

**Department of Artificial Intelligence and Machine Learning**

Design of visible spectrophotometer using rgb led

**Submitted By:**

**RVCE23BAI029-Gnanendra Naidu N**

**RVCE23BAI020-Amudhan S**

**RVCE23BAI023-Apoorva Krishna P**

**RVCE23BAI055-Dhaksha Muthukumaran**

**INTRODUCTION**

Spectrophotometry is a widely-used technique in analytical chemistry that allows the determination of the composition of substances based on their interaction with light. Visible spectrophotometry specifically focuses on the analysis of compounds that absorb or transmit light within the visible range of the electromagnetic spectrum. It plays a crucial role in fields such as environmental monitoring, pharmaceutical research, and quality control in the food industry.

Traditional spectrophotometers are often expensive and may require specialized training for operation. This study seeks to address these limitations by developing a visible spectrophotometer using an Arduino microcontroller and an RGB LED as the light source. The Arduino platform provides a user-friendly and affordable option for controlling the LED, enabling precise control over the emitted wavelengths. This DIY approach aims to democratize spectroscopic analysis, making it accessible to a broader audience.

In this study, a visible spectrophotometer was constructed using an Arduino-controlled RGB LED and evaluated its effectiveness in generating an absorption versus wavelength graph for molecular analysis. The spectrophotometer utilized the principles of absorption spectroscopy, where the amount of light absorbed by a sample at specific wavelengths provides information about the presence and characteristics of molecules within the solution.

The Arduino platform facilitated control over the RGB LED, enabling the emission of light at different wavelengths within the visible spectrum. The sample solutions were placed in a cuvette, allowing the transmitted light to be measured by a light sensor connected to the Arduino. By varying the wavelength of the emitted light and recording the intensity of the transmitted light, data points for the absorption versus wavelength graph were obtained. By comparing the obtained absorption spectrum with known spectra of different compounds, preliminary identifications of potential molecules were made.

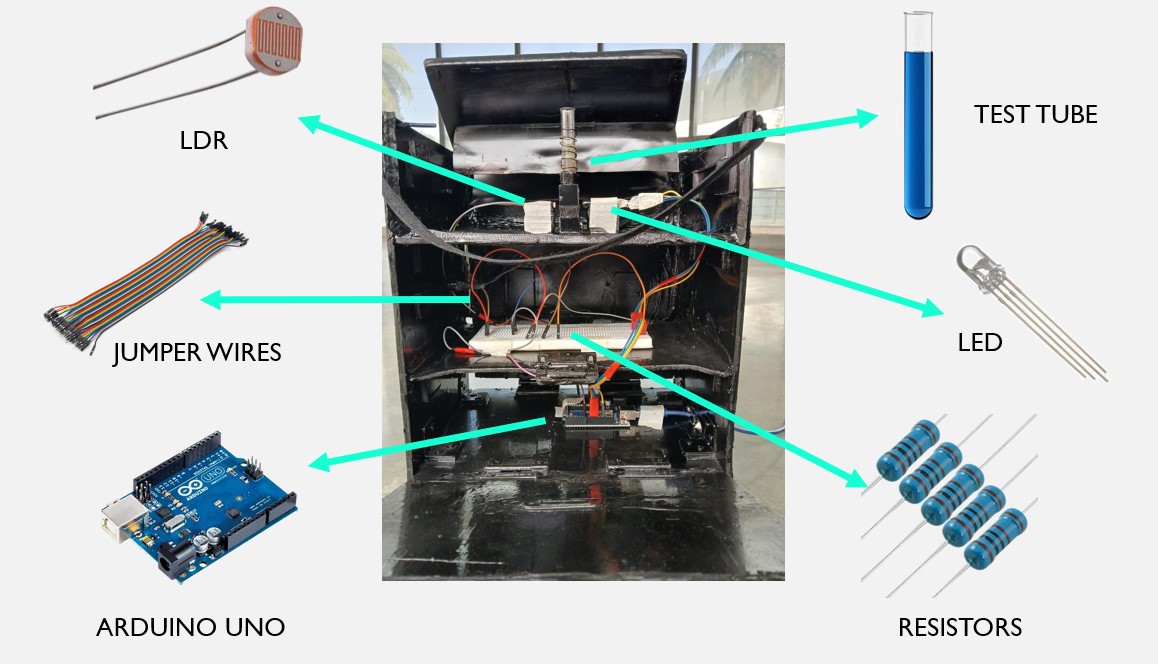
**PRINCIPLE**

Spectrophotometry is a novel technique that focuses on the sensitive and accurate detection of ions and molecules in various samples and has emerged as a powerful tool for qualitative analysis. It combines the principles of spectrometry and photometry to measure the absorbance or transmission of light by a sample at different wavelengths. Spectrometry involves the dispersion of light into its component wavelengths. In spectrophotometry, a spectrometer is used to split a beam of light into its individual wavelengths, creating a spectrum.

