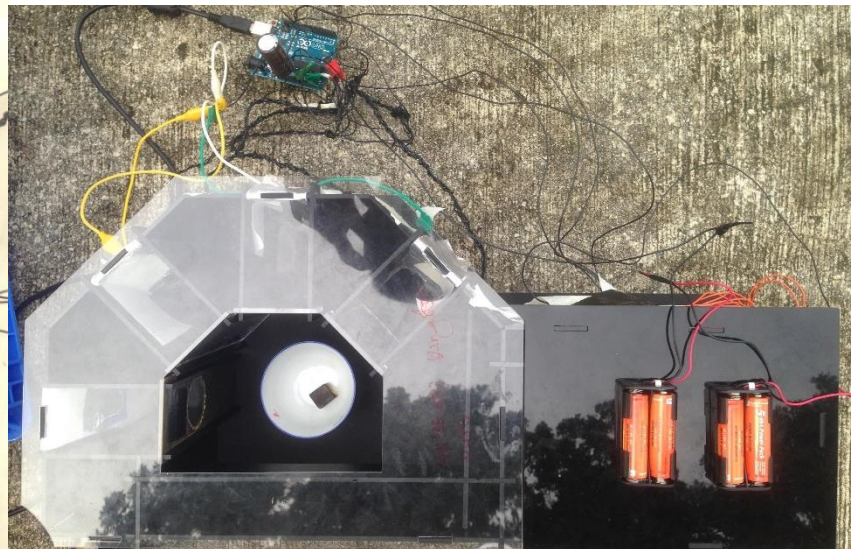
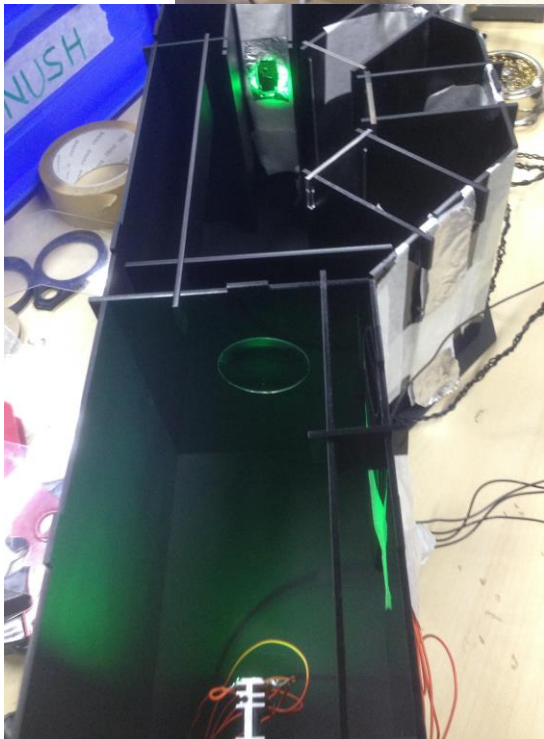
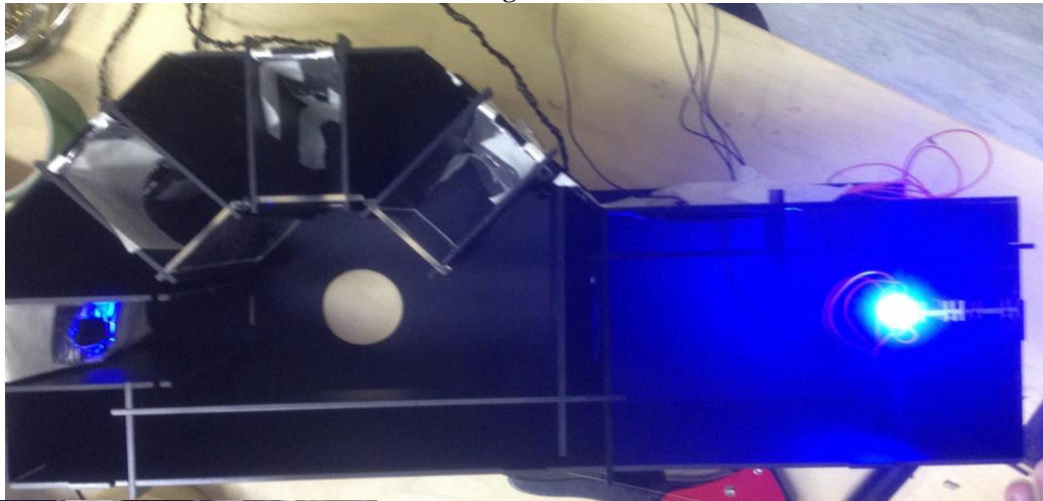