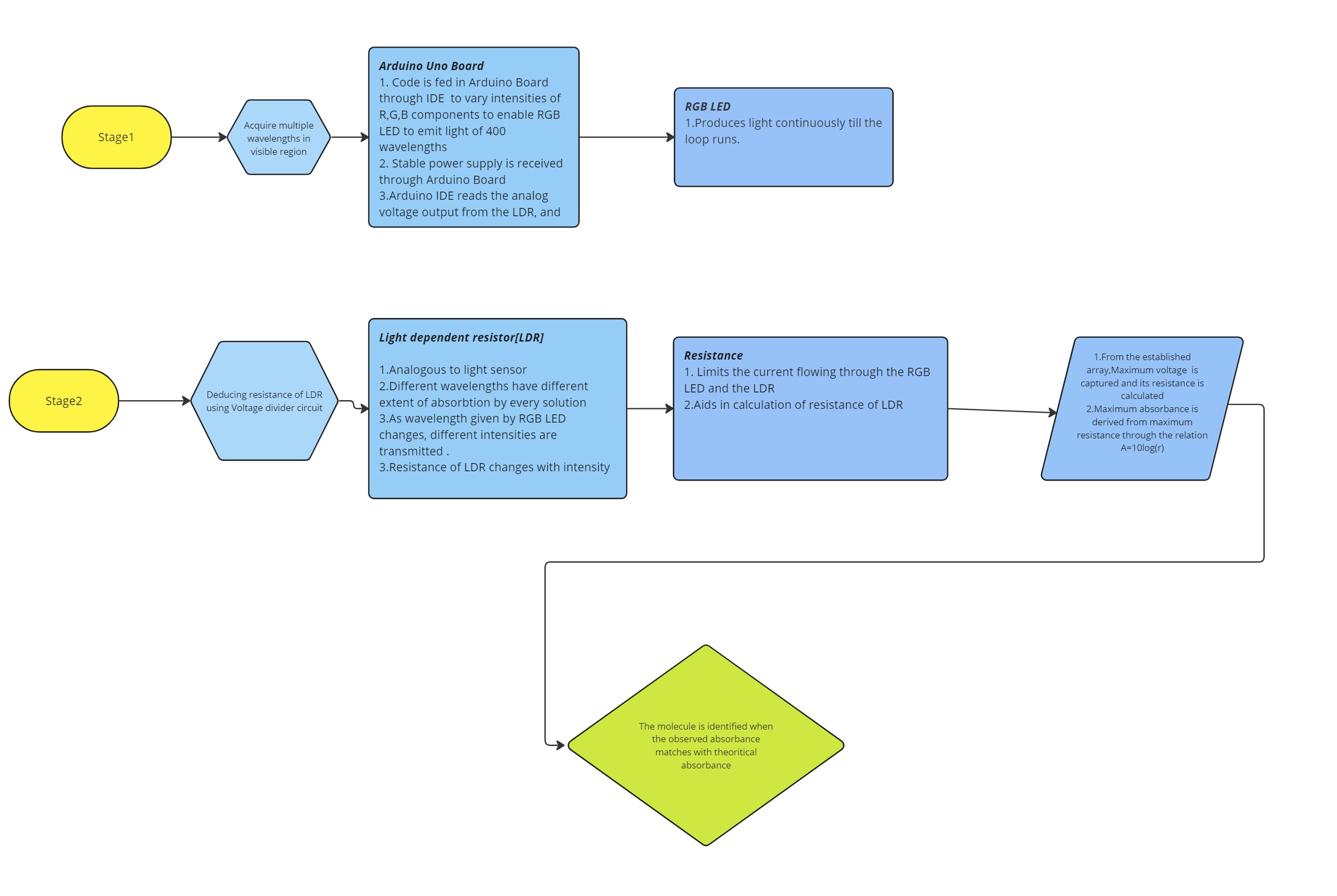
In a visible spectrometer this corresponds to the visible spectrum. In our study, we make use of an RGB LED controlled by an Arduino for this purpose. By scanning through different wavelengths, the spectrometer generates a series of monochromatic light beams. Photometry involves the measurement of the intensity of light. In spectrophotometry, photometric measurements are performed using a photodetector, such as a photodiode or a photomultiplier tube. The photodetector converts the light intensity into an electrical signal that can be measured and quantified. We make use of an LDR as a photodetector.

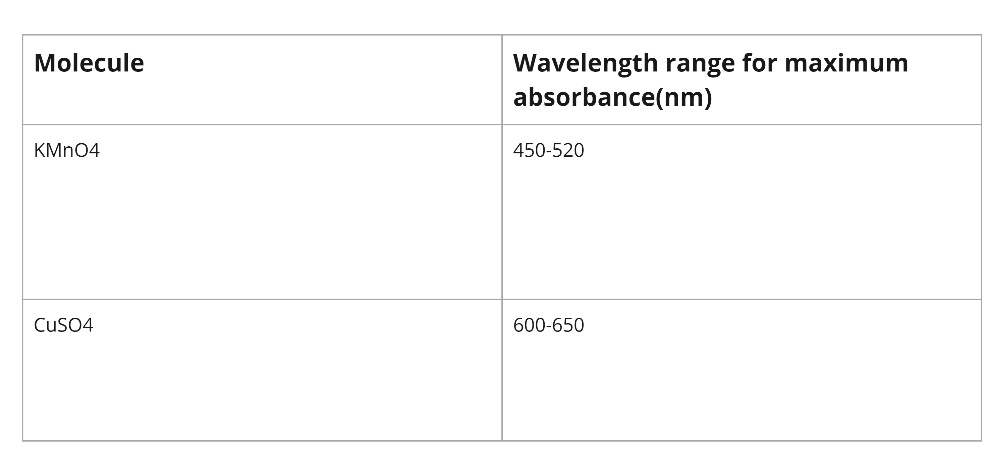
The response of the photodetector to the light at each specific wavelength is recorded, allowing for the determination of the intensity of light transmitted through the sample. By combining spectrometry and photometry, spectrophotometry enables the measurement of the absorption or transmission of light by a sample as a function of the wavelength or frequency of the light. It involves comparing the intensity of light before and after it interacts with the sample to determine the amount of light absorbed or transmitted, which provides information about the properties of the sample.