Figure 6

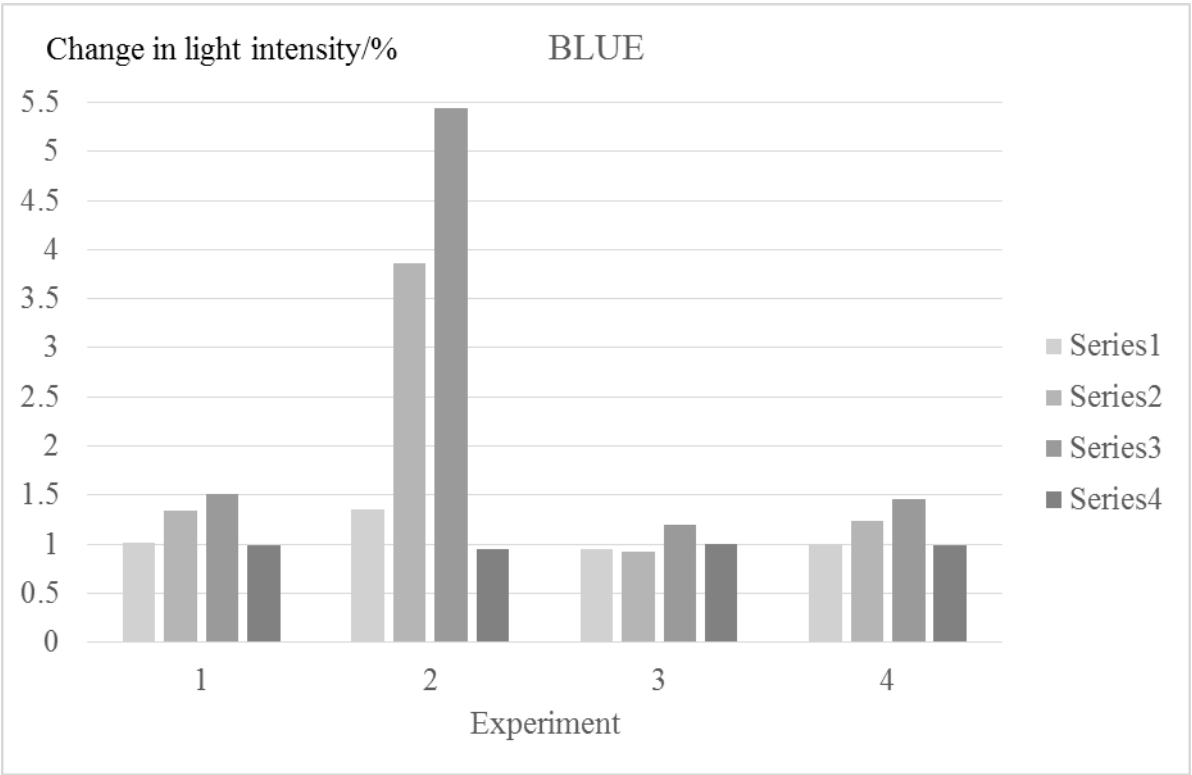


Figure 7.1

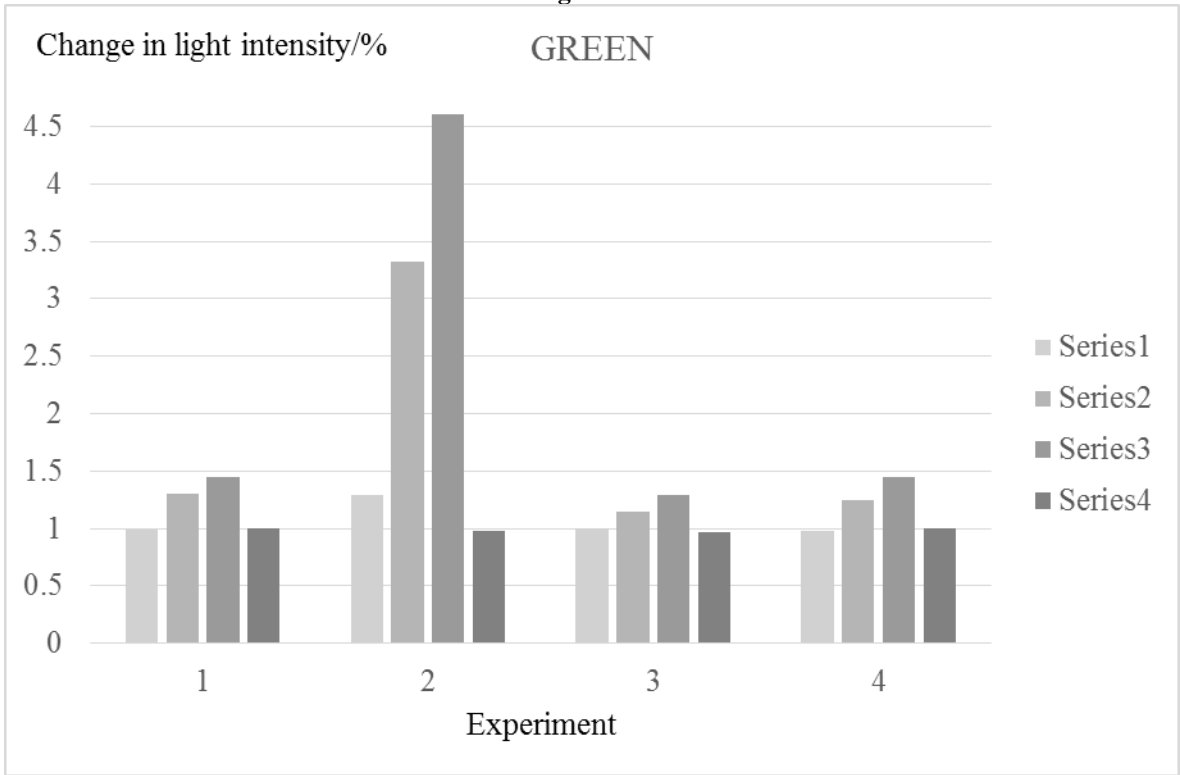


Figure 7.2

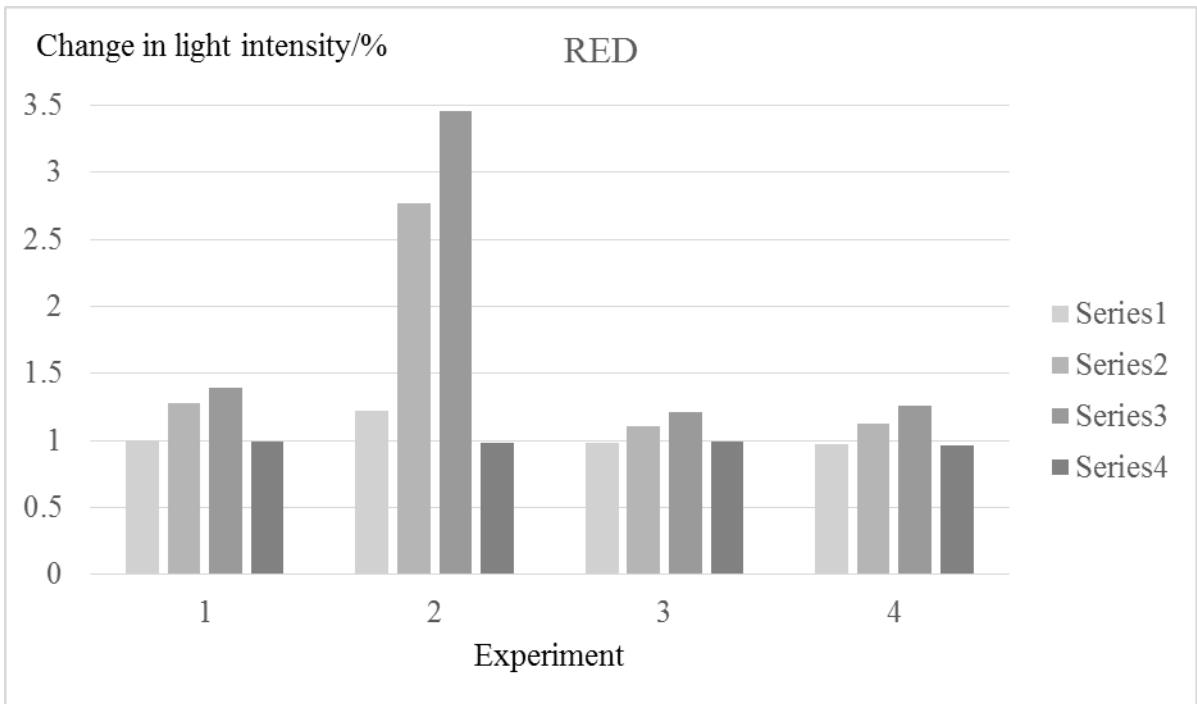


Figure 7.3

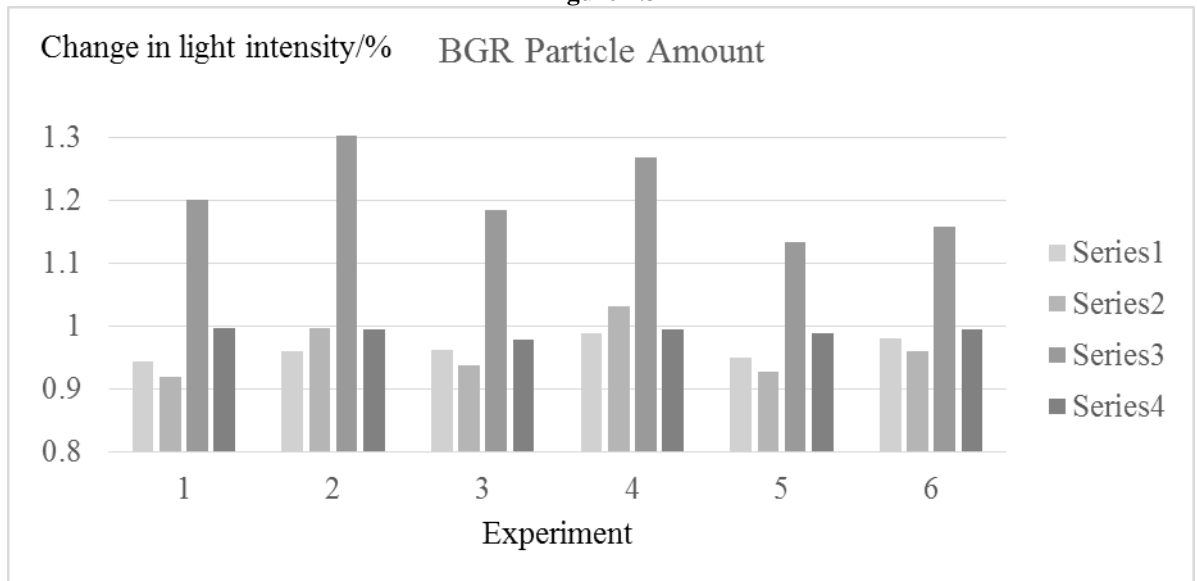


Figure 8.1

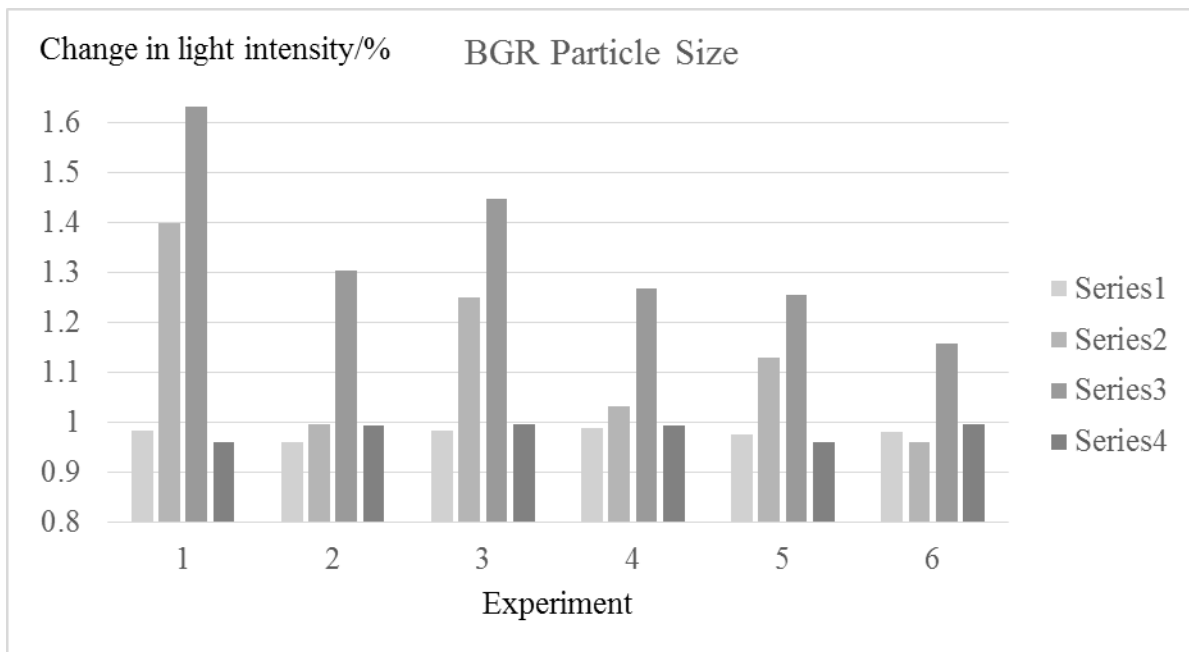


Figure 8.2

Table 1: Materials used for light sensor circuits with respective specifications & quantities.