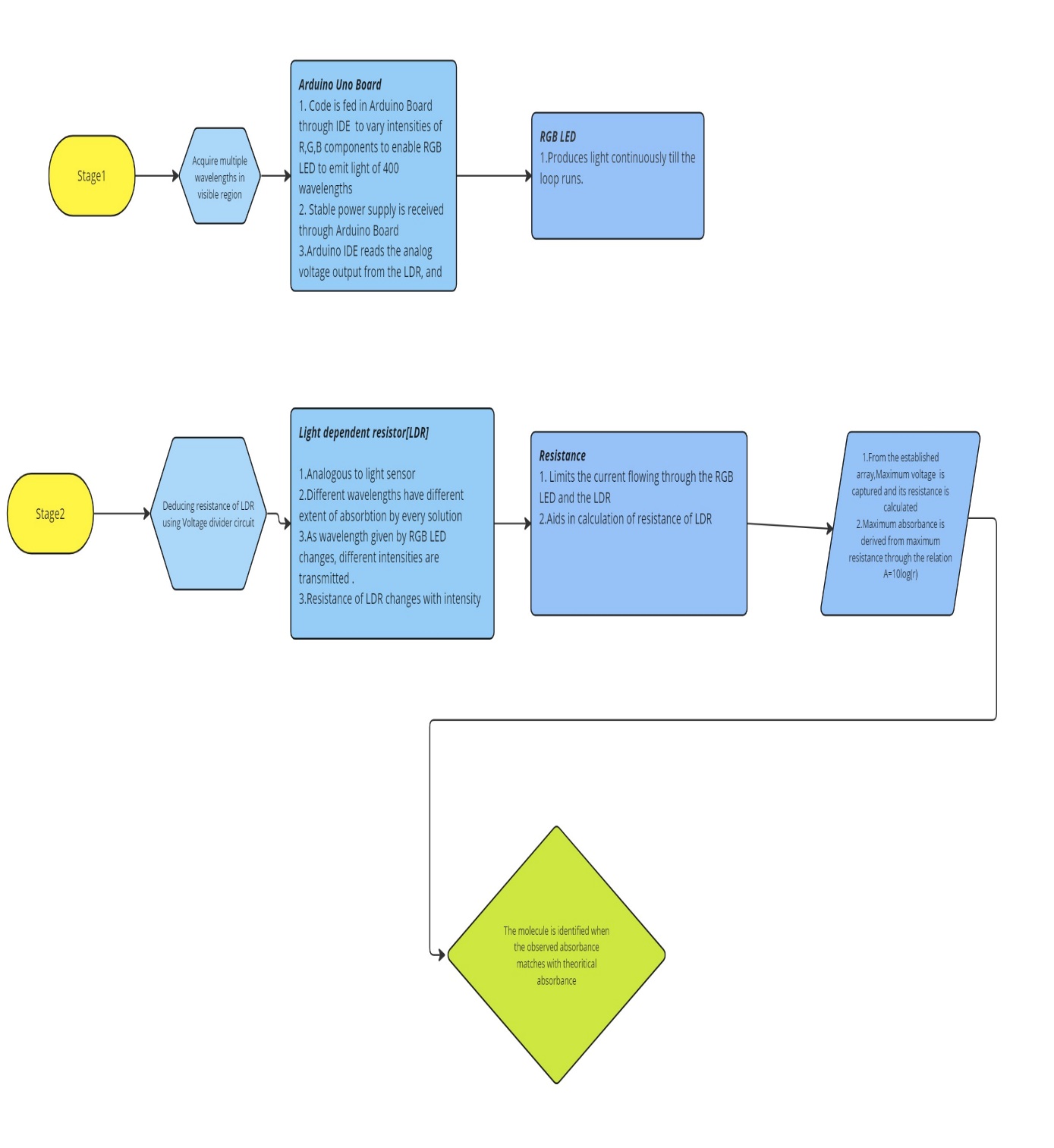
**MATERIALS**



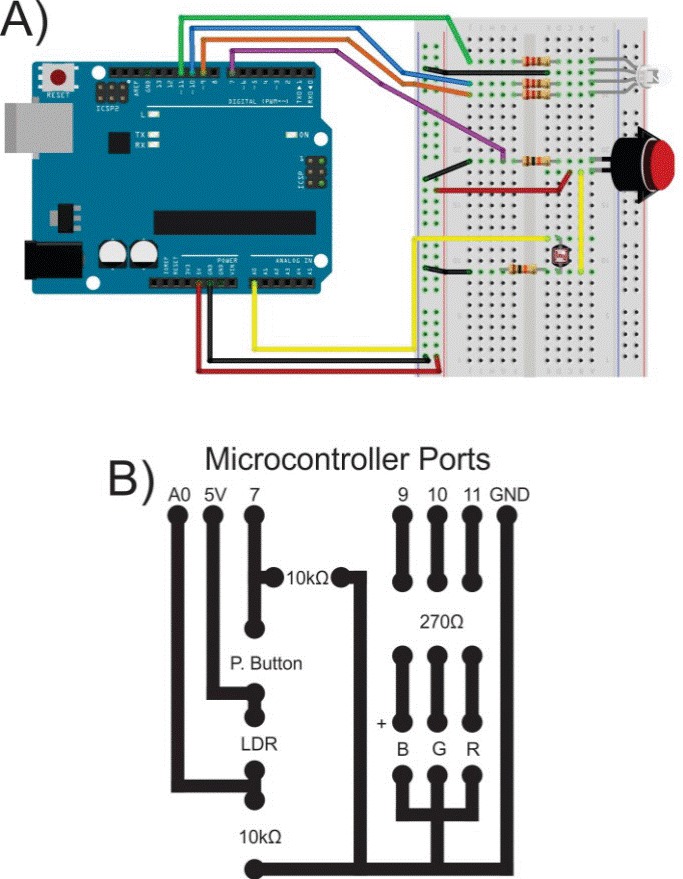
**WORKING**

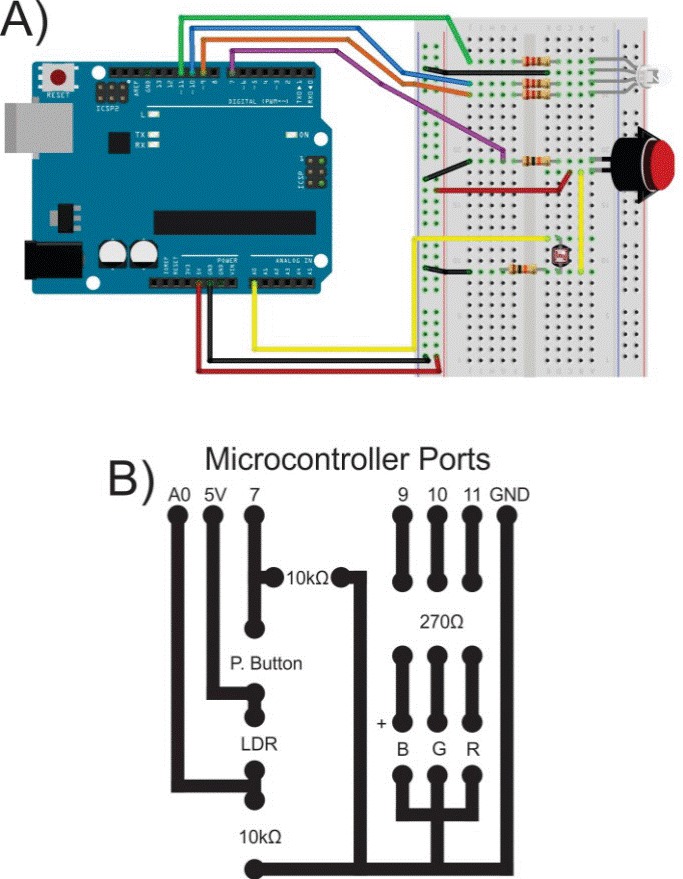






**CIRCUIT**





**CODE**

MATLAB Code:

clear

l=zeros(1,401);

l=['ldr values ']

V=l\*5/1023;

R=(V-5)/(5/100);

a=-(2)\*log10(1./R);

w=[450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699];

max(a)

indices = find(a == max(a));

correspondingValues = w(indices);

plot(w,a,'-')

xlabel('Wavelength');

ylabel('Absorbance');

title('Absorption Spectrum');

Arduino code:

#include <math.h>

class Value {

public:

    float R, G, B;

    Value()

     {

        R = 0.0;

        G = 0.0;

        B = 0.0;

     }

    float red(float a)

    {

        if (646 <= a && a <= 780) {

            R = 1;

        } else if (581 <= a && a <= 645) {

            R = 1;

        } else if (511 <= a && a <= 580) {

            R = (a - 510) / 70;

        } else if (491 <= a && a <= 510) {

            R = 0;

        }

        if (441 <= a && a <= 490) {

            R = 0;

        }

        if (380 <= a && a <= 440) {

            R = (440 - a) / 60;

        }

        return R;

    }

    float green(float a)

    {

        if (646 <= a && a <= 780) {

            G = 0;

        } else if (581 <= a && a <= 645) {

            G = (645 - a) / 65;

        } else if (511 <= a && a <= 580) {

            G = 1;

        } else if (489 <= a && a <= 510) {

            G = 1;

        }

        if (441 <= a && a <= 490) {

            G = (a - 440) / 50;

        }

        if (380 <= a && a <= 440) {

            G = 0;

        }

        return G;

    }

    float blue(float a)

    {

        if (646 <= a && a <= 780) {

            B = 0;

        } else if (581 <= a && a <= 645) {

            B = 0;

        } else if (511 <= a && a <= 580) {

            B = 0;

        } else if (491 <= a && a <= 510) {

            B = (510 - a) / 20;

        }

        if (441 <= a && a <= 490) {

            B = 1;

        }

        if (380 <= a && a <= 440) {

            B = 1;

        }

        return B;

    }

    float factor(float a)

    {

        float fac = 1;

        if (701 <= a && a <= 780) {

            fac = (3.0 / 10) + (((7.0 / 10) \* (780 - a)) / 80);

        } else if (421 <= a && a <= 700) {

            fac = 1;

        } else if (380 <= a && a <= 420) {

            fac = (3.0 / 10) + (((7.0 / 10) \* (a - 380)) / 40);

        }

        return fac;

    }

    float fina(float a, float b)

    {

        float f = 255 \* pow((a \* b), (8.0 / 10));

        return f;

    }

};

void setup()

{

  Serial.begin(9600);

  int j = 0;

  int R[402];

  int G[402];

  int B[402];

  int L[402];

  pinMode(9,OUTPUT);

  pinMode(10,OUTPUT);

  pinMode(11,INPUT);

  for(int i=380;i<=780;i++)

  {

    Value V;

    L[j] = i;

    R[j] = static\_cast<int>(V.fina(V.factor(i), V.red(i)));

    G[j] = static\_cast<int>(V.fina(V.factor(i), V.green(i)));

    B[j] = static\_cast<int>(V.fina(V.factor(i), V.blue(i)));

    //Serial.print("{");

    //Serial.print(R[j]);

    //Serial.print(",");

    //Serial.print(G[j]);

    //Serial.print(",");

    //Serial.print(B[j]);

    //Serial.print("}");

    j++;

  }

}

void loop()

{

  int LDR[402];

  int A[402];

  int R[402];

  int G[402];

  int B[402];

  int L[402];

  int Res[402];

  if(digitalRead(7)==LOW)

  {

  Serial.print("[");

  for(int c=0;c<=401;c++)

  {

    analogWrite(9,R[c]);

    analogWrite(10,G[c]);

    analogWrite(11,B[c]);

    delay(50);

    LDR[c]=analogRead(0);

    Res[c]=(500/LDR[c])-100;

    A[c]=-log(Res[c]);

    Serial.print(LDR[c]);

    Serial.print(",");

    delay(50);

  }

  Serial.print("]");

  }

  else

  {

    analogWrite(9,0);

    analogWrite(10,0);

    analogWrite(11,0);