<u>Materials</u>	<u>Specifications</u>	<u>Quantity</u>	<u>Materials</u>	<u>Specifications</u>	<u>Quantity</u>
Matt black acrylic	3.0 mm thick Width: 40 cm Length: 60 cm	2 sheets	JFET Input Operational Amplifier	TL062	3
Transparent acrylic	3.0 mm thick Width: 40 cm Length: 60 cm	2 sheets	JFET Input Operational Amplifier	LF356	1
Aluminium foil & paper	Length: 30cm Width: 4.0cm	4 each	Ceramic capacitor	220nF (nanofarads)	3
PVC pipe	Radius: 1.8 cm Length: 5 cm	1	Ceramic capacitor	18pF (picofarads)	3
Copper Prototype PCB	Length: 6.0 cm Width: 5.0 cm	4	Ceramic capacitor	100nF	11
Fridge stands		3	Ceramic capacitor	10nF	1
Screws & Nuts	1mm radius	16 Screws 48 Nuts	Electrolytic capacitor	100µF	8
Ceramic cups		4	Photodiode	PD401	4
Incense sticks	Varies	6	Resistor	1kΩ	16
DAISO Wood	Varies	3	Resistor	1MΩ	6

Arduino	Uno R3	1	Resistor	3.3k Ω	3
---------	--------	---	----------	---------------	---

Note: 330nF capacitor in **Research Materials and Methods Figure 1** circuit replaced with 100nF capacitor and 220nF capacitor placed in parallel due to shortage of parts

Equipment used:

Laser cutter LibreCAD software Soldering set PVC pipe cutter Dust blower Lighter
Laptop Analytical balance Drill bitsPCB cutter Arduino IDE CoolTerm