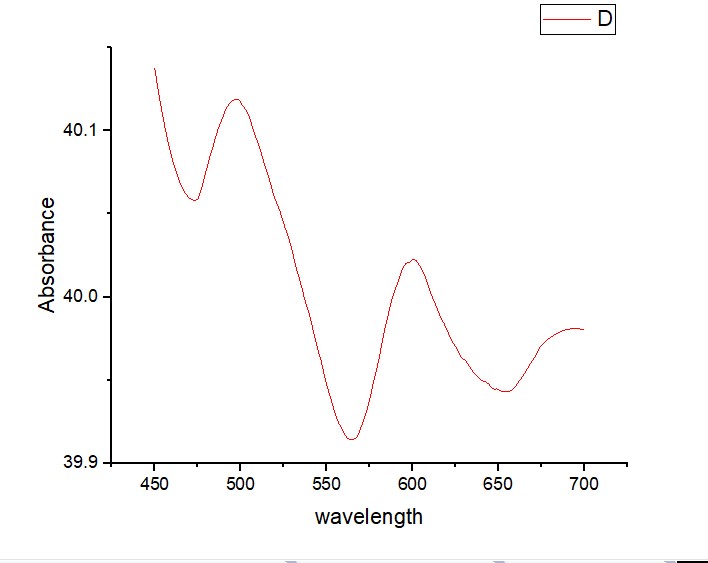
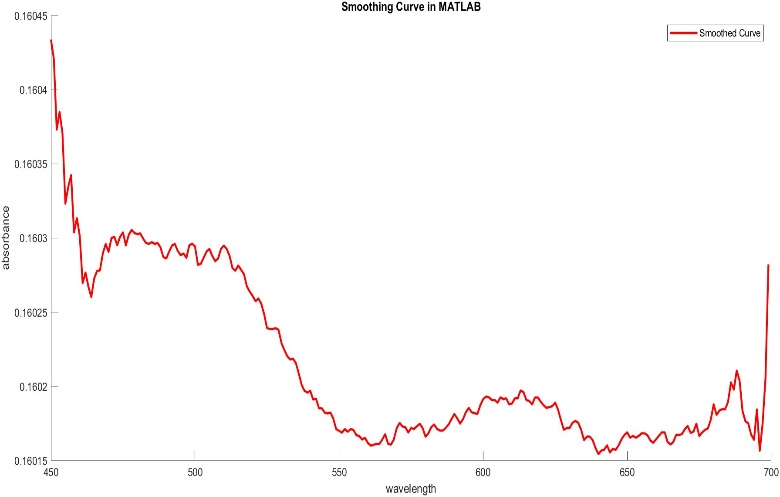
    exit;

  }

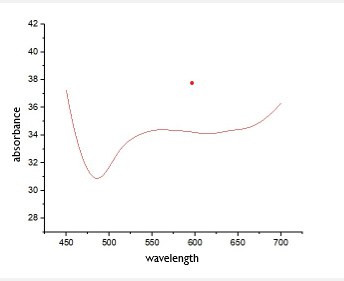
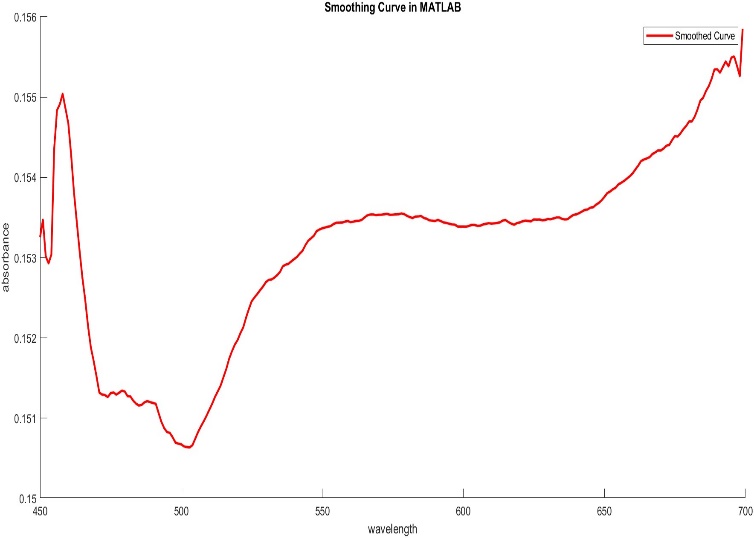
}

**GRAPHS**

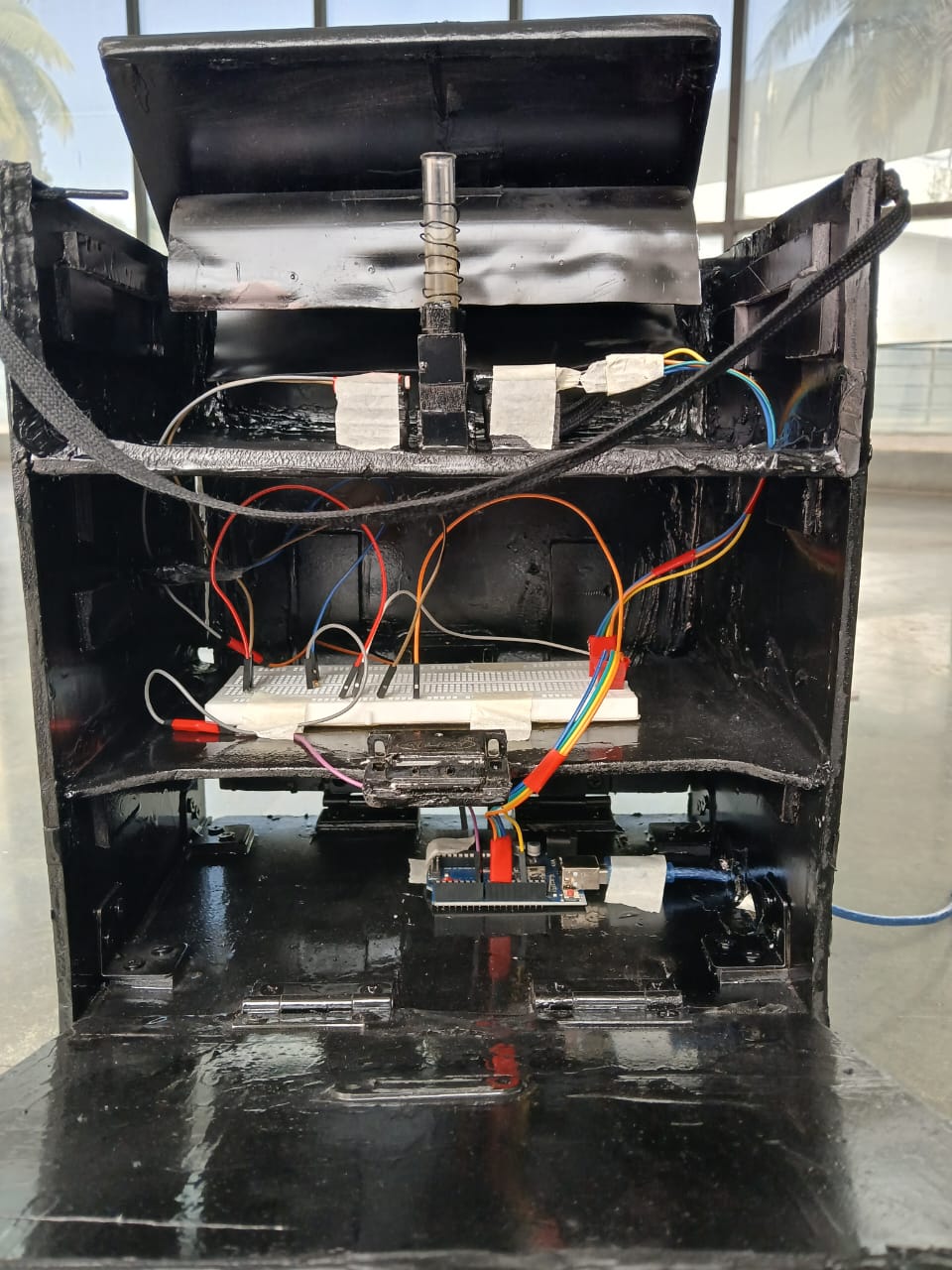
Absorption vs Wavelength Graph of CuSO4



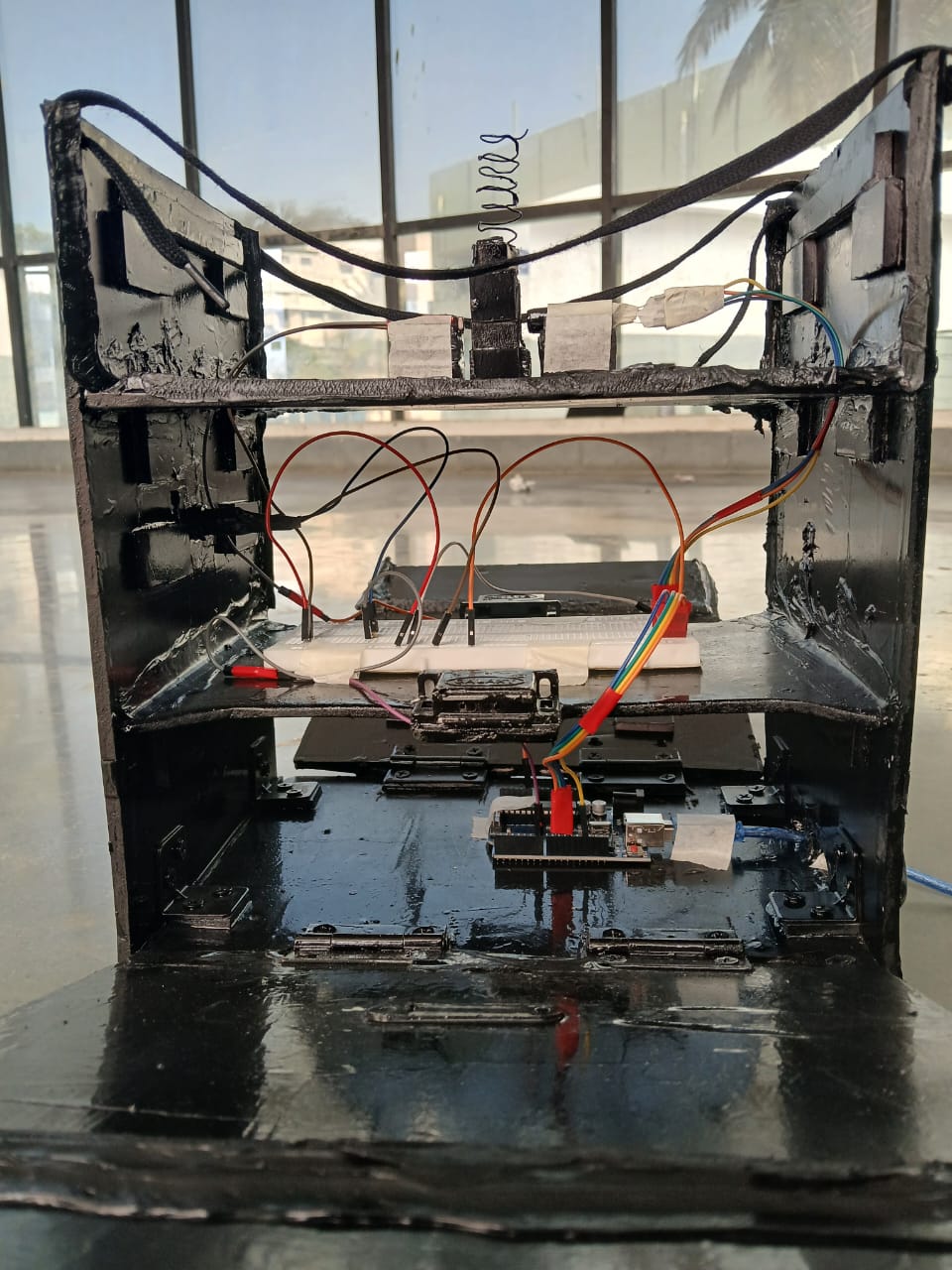
Absorption vs Wavelength Graph of KMnO4



**PHOTOS**









**RESULTS**

A series of measurements were taken by varying the wavelength of the emitted light. For each wavelength, the intensity of the transmitted light was recorded. This data was used to calculate the absorbance of the solution at different wavelengths. The absorbance values were then plotted against the corresponding wavelengths to generate the absorption versus wavelength graph.)

The visible spectrophotometer constructed using an Arduino-controlled RGB LED successfully generated an absorption versus wavelength graph for the analyzed solution. By comparing the obtained absorption spectrum with known spectra of different molecules, it was possible to identify the potential molecules present in the solution. The positions and shapes of the peaks in the absorption spectrum provided valuable information about the types and concentrations of the compounds.

The obtained absorption versus wavelength graph displayed distinct peaks and valleys, indicating the presence of specific molecules in the analyzed solution. Each peak or valley corresponded to a particular wavelength at which the molecules absorbed or transmitted light most effectively. The positions and shapes of these absorption features provided valuable information about the types of compounds present in the solution. The characteristic absorption peaks observed in the spectrum could be attributed to specific molecular species, allowing for qualitative analysis.

The visible spectrophotometer developed in this study demonstrates the potential for low-cost and accessible spectroscopic analysis in various scientific and educational applications. Further improvements and refinements to the instrument, as well as comprehensive calibration and validation processes, could enhance the accuracy and reliability of the results. Overall, this study lays the foundation for future investigations and applications of visible spectrophotometry using an Arduino-controlled RGB LED setup.

**FUTURE WORK**

The developed visible spectrophotometer holds promise for various applications in scientific research, education, and industry. Its low-cost and accessible design, powered by Arduino, opens up opportunities for wider adoption and customization. Future work will focus on optimization, calibration and further applications of the spectrophotometer to enhance its performance and broaden its potential uses.

Optimizing the sensor design of the visible spectrophotometer includes exploring different sensor technologies that offer improved sensitivity and dynamic range. Additionally, investigating the use of specialized coatings or filters to enhance the sensor's response in specific wavelength ranges can improve the overall performance of the spectrophotometer. Improving the spectral resolution of the visible spectrophotometer is also crucial for accurately detecting and characterizing narrow absorption features. Future work would include exploring techniques such as increasing the number of wavelength measurement points or utilizing more precise optical components to achieve higher resolution.

Establishing robust calibration strategies is essential for accurate quantification of the target compounds. Future work would focus on developing calibration methods that consider factors such as temperature variations, light source stability, and sensor drift. Implementing advanced signal processing techniques can help reduce noise and improve the signal-to-noise ratio of the obtained spectra. Future work should also focus on adapting the instrument and analysis methods to cater to specific application requirements.

Exploring miniaturization and portability options for the visible spectrophotometer can also expand its potential applications. By addressing these areas of future work, the visible spectrophotometer can be optimized, leading to improved performance, higher accuracy, and expanded functionality. These advancements would contribute to the broader field of spectroscopy and enable a wider range of applications in scientific research, industrial processes, and healthcare diagnostics.